Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for defining, configuring and monitoring Forwarding Equivalent Class (FEC) to Next Hop Label Forwarding Entry (NHLFE) mappings and corresponding actions for use with Multiprotocol Label Switching (MPLS).

Table of Contents

1. Introduction .............................................. 2
2. Terminology ............................................. 2
1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for specifying FEC to NHLFE mappings and corresponding actions for Multiprotocol Label Switching (MPLS).

On the ingress of an MPLS network, packets entering the MPLS domain are assigned to a FEC. Those packets belonging to a forwarding equivalency class (FEC) are associated with an NHLFE (i.e.: MPLS label) via the FEC-to-NHLFE (FTN) mapping [RFC3031]. This relationship defines how the an ingress LSR will impose MPLS labels onto incoming packets. It also explains how egress LSRs will de-capsulate the MPLS shim header from MPLS packets. It is important to note that an NHLFE entry can redirect packets to either an LSP or a Traffic Engineered (TE) Tunnel.

Conceptually, some of the FTN table functionality could be implemented using the Forwarding Information Base (FIB) to map all packets destined for a prefix to an LSP. However, this mapping is coarse in nature.

Similar functionality is already being used in other contexts, such as security filters, access filters, and for RSVP flow identification. All of these require various combinations of matching based on IP header and upper-layer header information to identify packets for a particular treatment. When packets match a particular rule, a corresponding action is executed against those packets. For example, two popular actions to take when a successful...
match is detected are allowing the packet to be forwarded or to
discard it. However, other actions are possible, such as modifying
the TOS byte, or redirecting a packet to a particular outgoing
interface.

This draft attempts to consolidate the various matching requirements
and associated action options needed for MPLS into a single
specification.

2. Terminology

Although all of the terminology used in this draft is either covered
in the MPLS Architecture [RFC3031] or in the SNMP Architecture
[RFC2571], it is informational to define some immediately pertinent
acronyms/terminology here.

- MPLS Multiprotocol Label Switching
- FEC Forwarding Equivalence Class
- NHLFE Next-Hop Label Forwarding Entry
- FTN FEC-to-NHLFE
- MIB Management Information Base

3. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC-2119 [2].

4. The SNMP Management Framework

The SNMP Management Framework presently consists of five major
components:

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

- Mechanisms for describing and naming objects and events for the
  purpose of management. The first version of this Structure of
  Management Information (SMI) is called SMIv1 and described in STD
  16, RFC 1155 [RFC1155], STD 16, RFC 1212 [RFC1212] and STD 16,
  RFC 1215 [RFC1215]. The second version, called SMIv2, is
  described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579
  [RFC2579] and STD 58, RFC 2580 [RFC2580].

- Message protocols for transferring management information. The
  first version of the SNMP message protocol is called SNMPv1 and
described in STD 15, RFC 1157 [RFC1157]. A second version of the
  SNMP message protocol, which is not an Internet standards track
  protocol, is called SNMPv2c and described in RFC 1901 [RFC1901]
  and RFC 1906 [RFC1906]. The third version of the message
Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [RFC1905].

- A set of fundamental applications described in RFC 2573 [RFC2573] and the view-based access control mechanism described in RFC 2575 [RFC2575].

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

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

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

5. Outline

This MIB resides on any LSR which does the FEC-to-NHLFE mapping in order to map traffic into the MPLS domain. The MIB consists of three tables: The mplsFTNTable defines the rule base against which incoming packets are matched and actions taken on matching packets. The mplsFTNMapTable defines the application of these rules to specific interfaces. Finally, the mplsFTNPerfTable provides performance counters for every FTN entry that is active, on a per-interface basis.

5.1. mplsFTNTable

This table allows FEC-to-NHLFE mappings to be specified and monitored. Each entry in this table defines a rule to be applied to incoming packets (on interfaces that the FTN entry is activated on using mplsFTNMapTable; see Section 5.2) and an action to be taken on matching packets. mplsFTNTable provides a 5-tuple (Source Addr, Dest Addr, Source Port, DestPort, layer-4 Protocol) matching and allows addresses, port
ranges and the exp bits to be specified. The action pointer points at either an MPLS-LSR MIB \([\text{LSRMIB}]\) mplsXCEntry when the NHLFE entry is a non-TE LSP, or it points at an mplsTunnelEntry in the MPLS-TE MIB \([\text{TEMIB}]\) if we wish to make the NHLFE the start of a TE tunnel.

