Network Working Group
Request for Comments: 2922
Category: Informational

A. Bierman
Cisco Systems, Inc.
K. Jones
Nortel Networks

September 2000

Physical Topology MIB

Status of this Memo

Copyright Notice

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects used for managing physical topology identification and discovery.

Table of Contents

1. The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [RFC2571].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [RFC1155], STD 16, RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [RFC1901] and RFC 1906 [RFC1906]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [RFC1906], RFC 2572 [RFC2572] and RFC 2574 [RFC2574].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [RFC1905].
- A set of fundamental applications described in RFC 2573 [RFC2573] and the view-based access control mechanism described in RFC 2575 [RFC2575].

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

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

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

2. Overview

There is a need for a standardized means of representing the physical network connections pertaining to a given management domain. The Physical Topology MIB (PTOPO-MIB) provides a standard way to identify connections between network ports and to discover network addresses of SNMP agents containing management information associated with each port.

A topology mechanism is used to discover the information required by the PTOPO-MIB. There is a need for a standardized topology mechanism to increase the likelihood of multi-vendor interoperability of such physical topology management information. The PTOPO-MIB does not, however, specify or restrict the discovery mechanism(s) used for an implementation of the PTOPO-MIB. Topology mechanisms exist for certain media types (such as FDDI) and proprietary mechanisms exist for other media such as shared media Ethernet, switched Ethernet, and Token Ring. Rather than specifying mechanisms for each type of technology, the PTOPO-MIB allows co-existence of multiple topology mechanisms. The required objects of the PTOPO-MIB define the core requirements for any topology mechanism.

The scope of the physical topology (PTOPO) mechanism is the identification of connections between two network ports. Network addresses of SNMP agents containing management information associated with each port can also be identified.

2.1. Terms

Some terms are used throughout this document:

Physical Topology
: Physical topology represents the topology model for layer 1 of the OSI stack - the physical layer. Physical topology consists of identifying the devices on the network and how they are physically interconnected. By definition of this document, physical topology does not imply a physical relationship between ports on the same device. Other means exist for

determining these relationships (e.g., Entity MIB [RFC2737]) exist for determining these relationships. Note that physical topology is independent of logical topology, which associates ports based on higher layer attributes, such as network layer address.

Chassis
: A chassis is a physical component which contains other physical components. It is identified by an entPhysicalEntry with an entPhysicalClass value of 'chassis(3)' and an entPhysicalContainedIn value of zero. A chassis identifier consists of a globally unique SnmpAdminString.

Local Chassis
: The particular chassis containing the SNMP agent implementing the PTOPO MIB.

Port
: A port is a physical component which can be connected to another port through some medium. It is identified by an entPhysicalEntry with an entPhysicalClass value of 'port(10)'. A port identifier consists of an SnmpAdminString which must be unique within the context of the chassis which contains the port.

Connection Endpoint
: A connection endpoint consists of a physical port, which is contained within a single physical chassis.

Connection Endpoint Identifier
: A connection endpoint is identified by a globally unique chassis identifier and a port identifier unique within the associated chassis.

Connection
: A connection consists of two physical ports, and the attached physical medium, configured for the purpose of transferring network traffic between the ports. A connection is identified by its endpoint identifiers.

Non-local Connection
: A connection for which neither endpoint is located on the local chassis.

Cloud
A cloud identifies a portion of the topology for which insufficient information is known to completely infer the interconnection of devices that make up that portion of the topology.

2.2. Design Goals

Several factors influenced the design of this physical topology function:

- Simplicity
  The physical topology discovery function should be as simple as possible, exposing only the information needed to identify connection endpoints and the SNMP agent(s) associated with each connection endpoint.

- Completeness
  At least one standard discovery protocol capable of supporting the standard physical topology MIB must be defined. Multi-vendor interoperability will not be achievable unless a simple and extensible discovery protocol is available. However, the PTOPO MIB should not specify or restrict the topology discovery mechanisms an agent can use.

- No Functional Overlap
  Existing standard MIBs should be utilized whenever possible. Physical topology information is tightly coupled to functionality found in the Interfaces MIB [RFC2233] and Entity MIB [RFC2737]. New physical topology MIB objects should not duplicate these MIBs.

- Identifier Stability
  Connection endpoint identifiers must be persistent (i.e. stable across device reboots). Dynamic primary key objects like ifIndex and entPhysicalIndex are not suitable for table indices in a physical topology MIB that is replicated and distributed throughout a managed system.

- Identifier Flexibility
  Persistent string-based component identifiers should be supported from many sources. Chassis identifiers may be found in the Entity MIB [RFC2737], and port identifiers may be found in the Interfaces MIB [RFC2233] or Entity MIB [RFC2737].

- Partial Topology Support
  Physical topology data for remote components may only be partially available to an agent. An enumerated INTEGER hierarchy of component identifier types allows for incomplete physical connection identifier information to be substituted with secondary information such as unicast source MAC address or network address associated with a particular port. A PTOPO Agent maintains information derived from the 'best' source of information for each connection. If a 'better' identifier source is detected, the PTOPO entries are updated accordingly. It is an implementation specific matter whether a PTOPO agent replaces 'old' entries or retains them, however an agent must remove information known to be incorrect.

