Network Working Group
Internet Draft
Expires: January 2001

Thomas D. Nadeau
Cisco Systems, Inc.

Cheenu Srinivasan
Tachion Networks, Inc.

Arun Viswanathan
Force10 Networks, Inc.

July 2000

Multiprotocol Label Switching (MPLS) Packet Classifier Management Information Base Using SMIv2

draft-nadeau-mpls-packet-classifier-mib-01.txt

Status of this Memo

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 specifying packet classification and corresponding actions for use with Multiprotocol Label Switching (MPLS).

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 specifying packet classification s and corresponding actions for Multiprotocol Label Switching.

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

2. Terminology

TBD.

3. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

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

- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in RFC 1155 [SMIv1], RFC 1212 [SNMPv1MIBDef] and RFC 1215 [SNMPv1Traps]. The second version, called SMIv2, is described in RFC 1902 [SMIv2], RFC 1903 [SNMPv2TC] and RFC 1904 [SNMPv2Conf].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [SNMPv1]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [SNMPv2c] and RFC 1906 [SNMPv2TM]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [SNMPv2TM], RFC 2272 [SNMPv3MP] and RFC 2274 [SNMPv3USM].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [SNMPv1]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [SNMPv2PO].

- A set of fundamental applications described in RFC 2273 [SNMPv3App] and the view-based access control mechanism described in RFC 2275 [SNMPv3VACM]. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms

defined in the SMI. This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be

produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine-readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine-readable information is not considered to change the semantics of the MIB.

3.1. Object Definitions

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

4. Motivation

The primary motivation for this proposal arose from requirements in the MPLS area. In MPLS, packets belonging to a forwarding equivalency class (FEC) are associated with an LSP (ER-LSP) via the FEC-To-NHLFE (FTN) mapping [MPLS-Arch]. This mapping of packets to an LSP is made at the ingress LSR of 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. Likewise, an LSR could use its classifier to redirect packets into LSPs or TE Tunnels. With the classifier-based mapping it is possible to specify FECs finer in granularity and based on a richer set of criteria than is possible via the FIB mapping. In essence, the FTN table is a combination of the FIB and classifier.

The packet classification 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 proposal is an attempt to consolidate the various matching requirements and associated action options into a single specification, such that they satisfy existing usage and requirements as well as new ones such as those required by MPLS.

5. Outline

This MIB consists of three tables. mplsPacketClassifierTable defines the rule base against which incoming packets are matched and actions taken on matching packets. mplsPacketClassifierMapTable defines the application of these to specific interfaces. Finally, the mplsPacketClassifierPerfTable provides performance counters for every that is active, on a per-interface basis.

5.1. mplsPacketClassifierTable

This table allows packet classifiers to be specified. A packet classifier defines a rule to be applied to incoming packets on interfaces that the packet classifier is activated on and an action to be taken on matching packets. mplsPacketClassifierTable provides a standard 5-tuple matching and allows address and port ranges to be specified.

5.2. mplsPacketClassifierMapTable

This table provides the capability to activate or map packet classifiers defined in mplsPacketClassiferTable to specific interfaces in the system. Packet classifiers 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' a packet classifier between two existing packet classifiers already applied on an interface.

5.3. mplsPacketClassifierPerfTable

This table provides performance counters for each that is active on a per-interface basis. High capacity counters are provided.

6. Example

TBD.

7. The Use of RowPointer

RowPointer is a textual convention used to identify a conceptual row in an SNMP Table by pointing to one of its objects. In this MIB, in mplsPacketClassifierTable, the RowPointer object mplsPacketClassifierActionPointer indicates the LSP or tunnel to redirect packets matching a classifier to. This object SHOULD point to the first column of the appropriate conceptual row.

