Network Working Group
Internet Draft
Expires: Aug 2001

Dave Danenberg
Scott C. Park
Litchfield Communications, Inc.

Thomas D. Nadeau
Cisco Systems, Inc.

Andrew G. Malis
Vivace Networks, Inc.

February 22 2001

SONET/SDH Circuit Emulation Service Over MPLS (CEM) Management Information Base Using SMIv2

draft-danenberg-sonet-ces-mpls-mib-00.txt

Status of this Memo

1 Abstract

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

Table of Contents

2 Introduction

This document describes a model for managing encapsulated time division multiplexed (TDM) digital signals for transmission over a packet-oriented MPLS network.

This document is closely related to [CEM], which describes a circuit emulation header used to encapsulate TDM signals and provide the Circuit Emulation Service over MPLS (CEM). This document is also related to [TRANS and ENCAP], describing the transport and encapsulation of Layer 2 circuits over MPLS, respectively.

The model for CEM management is a MIB. The CEM MIB described in this document works closely with the MIBs described in [TEMIB and LSRMIB].

Together, [TEMIB and LSRMIB], describe the modeling of an MPLS Tunnel, and a Tunnel's underlying cross-connects. In the spirit of the [IFMIB], a CEM connection will be a virtual connection (VC), and will therefore not be represented in the ifTable.

There are functionalities introduced here that are not discussed in

[CEM, ENCAP, or TRANS]. So consider them as points of discussion for now. For example, introduced here is the concept of switching CEM VCs

between Primary and Backup MPLS Tunnels. Considering the speeds of CEM VCs, there is likely a requirement for automatic protection switching (APS) for tunnels carrying CEM traffic. CEM defects will be

used as input to CEM APS decisions. It is for further study to use other mechanisms for CEM APS (see draft-chang-mpls-path-protection-02.txt for other work in this area).

CEM dynamic bandwidth allocation (DBA) is also introduced in this MIB. CEM DBA will send packets with only a CEM Header (i.e., no payload) that will signal local conditions (such as AIS and un-equipped) when user traffic is not present. The remote CEM will play out "canned" SONET payloads when DBA is signaled.

CEM is currently designed to carry SONET paths as a "structured" adaptation (see Terminology). "Unstructured" CEM is for future consideration. The CEM MIB will reference SONET paths as modeled within [SONETMIB].

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

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

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 [BCP14].

3 Terminology

This document uses terminology from the document describing the MPLS architecture [MPLSArch]. A Label Switched Path (LSP) is modeled as described in [LSRMIB and TEMIB] via a series of cross-connects through 1 or more Label switch routers (LSR).

CEM terminology comes from the CEM draft that describes a mechanism for transporting time division multiplexed (TDM) digital signals over a packet-oriented MPLS network. The mechanism outlined in the

CEM draft terminates the SONET section and line overhead and then breaks the SONET path's Synchronous Payload Envelope (SPE) into fragments for transmission over a packet-based network. A 32-bit TDM header is appended at the beginning of each fragment to provide information regarding where the SPE begins within the packet stream, a sequence number, and pointer adjustment information.

"Adaptation" refers to the method of adapting a "foreign" communications protocol such that it can be carried by a "native" protocol. In this case, the foreign protocol is SONET/SDH and the native protocol is MPLS.

"Outbound" references the traffic direction where a SONET path's payload (SPE) is received, adapted to MPLS, assigned a VC label, and sent into the MPLS network.

Conversely, "inbound" is the direction where packets are received from the MPLS network, packet payloads are reassembled back into an SPE, and inserted as a SONET path into the SONET section and line.

Since A SONET path is bi-directional and symmetrical, it uses the same SONET time-slot, SONET width, MPLS packet size, and VC Label for

outbound and inbound traffic.

CEM will normally transmit into an originating "head" end of a Tunnel

LSP, and receive from a terminating "tail" end a Tunnel LSP. A CEM connection typically uses a VC (virtual connection) Label within a Tunnel Label [TRANS]. Multiple CEM VCs each with a unique MPLS VC Label and similar traffic engineering requirements can share the same

MPLS Tunnel. For Layer 2 transport over MPLS, the Tunnel Label is known as the "outer" Label, while the VC Label is known as the "inner" Label. An exception to this is with adjacent LSRs. In this case, there is an option for CEM VCs to connect directly without an outer Label.

VCs can be configured to switch to a "Backup Tunnel". The active Tunnel may be referred to as "In-service", while inactive Tunnels are

"Standby".

4 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 2574 [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 2575 [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.

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 Feature Checklist

The Circuit Emulation over MPLS MIB (CEM-MIB) is designed to satisfy the following requirements and constraints:

- The MIB supports manually configured CEM VCs. Although, the VC ID parameters needed for LDP are contained within this MIB. Adaptation circuits (like CEM VCs) configured via any MPLS signaling protocol are for future study.

- The MIB supports point-to-point CEM connections. Point-to-multipoint connections are for future study.

- The MIB establishes the adaptation connection by referencing the SONET path (within the ifTable [IFMIB]) to be adapted, the VC (inner)

Label, and the two Tunnels [TEMIB] that carry the bi-directional SONET path.

- The MIB configures the Tunnel and VCs: for Tunnel APS.

- The MIB configures the connection: name, packet length, error actions, etc.

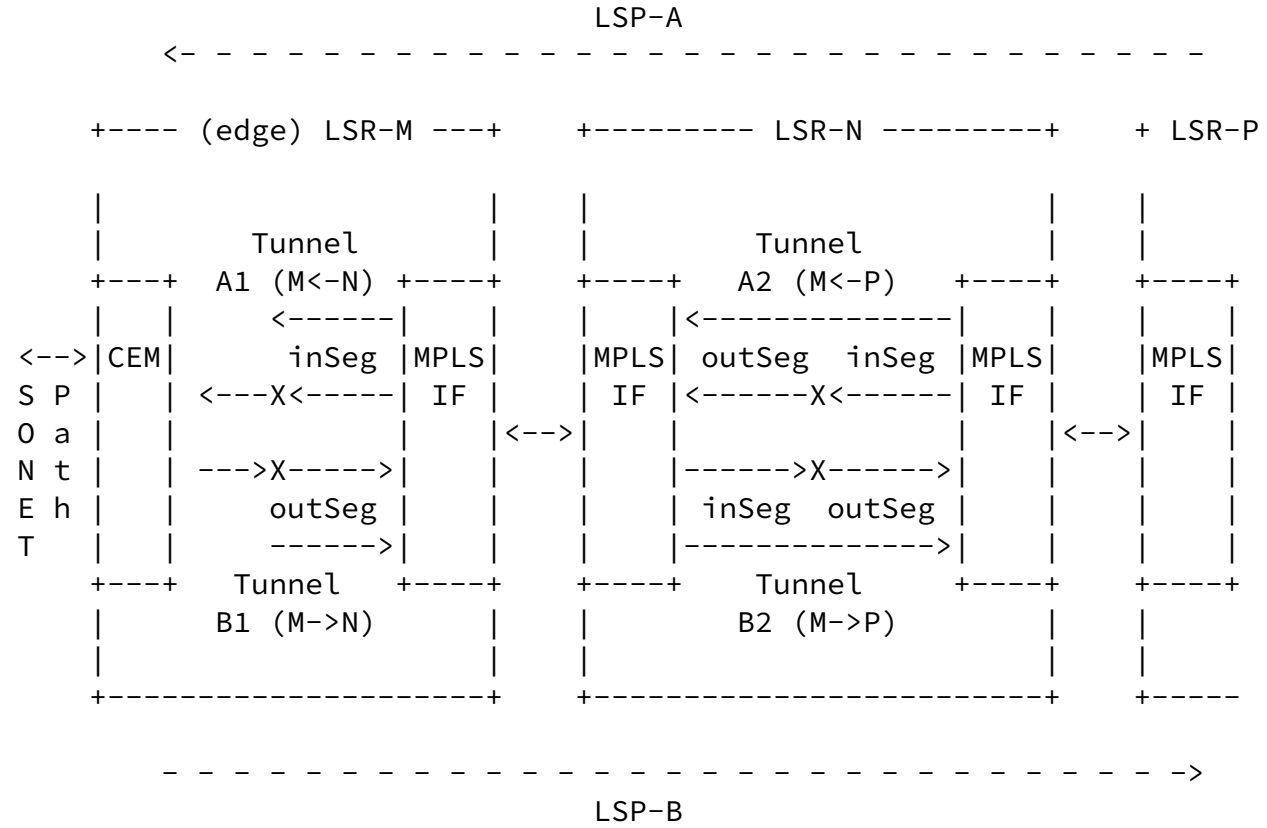
- The MIB reports: operational state, packet counts, error counts, etc.