- Low Polling Impact
  Physical topology polling should be minimized through techniques such as TimeFiltered data tables (from RMON-2 [RFC2021]), and last-change notifications.

3. Topology Framework

This section describes the physical topology framework in detail.

3.1. Devices and Topology Agents

The network devices, along with their physical connectivity, make up the physical topology. Some of these devices (but maybe not all) provide management agents that report their local physical topology information to a manager via the physical topology MIB.

These devices include communication infrastructure devices, such as hubs, switches, and routers, as well as 'leaf' devices such as workstations, printers, and servers. Generally, user data passes through infrastructure devices while leaf devices are sources and sinks of data. Both types of devices may implement the physical topology MIB, although implementation within leaf devices is much less critical.

Each managed device collects physical topology information from the network, based on the topology mechanism(s) it is configured to use. The data represents this agent's local view of the physical network. Part of the topology data collected must include the identification of other local agents which may contain additional topology information. The definition of 'local' varies based on the topology mechanism or mechanisms being used.

3.2. Topology Mechanisms

A topology mechanism is a means, possibly requiring some sort of protocol, by which devices determine topology information. The topology mechanism must provide sufficient information to populate the MIB described later in this document.

Topology mechanisms can be active or passive. Active mechanisms require a device to send and receive topology protocol packets. These packets provide the device ID of the source of the packet and may also indicate out which port the packet was transmitted. When receiving these packets, devices typically are required to identify on which port that packet was received.

Passive mechanisms take advantage of data on the network to populate the topology MIB. By maintaining a list of device identifiers seen on each port of all devices in a network, it is possible to populate the PTOPO-MIB.

Many instances of a particular topology mechanism may be in use on a given network, and many different mechanisms may be employed. In some cases, multiple mechanisms may overlap across part of the physical topology with individual ports supporting more than one topology mechanism. In general, this simply allows the port to collect more robust topology information. Agents may need to be configured so that they know which mechanism(s) are in use on any given portion of the network.

Most topology mechanisms need to be bounded to a subset of the network to contain their impact on the network and limit the size of topology tables maintained by the agent. Topology mechanisms are often naturally bounded by the media on which they run (e.g. FDDI topology mechanism) or by routers in the network that intentionally block the mechanism from crossing into other parts of the network.

3.3. Future Considerations

While the framework presented here is focused on physical topology, it may well be that the topology mechanisms and MIB described could be extended to include logical topology information as well. That is not a focus of this memo.

4. Physical Topology MIB

This section describes and defines the Physical Topology MIB.

4.1. Persistent Identifiers

The PTOPO MIB utilizes non-volatile identifiers to distinguish individual chassis and port components. These identifiers are associated with external objects in order to relate topology information to the existing managed objects.

In particular, an object from the Entity MIB [RFC2737] or Interfaces MIB [RFC2233] can be used as the 'reference-point' for a connection component identifier.

The Physical Topology MIB uses two identifier types pertaining to the PTOPO MIB:

- globally unique chassis identifiers.

- port identifiers; unique only within the chassis which contains the port.

Identifiers are stored as OCTET STRINGs, which are limited to 32 bytes in length, This supports flexible naming conventions and constrains the non-volatile storage requirements for an agent.

4.2. Relationship to Entity MIB

The first version of the Entity MIB [RFC2037] allows the physical component inventory and hierarchy to be identified. However, this MIB does not provide persistent component identifiers, which are required for the PTOPO MIB. Therefore, version 2 of the Entity MIB [RFC2737] is required to support that feature. Specifically, the entPhysicalAlias object is utilized as a persistent chassis identifier.

For agents implementing the PTOPO MIB, this new object must be used to represent the chassis identifier. Port identifiers can be based on the entPhysicalAlias object associated with the port, but only if the port is not represented as an interface in the ifXTable.

Implementation of the entPhysicalGroup [RFC2737] and the entPhysicalAlias object [RFC2737] are mandatory for SNMP agents which implement the PTOPO MIB. No other objects must be implemented from these MIBs to support the physical topology function.

4.3. Relationship to Interfaces MIB

The PTOPO MIB requires a persistent identifier for each port. The Interfaces MIB [RFC2233] provides a standard mechanism for managing network interfaces. Unfortunately, not all ports which may be represented in the PTOPO MIB are also represented in the Interfaces MIB (e.g., repeater ports).

For agents which implement the PTOPO MIB, for each port also represented in the Interfaces MIB, the agent must use the associated ifAlias value for the port identifier. For each port not represented in the Interfaces MIB, the associated entPhysicalAlias value must be used for the port identifier. Note that the PTOPO MIB requires only minimal support from the Interfaces MIB. Specifically, the ' ifGeneralInformationGroup' level of conformance must be provided for each port also identified in the PTOPO MIB. The agent may choose to support these objects with read-only access, as specified in the conformance section of the Interfaces MIB.