### 5.2. mplsFTNMapTable

This table provides the capability to activate or map FTN entries defined in mplsFTNTable to specific interfaces in the system. FTN entries are compared with incoming packets in the order in which they are applied on an interface. This order is specified by the order in which the rules are specified to be applied. Specifically, entries are arranged as a linked list of rules and are applied from the lowest indexed rule to the highest. Each entry in the list provides a forward and back pointer to possibly other mapping entries, as well as a pointer to the mplsFTNEntry in question.

Due to the linked-list structure of this table, it provides a mechanism to ‘insert’ an FTN entry between two existing FTN entries already applied on an interface.

Using this linked-list structure, one can retrieve FTN entries in the order of application on a per-interface basis as follows:

- To determine the first FTN entry on an interface with index ifIndex perform a GETNEXT retrieval operation on mplsFTNMapIndex.ifIndex.0.0; the returned object, if one exists, is (say) mplsFTNMapIndex.ifIndex.0.n. Then the index of the first FTN entry applied on this interface is n.

- To determine the FTN entry applied after the one indexed by n perform a GETNEXT retrieval operation on mplsFTNMapIndex.ifIndex.n.0; the returned object, if one exists, is (say) mplsFTNMapIndex.ifIndex.n.m. Then the index of the next FTN entry applied on this interface is m.

Use the above steps to retrieve all the applied FTN entries on a per-interface basis in application order. Note that the number of retrieval operations is the same as the number of applied FTN entries (i.e. the minimum number of GETNEXT operations needed using any indexing scheme).

### 5.3. mplsFTNPerfTable

This table provides performance counters for each FTN entry on a per-interface basis. High capacity counters are provided for situations where 32-bit counters would wrap around too quickly.

### 5.4 Pictoral Tabular Relationships
In this example, a single interface (ifIndex.1) is configured with three FTN entries utilizing three mapping table entries.

InterfacesMIB (RFC2863):

---ifEntry.1

mplsFTNMapTable:
<--(mplsFTNMapIndex = 1,
    mplsFTNMapPrevIndex = 0, --> (empty) <-----
    mplsFTNMapCurrIndex = 1) --------------      |

<--(mplsFTNMapIndex = 1,
    mplsFTNMapPrevIndex = 1, -------------> <--
    mplsFTNMapCurrIndex = 2) -------------

<--(mplsFTNMapIndex = 1,
    mplsFTNMapPrevIndex = 2, --------> <-
    mplsFTNMapCurrIndex = 3) ---

mplsFTNTable:
(mplsFTNIndex=1) <--------
(mplsFTNIndex=2) <--------
(mplsFTNIndex=3) <--------

mplsFTNPerfEntry:
(mplsFTNMapIndex = 1, mplsFTNMapCurrIndex = 1)
(mplsFTNMapIndex = 1, mplsFTNMapCurrIndex = 2)
(mplsFTNMapIndex = 1, mplsFTNMapCurrIndex = 3)

6. Examples

Suppose that we want to activate the following FTN entries.

1. in ifIndex=1, dest addr=1.2.0.0 -> out ifIndex=50, out label=150

2. in ifIndex=1, dest addr=1.3.0.0 -> tunnel=4

In this case the tables will look as follows in the MPLS LSR, TE and FTN MIBs. (Note that this is NOT an exhaustive list of all the objects in every table and we only show those objects of interest that help
illustrate each case.)

Entry #1 results in the following.

In mplsFTNTable:

{mplsFTNIndex = 1,
  mplsFTNDescr = "FTN-1 for net 1.2.0.0",
  mplsFTNMask = 0x40, -- destination address only
  mplsFTNAaddrType = ipv4,
  mplsFTNDestIpv4AddrMin = 1.2.0.0,
  mplsFTNDestIpv4AddrMax = 1.2.0.0,
  mplsFTNAActionType = redirectLsp,
  mplsFTNAActionPointer = mplsXCLspId.2.0.0.3
}

We indicate the LSP to redirect packets to by setting mplsFTNAActionPointer to the first column object of the XC entry corresponding to this LSP, in this case mplsXCLspId.2.0.0.3 which represents the following XC entry.