6 CEM MIB usage

6.1 How Tunnels and Segments work with CEM

The following sections do not cover setting up a full connection

across an MPLS network. They cover the configuration of the edge LSR - that is to say, the LSR providing the CEM function. Since Tunnels are uni-directional, a pair of Tunnels must be configured (one for inbound, one for outbound). The following graphic depicts a CEM VC that originates and terminates at LSR-M. It uses LSPs A and B formed by Tunnels Ax and Bx continuing through LSR-N to LSR-P. The concatenations of Tunnels create the LSPs. Note: since the CEM cross-connects terminate and originate at LSR-M, the in and out segments are not in tandem pairs (as they are in the transit LSR-N), this is per [LSRMIB]. Note: 'X' denotes a Tunnel's cross-connect.



6.2 Summary of CEM MIB

- The CEM SONET extension (mplsCemSonetPathExtTable) is used to indicate the time slot of the SONET path to be adapted.

- The CEM VC Table (mplsCemVcTable) is used for associating a SONET path with a pair of MPLS Tunnels (inbound and outbound). Its many objects are used to control VCs.

- The CEM Performance Table (mplsCemVcPerfTable) is an augmentation of the mplsCemVcTable and contains many objects for monitoring VCs.

- The CEM Mapping Table (mplsCemMappingTable) is used to map the inbound Tunnel and VC Label to the Circuit ID. The Circuit ID is the index to an entry in the mplsCemVcTable.

- The Tunnel Extension Table (mplsCemTunnelExtTable) is used to

configure any Tunnel switchover parameters.

- The Tunnel Extension Performance Table (mplsCemTunnelExtPerfTable) is used to monitor indications not available in the mplsTunnelPerfTable.

6.3 CEM configuration Step by Step

Configuring a CEM VC and a pair of Tunnels (at LSR-M) involves the following steps.

First configure the Tunnels and their cross-connects:

- Follow steps as defined in [TEMIB] to configure a Tunnel.

- Follow steps as defined in [LSRMIB] to set up the outbound cross-connect for that Tunnel. As this is the Tunnel origination (head-end), the cross-connect will not reference an InSegment and the ingress LSR ID will be set to that of the local (edge) LSR.

- Set up the inbound cross-connect per [LSRMIB]. As this is the Tunnel termination (tail-end), the cross-connect will not reference an OutSegment and the egress LSR ID will be set to that of the local (edge) LSR.

Configure the SONET parameters:

- Set the SONET path width in the sonetPathCurrentTable [SONETMIB].

- Set the SONET path starting time slot in the mplsCemSonetPathExtTable.

Configure the CEM VC:

- Create an entry in the mplsCemVcTable. The first index is obtained from the agent, the second is the VC instance. (Note: there may be multiple instances of an mplsCemVcTable entry for use with Backup Tunnels.) Now bind this entry to the SONET ifIndex [SONETMIB], the VC

Label, and the inbound and outbound tunnels [TEMIB] by setting those objects.

- Set other parameters in the mplsCemVcTable (packet length, etc.).

- Although MPLS signaling of CEM is outside the scope of this document, LDP parameters are defined in the mplsCemVcTable: VC ID length, VC group ID, and optional VC parameters (the VC Circuit ID can be the mplsCemVcTable index, the VC Type for CEM is specified in [TRANS]).

- Use the VC Label to create an entry in the mplsCemMappingTable. This table is indexed by the inbound Tunnel's indexes plus the VC Label. It associates the inbound Tunnel and VC Label to the mplsCemVcTable entry.

- Once a CEM VC is operational, the mplsCemVcPerfTable is used to monitor the various counts, indicators, and conditions of the VC.

Advanced CEM configuration (mplsCemVcTable and mplsCemTunnelExtTable):

- VC backup. In the mplsCemVcTable, as part of CEM APS, set the criteria for switching this VC to the Backup Tunnel.

- Tunnel Table. In the mplsCemTunnelExtTable, an entry is created here for each entry in the mplsCemMappingTable. This mplsCemTunnelExtTable table is indexed with the same indexes as the inbound Tunnel [TEMIB].

- Tunnel Backup. As with VC backup, there are objects for setting Tunnel Backup switchover criteria. For Tunnels, they are in the mplsCemTunnelExtTable. An entry is created in this table by the agent

for every entry in the mplsCemMappingTable, and is indexed as the inbound Tunnel is indexed [TEMIB]. Backup Tunnel switchovers will switch all VCs in bulk. It is for this reason that care must be taken

if Tunnel Backup and VC Backup mechanisms are both configured.

Note: CEM VCs are always configured with a pair of Tunnels (inbound and outbound). When a second set of CEM VC and Backup Tunnel is configured, then there will be a total of 4 Tunnels involved. To reduce complexities, it is recommended to design the agent's APS mechanism to switch completely from one CEM VC/Tunnel set to another.

For example, do not use an inbound Primary Tunnel with an outbound Backup Tunnel. To mix Primary and Backup Tunnels is for future study.

7 Example of CEM Setup

In this section we provide an example of using the MIB objects described in section 8 to set up a CEM VC. While this example is not meant to illustrate every permutation of the MIB, it is intended as an aid to understanding some of the key concepts. It is meant to be read after going through the MIB itself. See [LSRMIB] and [TEMIB] for

an example of setting up Tunnels, their segments and cross-connects.

First configure the SONET path width, starting time-slot, and associated CEM VC. In this case, an STS-3c starts at SONET time slot 1 (and is evenly distributed within the SONET frame). The ifIndex for

both the sonetPathCurrentEntry and mplsCemSonetPathExtTable is 23.

```
In sonetPathCurrentEntry:
{
   sonetPathCurrentWidth          = 3,
   sonetPathCurrentStatus
   ...
   ...
}
```



```
In mplsCemSonetPathExtTable:
{
   mplsCemSonetPathExtVcIndex      = 1
   mplsCemSonetPathExtTimeSlot     = 1
}
```

Then create a CEM configuration entry in mplsCemVcTable. The indexes are Index and Instance. Set PathIfIndex and VcLabel. The tunnel pointers [TEMIB] contain LSR IDs (normally IP addresses) as indexes.

```
In mplsCemVcTable:
{
   mplsCemVcIndex               = 1
   mplsCemVcInstance            = 1
   mplsCemVcSonetPathIfIndex    = 23
   mplsCemVcVcLabel             = 55
   mplsCemVcPriority            = 1
   mplsCemVcName                = "My Tunnel",
   mplsCemVcDescr               = "Here to there"
   mplsCemVcCreateTime          = Feb 7 2001
```

```
   mplsCemVcInTunnelIndex       = 1
   mplsCemVcInTunnelInstance    = 1
   mplsCemVcInTunnelIngLSR      = 123.123.125.1
   mplsCemVcInTunnelLclLSR      = 123.123.126.1
   mplsCemVcOutTunnelIndex      = 2
   mplsCemVcOutTunnelInstance   = 1
   mplsCemVcOutTunnelLclLSR     = 123.123.126.1
   mplsCemVcOutTunnelEgrLSR     = 123.123.125.1
   mplsCemVcPktLength           = 500 -- payload bytes
   mplsCemVcExpBits             = 0
   mplsCemVcPktResequence       = 0
   mplsCemVcEnableDBA           = 0
   mplsCemVcJtrBfrDepth         = 50 -- packets
   mplsCemVcErrorAction         = playAllOnes
   mplsCemVcDownAction          = playAllOnes
   mplsCemVcIntegrateToDown1    =  1000 -- microseconds
   mplsCemVcIntegrateToDown2    =   500
   mplsCemVcIntegrateToUp1      = 10000
   mplsCemVcIntegrateToUp2      = 0
   mplsCemVcApsEnable           = false
   mplsCemVcApsCriteria         = 0
   mplsCemVcApsHoldOffTimer     = 0
   mplsCemVcApsRevertTimer      = 0
   mplsCemVcLdpVcIdLength       = 0
   mplsCemVcLdpGroupId          = 0
   mplsCemVcLdpOptParam1        = 0
   mplsCemVcLdpOptParam2        = 0
   mplsCemVcTrapEnable          = true
   mplsCemVcAdminStatus         = up
   mplsCemVcRowStatus           = createAndGo
}
```


Now make the association of the inbound Tunnel and VC Label to the mplsCemVcTable entry. The Tunnel Index, Instance, Ingress & Local LSR

IDs, and VC Label are the 5 indexes for this table. Since the inbound

Tunnel terminates here, its egress LSR ID should be that of the local

(edge) LSR. Creating an entry in this table causes a corresponding entry to be made in the mplsCemTunnelExt and mplsCemTunnelExtPerf tables.

In mplsCemMappingTable:

```
{
```

```
   mplsCemMappingInTunnelIndex      = 1
   mplsCemMappingInTunnelInstance   = 1
   mplsCemMappingInTunnelIngressLSR = 123.123.125.1
   mplsCemMappingInTunnelLocalLSR   = 123.123.126.1
   mplsCemMappingVcLabel            = 55
   mplsCemMappingVcIndex            = 1
   mplsCemMappingVcInstance         = 1
   mplsCemMappingRowStatus          = createAndGo
}
```

Now configure the Tunnel extension table. It has the same indexes as the mplsTunnelTable entry for the inbound Tunnel. An entry in this table was created automatically when the mplsCemMapping table entry was created.

```
In MplsCemTunnelExtTable:
{
   mplsCemTunnelExtApsEnable           = false
   mplsCemTunnelExtApsCriteria         = 0
   mplsCemTunnelExtApsHoldoffTimer     = 0
   mplsCemTunnelExtApsRevertTimer      = 0
   mplsCemTunnelExtApsViaCemVcIndex    = 0
   mplsCemTunnelExtApsViaCemVcInstance = 0
}
```

7.1 Backup Tunnels

At this time, the type of APS supported by Backup Tunnels is for uni-directional protection switching. The approach used here supports 1+1

switching where APS decisions are made on the inbound ("sink") side based on CEM errors. Since CEM packets are emitted at regular intervals and contain sequence numbers, failures can be detected quickly and reliably.