4.4. Relationship to RMON-2 MIB

The RMON-2 MIB [RFC2021] contains address mapping information which can be integrated with physical topology information. The physical ports identified in a physical topology MIB can be related to the MAC and network layer addresses found in the addressMapTable.

4.5. Relationship to Bridge MIB

The Bridge MIB [RFC1493] contains information which may relate to physical ports represented in the ptopoConnTable. Entries in the dot1dBasePortTable and dot1dStpPortTable can by related to physical ports represented in the PTOPO MIB. Also, bridge port MAC addresses may be used as chassis and port identifiers in some situations.

4.6. Relationship to Repeater MIB

The Repeater MIB [RFC2108] contains information which may relate to physical ports represented in the PTOPO MIB. Entries in the rptrPortTable and rptrMonitorPortTable can by related to physical ports represented in the ptopoConnTable. Entries in the rptrInfoTable and rptrMonTable can be related to repeater backplanes possibly represented in the ptopoConnTable.

4.7. MIB Structure

The PTOPO MIB contains three MIB object groups:

- ptopoData
  Exposes physical topology data learned from discovery protocols and/or manual configuration.

- ptopoGeneral
  Contains general information regarding PTOPO MIB status.

- ptopoConfig
  Contains configuration variables for the PTOPO MIB agent function.

4.7.1. ptopoData Group

This group contains a single table to identity physical topology data.

The ptopoConnTable contains information about the connections learned or configured on behalf of the PTOPO MIB SNMP Agent.

4.7.2. ptopoGeneral Group

This group contains some scalar objects to report the status of the PTOPO MIB information currently known to the SNMP Agent. The global last change time, and table add and delete counters allow an NMS to set threshold alarms to trigger PTOPO polling.

4.7.3. ptopoConfig Group

This group contains tables to configure the behavior of the physical topology function. The transmission of ptopoLastChange notifications can be configured using the ptopoConfigTrapInterval scalar MIB object.