In mplsXCTable:

{mplsXCIndex = 2,
  mplsInSegmentIfIndex = 0, -- originating LSP
  mplsInSegmentLabel = 0, -- originating LSP
  mplsOutSegmentIndex = 3,
  mplsXCTLabelStackIndex = 0
}

Note that mplsInSegmentIfIndex and mplsInSegmentLabel values used to index this XC entry are zero as is required for an originating LSP [LSRMIB].

In mplsOutSegmentTable:

{mplsOutSegmentIndex = 3,
  mplsOutSegmentIfIndex = 50,
  mplsOutSegmentPushTopLabel = true,
  mplsOutSegmentTopLabel = 150
}

In mplsFTNMapTable:

{mplsFTNMapIndex = 1,
  mplsFTNPrevIndex = 0, -- first FTN entry on this interface
  mplsFTNMapCurrIndex = 1,
}

Entry #2 results in the following.
In mplsFTNTable:
{
  mplsFTNIndex = 2,
  mplsFTNDescr = "FTN-2 for net 1.2.0.0",
  mplsFTNMask = 0x40, -- destination address only
  mplsFTNAddrType = ipv4,
  mplsFTNDestAddrMin = 1.3.0.0,
  mplsFTNDestAddrMax = 1.3.0.0,
  mplsFTNActionType = redirectTunnel,
  -- We assume that the ingress and egress LSR IDs are 1.1.1.1
  -- and 2.2.2.2 respectively for this tunnel.
  mplsFTNActionPointer = mplsTunnelName.4.0.4.1.1.1.1.4.2.2.2.2
}

In mplsTunnelTable:
{
  mplsTunnelIndex = 4,
  mplsTunnelInstance = 0, -- primary tunnel
  mplsTunnelIngressLSRID = 1.1.1.1,
  mplsTunnelEgressLSRID = 2.2.2.2
}

In mplsFTNMapTable:
{
  mplsFTNMapIndex = 1,
  mplsFTNPrevIndex = 1,
  mplsFTNMapCurrIndex = 2
}

7. The Use of RowPointer

RowPointer is a textual convention used to identify a conceptual row in a conceptual Table in a MIB by pointing to one of its objects. In this MIB, in mplsFTNTable, the RowPointer object mplsFTNActionPointer indicates the LSP or TE Tunnel to redirect packets matching an FTN entry to. This object MUST point to the first instance of the first accessible columnar object in the appropriate conceptual row in order to allow the manager to find the appropriate corresponding entry in either the MPLS-LSR MIB [LSRMIB] or MPLS-TE MIB [TEMIB]. If this object returns zeroDotZero it implies that there is no currently defined action that is associated with that particular FTN entry.

8. MPLS-FTN MIB Definitions

MPLS-FTN-MIB DEFINITIONS ::= BEGIN

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, Counter64,

MPLS Working Group Expires May 2002 [Page 8]
DESCRIPTION

"This MIB module contains managed object definitions for specifying FEC to NHLFE (FTN) mappings and corresponding performance for MPLS."
-- Textual conventions used in this MIB.

MplsFTNIndex ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION "Index for a FEC-to-NHLFE (FTN) entry. A management
  station should consult the appropriate object containing
  the next available index associated with the table
  into which configuration is desired. For example, the
  MplsFTNTable has an associated mplsFTNIndexNext
  object that indicates the next available index."
  SYNTAX Unsigned32 (1..4294967295)

MplsFTNIndexOrZero ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION "Index for a FTN entry or zero. If
  the number of unassigned entries is
  exhausted, this object MUST return a
  value of 0."
  SYNTAX Unsigned32 (0..4294967295)

MplsFTNIndexIntegerNextFree ::= TEXTUAL-CONVENTION
  DISPLAY-HINT "d"
  STATUS current
  DESCRIPTION "An integer which may be used as a new Index in a table.

  The special value of 0 indicates that no more new entries can be
  created in the relevant table.

  When a MIB is used for configuration, an object with this SYNTAX
  always contains a legal value (if non-zero) for an index that is
  not currently used in the relevant table. The Command Generator
  (Network Management Application) reads this variable and uses the
  (non-zero) value read when creating a new row with an SNMP SET.
  When the SET is performed, the CommandResponder (agent) must
  determine whether the value is indeed still unused; Two Network
  Management Applications may attempt to create a row
  (configuration entry) simultaneously and use the same value. If
  it is currently unused, the SET succeeds and the Command
  Responder (agent) changes the value of this object, according to
an implementation-specific algorithm. If the value is in use, however, the SET fails. The Network Management Application must then re-read this variable to obtain a new usable value.