8. MPLS Packet Classifier MIB Definitions

```
MPLS-PACKET-CLASSIFIER-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   Integer32, Unsigned32, Counter32, experimental
      FROM SNMPv2-SMI
   MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
      FROM SNMPv2-CONF
   TEXTUAL-CONVENTION, TruthValue, RowStatus,
   StorageType, DisplayString
      FROM SNMPv2-TC
   InterfaceIndexOrZero
      FROM IF-MIB
   MplsTunnelIndex
      FROM MPLS-TE-MIB
   InetAddressIPv4, InetAddressIPv6, InetAddressType
      FROM INET-ADDRESS-MIB;

mplsPacketClassifierMIB MODULE-IDENTITY
   LAST-UPDATED "200007141200Z"  -- 14 July 2000 12:00:00 EST
   ORGANIZATION "Multiprotocol Label Switching (MPLS) Working Group"
   CONTACT-INFO
       "        Thomas D. Nadeau
        Postal: Cisco Systems, Inc.
                250 Apollo Drive
                Chelmsford, MA 01824
        Tel:    +1-978-244-3051
        Email:  tnadeau@cisco.com

                Cheenu Srinivasan
        Postal: Tachion Networks, Inc.
                Monmouth Park Corporate Center I
                Building C, 185 Monmouth Park Highway
                West Long Branch, NJ 07764
        Tel:    +1-732-542-7750 x1234
        Email:  cheenu@tachion.com

                Arun Viswanathan
```