4.8. Physical Topology MIB Definitions

```
PTOPO-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Integer32, Counter32, mib-2
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, AutonomousType, RowStatus, TimeStamp, TruthValue
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
```

```
        FROM SNMPv2-CONF
    TimeFilter
        FROM RMON2-MIB
    PhysicalIndex
        FROM ENTITY-MIB
    AddressFamilyNumbers
        FROM IANA-ADDRESS-FAMILY-NUMBERS-MIB;

ptopoMIB MODULE-IDENTITY
    LAST-UPDATED "200009210000Z"
    ORGANIZATION "IETF; PTOPOMIB Working Group"
    CONTACT-INFO
       "PTOPOMIB WG Discussion:
        ptopo@3com.com
        Subscription:
        majordomo@3com.com
          msg body: [un]subscribe ptopomib

        Andy Bierman
        Cisco Systems Inc.
        170 West Tasman Drive
        San Jose, CA 95134
        408-527-3711
        abierman@cisco.com

        Kendall S. Jones
        Nortel Networks
        4401 Great America Parkway
        Santa Clara, CA 95054
        408-495-7356
        kejones@nortelnetworks.com"
    DESCRIPTION
            "The MIB module for physical topology information."
    REVISION        "200009210000Z"
    DESCRIPTION
            "Initial Version of the Physical Topology MIB.  This version
            published as RFC 2922."
    ::= { mib-2 79 }

ptopoMIBObjects   OBJECT IDENTIFIER ::= { ptopoMIB 1 }


-- MIB groups
ptopoData         OBJECT IDENTIFIER ::= { ptopoMIBObjects 1 }
ptopoGeneral      OBJECT IDENTIFIER ::= { ptopoMIBObjects 2 }
ptopoConfig       OBJECT IDENTIFIER ::= { ptopoMIBObjects 3 }

-- textual conventions
```

```
PtopoGenAddr ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
            "The value of an address."
    SYNTAX      OCTET STRING (SIZE (0..20))

PtopoChassisIdType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
            "This TC describes the source of a chassis identifier.

            The enumeration 'chasIdEntPhysicalAlias(1)' represents a
            chassis identifier based on the value of entPhysicalAlias
            for a chassis component (i.e., an entPhysicalClass value of
            'chassis(3)').

            The enumeration 'chasIdIfAlias(2)' represents a chassis
            identifier based on the value of ifAlias for an interface
            on the containing chassis.

            The enumeration 'chasIdPortEntPhysicalAlias(3)' represents
            a chassis identifier based on the value of entPhysicalAlias
            for a port or backplane component (i.e., entPhysicalClass
            value of 'port(10)' or 'backplane(4)'), within the
            containing chassis.

            The enumeration 'chasIdMacAddress(4)' represents a chassis
            identifier based on the value of a unicast source MAC
            address (encoded in network byte order and IEEE 802.3
            canonical bit order), of a port on the containing chassis.

            The enumeration 'chasIdPtopoGenAddr(5)' represents a
            chassis identifier based on a network address, associated
            with a particular chassis.  The encoded address is actually
            composed of two fields.  The first field is a single octet,
            representing the IANA AddressFamilyNumbers value for the
            specific address type, and the second field is the
            PtopoGenAddr address value."
    SYNTAX      INTEGER {
            chasIdEntPhysicalAlias(1),
            chasIdIfAlias(2),
            chasIdPortEntPhysicalAlias(3),
            chasIdMacAddress(4),
            chasIdPtopoGenAddr(5)
    }

PtopoChassisId ::= TEXTUAL-CONVENTION
    STATUS      current
```

```
    DESCRIPTION
            "This TC describes the format of a chassis identifier
            string.  Objects of this type are always used with an
            associated PtopoChassisIdType object, which identifies the
            format of the particular PtopoChassisId object instance.

            If the associated PtopoChassisIdType object has a value of
            'chasIdEntPhysicalAlias(1)', then the octet string
            identifies a particular instance of the entPhysicalAlias
            object for a chassis component (i.e., an entPhysicalClass
            value of 'chassis(3)').

            If the associated PtopoChassisIdType object has a value of
            'chasIdIfAlias(2)', then the octet string identifies a
            particular instance of the ifAlias object for an interface
            on the containing chassis.

            If the associated PtopoChassisIdType object has a value of
            'chasIdPortEntPhysicalAlias(3)', then the octet string
            identifies a particular instance of the entPhysicalAlias
            object for a port or backplane component within the
            containing chassis.

            If the associated PtopoChassisIdType object has a value of
            'chasIdMacAddress(4)', then this string identifies a
            particular unicast source MAC address (encoded in network
            byte order and IEEE 802.3 canonical bit order), of a port on
            the containing chassis.

            If the associated PtopoChassisIdType object has a value of
            'chasIdPtopoGenAddr(5)', then this string identifies a
            particular network address, encoded in network byte order,
            associated with one or more ports on the containing chassis.
            The first octet contains the IANA Address Family Numbers
            enumeration value for the specific address type, and octets
            2 through N contain the PtopoGenAddr address value in
            network byte order."
    SYNTAX      OCTET STRING (SIZE (1..32))

PtopoPortIdType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
            "This TC describes the source of a particular type of port
            identifier used in the PTOPO MIB.

            The enumeration 'portIdIfAlias(1)' represents a port
            identifier based on the ifAlias MIB object.
```

```
            The enumeration 'portIdPortEntPhysicalAlias(2)' represents a
            port identifier based on the value of entPhysicalAlias for a
            port or backplane component (i.e., entPhysicalClass value of
            'port(10)' or 'backplane(4)'), within the containing
            chassis.

            The enumeration 'portIdMacAddr(3)' represents a port
            identifier based on a unicast source MAC address, which has
            been detected by the agent and associated with a particular
            port.

            The enumeration 'portIdPtopoGenAddr(4)' represents a port
            identifier based on a network address, detected by the agent
            and associated with a particular port."
    SYNTAX      INTEGER {
            portIdIfAlias(1),
            portIdEntPhysicalAlias(2),
            portIdMacAddr(3),
            portIdPtopoGenAddr(4)
    }

PtopoPortId ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
            "This TC describes the format of a port identifier string.
            Objects of this type are always used with an associated
            PtopoPortIdType object, which identifies the format of the
            particular PtopoPortId object instance.

            If the associated PtopoPortIdType object has a value of
            'portIdIfAlias(1)', then the octet string identifies a