An area for further study is 1:1 switching - normally this requires a

backward defect indication. However, this may be accomplished by forcing an upstream error, when downstream errors occur. In other words, when the sink side sees CEM errors and switches to standby, it

could turn off its outbound packet stream, thereby forcing remote CEM

errors, and causing the remote to also switch to standby. Also for further study is obtaining failure indications from other than CEM errors (e.g., OAM).

If Backup Tunnels are desired, they should be configured in the mplsTunnelTable [TEMIB]. They MUST be configured in this table as entries using the same mplsTunnelIndex primary index, but with unique mplsTunnelInstances as secondary indexes. This will result in the tunnel instances being configured in a group. It is also important to note that these secondary entries may contain different LSR IDs, as well as different corresponding tunnelHopTable, tunnelARHopTable, and tunnelCHop entries. For example, if it were desired to create a tunnel with one back up path, the mplsTunnelTable would contain two entries with a primary index of 2. Their two entries would have secondary indexes (mplsTunnelInstance) of 5 and 8. Hence, there would be two entries: (2,5) and (2,8). The secondary tunnel (2,8) should have a different cross-connect (mplsTunnelXCPointer) as it should take a different path (i.e.: to protect nodes or links) through the network to circumvent a network failure. It should also be the case that the Tunnel Label differs between both instances of the tunnel.

For a CEM VC to utilize a Backup Tunnel, there will be another instance of the CEM VC within the mplsCemVcTable (forming a CEM VC group). Each group member will have the same Index, PathIfIndex, and VcLabel, but with a unique mplsCemVcInstance. Each CEM VC instance will reference a different Tunnel (within a group of Tunnels). As with the Tunnels, a primary CEM VC may be instance-1 while a backup VC may be instance-2.

The switchover criteria in the mplsCemVc and mplsCemTunnelExt tables control the switchover when groups of CEM VCs and Tunnels have been configured. If the switchover is tunnel-based (that is, bulk VC switchover is being employed), then the mplsCemVc switchover criteria

should be none (and vice-versa).

Whether Tunnel Backup decisions are VC-based or tunnel-based, it is the APS criterion that initiates switchover processing. The system may then look for other Tunnels (within the group) that have no failures (according to criteria). If a switchover is to occur and there is more than one "good" Tunnel to switch to, then the Tunnel and CEM VC instances are used to decide (lower instances have higher priority). Things causing a bad tunnel are: administrative settings (CEM VC or Tunnel admin down), CEM jitter buffer errors, CEM missing packets, and CEM header errors (other criteria are for future study).

The "revertive" setting can also initiate a switch. This parameter is

useful for non-primary Tunnels where you may want to switch back to a

Primary Tunnel even if the Backup Tunnel is OK. Finally there is "hunt" mode for initiating switches in situations where the active

Tunnel is bad, but there are no defect indications available from the

standby Tunnels.

APS timers: There is a timer for use with "revertive" to delay switching back a Primary Tunnel once that Tunnel looks good. There is

also a hold-off timer to delay switching from an active Tunnel once it looks bad.

It is possible that the bandwidth for a Backup Tunnel may be less than that of the Primary Tunnel. In this case, when VCs are switched over to the Backup, lower priority VCs may have to be dropped (set dormant). The mplsCemVcPriority object is provided for this purpose. Sorting and dropping is normally the responsibility the CAC (connection admission control) function within an LSR.

7.2 Adjacent LSRs

This section explains the how to configure CEM VCs that are connected

via adjacent LSRs.

As [TRANS] points out, a VC label within a Tunnel label may not be necessary in the Adjacent LSR case. To configure such a connection, all the steps described above would be the same except when configuring the OutSegment [LSRMIB]. In this case, PushTopLabel in the [LSRMIB] object would be set false.

8 CEM MIB Definitions

```
MPLS-CEM-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   experimental, Integer32, Counter32, Unsigned32,
   Counter64, Gauge32
      FROM SNMPv2-SMI

   MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
      FROM SNMPv2-CONF

   TEXTUAL-CONVENTION, TruthValue, RowStatus, StorageType,
   TimeStamp, DisplayString
      FROM SNMPv2-TC
```

```
   InterfaceIndex, InterfaceIndexOrZero
      FROM IF-MIB

   MplsLSPID, MplsLabel
      FROM MPLS-LSR-MIB

   MplsLsrId, MplsTunnelIndex, MplsTunnelInstanceIndex
      FROM MPLS-TE-MIB;

mplsCemMIB MODULE-IDENTITY
   LAST-UPDATED "200102211200Z"  -- 21 Feb 2001 12:00:00 EST
   ORGANIZATION "Multiprotocol Label Switching (MPLS) Working Group"
   CONTACT-INFO
       "        Dave Danenberg
        Postal: Litchfield Communications, Inc.
                76 Westbury Park Rd
                Princeton Building East
                Watertown, CT 06795
        Tel:    +1-860-945-1573 x3180
```



```
        Email:  dave_danenberg@litchfieldcomm.com

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

                Andrew G. Malis
        Postal: Vivace Networks, Inc.
                2730 Orchard Parkway
                San Jose, CA 95134
        Email:  Andy.Malis@vivacenetworks.com

                Scott Park
        Postal: Litchfield Communications, Inc.
                76 Westbury Park Rd
                Princeton Building East
                Watertown, CT 06795
        Tel:    +1-860-945-1573 x3185
        Email:  scott_park@litchfieldcomm.com

        The MPLS Working Group (email distribution mpls@uu.net)"

   DESCRIPTION
```

```
        "This MIB contains managed object definitions for Circuit
         Emulation over MPLS as in: Malis, A., Vogelsang, S., and

         Martini, L. 'SONET/SDH Circuit Emulation Service Over MPLS
         (CEM) encapsulation', Internet Draft <draft-malis-sonet-
         ces-mpls-02.txt>, February 2001. This MIB is dependant on
         the MIBs as defined by T. Nadeau, C. Srinivasan, and A.
         Viswanathan <draft-ietf-mpls-lsr-mib-07.txt> and
         <draft-ietf-mpls-te-mib-05.txt>"

   -- Revision history.
   REVISION
       "200102141200Z "  -- 14 Feb 2001 12:00:00 EST
   DESCRIPTION
      "Initial version for closed review."
   REVISION
       "200102221200Z "  -- 22 Feb 2001 12:00:00 EST
   DESCRIPTION
      "Updates, cleanup, and clarifications after review amongst
       authors."
   ::= { experimental 9999 }

-- Top level components of this MIB.

-- Traps
mplsCemNotifications OBJECT IDENTIFIER ::= { mplsCemMIB 0 }
mplsCemNotifyPrefix  OBJECT IDENTIFIER ::= { mplsCemNotifications 0
}
```



```
-- Tables, Scalars
mplsCemObjects       OBJECT IDENTIFIER ::= { mplsCemMIB 1 }

-- Conformance
-- mplsCemConformance   OBJECT IDENTIFIER ::= { mplsCemMIB 2 }


-- MPLS CEM Virtual Connection (VC) Table.

mplsCemVcIndexNext OBJECT-TYPE
   SYNTAX            Unsigned32
   MAX-ACCESS        read-only
   STATUS            current
   DESCRIPTION
       "This object contains an appropriate value to be used
        for mplsCemVcIndex when creating entries in the
        mplsCemVcTable. The value 0 indicates that no
```

```
        unassigned entries are available.  To obtain the
        value of mplsCemVcIndex for a new entry in the
        mplsCemVcTable, the manager issues a management
        protocol retrieval operation to obtain the current
        value of mplsCemVcIndex.  After each retrieval
        operation, the agent should modify the value to
        reflect the next unassigned index.  After a manager
        retrieves a value the agent will determine through
        its local policy when this index value will be made
        available for reuse."
   ::= { mplsCemObjects 1 }

mplsCemVcTable   OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsCemVcEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table specifies information for connecting
        SONET paths and MPLS Tunnels."
   ::= { mplsCemObjects 2 }

mplsCemVcEntry   OBJECT-TYPE
   SYNTAX        MplsCemVcEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
        "A row in this table represents a connection for CEM. It
         is indexed by :

          - The mplsCemVcIndex. Uniquely identifying a singular
            CEM connection or a group. If a group, individual CEM
            connections are identified by the Instance.

          - The mplsCemVcInstance."

   INDEX  { mplsCemVcIndex, mplsCemVcInstance }
```



```
      ::= { mplsCemVcTable 1 }

MplsCemVcEntry ::= SEQUENCE {
      mplsCemVcIndex              Unsigned32,
      mplsCemVcInstance           Unsigned32,
      mplsCemVcSonetPathIfIndex   InterfaceIndexOrZero,
      mplsCemVcLabel              MplsLabel,
      mplsCemVcPriority           Unsigned32,
      mplsCemVcName               DisplayString,
```

```
      mplsCemVcDescr              DisplayString,
      mplsCemVcCreateTime         TimeStamp,
      mplsCemVcUpTime             TimeTicks,

      mplsCemVcInTunnelIndex      MplsTunnelIndex,
      mplsCemVcInTunnelInstance   MplsTunnelInstanceIndex,
      mplsCemVcInTunnelIngLSR     MplsLsrId,
      mplsCemVcInTunnelLclLSR     MplsLsrId,

      mplsCemVcOutTunnelIndex     MplsTunnelIndex,
      mplsCemVcOutTunnelInstance  MplsTunnelInstanceIndex,
      mplsCemVcOutTunnelLclLSR    MplsLsrId,
      mplsCemVcOutTunnelEgrLSR    MplsLsrId,

      mplsCemVcPktLength          Unsigned32,
      mplsCemVcExpBits            Unsigned32,
      mplsCemVcPktResequence      TruthValue,
      mplsCemVcEnableDBA          BITS,
      mplsCemVcJtrBfrDepth        Unsigned32,
      mplsCemVcErrorAction        INTEGER,
      mplsCemVcDownAction         INTEGER,
      mplsCemVcIntegrateToDown1   Unsigned32,
      mplsCemVcIntegrateToDown2   Unsigned32,
      mplsCemVcIntegrateToUp1     Unsigned32,
      mplsCemVcIntegrateToUp2     Unsigned32,
      mplsCemVcApsEnable          TruthValue,
      mplsCemVcApsCriteria        BITS,
      mplsCemVcDefects            BITS,
      mplsCemVcApsHoldoffTimer    Unsigned32,
      mplsCemVcApsRevertTimer     Unsigned32,
      mplsCemVcApsStatus          INTEGER,
      mplsCemVcLdpVcIdLength      Unsigned32,
      mplsCemVcLdpGroupId         Unsigned32,
      mplsCemVcLdpOptParam1       Unsigned32,
      mplsCemVcLdpOptParam2       Unsigned32,
      mplsCemVcTrapEnable         TruthValue,
      mplsCemVcAdminStatus        INTEGER,
      mplsCemVcOperStatus         INTEGER,
      mplsCemVcRowStatus          RowStatus,
      mplsCemVcStorageType        StorageType
   }

mplsCemVcIndex OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    accessible-for-notify
```



```
   STATUS        current
```

```
    DESCRIPTION
        "Primary index for the conceptual row identifying
         a group of CEM VCs."
    ::= { mplsCemVcEntry 1 }

mplsCemVcInstance OBJECT-TYPE
    SYNTAX        Unsigned32
    MAX-ACCESS    accessible-for-notify
    STATUS        current
    DESCRIPTION
        "Uniquely identifies an instance of a CEM VC. It is
         useful to identify multiple instances for the purpose of
         backup VCs."
    ::= { mplsCemVcEntry 2 }

mplsCemVcSonetPathIfIndex OBJECT-TYPE
    SYNTAX        InterfaceIndexOrZero
    MAX-ACCESS    accessible-for-notify
    STATUS        current
    DESCRIPTION
        "This is a unique index within the ifTable. It represents
         the interface index for the SONET path. A value of zero
         indicates an ifIndex that has yet to be configured or
         has since disappeared."
    ::= { mplsCemVcEntry 3 }

mplsCemVcLabel   OBJECT-TYPE
    SYNTAX        MplsLabel
    MAX-ACCESS    accessible-for-notify
    STATUS        current
    DESCRIPTION
        "The incoming and outgoing label for this VC. This
         value is placed in the Label field of the outgoing MPLS
         shim header."
    ::= { mplsCemVcEntry 4 }

mplsCemVcPriority OBJECT-TYPE
    SYNTAX        Unsigned32
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "Used when resources on a tunnel or backup tunnel are
         overbooked. The agent (connection admission control) can
         sort CEM VC entries that share an overbooked tunnel.
         Lower mplsCemVcPriority values have higher priority.
         VCs dropped will be set 'dormant' (as indicated in
         mplsCemVcOperStatus)."
    ::= { mplsCemVcEntry 5 }

mplsCemVcName  OBJECT-TYPE
    SYNTAX        DisplayString
```