Postal: Force10 Networks, Inc.
1440 McCarthy Blvd

                Milpitas, CA 95035
         Tel: +1-408-571-3516
         Email: arun@force10networks.com"

```
   DESCRIPTION
       "This MIB module contains managed object definitions for
        specifying packet classification for MPLS."

   -- Revision history.

   REVISION
       "200007141200Z"  -- 14 July 2000 12:00:00 EST
   DESCRIPTION
       "Initial draft version."
   REVISION
       "200003032030Z"  -- 03 March 2000 20:30:00 EST
   DESCRIPTION
       "Initial draft version."

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

-- Textual Conventions.

MplsPortAddr ::= TEXTUAL-CONVENTION
   STATUS              current
   DESCRIPTION
       "A TCP or UDP port number. Along with an IP address
        identifies a stream of IP traffic uniquely."
   SYNTAX              INTEGER (0..65535)

MplsPacketClassifierIndex ::= TEXTUAL-CONVENTION
   STATUS              current
   DESCRIPTION
       "Index for a packet classifier."
   SYNTAX              Integer32(1..2147483647)

MplsPacketClassifierIndexOrZero ::= TEXTUAL-CONVENTION
   STATUS              current
   DESCRIPTION
       "Index for a packet classifier or zero."
   SYNTAX              Integer32(0..2147483647)

-- Top level components of the MIB.

-- tables, scalars
mplsPacketClassifierObjects OBJECT IDENTIFIER
   ::= { mplsPacketClassifierMIB 1 }
```

```
-- traps
mplsPacketClassifierNotifications OBJECT IDENTIFIER
```

```
   ::= { mplsPacketClassifierMIB 2 }
-- notification prefix
mplsPacketClassifierNotifPrefix OBJECT IDENTIFIER
   ::= { mplsPacketClassifierNotifications 0 }

-- conformance
mplsPacketClassifierConformance OBJECT IDENTIFIER
   ::= { mplsPacketClassifierMIB 3 }

-- Packet classifier table.

mplsPacketClassifierIndexNext OBJECT-TYPE
   SYNTAX             MplsPacketClassifierIndexOrZero
   MAX-ACCESS         read-only
   STATUS             current
   DESCRIPTION
       "This  object contains the next appropriate value to  be
        used   for   mplsPacketClassifierIndex  when   creating
        entries  in  the  mplsPacketClassifierTable.   If   the
        number  of unassigned entries is exhausted, this object
        MUST   return   a   value  of   0.    To   obtain   the
        mplsPacketClassifierIndex 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."
   ::= { mplsPacketClassifierObjects 1 }

mplsPacketClassifierTable  OBJECT-TYPE
   SYNTAX          SEQUENCE OF MplsPacketClassifierEntry
   MAX-ACCESS      not-accessible
   STATUS          current
   DESCRIPTION
       "This table contains the currently defined packet
        classifiers."
   ::=  { mplsPacketClassifierObjects  2 }

mplsPacketClassifierEntry  OBJECT-TYPE
   SYNTAX          MplsPacketClassifierEntry
   MAX-ACCESS      not-accessible
   STATUS          current
   DESCRIPTION
       "Each entry represents one packet classifier which
        defines a rule to compare against incoming packets and
```

```
        an action to be taken on matching packets."
    INDEX { mplsPacketClassifierIndex }
    ::=  { mplsPacketClassifierTable 1 }
```

```
MplsPacketClassifierEntry  ::=  SEQUENCE {
      mplsPacketClassifierIndex
   MplsPacketClassifierIndex,
      mplsPacketClassifierRowStatus          RowStatus,
      mplsPacketClassifierDescr              DisplayString,
      mplsPacketClassifierApplied            TruthValue,