            particular instance of the ifAlias object.

            If the associated PtopoPortIdType object has a value of
            'portIdEntPhysicalAlias(2)', then the octet string
            identifies a particular instance of the entPhysicalAlias
            object for a port component (i.e., entPhysicalClass value of
            'port(10)').

            If the associated PtopoPortIdType object has a value of
            'portIdMacAddr(3)', then this string identifies a particular
            unicast source MAC address associated with the port.

            If the associated PtopoPortIdType object has a value of
            'portIdPtopoGenAddr(4)', then this string identifies a
            network address associated with the port.  The first octet
            contains the IANA AddressFamilyNumbers enumeration value for
            the specific address type, and octets 2 through N contain
```

```
            the PtopoGenAddr address value in network byte order."
    SYNTAX      OCTET STRING (SIZE (1..32))


PtopoAddrSeenState ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
            "This TC describes the state of address detection for a
            particular type of port identifier used in the PTOPO MIB.

            The enumeration 'notUsed(1)' represents an entry for which
            the particular MIB object is not applicable to the remote
            connection endpoint,

            The enumeration 'unknown(2)' represents an entry for which
            the particular address collection state is not known.

            The enumeration 'oneAddr(3)'  represents an entry for which
            exactly one source address (of the type indicated by the
            particular MIB object), has been detected.

            The enumeration 'multiAddr(4)'  represents an entry for
            which more than one source address (of the type indicated by
            the particular MIB object), has been detected.

            An agent is expected to set the initial state of the
            PtopoAddrSeenState to 'notUsed(1)' or 'unknown(2)'.

            Note that the PTOPO MIB does not restrict or specify the
            means in which the PtopoAddrSeenState is known to an agent.
            In particular, an agent may detect this information through
            configuration data, or some means other than directly
            monitoring all port traffic."
    SYNTAX      INTEGER {
            notUsed(1),
            unknown(2),
            oneAddr(3),
            multiAddr(4)
    }

--  ********************************************************
--
--           P T O P O    D A T A     G R O U P
--
--  ********************************************************

-- Connection Table
```

```
ptopoConnTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PtopoConnEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one or more rows per physical network
            connection known to this agent.  The agent may wish to
            ensure that only one ptopoConnEntry is present for each
            local port, or it may choose to maintain multiple
            ptopoConnEntries for the same local port.

            Entries based on lower numbered identifier types are
            preferred over higher numbered identifier types, i.e., lower
            values of the ptopoConnRemoteChassisType and
            ptopoConnRemotePortType objects."
    ::= { ptopoData 1 }

ptopoConnEntry       OBJECT-TYPE
    SYNTAX      PtopoConnEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular physical network connection.
            Entries may be created and deleted in this table, either
            manually or by the agent, if a physical topology discovery
            process is active."
    INDEX   {
           ptopoConnTimeMark,
           ptopoConnLocalChassis,
           ptopoConnLocalPort,
           ptopoConnIndex
    }
    ::= { ptopoConnTable 1 }

PtopoConnEntry ::= SEQUENCE {
      ptopoConnTimeMark            TimeFilter,
      ptopoConnLocalChassis        PhysicalIndex,
      ptopoConnLocalPort           PhysicalIndex,
      ptopoConnIndex               Integer32,
      ptopoConnRemoteChassisType   PtopoChassisIdType,
      ptopoConnRemoteChassis       PtopoChassisId,
      ptopoConnRemotePortType      PtopoPortIdType,
      ptopoConnRemotePort          PtopoPortId,
      ptopoConnDiscAlgorithm       AutonomousType,
      ptopoConnAgentNetAddrType    AddressFamilyNumbers,
      ptopoConnAgentNetAddr        PtopoGenAddr,
      ptopoConnMultiMacSASeen      PtopoAddrSeenState,
      ptopoConnMultiNetSASeen      PtopoAddrSeenState,
```

```
      ptopoConnIsStatic            TruthValue,
      ptopoConnLastVerifyTime      TimeStamp,
      ptopoConnRowStatus           RowStatus
}

ptopoConnTimeMark  OBJECT-TYPE
    SYNTAX      TimeFilter
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A TimeFilter for this entry.  See the TimeFilter textual
            convention in RFC 2021 to see how this works."
    ::= { ptopoConnEntry 1 }

ptopoConnLocalChassis  OBJECT-TYPE
    SYNTAX      PhysicalIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The entPhysicalIndex value used to identify the chassis
            component associated with the local connection endpoint."
    ::= { ptopoConnEntry 2 }

ptopoConnLocalPort     OBJECT-TYPE
    SYNTAX      PhysicalIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The entPhysicalIndex value used to identify the port
            component associated with the local connection endpoint."
    ::= { ptopoConnEntry 3 }

ptopoConnIndex    OBJECT-TYPE
    SYNTAX      Integer32  (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object represents an arbitrary local integer value
            used by this agent to identify a particular connection
            instance, unique only for the indicated local connection
            endpoint.

            A particular ptopoConnIndex value may be reused in the event
            an entry is aged out and later re-learned with the same (or
            different) remote chassis and port identifiers.

            An agent is encouraged to assign monotonically increasing
            index values to new entries, starting with one, after each
```

```
            reboot.  It is considered unlikely that the ptopoConnIndex
            will wrap between reboots."
    ::= { ptopoConnEntry 4 }

ptopoConnRemoteChassisType  OBJECT-TYPE
    SYNTAX      PtopoChassisIdType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The type of encoding used to identify the chassis
            associated with the remote connection endpoint.

            This object may not be modified if the associated
            ptopoConnRowStatus object has a value of active(1)."
    ::= { ptopoConnEntry 5 }

ptopoConnRemoteChassis  OBJECT-TYPE
    SYNTAX      PtopoChassisId
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The string value used to identify the chassis component
            associated with the remote connection endpoint.

            This object may not be modified if the associated
            ptopoConnRowStatus object has a value of active(1)."
    ::= { ptopoConnEntry 6 }

ptopoConnRemotePortType  OBJECT-TYPE
    SYNTAX      PtopoPortIdType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The type of port identifier encoding used in the associated