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



```
   DESCRIPTION
       "The canonical name assigned to the CEM VC. This
        name can be used to refer to the CEM VC on the
        LSRs console port."
   ::= { mplsCemVcEntry 6 }

mplsCemVcDescr OBJECT-TYPE
   SYNTAX        DisplayString
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "A textual string containing information about the
        CEM VC.  If there is no description this object
        contains a zero length string."
   ::= { mplsCemVcEntry 7 }

mplsCemVcCreateTime OBJECT-TYPE
   SYNTAX        TimeStamp
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "System time when theis CEM VC was created."
   ::= { mplsCemVcEntry 8 }

mplsCemVcUpTime  OBJECT-TYPE
   SYNTAX        TimeTicks
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of consecutive ticks this CEM VC has been 'up'
        as observed in mplsCemVcOperStatus."
   ::= { mplsCemVcEntry 9 }

-- The following 8 objects represent the indexes for the
-- inbound and outbound tunnels for this CEM VC.
mplsCemVcInTunnelIndex OBJECT-TYPE
   SYNTAX        MplsTunnelIndex
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION "Part of set of indexes for inbound tunnel"
   REFERENCE
       "Srinivasan, C., Viswanathan, A., and T. Nadeau,
        MPLS Traffic Engineering Management Information
        Base Using SMIv2 <draft-ietf-mpls-te-mib-05.txt>,
```

```
         November 2000."
   ::= { mplsCemVcEntry 10 }
mplsCemVcInTunnelInstance OBJECT-TYPE
   SYNTAX        MplsTunnelInstanceIndex
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION "Part of set of indexes for inbound tunnel"
   ::= { mplsCemVcEntry 11 }
mplsCemVcInTunnelIngLSR OBJECT-TYPE
   SYNTAX        MplsLsrId
```



```
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION "Part of set of indexes for inbound tunnel"
   ::= { mplsCemVcEntry 12 }
mplsCemVcInTunnelLclLSR OBJECT-TYPE
   SYNTAX        MplsLsrId
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION "Part of set of indexes for inbound tunnel"
   ::= { mplsCemVcEntry 13 }
mplsCemVcOutTunnelIndex OBJECT-TYPE
   SYNTAX        MplsTunnelIndex
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION "Part of set of indexes for outbound tunnel"
   ::= { mplsCemVcEntry 14 }
mplsCemVcOutTunnelInstance OBJECT-TYPE
   SYNTAX        MplsTunnelInstanceIndex
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION "Part of set of indexes for outbound tunnel"
   ::= { mplsCemVcEntry 15 }
mplsCemVcOutTunnelLclLSR OBJECT-TYPE
   SYNTAX        MplsLsrId
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION "Part of set of indexes for outbound tunnel"
   ::= { mplsCemVcEntry 16 }
mplsCemVcOutTunnelEgrLSR OBJECT-TYPE
   SYNTAX        MplsLsrId
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION "Part of set of indexes for outbound tunnel"
   ::= { mplsCemVcEntry 17 }
```

```
mplsCemVcPktLength OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This is the packet or payload length for this CEM
        VC. It is fixed and applies to inbound and
        outbound packets carrying user payload. Note: DBA
        packets have their own length and are not effected
        by this."
   ::= { mplsCemVcEntry 18 }

mplsCemVcExpBits OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This value is placed in the EXP bit field of the
```



```
        outbound MPLS shim header (with the VC Label). These
        EXP bits convey to the LSR the PHB to be applied to
        these packets."
   REFERENCE
        "Faucher, F, et al, MPLS Support of Differentiated
        Services <draft-ietf-mpls-diff-ext-08.txt> Feb 2001."
   ::= { mplsCemVcEntry 19 }

mplsCemVcPktResequence OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Reports if implementation provides packet re-sequencing.
        With this, as inbound packets are queued in the jitter
        buffer, out of order packets are re-sequenced. The
        maximum sequence number differential (order correction
        can occur within) is dependant on the depth of the
        jitter buffer. See mplsCemVcJtrBfrDepth."
   ::= { mplsCemVcEntry 20 }

mplsCemVcEnableDBA OBJECT-TYPE
   SYNTAX BITS {
         allOnesOnAis(0),
         allZerosOnUnequipped(1)
   }
   MAX-ACCESS    read-create
   STATUS        current
```

```
   DESCRIPTION
       "Any bits set here MUST enable the DBA (dynamic bandwidth
        allocation) feature for the specified condition. Setting
        allOnesOnAis will cause CEM packet payload suppression
        when AIS is detected on the associated SONET path.
        Similarly, allZerosOnUnequipped will cause payload
        suppression when the SONET path is un-equipped. During
        these conditions, CEM packets will continue to be sent,
        but with indicators set in the CEM header instructing the
        remote to play all ones or zeros onto its SONET path.
        Note: some implementations may not support this feature."
   ::= { mplsCemVcEntry 21 }

mplsCemVcJtrBfrDepth OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This setting configures the number of packet buffers
        reserved for this CEM VC. (This object would not apply
        in implementations that support CEM VC groups, but
        cannot process inbound packets on CEM VCs that are
        currently in standby.) This object essentially sets
        the maximum amount of time allowed between CEM packets
        before the jitter buffer empties. This variable should
        be set based on the SONET path width (speed) and the
```



```
        amount of delay variation expected to be introduced by
        the network. Like bandwidth, jitter buffers are likely
        to be a limited resource to be managed."
   ::= { mplsCemVcEntry 22 }

mplsCemVcErrorAction OBJECT-TYPE
   SYNTAX INTEGER {
         playAllOnes(1),
         playAllZeros(2),
         playPseudoRandom(3)
   }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "These are the actions to take when inbound packets are
        missing due to gap in sequence numbers (uncorrectable via
        available re-sequencing), jitter buffer underruns, or
        packets with bad CEM headers. These patterns are sent
        (played) on the SONET path. These settings are used for
        immediate errors and (unless the mplsCemVcDownAction is
```

```
         'none') are not in effect once the CEM VC is 'down'."
   ::= { mplsCemVcEntry 23 }

mplsCemVcDownAction OBJECT-TYPE
   SYNTAX        INTEGER {
         errorAction(1),
         playAllOnes(2),
         playAllZeros(3),
         playPseudoRandom(4)
   }
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "These are the actions to take once the CEM VC has been
        declared 'down' (as observed in mplsCemVcOperStatus).
        These patterns are sent (played) on the SONET path.
        See Integration Timers for events causing CEM VC 'down'.
        If 'errorAction' is selected, then the mplsCemVcErrorAction
        settings stay in effect even after the CEM VC is 'down'."
   ::= { mplsCemVcEntry 24 }

--
-- The following 4 timers work together to integrate (filter)
-- errors and the lack of errors on the CEM VC. Errors are:
-- missing packet, packet out of sequence, CEM header error,
-- jitter buffer error. Example of usage :
--
-- When an error occurs, Down1 and Down2 start counting, if
-- no errors occur within Down2, Down1 is cancelled. If errors
-- do occur within Down2, the Down2 timer is restarted. The CEM
-- VC is 'down' if Down1 expires - then Up1 starts counting.
-- If an error occurs within Up1, Up1 is restarted. The CEM VC
-- is 'up' if Up1 expires. Based on this, Down2 must be less
-- than Down1. Up2 timer usage is for further study.
```



```
--
-- Other usage of these integration objects is for further study.
-- For example, these objects may be moved to their own new table
-- where rows in the new table would be referenced in the CEM VC
-- table to support separate integration of the various CEM errors
-- (buffer errors, missing packets, CEM header errors).
--
mplsCemVcIntegrateToDown1 OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION   "See comment above - units are microseconds."
```

```
   ::= { mplsCemVcEntry 25 }

mplsCemVcIntegrateToDown2 OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION   "See comment above - units are microseconds."
   ::= { mplsCemVcEntry 26 }

mplsCemVcIntegrateToUp1 OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION   "See comment above - units are microseconds."
   ::= { mplsCemVcEntry 27 }

mplsCemVcIntegrateToUp2 OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION   "See comment above - units are microseconds."
   ::= { mplsCemVcEntry 28 }

mplsCemVcApsEnable OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Enables this CEM VC (as part of a group) to be used for
        APS. A CEM VC group is created when a unique CEM VC index
        (in mplsCemVcTable) has multiple instances. Setting
        mplsCemVcApsEnable to 'false' may be useful when a CEM VC
        is suspect, being debugged, or not fully configured.
        Note: some implementations may not support APS."
   ::= { mplsCemVcEntry 29 }

mplsCemVcApsCriteria OBJECT-TYPE
   SYNTAX BITS {
         cemError(0),
         cemErrorStbyOK(256),
         revertive(512),
         hunt(1024)
```