      mplsPacketClassifierMask               BITS,
      mplsPacketClassifierAddrType           InetAddressType,
      mplsPacketClassifierSourceIpv4AddrMin  InetAddressIPv4,
      mplsPacketClassifierSourceIpv6AddrMin  InetAddressIPv6,
      mplsPacketClassifierSourceIpv4AddrMax  InetAddressIPv4,
      mplsPacketClassifierSourceIpv6AddrMax  InetAddressIPv6,
      mplsPacketClassifierDestIpv4AddrMin    InetAddressIPv4,
      mplsPacketClassifierDestIpv6AddrMin    InetAddressIPv6,
      mplsPacketClassifierDestIpv4AddrMax    InetAddressIPv4,
      mplsPacketClassifierDestIpv6AddrMax    InetAddressIPv6,
      mplsPacketClassifierSourcePortMin      MplsPortAddr,
      mplsPacketClassifierSourcePortMax      MplsPortAddr,
      mplsPacketClassifierDestPortMin        MplsPortAddr,
      mplsPacketClassifierDestPortMax        MplsPortAddr,
      mplsPacketClassifierProtocol           INTEGER,
      mplsPacketClassifierActionType         INTEGER,
      mplsPacketClassifierActionPointer      RowPointer,
      mplsPacketClassifierStorageType        StorageType
   }

mplsPacketClassifierIndex   OBJECT-TYPE
   SYNTAX             MplsPacketClassifierIndex
   MAX-ACCESS         not-accessible
   STATUS             current
   DESCRIPTION
       "Unique index for the this packet classifier entry."
   ::= { mplsPacketClassifierEntry 1 }

mplsPacketClassifierRowStatus OBJECT-TYPE
   SYNTAX             RowStatus
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "For controlling the creation and deletion of this row."
   ::= { mplsPacketClassifierEntry 2 }

mplsPacketClassifierDescr   OBJECT-TYPE
   SYNTAX             DisplayString
   MAX-ACCESS         read-create
   STATUS             current
```

```
    DESCRIPTION
        "Description of this packet classifier."
    ::= { mplsPacketClassifierEntry 3 }
```

```
mplsPacketClassifierApplied OBJECT-TYPE
   SYNTAX              TruthValue
   MAX-ACCESS          read-only
   STATUS              current
   DESCRIPTION
       "Indicates whether this packet classifier has been
        applied on any interface or not."
   ::= { mplsPacketClassifierEntry 4 }

mplsPacketClassifierMask OBJECT-TYPE
   SYNTAX             BITS {
                       sourceAddr(0),
                       destAddr(1),
                       sourcePort(2),
                       destPort(3),
                       protocol(4)
                      }
   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, and
        protocol is active for this . If a particular bit is
        inactive (i.e., set to zero) then the corresponding
        field in the packet is ignored for comparison
        purposes."
   ::= { mplsPacketClassifierEntry 5 }

mplsPacketClassifierAddrType OBJECT-TYPE
   SYNTAX             InetAddressType
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "Type  of  IP  packet  that this classifier  will  match
        against. If this object has the value ipv4(1) then  the
        objects  in this entry of type InetAddressIpv6 MUST  be
        ignored by management applications. If this object  has
        the  value  ipv6(1) then the objects in this  entry  of
        type  InetAddressIpv4  MUST be  ignored  by  management
        applications."
   DEFVAL { ipv4 }
   ::= { mplsPacketClassifierEntry 6 }

mplsPacketClassifierSourceIpv4AddrMin OBJECT-TYPE
   SYNTAX             InetAddressIPv4
```