            'ptopoConnRemotePort' object.

            This object may not be modified if the associated
            ptopoConnRowStatus object has a value of active(1)."
    ::= { ptopoConnEntry 7 }

ptopoConnRemotePort  OBJECT-TYPE
    SYNTAX      PtopoPortId
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The string value used to identify the port component
            associated with the remote connection endpoint.
```

```
            This object may not be modified if the associated
            ptopoConnRowStatus object has a value of active(1)."
    ::= { ptopoConnEntry 8 }

ptopoConnDiscAlgorithm OBJECT-TYPE
    SYNTAX      AutonomousType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "An indication of the algorithm used to discover the
            information contained in this conceptual row.

            A value of ptopoDiscoveryLocal indicates this entry was
            configured by the local agent, without use of a discovery
            protocol.

            A value of { 0 0 } indicates this entry was created manually
            by an NMS via the associated RowStatus object. "
    ::= { ptopoConnEntry 9 }

ptopoConnAgentNetAddrType  OBJECT-TYPE
    SYNTAX      AddressFamilyNumbers
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "This network address type of the associated
            ptopoConnNetAddr object, unless that object contains a zero
            length string.  In such a case, an NMS application should
            ignore any returned value for this object.

            This object may not be modified if the associated
            ptopoConnRowStatus object has a value of active(1)."
    ::= { ptopoConnEntry 10 }

ptopoConnAgentNetAddr  OBJECT-TYPE
    SYNTAX      PtopoGenAddr
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "This object identifies a network address which may be used
            to reach an SNMP agent entity containing information for the
            chassis and port components represented by the associated
            'ptopoConnRemoteChassis' and 'ptopoConnRemotePort' objects.
            If no such address is known, then this object shall contain
            an empty string.

            This object may not be modified if the associated
            ptopoConnRowStatus object has a value of active(1)."
```

```
    ::= { ptopoConnEntry 11 }

ptopoConnMultiMacSASeen  OBJECT-TYPE
    SYNTAX      PtopoAddrSeenState
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object indicates if multiple unicast source MAC
            addresses have been detected by the agent from the remote
            connection endpoint, since the creation of this entry.

            If this entry has an associated ptopoConnRemoteChassisType
            and/or ptopoConnRemotePortType value other than
            'portIdMacAddr(3)', then the value 'notUsed(1)' is returned.

            Otherwise, one of the following conditions must be true:

            If the agent has not yet detected any unicast source MAC
            addresses from the remote port, then the value 'unknown(2)'
            is returned.

            If the agent has detected exactly one unicast source MAC
            address from the remote port, then the value 'oneAddr(3)' is
            returned.

            If the agent has detected more than one unicast source MAC
            address from the remote port, then the value 'multiAddr(4)'
            is returned."
    ::= { ptopoConnEntry 12 }

ptopoConnMultiNetSASeen  OBJECT-TYPE
    SYNTAX      PtopoAddrSeenState
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object indicates if multiple network layer source
            addresses have been detected by the agent from the remote
            connection endpoint, since the creation of this entry.

            If this entry has an associated ptopoConnRemoteChassisType
            or ptopoConnRemotePortType value other than
            'portIdGenAddr(4)' then the value 'notUsed(1)' is returned.

            Otherwise, one of the following conditions must be true:

            If the agent has not yet detected any network source
            addresses of the appropriate type from the remote port, then
            the value 'unknown(2)' is returned.
```

```
            If the agent has detected exactly one network source address
            of the appropriate type from the remote port, then the value
            'oneAddr(3)' is returned.

            If the agent has detected more than one network source
            address (of the same appropriate type) from the remote port,
            this the value 'multiAddr(4)' is returned."
    ::= { ptopoConnEntry 13 }

ptopoConnIsStatic  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "This object identifies static ptopoConnEntries.  If this
            object has the value 'true(1)', then this entry is not
            subject to any age-out mechanisms implemented by the agent.

            If this object has the value 'false(2)', then this entry is
            subject to all age-out mechanisms implemented by the agent.

            This object may not be modified if the associated
            ptopoConnRowStatus object has a value of active(1)."
    DEFVAL { false }
    ::= { ptopoConnEntry 14 }

ptopoConnLastVerifyTime  OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "If the associated value of ptopoConnIsStatic is equal to
            'false(2)', then this object contains the value of sysUpTime
            at the time the conceptual row was last verified by the
            agent, e.g., via reception of a topology protocol message,
            pertaining to the associated remote chassis and port.

            If the associated value of ptopoConnIsStatic is equal to
            'true(1)', then this object shall contain the value of
            sysUpTime at the time this entry was last activated (i.e.,
            ptopoConnRowStatus set to 'active(1)')."
    ::= { ptopoConnEntry 15 }

ptopoConnRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
```

```
            "The status of this conceptual row."
    ::= { ptopoConnEntry 16 }

--  ********************************************************
--
--           P T O P O    G E N E R A L     G R O U P
--
--  ********************************************************

-- last change time stamp for the whole MIB

ptopoLastChangeTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The value of sysUpTime at the time a conceptual row is
            created, modified, or deleted in the ptopoConnTable.

            An NMS can use this object to reduce polling of the
            ptopoData group objects."
    ::= { ptopoGeneral 1 }

ptopoConnTabInserts OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "table entries"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of times an entry has been inserted into the
            ptopoConnTable."
    ::= { ptopoGeneral 2 }

ptopoConnTabDeletes OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "table entries"
    MAX-ACCESS  read-only
    STATUS      current

    DESCRIPTION
            "The number of times an entry has been deleted from the
            ptopoConnTable."
    ::= { ptopoGeneral 3 }

ptopoConnTabDrops OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "table entries"
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