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

```
        "Bits set here represent defects and modes for switching
         the SONET path from the active CEM VC to another within
         this VC's  group. Defects include: CEM errors (jitter
         buffer errors, missing packets, bad CEM headers).
         Separating these CEM defects into individually selectable
         bits, or adding other types of defect indications (e.g.
         OAM status) is for future study.

         Modes:
         The 'cemErrorStbyOK' mode setting is useful in APS network
         configurations where CEM traffic is not present on standby
         inbound VCs. Therefore they are expected to have CEM
         errors, but are still to be considered as viable switch-
         over candidates. cemErrorStbyOK essentially masks CEM
         errors when this CEM VC is in standby.

         'revertive' mode is useful for non-preferred CEM VCs where
         you may want to switch back to a preferred VC that has no
         defects while the currently active VC also has no defects
         (see mplsCemVcApsRevertTimer). 'revertive' should NOT be
         used when standby VCs have no defect indications available.
         If 'revertive' is not set, then manual reversion is possible

         by simply setting the active CEM VC's admin status 'down',
         then 'up'. Preferred should NOT have 'revertive' set.

         'hunt' mode is useful when standby VCs offer no defect
         indications (see mplsCemVcApsHoldOffTimer). 'hunt' should
         not be used when standby VCs have available defect
         indications. Hunting assumes standby VCs are good, if not
         the down timers will determine it's bad, and hunting
         continues.

         'hunt' and 'revertive' are mutually exclusive."
    ::= { mplsCemVcEntry 30 }

mplsCemVcDefects OBJECT-TYPE
    SYNTAX BITS {
          cemError(0)
    }
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Current state of these CEM VC defects. The cemError
         indicator is a summary of any error associated with
         processing inbound CEM packets. Separately displaying
         CEM errors types (buffer errors, missing packets, and
         header errors) are for future study. Also other bits
         here may be defined here to show results of error trend
         analysis. These bits MUST be aligned with
         mplsCemVcApsCriteria bits so mplsCemVcApsCriteria can
```

```
        be used as a mask.
        Note: other defect indications (e.g. from OAM) are for
        future study."
   ::= { mplsCemVcEntry 31 }

mplsCemVcApsHoldoffTimer OBJECT-TYPE -- units are in seconds
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Used to hold-off an APS switch after the CEM VC has been
        determined 'down'. The purpose is to allow potential
        recovery schemes within lower communications layers a
        chance to recover.

        Also useful in 'hunt' mode to hold off switching to the
        next CEM VC (slows oscillation when all VCs are down)."
   ::= { mplsCemVcEntry 32 }

mplsCemVcApsRevertTimer OBJECT-TYPE -- units are in seconds
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "If this CEM VC is active and 'up', then this timer is
        used in conjunction with 'revertive' (if set above).
        The reversion would be delayed for this time."
   ::= { mplsCemVcEntry 33 }

mplsCemVcApsStatus OBJECT-TYPE
   SYNTAX INTEGER {
        active(1),
        standby(2)
   }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
        "If part of a CEM VC group, this indicates if the
         CEM VC has been selected (via APS algorithm/protocol)
         to connect the associated Tunnel to the SONET path.
         If not part of a group, then always 'active'."
   ::= { mplsCemVcEntry 34 }

mplsCemVcLdpVcIdLength OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
```

```
   STATUS        current
   DESCRIPTION
       "For use by LDP signaling"
   ::= { mplsCemVcEntry 35 }

mplsCemVcLdpGroupId OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
```



```
   STATUS        current
   DESCRIPTION
       "For use by LDP signaling"
   ::= { mplsCemVcEntry 36 }

mplsCemVcLdpOptParam1 OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "For use by LDP signaling"
   ::= { mplsCemVcEntry 37 }

mplsCemVcLdpOptParam2 OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "For use by LDP signaling"
   ::= { mplsCemVcEntry 38 }

mplsCemVcTrapEnable OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "If this object is true, generation of mplsCemVcUp
        and mplsCemVcDown traps aare enabled for this CEM VC,
        otherwise these traps are not emitted."
   DEFVAL { false }
   ::= { mplsCemVcEntry 39}

mplsCemVcAdminStatus OBJECT-TYPE
   SYNTAX   INTEGER {
                up(1),     -- ready to pass packets
                down(2),
                testing(3) -- in some test mode
   }
```

```
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "The desired operational status of this CEM VC."
   ::= { mplsCemVcEntry 40 }

mplsCemVcOperStatus OBJECT-TYPE
   SYNTAX INTEGER {
        up(1),            -- ready to pass packets
        down(2),
        testing(3),       -- in some test mode
        unknown(4),       -- status cannot be determined
        dormant(5),
        notPresent(6),    -- some component is missing
        lowerLayerDown(7) -- down due to the state of
                          -- lower layer interfaces
```



```
   }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
        "Indicates the actual operational status of this CEM VC."
   ::= { mplsCemVcEntry 41 }

mplsCemVcRowStatus OBJECT-TYPE
   SYNTAX        RowStatus
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "For creating, modifying, and deleting this row."
   ::= { mplsCemVcEntry 42 }

mplsCemVcStorageType OBJECT-TYPE
   SYNTAX        StorageType
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This variable indicates the storage type for this
        object."
   ::= { mplsCemVcEntry 43 }

-- End of MPLS CEM Virtual Connection (VC) Table


-- MPLS CEM VC Performance Table.

mplsCemVcPerfTable OBJECT-TYPE
```

```
   SYNTAX        SEQUENCE OF MplsCemVcPerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "This table provides per CEM VC performance information.
        Note: outbound errors are not being considered. It is
        assumed (at this time) that CEM packets will be forwarded
        as they are generated. i.e., it is assumed that there
        are no local outbound packet congestion issues. HC (high
        capacity) counters are needed for packet counts due to
        the high speeds expected with CEM. A SONET path of width
        48 can rollover a non-HC counter in a few minutes."
   ::= { mplsCemObjects 3 }

mplsCemVcPerfEntry OBJECT-TYPE
   SYNTAX        MplsCemVcPerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table is created by the agent for every
        mplsCemVcEntry. It is an extension to mplsCemVcEntry."

   AUGMENTS { mplsCemVcEntry }
```



```
   ::= { mplsCemVcPerfTable 1 }

MplsCemVcPerfEntry ::= SEQUENCE {
      mplsCemVcPerfTotalInPacketsHC  Counter64,
      mplsCemVcPerfTotalOutPacketsHC Counter64,
      mplsCemVcPerfDbaInPacketsHC    Counter64,
      mplsCemVcPerfDbaOutPacketsHC   Counter64,
      mplsCemVcPerfInNegPtrAdjustHC  Counter64,
      mplsCemVcPerfInPosPtrAdjustHC  Counter64,
      mplsCemVcPerfOutNegPtrAdjustHC Counter64,
      mplsCemVcPerfOutPosPtrAdjustHC Counter64,
      mplsCemVcPerfCrctHdrErrors     Counter32,
      mplsCemVcPerfUncrctHdrErrors   Counter32,
      mplsCemVcPerfMissingPkts       Counter32,
      mplsCemVcPerfPktsOoseq         Counter32,
      mplsCemVcPerfJtrBfrUnderruns   Counter32,
      mplsCemVcPerfJtrBfrOverruns    Counter32,
      mplsCemVcPerfDiscontinuityTime TimeStamp
   }

mplsCemVcPerfTotalInPacketsHC OBJECT-TYPE
   SYNTAX        Counter64
```

```
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of inbound packets received."
   ::= { mplsCemVcPerfEntry 1 }

mplsCemVcPerfTotalOutPacketsHC OBJECT-TYPE
   SYNTAX        Counter64
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of outbound packets sent."
   ::= { mplsCemVcPerfEntry 2 }

mplsCemVcPerfDbaInPacketsHC OBJECT-TYPE
   SYNTAX        Counter64
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of DBA packets received."
   ::= { mplsCemVcPerfEntry 3 }

mplsCemVcPerfDbaOutPacketsHC OBJECT-TYPE
   SYNTAX        Counter64
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of DBA packets sent."
   ::= { mplsCemVcPerfEntry 4 }

mplsCemVcPerfInNegPtrAdjustHC OBJECT-TYPE
   SYNTAX        Counter64
```



```
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of negative pointer adjustments made on the
        SONET path based on CEM pointer adjustments received."
   ::= { mplsCemVcPerfEntry 5 }

mplsCemVcPerfInPosPtrAdjustHC OBJECT-TYPE
   SYNTAX        Counter64
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of postive pointer adjustments made on the
        SONET path based on CEM pointer adjustments received."
```

```
    ::= { mplsCemVcPerfEntry 6 }

mplsCemVcPerfOutNegPtrAdjustHC OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of negative pointer adjustments seen on the
         SONET path and encoded onto sent CEM packets."
    ::= { mplsCemVcPerfEntry 7 }

mplsCemVcPerfOutPosPtrAdjustHC OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of positive pointer adjustments seen on the
         SONET path and encoded onto sent CEM packets."
    ::= { mplsCemVcPerfEntry 8 }

mplsCemVcPerfCrctHdrErrors OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of correctable CEM header errors detected on
         inbound CEM packets."