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

```
        "Lower end of source address range - IPv4 version."
    ::= { mplsPacketClassifierEntry 7 }

mplsPacketClassifierSourceIpv6AddrMin OBJECT-TYPE
    SYNTAX             InetAddressIPv6
    MAX-ACCESS         read-create
    STATUS             current
    DESCRIPTION
        " Lower end of source address range - IPv6 version."
    ::= { mplsPacketClassifierEntry 8 }

mplsPacketClassifierSourceIpv4AddrMax OBJECT-TYPE
    SYNTAX             InetAddressIPv4
    MAX-ACCESS         read-create
    STATUS             current
    DESCRIPTION
        "Upper end of source address range - IPv4 version."
    ::= { mplsPacketClassifierEntry 9 }

mplsPacketClassifierSourceIpv6AddrMax OBJECT-TYPE
    SYNTAX             InetAddressIPv6
    MAX-ACCESS         read-create
    STATUS             current
    DESCRIPTION
        "Upper end of source address range - IPv4 version."
    ::= { mplsPacketClassifierEntry 10 }

mplsPacketClassifierDestIpv4AddrMin OBJECT-TYPE
    SYNTAX             InetAddressIPv4
    MAX-ACCESS         read-create
    STATUS             current
    DESCRIPTION
        "Lower end of destination address range - IPv4 version."
    ::= { mplsPacketClassifierEntry 11 }

    mplsPacketClassifierDestIpv6AddrMin OBJECT-TYPE
    SYNTAX             InetAddressIPv6
    MAX-ACCESS         read-create
    STATUS             current
    DESCRIPTION
        "Lower end of destination address range - IPv6 version."
    ::= { mplsPacketClassifierEntry 12 }

mplsPacketClassifierDestIpv4AddrMax OBJECT-TYPE
    SYNTAX             InetAddressIPv4
    MAX-ACCESS         read-create
    STATUS             current
```

```
    DESCRIPTION
        "Upper end of destination address range - IPv4 version "
    ::= { mplsPacketClassifierEntry 13 }
```

```
mplsPacketClassifierDestIpv6AddrMax OBJECT-TYPE
   SYNTAX             InetAddressIPv6
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "Upper end of destination address range - IPv6 version "
   ::= { mplsPacketClassifierEntry 14 }

mplsPacketClassifierSourcePortMin OBJECT-TYPE
   SYNTAX             MplsPortAddr
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "Lower end of source port range."
   ::= { mplsPacketClassifierEntry 15 }

mplsPacketClassifierSourcePortMax OBJECT-TYPE
   SYNTAX             MplsPortAddr
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "Higher end of source port range "
   ::= { mplsPacketClassifierEntry 16 }

mplsPacketClassifierDestPortMin OBJECT-TYPE
   SYNTAX             MplsPortAddr
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "Lower end of the destination port range."
   ::= { mplsPacketClassifierEntry 17 }

mplsPacketClassifierDestPortMax OBJECT-TYPE
   SYNTAX             MplsPortAddr
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "Higher end of the destination port range."
   ::= { mplsPacketClassifierEntry 18 }

mplsPacketClassifierProtocol OBJECT-TYPE
   SYNTAX             INTEGER (0..65535)
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "Protocol."
```

```
   ::= { mplsPacketClassifierEntry 19 }

mplsPacketClassifierActionType OBJECT-TYPE
```

```
   SYNTAX             INTEGER {
                   drop(1),          -- discard this packet
                   redirectLsp(2),   -- redirect into specified LSP
                   redirectTunnel(3) -- redirect into specified tunnel
                      }
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "The type of action to be taken on packets matching this
        filter."
   ::= { mplsPacketClassifierEntry 20 }

mplsPacketClassifierActionPointer OBJECT-TYPE
   SYNTAX             RowPointer
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "If mplsPacketClassifierActionType is redirectLsp(2),
        then this object indicates the instance of mplsXCEntry
        for the LSP to redirect matching packets to. If
        mplsPacketClassifierActionType is redirectTunnel(3),
        then this object indicates the instance of
        mplsTunnelEntry for the MPLS tunnel to redirect
        matching packets to. For other values of
        mplsPacketClassifierActionType this object MUST be
        ignored by management applications. Agents SHOULD
        return 0 as the value of this object."
   ::= { mplsPacketClassifierEntry 21 }