An OBJECT-TYPE definition using this SYNTAX MUST specify the relevant table for which the object is providing this functionality.

SYNTAX   Unsigned32 (0..4294967295)

-- Top-Level Components of this MIB.

mplsFTNNotifications OBJECT IDENTIFIER ::= { mplsFTNMIB 0 }
mplsFTNObjects       OBJECT IDENTIFIER ::= { mplsFTNMIB 1 }
mplsFTNConformance   OBJECT IDENTIFIER ::= { mplsFTNMIB 2 }

-- FTN table.

mplsFTNIndexNext OBJECT-TYPE
SYNTAX              MplsFTNIndexIntegerNextFree
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
"This object contains the next appropriate value to be used for mplsFTNIndex when creating entries in the mplsFTNTable. If the number of unassigned entries is exhausted, this object MUST return a value of 0. To obtain the mplsFTNIndex value for a new entry, the manager must first issue a management protocol retrieval operation to obtain the current value of this object. The agent should modify the value to reflect the next unassigned index after each retrieval operation. After a manager retrieves a value the agent will determine through its local policy when this index value will be made available for reuse."
 ::= { mplsFTNObjects 1 }

mplsFTNTable  OBJECT-TYPE
SYNTAX          SEQUENCE OF MplsFTNEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
"This table contains the currently defined FTN entries. This table allows FEC to NHLFE mappings to be specified. Each entry in this table defines a rule to be applied to incoming packets (on interfaces that the FTN entry is activated on using mplsFTNMapTable) and an action to be taken on matching packets (mplsFTNActionPointer). The MplsFTNTable provides a 5-tuple matching and allows addresses, port ranges and the exp bits to be specified."
The action pointer points at either an MPLS-LSR MIB mplsXCEntry when the NHLFE entry is a non-TE LSP, or it points at an mplsTunnelEntry in the MPLS-TE MIB if we wish to make the NHLFE the start of a TE tunnel.

REFERENCE
"Srinivansan, C., and A. Viswanathan, T. Nadeau, MPLS Label Switch Router Management Information Base Using SMIv2, draft-ietf-mpls-lsr-mib-09.txt


::=  { mplsFTNObjects  2 }

mplsFTNEntry  OBJECT-TYPE
SYNTAX          MplsFTNEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
"Each entry represents one FTN entry which defines a rule to compare against incoming packets and an action to be taken on matching packets."
INDEX { mplsFTNIndexIndex }
::=  { mplsFTNTable 1 }

MplsFTNEntry  ::=  SEQUENCE {
  mplsFTNIndexIndex          MplsFTNIndex,
  mplsFTNRowStatus           RowStatus,
  mplsFTNDescr               SnmpAdminString,
  mplsFTNMask                BITS,
  mplsFTNAddrType            InetAddressType,
  mplsFTNSourceAddrMin       InetAddress,
  mplsFTNSourceAddrMax       InetAddress,
  mplsFTNDestAddrMin         InetAddress,
  mplsFTNDestAddrMax         InetAddress,
  mplsFTNSourcePortMin       InetPortNumber,
  mplsFTNSourcePortMax       InetPortNumber,
  mplsFTNDestPortMin         InetPortNumber,
  mplsFTNDestPortMax         InetPortNumber,
  mplsFTNProtocol            INTEGER,
  mplsFTNActionType          INTEGER,
  mplsFTNActionPointer       RowPointer,
  mplsFTNExpBits             Unsigned32,
  mplsFTNStorageType         StorageType
}

mplsFTNIndexIndex  OBJECT-TYPE
SYNTAX          MplsFTNIndex
MAX-ACCESS      not-accessible
STATUS current
DESCRIPTION "Unique index for the this entry."
 ::= {mplsFTNEntry 1 }

mplsFTNRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Used for controlling the creation and deletion of this row. All writable objects in this row may be modified at any time. Setting this variable to 'destroy' when the MIB contains one or more RowPointers pointing to it results in destruction being delayed until the row is no longer used."
 ::= {mplsFTNEntry 2 }

mplsFTNDescr OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-create
STATUS current
DESCRIPTION "The description of this FTN entry. Due to the arbitrary indexing of this table, this object should contain some meaningful text that an operator could use to further distinguish entries in this table."
 ::= {mplsFTNEntry 3 }