    ::= { mplsCemVcPerfEntry 9 }

mplsCemVcPerfUncrctHdrErrors OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of uncorrectable CEM header errors detected on
         inbound CEM packets."
    ::= { mplsCemVcPerfEntry 10 }

mplsCemVcPerfMissingPkts OBJECT-TYPE
    SYNTAX        Counter32
```



```
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of missing packets (as detected via CEM header
         sequence number gaps)."
    ::= { mplsCemVcPerfEntry 11 }
```

```
mplsCemVcPerfPktsOoseq OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of packets detected out of sequence (via CEM
        header sequence numbers), but were able to be re-sequenced.
        That is, the differential in sequence numbers was less
        than the jitter buffer depth. Note: some implementations
        may not support this feature (see mplsCemVcPktResequence).
        Any packets so far out of sequence that a re-sequencer
        can not correct for would be counted as missing packet."
   ::= { mplsCemVcPerfEntry 12 }

mplsCemVcPerfJtrBfrUnderruns OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of times the jitter buffer transitioned to empty."
   ::= { mplsCemVcPerfEntry 13 }

mplsCemVcPerfJtrBfrOverruns OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of packets received while jitter buffer is full."
   ::= { mplsCemVcPerfEntry 14 }

mplsCemVcPerfDiscontinuityTime 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 segment's Counter32 or
         Counter64 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."
    ::= { mplsCemVcPerfEntry 15 }

-- End MPLS CEM VC Performance Table


-- MPLS CEM Mapping Table.
```

```
mplsCemMappingTable OBJECT-TYPE
   SYNTAX           SEQUENCE OF MplsCemMappingEntry
   MAX-ACCESS       not-accessible
   STATUS           current
   DESCRIPTION
       "This table maps an inbound Tunnel/VcLabel to a CEM VC."
   ::= { mplsCemObjects 4 }

mplsCemMappingEntry OBJECT-TYPE
   SYNTAX           MplsCemMappingEntry
   MAX-ACCESS       not-accessible
   STATUS           current
   DESCRIPTION
        "A row in this table represents the inbound connection
         between a Tunnel/VcLabel and CEM VC/instance. It is
         indexed by the same 4 indexes that index the inbound
         Tunnel, then adds the VC Label as a 5th index. The Tunnel
         referenced here is in the mplsTunnelTable.

          - The mplsCemMappingInTunnelIndex uniquely identifies a
            singular Tunnel or a group. If a group, individual
            Tunnels are identified by the Instance.

          - The mplsCemMappingInTunnelInstance.

          - ID (normally an IP address) for the ingress LSR,
            mplsCemMappingInTunnelIngressLSR.

          - ID (normally an IP address) of the Local (egress) LSR,
             mplsCemMappingInTunnelLocalLSR,

          - and the VC Label, mplsCemMappingVcLabel."

   INDEX  { mplsCemMappingInTunnelIndex,
            mplsCemMappingInTunnelInstance,
            mplsCemMappingInTunnelIngressLSR,
            mplsCemMappingInTunnelLocalLSR,
            mplsCemMappingVcLabel }

      ::= { mplsCemMappingTable 1 }

MplsCemMappingEntry ::= SEQUENCE {
      mplsCemMappingInTunnelIndex       MplsTunnelIndex,
      mplsCemMappingInTunnelInstance    MplsTunnelInstanceIndex,
      mplsCemMappingInTunnelIngressLSR  MplsLsrId,
      mplsCemMappingInTunnelLocalLSR    MplsLsrId,
      mplsCemMappingVcLabel             MplsLabel,
      mplsCemMappingVcIndex             Integer32,
```

```
      mplsCemMappingVcInstance          Integer32,
      mplsCemMappingRowStatus           RowStatus,
      mplsCemMappingStorageType         StorageType
   }
```



```
mplsCemMappingInTunnelIndex OBJECT-TYPE
   SYNTAX        MplsTunnelIndex
   MAX-ACCESS    accessible-for-notify
   STATUS        current
   DESCRIPTION
       "Primary index for the conceptual row identifying
        a group of Tunnel/VcLabel to CEM VC mappings."
   ::= { mplsCemMappingEntry 1 }

mplsCemMappingInTunnelInstance OBJECT-TYPE
   SYNTAX        MplsTunnelInstanceIndex
   MAX-ACCESS    accessible-for-notify
   STATUS        current
   DESCRIPTION
       "Uniquely identifies an instance of a mapping"
   ::= { mplsCemMappingEntry 2 }

mplsCemMappingInTunnelIngressLSR  OBJECT-TYPE
   SYNTAX        MplsLsrId
   MAX-ACCESS    accessible-for-notify
   STATUS        current
   DESCRIPTION
       "Uniquely identifies an ingress LSR"
   ::= { mplsCemMappingEntry 3 }

mplsCemMappingInTunnelLocalLSR  OBJECT-TYPE
   SYNTAX        MplsLsrId
   MAX-ACCESS    accessible-for-notify
   STATUS        current
   DESCRIPTION
       "Uniquely identifies the local LSR"
   ::= { mplsCemMappingEntry 4 }

mplsCemMappingVcLabel OBJECT-TYPE
   SYNTAX        MplsLabel
   MAX-ACCESS    accessible-for-notify
   STATUS        current
   DESCRIPTION
       "Identifies a unique label on this tunnel"
   ::= { mplsCemMappingEntry 5 }
```

```
mplsCemMappingVcIndex  OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Index for entry (or group of entries, in the case of backup
        VCs) in the mplsCemVcTable. This value can also be thought
        of as the Circuit ID."
   ::= { mplsCemMappingEntry 6 }

mplsCemMappingVcInstance  OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-create
```



```
   STATUS        current
   DESCRIPTION
       "Identifies a unique member within a CEM VC group.
        (This could of course be a group of one.)"
   ::= { mplsCemMappingEntry 7 }

mplsCemMappingRowStatus OBJECT-TYPE
   SYNTAX        RowStatus
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "For creating, modifying, and deleting this row."
   ::= { mplsCemMappingEntry 8 }

mplsCemMappingStorageType OBJECT-TYPE
   SYNTAX        StorageType
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "This variable indicates the storage type for this
        object."
   ::= { mplsCemMappingEntry 9 }

-- End of MPLS CEM Mapping Table


-- MPLS Tunnel Extension Table

mplsCemTunnelExtTable OBJECT-TYPE
   SYNTAX             SEQUENCE OF MplsCemTunnelExtEntry
   MAX-ACCESS         not-accessible
   STATUS             current
   DESCRIPTION
```

```
        "This table is an extension to the mplsTunnelTable. This
         table exists to configure parameters useful for CEM APS."
    ::= { mplsCemObjects 5 }

mplsCemTunnelExtEntry OBJECT-TYPE
    SYNTAX              MplsCemTunnelExtEntry
    MAX-ACCESS          not-accessible
    STATUS              current
    DESCRIPTION
         "A row in this table represents a Tunnel used for CEM VCs.
          The Tunnel referenced here is in the mplsTunnelTable.
          It is indexed by the same 4 indexes of the inbound Tunnel:

           - The mplsCemTunnelExtIndex uniquely identifies a
             singular Tunnel or a group. If a group, individual
             Tunnels are identified by the Instance.

           - The mplsCemTunnelExtInstance.

           - ID (normally an IP address) for the ingress LSR,
             mplsCemTunnelExtIngressLSR.
```



```
           - ID (normally an IP address) of the Local (egress) LSR,
              mplsCemTunnelExtLocalLSR. In the mplsTunnelTable, this
              index would be referred to as the Egress LSR ID. Since
              CEM is always at a tunnel termination, the egress LSR
              is always the local (edge) LSR.

          The local agent creates an entry here for every entry
          created in the mplsCemMappingTable."

    INDEX  { mplsCemTunnelExtIndex,
             mplsCemTunnelExtInstance,
             mplsCemTunnelExtIngressLSR,
             mplsCemTunnelExtLocalLSR }

       ::= { mplsCemTunnelExtTable 1 }

MplsCemTunnelExtEntry::= SEQUENCE {
      mplsCemTunnelExtApsEnable           TruthValue,
      mplsCemTunnelExtApsCriteria         BITS,
      mplsCemTunnelExtDefects             BITS,
      mplsCemTunnelExtApsHoldoffTimer     Unsigned32,
      mplsCemTunnelExtApsRevertTimer      Unsigned32,
      mplsCemTunnelExtApsViaCemVcIndex    Unsigned32,
      mplsCemTunnelExtApsViaCemVcInstance Unsigned32,
```

```
      mplsCemTunnelExtStorageType         StorageType
   }

mplsCemTunnelExtApsEnable OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "Enables this Tunnel (as part of a group) to be used for
        APS. Remember a Tunnel group is created when a unique
        Tunnel index (in mplsTunnelTable) has multiple instances.
        Tunnel disabling may be useful when a Tunnel is suspect,
        being debugged, or not fully configured.
        Note: some implementations may not support APS."
   ::= { mplsCemTunnelExtEntry 1 }

mplsCemTunnelExtApsCriteria OBJECT-TYPE
   SYNTAX BITS {
         cemVcCriteria(0),
         revertive(512),
         hunt(1024)
   }
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "Bits set here represent defects and modes for switching
        all VCs within this Tunnel to another Tunnel that has no
        APS defects. Currently, only CEM defects are available.
        Separating and individually selecting CEM defects (buffer
```



```
        errors, header errors, etc.) is for future study. Also,
        for future study: determining a Tunnel's state of
        usefulness via other indications (such as OAM).

        If 'cemVcCriteria' is set, mplsCemTunnelExtApsViaCemVc
        is used to index the CEM VC for which defects are
        included. The purpose of this is in situations when
        CEM VCs can generate better defects than Tunnels (for
        example: buffer errors).

        Revertive and Hunt modes:
        'revertive' is useful for non-primary Tunnels where you
        may want to switch back to a primary Tunnel that has no
        defects while the currently active also has no defects