mplsPacketClassifierStorageType OBJECT-TYPE
   SYNTAX             StorageType
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION
       "The storage type for this entry."
   ::= { mplsPacketClassifierEntry 22 }

-- End of mplsPacketClassifierTable.


-- Packet classifier mapping table.

mplsPacketClassifierMapTable OBJECT-TYPE
   SYNTAX              SEQUENCE OF MplsPacketClassifierMapEntry
   MAX-ACCESS          not-accessible
   STATUS              current
   DESCRIPTION
```

```
        "This table contains objects for mapping previously
         defined packet classifiers to interfaces."
    ::=  { mplsPacketClassifierObjects 3 }
```

```
mplsPacketClassifierMapEntry OBJECT-TYPE
   SYNTAX              MplsPacketClassifierMapEntry
   MAX-ACCESS          not-accessible
   STATUS              current
   DESCRIPTION
       "Each entry indicates the application of a particular
        packet classifier on an interface. The order of
        application of packet classifiers 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 classifier
        is applied to, with the value 0 representing all
        interfaces, the index of the previous packet classifier
        applied on the interface and the index of the current
        packet classifier. This linked-list structure allows
        classifiers to be inserted at arbitrary positions in
        the list. Agents MUST NOT allow the same classifiers to
        be applied multiple times to the same interface."
   INDEX  {
             mplsPacketClassifierMapIfIndex,
             mplsPacketClassifierMapPrevIndex,
             mplsPacketClassifierMapCurrIndex
           }
   ::=  { mplsPacketClassifierMapTable 1 }

MplsPacketClassifierMapEntry  ::=  SEQUENCE {
      mplsPacketClassifierMapIfIndex   InterfaceIndexOrZero,
      mplsPacketClassifierMapPrevIndex MplsPacketClassifierIndexOrZero,
      mplsPacketClassifierMapCurrIndex MplsPacketClassifierIndex,
      mplsPacketClassifierMapRowStatus RowStatus,
      mplsPacketClassifierMapStorageType
                                       StorageType
   }

mplsPacketClassifierMapIfIndex OBJECT-TYPE
   SYNTAX              InterfaceIndexOrZero
   MAX-ACCESS          read-create
   STATUS              current
   DESCRIPTION
       "Interface index that this classifier is being applied
        to. Zero represents all interfaces."
   ::= { mplsPacketClassifierMapEntry 1 }

mplsPacketClassifierMapPrevIndex OBJECT-TYPE
   SYNTAX              MplsPacketClassifierIndexOrZero
   MAX-ACCESS          read-create
```

```
    STATUS              current
    DESCRIPTION
        "Index of the previous classifier that was applied to
```

```
        this interface. Zero indicates that this should be the
        first classifier in the list."
   ::=  { mplsPacketClassifierMapEntry 2 }

mplsPacketClassifierMapCurrIndex OBJECT-TYPE
   SYNTAX              MplsPacketClassifierIndex
   MAX-ACCESS          read-create
   STATUS              current
   DESCRIPTION
       "Index of the current classifier that is being applied
        to this interface."
   ::=  { mplsPacketClassifierMapEntry 3 }

mplsPacketClassifierMapRowStatus OBJECT-TYPE
   SYNTAX              RowStatus
   MAX-ACCESS          read-create
   STATUS              current
   DESCRIPTION
       "For controlling the creation and deletion of this row."
   ::=  { mplsPacketClassifierMapEntry 4 }

mplsPacketClassifierMapStorageType OBJECT-TYPE
   SYNTAX              StorageType
   MAX-ACCESS          read-create
   STATUS              current
   DESCRIPTION
       "The storage type for this entry."
::= { mplsPacketClassifierMapEntry 5 }

-- End of packetClassifierMapTable

-- Packet classifier performance table

mplsPacketClassifierPerfTable OBJECT-TYPE
   SYNTAX              SEQUENCE OF MplsPacketClassifierPerfEntry
   MAX-ACCESS          not-accessible
   STATUS              current
   DESCRIPTION
       "This table contains performance statistics on packet
        classifiers on a per-interface basis."
   ::= { mplsPacketClassifierObjects 4 }

mplsPacketClassifierPerfEntry OBJECT-TYPE
   SYNTAX              MplsPacketClassifierPerfEntry
   MAX-ACCESS          not-accessible
   STATUS              current
   DESCRIPTION
       "Each entry contains performance information for the
```