mplsFTNMask OBJECT-TYPE
SYNTAX BITS {
  sourceAddr(0),
  destAddr(1),
  sourcePort(2),
  destPort(3),
  protocol(4),
  expBits(5)
}
MAX-ACCESS read-create
STATUS current
DESCRIPTION "This bit map indicates which of the fields described next, namely source address range, destination address range, source port range, destination port range, IPv4 layer-4 protocol or Ipv6 next header field and exp bits is active for this FTN entry. If a particular bit is inactive (i.e., set to zero) then the corresponding field in the packet is ignored for comparison purposes."
 ::= {mplsFTNEntry 4 }
mplsFTNAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"Type of IP packet that this entry will match against. Values unknown(0) and dns(16) are disallowed. All source and destination IP address types MUST use the same address type."
DEFVAL { ipv4 }
::= { mplsFTNEntry 5 }

mplsFTNSourceAddrMin OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The lower end of source address range.
The IP address type for this object MUST use the mplsFTNAddrType object defined for this row."
::= { mplsFTNEntry 6 }

mplsFTNSourceAddrMax OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The upper end of source address range."
::= { mplsFTNEntry 7 }

mplsFTNDestAddrMin OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The lower end of destination address range.
The IP address type for this object MUST use the mplsFTNAddrType object defined for this row."
::= { mplsFTNEntry 8 }

mplsFTNDestAddrMax OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The upper end of destination address range version.
The IP address type for this object MUST
use the mplsFTNAddrType object defined for this row.
::= { mplsFTNEntry 9 }

mplsFTNSourcePortMin OBJECT-TYPE
SYNTAX             InetPortNumber
MAX-ACCESS         read-create
STATUS             current
DESCRIPTION
   "The lower end of source port range."
 ::= { mplsFTNEntry 10 }

mplsFTNSourcePortMax OBJECT-TYPE
SYNTAX             InetPortNumber
MAX-ACCESS         read-create
STATUS             current
DESCRIPTION
   "The higher end of source port range."
 ::= { mplsFTNEntry 11 }

mplsFTNDestPortMin OBJECT-TYPE
SYNTAX             InetPortNumber
MAX-ACCESS         read-create
STATUS             current
DESCRIPTION
   "The lower end of the destination port range."
 ::= { mplsFTNEntry 12 }

mplsFTNDestPortMax OBJECT-TYPE
SYNTAX             InetPortNumber
MAX-ACCESS         read-create
STATUS             current
DESCRIPTION
   "The higher end of the destination port range."
 ::= { mplsFTNEntry 13 }

mplsFTNProtocol OBJECT-TYPE
SYNTAX             Integer32 (0..65535)
MAX-ACCESS         read-create
STATUS             current
DESCRIPTION
   "The contents of the protocol ID field in the IP header."
 ::= { mplsFTNEntry 14 }

mplsFTNActionType OBJECT-TYPE
SYNTAX             INTEGER {
drop(1),          -- discard this packet
redirectLsp(2),   -- redirect into LSP
redirectTunnel(3) -- redirect into tunnel
}
The type of action to be taken on packets matching this FTN entry.

::= { mplsFTNEntry 15 }

mplsFTNActionPointer OBJECT-TYPE
SYNTAX RowPointer
MAX-ACCESS read-create
STATUS current
DESCRIPTION
'If mplsFTNACTIONType is redirectLsp(2), then this object indicates the instance of mplsXCEntry for the LSP to redirect matching packets to. If mplsFTNACTIONType is redirectTunnel(3), then this object indicates the instance of mplsTunnelEntry for the MPLS TE tunnel to redirect matching packets to. For other values of mplsFTNACTIONType the agent MUST not allow the configuration/creation of inconsistent values, and thus MUST return an appropriate error in this case. An agent MUST report the invalid or incorrect value so that a management station can determine an appropriate course of action.'

::= { mplsFTNEntry 16 }

mplsFTNExpBits OBJECT-TYPE
SYNTAX Unsigned32 (0..127)
MAX-ACCESS read-create
STATUS current
DESCRIPTION
'This object indicates the contents of the exp bits field to match incoming traffic against.'
REFERENCE

::= { mplsFTNEntry 17 }

mplsFTNStorageType OBJECT-TYPE
SYNTAX StorageType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
'The storage type for this FTN entry. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row.'

::= { mplsFTNEntry 18 }
-- End of mplsFTNTable.