        (see mplsCemTunnelExtApsRevertTimer). 'revertive' should
        NOT be used when standby Tunnels have no available defect
        indications. If 'revertive' is not set, then manual
```

```
        reversion is possible by simply setting the active
        Tunnel's admin status 'down', then 'up'.

        'hunt' is useful when standby Tunnels offer no defect
        indications (see mplsCemTunnelExtApsHoldOffTimer).
        'hunt' should not be used when standby Tunnels have
        available defect indications. Hunting assumes standby
        Tunnels are good, if not the down timers will determine
        it's bad, and hunting continues.


        'hunt' and 'revertive' are mutually exclusive.

        Note: criteria defect bits here MUST align with
        mplsCemTunnelExtDefects so it can be used as a mask."
   ::= { mplsCemTunnelExtEntry 6 }

mplsCemTunnelExtDefects OBJECT-TYPE
   SYNTAX BITS {
         cemVcCriteria(0)
   }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Current state of Tunnel defect indications. Other bits
        here may be defined here to show results of error trend
        analysis. Note: These bits MUST be aligned with
        mplsCemTunnelExtApsCriteria bits so
        mplsCemTunnelExtApsCriteria can be used as a mask.
        Note: other defect indications (e.g. from OAM) are for
        future study."
   ::= { mplsCemTunnelExtEntry 7 }

mplsCemTunnelExtApsHoldoffTimer OBJECT-TYPE -- units are in seconds
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
```



```
   DESCRIPTION
       "Primarily used to hold-off an APS switch after the Tunnel
        has been determined 'down'. The purpose is to allow
        potential recovery schemes within lower communications
        layers a chance to recover.

        Also useful in 'hunt' mode to hold off switching to the
        next Tunnel (slows oscillation when all Tunnels are down)."
   ::= { mplsCemTunnelExtEntry 8 }
```

```
mplsCemTunnelExtApsRevertTimer OBJECT-TYPE -- units are in seconds
   SYNTAX        Unsigned32
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION
       "If this Tunnel is active and 'up', then this timer
        is used in conjunction with 'revertive' (if set above).
        The reversion would be delayed for this time."
   ::= { mplsCemTunnelExtEntry 9 }

mplsCemTunnelExtApsViaCemVcIndex OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "Index of the CEM VC from which APS defect criteria are
        included with this Tunnel's APS criteria."
   ::= { mplsCemTunnelExtEntry 10 }

mplsCemTunnelExtApsViaCemVcInstance OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "Instance of the CEM VC from which APS defect criteria
        are included with this Tunnel's APS criteria."
   ::= { mplsCemTunnelExtEntry 11 }

mplsCemTunnelExtStorageType OBJECT-TYPE
   SYNTAX        StorageType
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "This variable indicates the storage type for this object."
   ::= { mplsCemTunnelExtEntry 12 }

-- End of MPLS Tunnel Extension Table


-- MPLS CEM Tunnel Extension Performance Table

mplsCemTunnelExtPerfTable OBJECT-TYPE
   SYNTAX        SEQUENCE OF MplsCemTunnelExtPerfEntry
   MAX-ACCESS    not-accessible
```



```
   STATUS        current
```

```
   DESCRIPTION
       "This table is an extension to the mplsTunnelTable. It
        provides per Tunnel performance information."
   ::= { mplsCemObjects 6 }

mplsCemTunnelExtPerfEntry OBJECT-TYPE
   SYNTAX        MplsCemTunnelExtPerfEntry
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
       "An entry in this table is created by the agent for every
        mplsCemTunnelExtEntry. It is an extension to
        mplsCemTunnelExtEntry."

   AUGMENTS { mplsCemTunnelExtEntry }

   ::= { mplsCemTunnelExtPerfTable 1 }

MplsCemTunnelExtPerfEntry ::= SEQUENCE {
      mplsCemTunnelExtPerfFailedLabelLookups  Counter32,
      mplsCemTunnelExtPerfLastFailedLookup    MplsLabel
   }

mplsCemTunnelExtPerfFailedLabelLookups OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Number of packets with unexpected labels received. This
        count is based on inbound VC labels seen that are not
        configured for this Tunnel."
   ::= { mplsCemTunnelExtPerfEntry 1 }

mplsCemTunnelExtPerfLastFailedLookup OBJECT-TYPE
   SYNTAX        MplsLabel
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
       "Contains label from the packet that last failed a label
       lookup on this Tunnel."
   ::= { mplsCemTunnelExtPerfEntry 2 }

-- End of MPLS Tunnel Extension Performance Table


-- MPLS CEM SONET Path Extension Table.

mplsCemSonetPathExtTable OBJECT-TYPE
   SYNTAX           SEQUENCE OF MplsCemSonetPathExtEntry
   MAX-ACCESS       not-accessible
   STATUS           current
```

```
   DESCRIPTION
       "This table associates a SONET path with a CEM VC
```



```
        (or CEM VC group) and provides the starting time-slot
        of the SONET path. It as assumed that the transmit
        and receive sides of the SONET path have the same
        starting time-slots. It is also assumed that STSn
        paths have contiguous time-slots."
   ::= { mplsCemObjects 7 }

mplsCemSonetPathExtEntry OBJECT-TYPE
   SYNTAX          MplsCemSonetPathExtEntry
   MAX-ACCESS      not-accessible
   STATUS          current
   DESCRIPTION
        "This table is indexed by the SONET path ifIndex. An
         entry to this table is created by the agent when an
         entry is created in the mplsCemVcTable."

   INDEX  { mplsCemSonetPathExtSonetIfIndex }

      ::= { mplsCemSonetPathExtTable 1 }

MplsCemSonetPathExtEntry ::= SEQUENCE {
      mplsCemSonetPathExtVcIndex        Unsigned32,
      mplsCemSonetPathExtTimeSlot       Unsigned32,
      mplsCemSonetPathExtStorageType    StorageType
   }

mplsCemSonetPathExtVcIndex OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "Index to entry in the mplsCemVcTable."
   ::= { mplsCemSonetPathExtEntry 1 }

mplsCemSonetPathExtTimeSlot OBJECT-TYPE
   SYNTAX        Unsigned32 (1..192)
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "Starting time-slot for this SONET path within the
        SONET line and section. For OC-48, this value could
        range from 1 to 48. The SONET path width must be taken
        into consideration here, for example, in an OC-48 an
        STS-3c could not start at time-slot 47."
```

```
   ::= { mplsCemSonetPathExtEntry 2 }

mplsCemSonetPathExtStorageType OBJECT-TYPE
   SYNTAX        StorageType
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
       "This variable indicates the storage type for this
        object."
   ::= { mplsCemSonetPathExtEntry 3 }
```



```
-- End of MPLS CEM SONET Path Extension Table.


-- Notifications - CEM VC

mplsCemTrapEnable OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
          "If this object is true, then it enables the
           generation of mplsCemVcUp and mplsCemVcDown
           traps, otherwise these traps are not emitted.
           These traps are also individually enabled
           for each CEM VC."
   DEFVAL { false }
   ::= { mplsCemObjects 8 }

mplsCemVcUp NOTIFICATION-TYPE
   OBJECTS   { mplsCemVcIndex,
               mplsCemVcInstance,
               mplsCemVcOperStatus,
               mplsCemVcAdminStatus,
               mplsCemVcSonetPathIfIndex }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsCemVcOperStatus object for one of the configured
        CEM VC entries is about to leave the down state and
        transition into some other state (but not into the
        notPresent state).  This other state is indicated by
        the included value of mplsCemVcOperStatus."
   ::= { mplsCemNotifyPrefix 1 }

mplsCemVcDown NOTIFICATION-TYPE
```

```
   OBJECTS   { mplsCemVcIndex,
               mplsCemVcInstance,
               mplsCemVcOperStatus,
               mplsCemVcAdminStatus,
               mplsCemVcSonetPathIfIndex }
   STATUS      current
   DESCRIPTION
       "This notification is generated when a
        mplsCemVcOperStatus object for one of the configured
        CEM VC entries is about to enter the down state and
        transition into some other state (but not from the
        notPresent state).  This other state is indicated by
        the included value of mplsCemVcOperStatus."
   ::= { mplsCemNotifyPrefix 2 }

mplsCemVcApsFrom NOTIFICATION-TYPE
   OBJECTS   { mplsCemVcIndex,
               mplsCemVcInstance,
```



```
               mplsCemVcOperStatus,
               mplsCemVcAdminStatus,
               mplsCemVcSonetPathIfIndex }
   STATUS      current
   DESCRIPTION
       "This notification is generated when the CEM VC APS
        mechanism has initiated a switch from the specified
        Index/Instance to find a 'good' CEM VC within the same
        group. To reduce the number of Traps, it is recommended
        to only generate mplsCemVcApsFrom when first initiating
        attempt to find a good CEM VC instance. That is, do not
        generate Traps if the APS mechanism is continuing to
        switch looking for an CEM VC group member that is good."
   ::= { mplsCemNotifyPrefix 3 }

mplsCemVcApsTo NOTIFICATION-TYPE
   OBJECTS   { mplsCemVcIndex,
               mplsCemVcInstance,
               mplsCemVcOperStatus,
               mplsCemVcAdminStatus,
               mplsCemVcSonetPathIfIndex }
   STATUS      current
   DESCRIPTION
       "This notification is generated when the CEM VC APS
        mechanism has switched to and settled on a CEM VC Index
        and Instance that is 'good' (whether from a bad VC or
        when reverting)."
   ::= { mplsCemNotifyPrefix 4 }
```

```
   -- Note: generating Traps for Tunnel (bulk VC) APS switching is
   -- for further study.

   -- End of notifications.

END
```