specified interface and packet classifier
activated/mapped to this interface."

```
   INDEX  { mplsPacketClassifierMapIfIndex,
mplsPacketClassifierMapCurrIndex }
   ::=  { mplsPacketClassifierPerfTable 1 }

MplsPacketClassifierPerfEntry  ::=  SEQUENCE {
      mplsPacketClassifierMatchedPackets          Counter32,
      mplsPacketClassifierMatchedOctets           Counter32,
      mplsPacketClassifierMatchedHCPackets        Counter64,
      mplsPacketClassifierMatchedHCOctets         Counter64
   }

mplsPacketClassifierMatchedPackets OBJECT-TYPE
   SYNTAX              Counter32
   MAX-ACCESS          read-only
   STATUS              current
   DESCRIPTION
       "Number of packets that matched the specified packet
        classifier if it is applied/mapped to the specified
        interface."
   ::= { mplsPacketClassifierPerfEntry 1 }

mplsPacketClassifierMatchedOctets OBJECT-TYPE
   SYNTAX              Counter32
   MAX-ACCESS          read-only
   STATUS              current
   DESCRIPTION
       "Number of octets that matched the specified packet
        classifier if it is applied/mapped to the specified
        interface."
   ::= { mplsPacketClassifierPerfEntry 2 }

mplsPacketClassifierMatchedHCPackets OBJECT-TYPE
   SYNTAX              Counter64
   MAX-ACCESS          read-only
   STATUS              current
   DESCRIPTION
       "High-capacity counter for the number of packets that
        matched the specified packet classifier if it is
        applied/mapped to the specified interface."
   ::= { mplsPacketClassifierPerfEntry 3 }

mplsPacketClassifierMatchedHCOctets OBJECT-TYPE
   SYNTAX              Counter64
   MAX-ACCESS          read-only
   STATUS              current
   DESCRIPTION
       "High-capacity counter for the number of octets that
```

```
        matched the specified packet classifier if it is
        applied/mapped to the specified interface."
    ::= { mplsPacketClassifierPerfEntry 4 }
```

```
-- End of mplsPacketClassifierPerfTable

-- Module compliance.

mplsPacketClassifierGroups
   OBJECT IDENTIFIER ::= { mplsPacketClassifierConformance 1 }

mplsPacketClassifierCompliances
   OBJECT IDENTIFIER ::= { mplsPacketClassifierConformance 2 }

mplsPacketClassifierModuleCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
       "Compliance statement for agents that support  the  MPLS
        Packet Classifier MIB."

MODULE -- this module

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

   MANDATORY-GROUPS {
      mplsPacketClassifierRuleGroup,
      mplsPacketClassifierMapGroup
   }

   GROUP mplsPacketClassifierHCPerfGroup
   DESCRIPTION
       "This  group  is  mandatory for those perf  entries  for
        which  the  object  mplsPacketClassifierMatchedHCOctets
        and  mplsPacketClassifierMatchedHCPackets  wrap  around
        too quickly."

   ::= { mplsPacketClassifierCompliances 1 }

-- Units of conformance.
mplsPacketClassifierRuleGroup OBJECT-GROUP
   OBJECTS {
      mplsPacketClassifierIndexNext,
      mplsPacketClassifierRowStatus,
      mplsPacketClassifierDescr,
      mplsPacketClassifierApplied,
      mplsPacketClassifierMask,
      mplsPacketClassifierAddrType,
      mplsPacketClassifierSourceIpv4AddrMin,
```

```
      mplsPacketClassifierSourceIpv6AddrMin,
      mplsPacketClassifierSourceIpv4AddrMax,
```

```
      mplsPacketClassifierSourceIpv6AddrMax,
      mplsPacketClassifierDestIpv4AddrMin,
      mplsPacketClassifierDestIpv6AddrMin,
      mplsPacketClassifierDestIpv4AddrMax,
      mplsPacketClassifierDestIpv6AddrMax,
      mplsPacketClassifierSourcePortMin,
      mplsPacketClassifierSourcePortMax,
      mplsPacketClassifierDestPortMin,
      mplsPacketClassifierDestPortMax,
      mplsPacketClassifierProtocol,
      mplsPacketClassifierActionType,
      mplsPacketClassifierActionPointer,
      mplsPacketClassifierStorageType
   }
   STATUS current
   DESCRIPTION
       "Collection   of  objects  needed  for  MPLS  classifier
        configuration and monitoring."
   ::= { mplsPacketClassifierGroups 1 }

mplsPacketClassifierMapGroup OBJECT-GROUP
   OBJECTS {
      mplsPacketClassifierMapIfIndex,
      mplsPacketClassifierMapPrevIndex,
      mplsPacketClassifierMapCurrIndex,
      mplsPacketClassifierMapRowStatus,
      mplsPacketClassifierMapStorageType
   }
   STATUS current
   DESCRIPTION
       "Collection   of  objects  needed  for  MPLS  classifier
        configuration and monitoring."
   ::= { mplsPacketClassifierGroups 2 }

mplsPacketClassifierPerfGroup OBJECT-GROUP
   OBJECTS {
      mplsPacketClassifierMatchedPackets,
      mplsPacketClassifierMatchedOctets
   }
   STATUS current
   DESCRIPTION
       "Collection of objects needed for MPLS packet classifier
        performance monitoring."
   ::= { mplsPacketClassifierGroups 3 }

mplsPacketClassifierHCPerfGroup OBJECT-GROUP
   OBJECTS {
```

```
      mplsPacketClassifierMatchedHCPackets,
      mplsPacketClassifierMatchedHCOctets
   }
```

```
   STATUS current
   DESCRIPTION
       "Collection of objects needed for MPLS packet classifier
        performance   monitoring   when   using   high-capacity
        counters."
   ::= { mplsPacketClassifierGroups 4 }

-- End of MPLS-PACKET-CLASSIFIER-MIB

END
```