-- FTN to interface mapping table.

mplsFTNMapTable OBJECT-TYPE
SYNTAX SEQUENCE OF MplsFTNMapEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table contains objects for mapping previously
defined entries in mplsFTNTable to interfaces.
This table provides the capability to activate or map FTN entries defined in mplsFTNTable to specific interfaces in the system. FTN entries are compared with incoming packets in the order in which they are applied on an interface. For this reason, this table provides a mechanism to 'insert' an FTN entry between two existing FTN entries already applied on an interface.

Using this linked-list structure, one can retrieve FTN entries in the order of application on a per-interface basis as follows:

- To determine the first FTN entry on an interface with index ifIndex perform a GETNEXT retrieval operation on mplsFTNMapIndex.ifIndex.0.0; the returned object, if one exists, is (say) mplsFTNMapIndex.ifIndex.0.n. Then the index of the first FTN entry applied on this interface is n.

- To determine the FTN entry applied after the one indexed by n perform a GETNEXT retrieval operation on mplsFTNMapIndex.ifIndex.n.0; the returned object, if one exists, is (say) mplsFTNMapIndex.ifIndex.n.m. Then the index of the next FTN entry applied on this interface is m.

Use the above steps to retrieve all the applied FTN entries on a per-interface basis in application order. Note that the number of retrieval operations is the same as the number of applied FTN entries (i.e. the minimum number of GETNEXT operations needed using any indexing scheme)."
::= { mplsFTNObjects 3 }

mplsFTNMapEntry OBJECT-TYPE
SYNTAX MplsFTNMapEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Each entry indicates the application of a particular entry as defined in mplsFTNTable on an interface. The
order of application of FTN entries on an interface is
the order in which they will be compared against
incoming packets for a match. Each entry of this table
is indexed by the interface index that the FTN entry is
applied to, with the value 0 representing all
interfaces, the index of the previous FTN entry applied
on the interface and the index of the current FTN
entry. This linked-list structure allows FTN entries to
be inserted at arbitrary positions in the list. Agents
MUST NOT allow the same FTN entries to be applied
multiple times to the same interface. Agents MUST not
allow the creation of rows in this table until the
corresponding rows are created in the mplsFTNTable. If
the corresponding row in the FTN table is destroyed,
the agent MUST destroy the corresponding entries in
this table as well. Although it is strongly recommended
that managers destroy rows in this table in such a way
that keeps the remainder of the table consistent, agents
MUST ultimately make sure that if an entry in this table is
destroyed, that inconsistencies in the table are not allowed
to occur. To this end, rows that are no longer appropriate
should be taken out of service."

INDEX {
    mplsFTNMapIndex,
    mplsFTNMapPrevIndex,
    mplsFTNMapCurrIndex
}
 ::= { mplsFTNMapTable 1 }

MplsFTNMapEntry ::= SEQUENCE {
    mplsFTNMapIndex        InterfaceIndexOrZero,
    mplsFTNMapPrevIndex    MplsFTNIndexOrZero,
    mplsFTNMapCurrIndex    MplsFTNIndex,
    mplsFTNMapLastChanged  TimeStamp,
    mplsFTNMapRowStatus    RowStatus,
    mplsFTNMapStorageType  StorageType
}

mplsFTNMapIndex OBJECT-TYPE
SYNTAX              InterfaceIndexOrZero
MAX-ACCESS          not-accessible
STATUS              current
DESCRIPTION
    "The interface index that this FTN entry is being
    applied to. An index with value zero is applied all
    interfaces. If this entry is created with index zero
    and other indexes are created with non-zero values,
    these entries superceede the entry with index zero."
 ::= { mplsFTNMapEntry 1 }
mplsFTNMapPrevIndex OBJECT-TYPE
  SYNTAX             MplsFTNIndexOrZero
  MAX-ACCESS         not-accessible
  STATUS             current
  DESCRIPTION
    "The index of the previous FTN entry that was applied to
    this interface. Zero indicates that this should be the
    first FTN entry in the list."
  ::=  { mplsFTNMapEntry 2 }

mplsFTNMapCurrIndex OBJECT-TYPE
  SYNTAX             MplsFTNIndex
  MAX-ACCESS         not-accessible
  STATUS             current
  DESCRIPTION
    "Index of the current FTN entry that is being applied to
    this interface."
  ::=  { mplsFTNMapEntry 3 }