            "The number of times an entry would have been added to the
            ptopoConnTable, (e.g., via information learned from a
            topology protocol), but was not because of insufficient
            resources."
    ::= { ptopoGeneral 4 }

ptopoConnTabAgeouts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of times an entry has been deleted from the
            ptopoConnTable because the information timeliness interval
            for that entry has expired."
    ::= { ptopoGeneral 5 }

--  ********************************************************
--
--           P T O P O    C O N F I G     G R O U P
--
--  ********************************************************

ptopoConfigTrapInterval OBJECT-TYPE
    SYNTAX      Integer32 (0 | 5..3600)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This object controls the transmission of PTOPO
            notifications.

            If this object has a value of zero, then no
            ptopoConfigChange notifications will be transmitted by the
            agent.

            If this object has a non-zero value, then the agent must not
            generate more than one ptopoConfigChange trap-event in the
            indicated period, where a 'trap-event' is the transmission
            of a single notification PDU type to a list of notification
            destinations.  If additional configuration changes occur
            within the indicated throttling period, then these trap-
            events must be suppressed by the agent. An NMS should
            periodically check the value of ptopoLastChangeTime to
            detect any missed ptopoConfigChange trap-events, e.g. due to
            throttling or transmission loss.
```

```
            If notification transmission is enabled, the suggested
            default throttling period is 60 seconds, but transmission
            should be disabled by default.

            If the agent is capable of storing non-volatile
            configuration, then the value of this object must be
            restored after a re-initialization of the management
            system."
    DEFVAL { 0 }
    ::= { ptopoConfig 1 }

ptopoConfigMaxHoldTime OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This object specifies the desired time interval for which
            an agent will maintain dynamic ptopoConnEntries.

            After the specified number of seconds since the last time an
            entry was verified, in the absence of new verification
            (e.g., receipt of a topology protocol message), the agent
            shall remove the entry.  Note that entries may not always be
            removed immediately, but may possibly be removed at periodic
            garbage collection intervals.
            This object only affects dynamic ptopoConnEntries, i.e.  for
            which ptopoConnIsStatic equals 'false(2)'. Static entries
            are not aged out.

            Note that dynamic ptopoConnEntries may also be removed by
            the agent due to the expired timeliness of learned topology
            information (e.g., timeliness interval for a remote port
            expires).  The actual age-out interval for a given entry is
            defined by the following formula:

              age-out-time =
                min(ptopoConfigMaxHoldTime, <entry-specific hold-time>)

            where <entry-specific hold-time> is determined by the
            discovery algorithm, and may be different for each entry."
    DEFVAL { 300 }
    ::= { ptopoConfig 2 }


-- PTOPO MIB Notification Definitions
ptopoMIBNotifications  OBJECT IDENTIFIER ::= { ptopoMIB 2 }
ptopoMIBTrapPrefix     OBJECT IDENTIFIER ::=
```

```
      { ptopoMIBNotifications 0 }

ptopoConfigChange NOTIFICATION-TYPE
    OBJECTS       {
             ptopoConnTabInserts,
             ptopoConnTabDeletes,
             ptopoConnTabDrops,
             ptopoConnTabAgeouts
    }
    STATUS        current
    DESCRIPTION
            "A ptopoConfigChange notification is sent when the value of
            ptopoLastChangeTime changes. It can be utilized by an NMS to
            trigger physical topology table maintenance polls.

            Note that transmission of ptopoConfigChange notifications
            are throttled by the agent, as specified by the
            'ptopoConfigTrapInterval' object."
   ::= { ptopoMIBTrapPrefix 1 }


-- PTOPO Registration Points
ptopoRegistrationPoints  OBJECT IDENTIFIER ::= { ptopoMIB 3 }

-- values used with ptopoConnDiscAlgorithm object
ptopoDiscoveryMechanisms OBJECT IDENTIFIER ::=
      { ptopoRegistrationPoints 1 }

ptopoDiscoveryLocal      OBJECT IDENTIFIER ::=
      { ptopoDiscoveryMechanisms 1 }


-- conformance information
ptopoConformance OBJECT IDENTIFIER ::= { ptopoMIB 4 }

ptopoCompliances OBJECT IDENTIFIER ::= { ptopoConformance 1 }
ptopoGroups      OBJECT IDENTIFIER ::= { ptopoConformance 2 }


-- compliance statements
ptopoCompliance MODULE-COMPLIANCE
   STATUS  current
    DESCRIPTION
            "The compliance statement for SNMP entities which implement
            the PTOPO MIB."
    MODULE  -- this module
        MANDATORY-GROUPS {
              ptopoDataGroup,
```

```
                ptopoGeneralGroup,
                ptopoConfigGroup,
                ptopoNotificationsGroup
            }
        ::= { ptopoCompliances 1 }

-- MIB groupings
ptopoDataGroup   OBJECT-GROUP
    OBJECTS {
         ptopoConnRemoteChassisType,
         ptopoConnRemoteChassis,
         ptopoConnRemotePortType,
         ptopoConnRemotePort,
         ptopoConnDiscAlgorithm,
         ptopoConnAgentNetAddrType,
         ptopoConnAgentNetAddr,
         ptopoConnMultiMacSASeen,
         ptopoConnMultiNetSASeen,
         ptopoConnIsStatic,
         ptopoConnLastVerifyTime,
         ptopoConnRowStatus
    }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to represent
            physical topology information for which a single agent
            provides management information.

            This group is mandatory for all implementations of the PTOPO
            MIB."
    ::= { ptopoGroups 1 }

ptopoGeneralGroup    OBJECT-GROUP
    OBJECTS {
         ptopoLastChangeTime,
         ptopoConnTabInserts,
         ptopoConnTabDeletes,
         ptopoConnTabDrops,
         ptopoConnTabAgeouts
    }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to report the
            general status of the PTOPO MIB implementation.

            This group is mandatory for all agents which implement the
            PTOPO MIB."
    ::= { ptopoGroups 2 }
```

```
ptopoConfigGroup    OBJECT-GROUP
    OBJECTS {
         ptopoConfigTrapInterval,
         ptopoConfigMaxHoldTime
    }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to configure the
            PTOPO MIB implementation behavior.

            This group is mandatory for agents which implement the PTOPO
            MIB."
    ::= { ptopoGroups 3 }

ptopoNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
         ptopoConfigChange
    }
    STATUS        current
    DESCRIPTION
            "The collection of notifications used to indicate PTOPO MIB
            data consistency and general status information.

            This group is mandatory for agents which implement the PTOPO
            MIB."
    ::= { ptopoGroups 4 }

END
```