9. Security Considerations

It is clear that this MIB can be used for configuration of certain objects, and anything that can be configured can be incorrectly configured, with potentially disastrous results.

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

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

SNMPv1 or SNMPv2 are by themselves not a secure environment. Even if the network itself is secure (for example by using IPSec [IPSEC]), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB. It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [SNMPv3USM] and the View-based Access Control [SNMPv3VACM] is recommended. It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

There are a number of managed objects in this MIB that may contain

information that may be sensitive from a business perspective, in that they represent a customer's interface to the MPLS network.

Allowing uncontrolled access to these objects could result in malicious and unwanted disruptions of network traffic or incorrect configurations for these customers. There are no objects that are particularly sensitive in their own right, such as passwords or monetary amounts.

10. References

[MPLSArch] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", Internet Draft <draft-ietf-mpls-arch-03.txt>, February 1999.

[MPLSFW] Callon, R., Doolan, P., Feldman, N., Fredette, A., Swallow, G., and A. Viswanathan, "A Framework for Multiprotocol Label Switching", Internet Draft <draft-ietf-mpls-framework-02.txt>, November 1997.

[LSRMIB] Srinivasan, C., Viswanathan, A. and T. Nadeau, "MPLS Label Switch Router Management Information Base Using SMIv2", Internet Draft <draft-ietf-mpls-lsr-mib-02.txt>, September 2000.

[TEMIB] Srinivasan, C., Viswanathan, A. and Nadeau, T., "MPLS Traffic Engineering Management Information Base Using SMIv2", Internet Draft <draft-ietf-mpls-te-mib-02.txt>, September 2000.

[LDPMIB] Cucchiara, J., Sjostrand, H., and J. Luciani, " Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)", Internet Draft <draft-ietf-mpls-ldp-mib-05.txt>, August 1998.

[LblStk] Rosen, E., Rekhter, Y., Tappan, D., Farinacci, D., Federokow, G., Li, T., and A. Conta, "MPLS Label Stack Encoding", Internet Draft <draft-ietf-mpls-label-encaps-03.txt>, September 1998.

[RSVPTun] Awaduche, D., Berger, L., Der-Haw, G., Li, T., Swallow, G., and V. Srinivasan, "Extensions to RSVP for LSP Tunnels", Internet Draft <draft-mpls-rsvp-lsp-tunnel-00.txt>, November 1998.

[CRLDP] Andersson, L., Fredette, A., Jamoussi, B., Callon, R., Doolan, P., Feldman, N., Gray, E., Halpern, J., Heinenan, J., Kilty, T., Malis, A., Girish, M., Sundell, K., Vaananen, P., T. Worster, Wu, L., and

Dantu, R., "Explicit Routing Over LDP Specification",
Internet Draft <draft-jamoussi-mpls-cr-ldp-00.txt>,

November 1998.

[Assigned] Reynolds, J., and J. Postel, "Assigned Numbers", RFC 1700, October 1994.

[SNMPArch] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2271, January 1998.

[SMIv1] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, May 1990.

[SNMPv1MIBDef] Rose, M., and K. McCloghrie, "Concise MIB Definitions", RFC 1212, March 1991.

[SNMPv1Traps] M. Rose, "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[SMIv2] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.

[SNMPv2TC] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, SNMP Research, Inc., Cisco Systems, Inc., January 1996.

[SNMPv2Conf] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.

[SNMPv1] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1157, May 1990.

[SNMPv2c] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[SNMPv2TM] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

[SNMPv3MP] Case, J., Harrington D., Presuhn R., and B. Wijnen,

"Message Processing and Dispatching for the Simple
Network Management Protocol (SNMP)", RFC 2272, January

1998.

[SNMPv3USM] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2274, January 1998.

[SNMPv2PO] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[SNMPv3App] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2273, January 1998

[SNMPv3VACM] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, January 1998

11. Authors' Addresses

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

Cheenu Srinivasan
Tachion Networks, Inc.
185 Monmouth Park Highway
West Long Branch, NJ 07764
Phone: +1-732-542-7750 x1234
Email: cheenu@tachion.com

Arun Viswanathan
Force10 Networks, Inc.
1440 McCarthy Blvd
Milpitas, CA 95035
Phone: +1-408-571-3516
Email: arun@force10networks.com

12. Full Copyright Statement