9 References

[MPLSArch] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", Internet Draft <draft-ietf-mpls-arch-06.txt>, August 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-05.txt>, September 1999.

[CEM] Malis, A., Vogelsang, S., and Martini, L. "SONET/SDH Circuit Emulation Service Over MPLS (CEM) encapsulation", Internet Draft <draft-malis-sonet-ces-mpls-02.txt>, February 2001.

[TRANS] Martini et al, "Transport of Layer 2 Frames Over MPLS" <draft-martini-l2circuit-trans-mpls-05.txt>, November 2000.

[ENCAP] Martini et al, "Encapsulation Methods for Transport of Layer 2 Frames Over MPLS", <draft-martini-l2circuit-encap-mpls-01.txt>, November 2000.

[LSRMIB] Srinivasan, C., Viswanathan, A., and Nadeau, T. "MPLS Label Switch Router Management Information Base Using SMIv2", draft-ietf-mpls-lsr-mib-07.txt, January 2001.

[TEMIB] Srinivasan, C., Viswanathan, A., and Nadeau, T. "MPLS Traffic Engineering Management Information

Base Using SMIv2", <draft-ietf-mpls-te-mib-05.txt>,

November 2000.

[SONETMIB] Brown, T. and Tesink, K. "Definitions of Managed Objects for the SONET/SDH Interface Type", RFC 1595.

[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-07.txt>, September 1999.

[Assigned] Reynolds, J., and J. Postel, "Assigned Numbers", RFC 1700, October 1994. See also: http://www.isi.edu/in-notes/iana/assignments/smi-numbers

[IANAFamily] Internet Assigned Numbers Authority (IANA), ADDRESS FAMILY NUMBERS,(http://www.isi.edu/in-notes/iana/assignements/address-family-numbers), for MIB see: ftp://ftp.isi.edu/mib/ianaaddressfamilynumbers.mib

[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.

[RFC2572] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

[RFC2574] Blumenthal, U., and B. Wijnen, "User-based Security

Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

[RFC1905] 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.

[RFC2573] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.

[RFC2575] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.

[RFC2570] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.

[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 2574, April 1999.

[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 2575, April 1999.

[IPSEC] Kent, S., and Atkinson, R., "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[IFMIB] McCloghrie, K., and F. Kastenholtz, "The Interfaces Group MIB using SMIv2", RFC 2233, Nov. 1997

[ATOMMIB] Tesink, K., "Definitions of Managed Objects for ATM Management", RFC 2515, Feb. 1999

[BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10 Author's Addresses

Dave Danenberg
Litchfield Communications, Inc.
76 Westbury Park Rd
Princeton Building East
Watertown, CT 06795
Email: dave_danenberg@litchfieldcomm.com

Thomas D. Nadeau
Cisco Systems, Inc.
250 Apollo Drive

Chelmsford, MA 01824
Email: tnadeau@cisco.com

Andrew G. Malis
Vivace Networks, Inc.
2730 Orchard Parkway
San Jose, CA 95134

Email: Andy.Malis@vivacenetworks.com

Scott C. Park
Litchfield Communications, Inc.
76 Westbury Park Rd
Princeton Building East
Watertown, CT 06795
Email: scott_park@litchfieldcomm.com

11 Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.