5. Intellectual Property

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

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice

this standard. Please address the information to the IETF Executive Director.

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

6. Acknowledgements

The PTOPO Discovery Protocol is a product of the IETF PTOPOMIB Working Group.

7. References

[RFC1155] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[RFC1157] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[RFC1212] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[RFC1215] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[RFC1493] Decker, E., Langille, P., Rijsinghani, A. and K. McCloghrie, "Definitions of Managed Objects for Bridges", RFC 1493, July 1993.

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

[RFC1901] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", January 1996.

[RFC1902] 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.

[RFC1903] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.

[RFC1904] 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.

[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.

[RFC1906] 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.

[RFC2021] Waldbusser, S., "Remote Network Monitoring MIB (RMON-2)", RFC 2021, January 1997.

[RFC2037] McCloghrie, K. and A. Bierman, "Entity MIB using SMIv2", RFC 2037, October 1996.

[RFC2108] de Graaf, K., Romascanu, D., McMaster, D. and K. McCloghrie, "Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIv2", RFC 2108, February 1997.

[RFC2233] McCloghrie, K. and F. Kastenholtz, "The Interfaces Group MIB using SMIv2", RFC 2233, November 1997.

[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.

[RFC2571] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

[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.

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

[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.

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC2737] McCloghrie, K. and A. Bierman, "Entity MIB (Version 2)", RFC 2737, Cisco Systems, December 1999.

8. Security Considerations

There are a number of management objects defined in this MIB that 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.

There are a number of managed objects in this MIB that may contain sensitive information. These are:

read-create objects: ptopoConnRemoteChassisType ptopoConnRemoteChassis ptopoConnRemotePortType ptopoConnRemotePort ptopoConnAgentNetAddrType ptopoConnAgentNetAddr ptopoConnIsStatic ptopoConfigTrapInterval ptopoConfigMaxHoldTime

read-only objects: ptopoConnDiscAlgorithm ptopoConnMultiMacSASeen ptopoConnMultiNetSASeen ptopoConnLastVerifyTime ptopoLastChangeTime

notifications: ptopoConfigChange

These MIB objects expose information about the physical connectivity for a particular portion of a network.

A network administrator may also wish to inhibit transmission of any ptopoConfigChange notification by setting the ptopoConfigTrapInterval object to zero.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, 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 RFC 2574 [RFC2574] and the View-based Access Control Model RFC 2575 [RFC2575] 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.

9. Authors' Addresses

Andy Bierman
Cisco Systems
170 West Tasman Drive
San Jose, CA USA 95134

Phone: +1 408-527-3711
EMail: abierman@cisco.com

Kendall S. Jones
Nortel Networks
4401 Great America Parkway
Santa Clara, CA USA 95054

Phone: +1 408-495-7356
EMail: kejones@nortelnetworks.com

10. Full Copyright Statement

Acknowledgement

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