mplsFTNMapLastChanged OBJECT-TYPE
  SYNTAX             TimeStamp
  MAX-ACCESS         read-only
  STATUS             current
  DESCRIPTION
    "Indicates the last time an entry was added, deleted
    or modified in this table. Management stations should
    consult this object to determine if the table has
    been modified and requires their attention."
  ::=  { mplsFTNMapEntry 4 }

mplsFTNMapRowStatus OBJECT-TYPE
  SYNTAX             RowStatus
  MAX-ACCESS         read-create
  STATUS             current
  DESCRIPTION
    "Used for controlling the creation and deletion of this
    row. All writable objects in this row may be
    modified at any time. Setting this variable to
    'destroy' when the MIB contains one or more RowPointers
    pointing to it results in destruction being
    delayed until the row is no longer used."
  ::=  { mplsFTNMapEntry 5 }

mplsFTNMapStorageType OBJECT-TYPE
  SYNTAX             StorageType
  MAX-ACCESS         read-create
  STATUS             current
  DESCRIPTION
    "The storage type for this entry. Conceptual rows having the value 'permanent'"
need not allow write-access to any
columnar objects in the row.
::= { mplsFTNMapEntry 6 }

-- End of mplsFTNMapTable

-- FTN entry performance table

mplsFTNPerfTable OBJECT-TYPE
SYNTAX            SEQUENCE OF MplsFTNPerfEntry
MAX-ACCESS        not-accessible
STATUS            current
DESCRIPTION       "This table contains performance statistics on FTN
entries on a per-interface basis."
::= { mplsFTNObjects 4 }

MplsFTNPerfEntry OBJECT-TYPE
SYNTAX            MplsFTNPerfEntry
MAX-ACCESS        not-accessible
STATUS            current
DESCRIPTION       "Each entry contains performance information for the
specified interface and FTN entry activated/mapped to
this interface."
INDEX  { mplsFTNMapIndex, mplsFTNMapCurrIndex }
::= { mplsFTNPerfTable 1 }

MplsFTNPerfEntry ::= SEQUENCE {
  mplsFTNMatchedPackets Counter64,
  mplsFTNMatchedOctets Counter64,
  mplsFTNDiscontinuityTime TimeStamp
}

mplsFTNMatchedPackets OBJECT-TYPE
SYNTAX            Counter32
MAX-ACCESS        read-only
STATUS            current
DESCRIPTION       "Number of packets that matched the specified FTN entry
if it is applied/mapped to the specified interface.
Discontinuities in the value of this counter can occur
at re-initialization of the management system, and at
other times as indicated by the value of
mplsFTNDiscontinuityTime."
::= { mplsFTNPerfEntry 1 }

mplsFTNMatchedOctets OBJECT-TYPE
SYNTAX            Counter32
MAX-ACCESS        read-only
Number of octets that matched the specified FTN entry if it is applied/mapped to the specified interface. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of mplsFTNDiscontinuityTime.

::= { mplsFTNPerfEntry 2 }

mplsFTNDiscontinuityTime OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of sysUpTime on the most recent occasion at which any one or more of this table’s counters suffered a discontinuity. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value."

 ::= { mplsFTNPerfEntry 3 }

-- End of mplsFTNPerfTable

-- Module compliance.

mplsFTNGroups
OBJECT IDENTIFIER ::= { mplsFTNConformance 1 }

mplsFTNCompliances
OBJECT IDENTIFIER ::= { mplsFTNConformance 2 }

mplsFTNModuleFullCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"Compliance statement for agents that fully support the MPLS-FTN MIB."

MODULE IF-MIB -- The interfaces MIB, RFC2863
MANDATORY-GROUPS {
    ifGeneralInformationGroup,
    ifCounterDiscontinuityGroup
}

MODULE -- this module

-- The mandatory groups have to be implemented
-- by all LSRs. However, the configuration objects
-- may all be supported as read-only objects in
-- the case where manual configuration is unsupported.
MANDATORY-GROUPS {
    mplsFTNRuleGroup,
    mplsFTNMapGroup,
    mplsFTNPerfGroup
}

OBJECT mplsFTNAddrType
SYNTAX  InetAddressType { unknown(0), ipv4(1), ipv6(2) }
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required. An implementation is only required
to support IPv4 and IPv6 addresses."

::= { mplsFTNCompliances 1 }

-- Conformance for read-only implementations

mplsFTNModuleReadOnlyCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
    "When this MIB is implemented without support for read-create
(i.e. in read-only mode), then such an implementation can claim
read-only compliance. Such a device can then be monitored but can
not be configured with this MIB."

MODULE IF-MIB -- The interfaces MIB, RFC2863
MANDATORY-GROUPS {
    ifGeneralInformationGroup,
    ifCounterDiscontinuityGroup
}

MODULE -- this module

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

MANDATORY-GROUPS { mplsFTNPerfGroup }

GROUP mplsFTNRuleGroup
DESCRIPTION
    "This group is mandatory for implementations that support
the viewing of FTN rules."

OBJECT mplsFTNIndexNext
MIN-ACCESS not-accessible
DESCRIPTION
    "Not needed when mplsFTNTable implemented read-only."
OBJECT mplsFTNRowStatus
SYNTAX RowStatus { active(1) }
MIN-ACCESS read-only
DESCRIPTION "Write access is not required, and active is the only status that needs to be supported."

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

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

OBJECT mplsFTNAddrType
SYNTAX InetAddressType { unknown(0), ipv4(1), ipv6(2) }
MIN-ACCESS read-only
DESCRIPTION "Write access is not required. An implementation is only required to support IPv4 and IPv6 addresses."

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

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

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

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

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

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

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

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

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

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

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

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

GROUP mplsFTNMapGroup
DESCRIPTION "This group is mandatory for implementations that support the viewing of the FTN mapping table."

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

OBJECT mplsFTNMapStorageType
DESCRIPTION
   "This group is mandatory for implementations that support
   the viewing of the FTN mapping table."

::= { mplsFTNCompliances 2 }

-- Units of conformance.

mplsFTNRuleGroup OBJECT-GROUP
OBJECTS {
   mplsFTNIndexNext,
   mplsFTNRowStatus,
   mplsFTNDescr,
   mplsFTNMask,
   mplsFTNAddrType,
   mplsFTNSourceAddrMin,
   mplsFTNSourceAddrMax,
   mplsFTNDestAddrMin,
   mplsFTNDestAddrMax,
   mplsFTNSourcePortMin,
   mplsFTNSourcePortMax,
   mplsFTNDestPortMin,
   mplsFTNDestPortMax,
   mplsFTNProtocol,
   mplsFTNActionType,
   mplsFTNActionPointer,
   mplsFTNExpBits,
   mplsFTNStorageType
}
STATUS current
DESCRIPTION
   "Collection of objects needed for MPLS FTN
   configuration."
::= { mplsFTNGroups 1 }

mplsFTNMapGroup OBJECT-GROUP
OBJECTS {
   mplsFTNMapLastChanged,
   mplsFTNMapRowStatus,
   mplsFTNMapStorageType
}
STATUS current
DESCRIPTION
   "Collection of objects needed for MPLS FTN activation."
::= { mplsFTNGroups 2 }

mplsFTNPerfGroup OBJECT-GROUP
OBJECTS {  
    mplsFTNMatchedPackets,  
    mplsFTNMatchedOctets,  
    mplsFTNDiscontinuityTime  
}  
STATUS current  
DESCRIPTION  
"Collection of objects needed for MPLS FTN performance monitoring."  
::= { mplsFTNGroups 3 }  
END

9. Security Considerations

It is clear that this MIB is potentially useful for configuration. Anything that can be configured can be misconfigured, with potentially disastrous effects.

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

There are many read-write and read-create management objects defined in this MIB. Such objects are often sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. The use of SNMP Version 3 is recommended over prior versions for configuration control as its security model is improved.

There are a number of managed objects in this MIB that may contain information that may be sensitive from a business perspective, in that they may represent a customer’s service contract or the filters that the service provider chooses to apply to a customer’s ingress or egress traffic. There are no objects which are sensitive in their own right, such as passwords or monetary amounts.

10. References

10.1 Normative References

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon,  


10.2 Informative References


[RFC2570] Case, J., Mundy, R., Partain, D., and B. Stewart,


11. Authors' Addresses

Thomas D. Nadeau
Cisco Systems, Inc.
300 Apollo Drive
Chelmsford, MA 01824
Phone: +1-978-244-3051
Email: tnadeau@cisco.com
12. Acknowledgements

We would like to thank Joan Cucchiara, Mike Piecuch, and Adrien Grise for their insightful comments and additions to this draft.

13. Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

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

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.