<draft-bierman-ptopo-mib-proto-00.txt>

Physical Topology MIB and Discovery Protocol Proposal

25 March 1997

Andy Bierman
Cisco Systems Inc.
abierman@cisco.com

Keith McCloghrie
Cisco Systems Inc.
kzm@cisco.com

Status of this Memo

1. Introduction

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

2. The SNMP Network Management Framework

The SNMP Network Management Framework presently consists of three major components. They are:

- the SMI, described in RFC 1902 [1], - the mechanisms used for describing and naming objects for the purpose of management.
- the MIB-II, STD 17, RFC 1213 [2], - the core set of managed objects for the Internet suite of protocols.
- the protocol, RFC 1157 [6] and/or RFC 1905 [4], - the protocol for accessing managed information.

Textual conventions are defined in RFC 1903 [3], and conformance statements are defined in RFC 1904 [5].

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

This memo specifies a MIB module that is compliant to the SNMPv2 SMI. A semantically identical MIB conforming to the SNMPv1 SMI can be produced through the appropriate translation.

2.1. 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 refer to the object type.

3. Overview

There is a need for a standardized way of representing the physical network connections pertaining to a given management domain. A standardized discovery mechanism is also required to increase the likelihood of multi-vendor interoperability of such physical topology

management information.

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

This document contains three main sections:

Physical Topology Discovery Protocol
: The PTOPO Discovery Protocol (PDP) is defined, which provides a simple and interoperable means of supporting the MIB objects defined in the PTOPO MIB.

Entity MIB Extension
: The Entity MIB physical inventory and interface mapping information is utilized in the PTOPO MIB, and an extension module is defined to provide persistent names for physical components.

Physical Topology MIB
: The PTOPO MIB is used for configuring the physical topology function and retrieving learned physical topology information.

3.1. Terms

Some terms are used throughout this document:

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.

Chassis Identifier
: A non-volatile DisplayString, unique within a particular administrative domain, used to name a chassis. Preferably, this is a globally unique string as well.

Local Chassis
: The particular chassis containing an SNMP agent implementing the PTOPO MIB and associated Entity 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)'.

Port Identifier
A port identifier consists of a non-volatile DisplayString, which must be unique within the context of the chassis which contains the port.

Physical Connection Endpoint Identifier
A physical connection endpoint consists of a physical port, which is contained within a single physical chassis. It is distinguished by its chassis identifier and port identifier strings.

Physical Connection
A physical connection consists of two physical ports, each in a different chassis, configured for the purpose of transferring network traffic between the ports. A physical connection is identified by its endpoints.

3.2. 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 physical connection endpoints and the SNMP agent(s) associated with each physical connection endpoint. The PTOPO MIB and discovery protocol provide neighbor discovery. Only physical connections terminating on a local chassis are supported. This allows the MIB and protocol to be bounded and simple, since topology information does not have to be forwarded.

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.

No Functional Overlap
Existing standard MIBs should be utilized whenever possible. Physical topology information is tightly coupled to functionality found in the Interfaces MIB [7] and Entity MIB [8]. New physical

topology MIB objects should not duplicate these MIBs.

Identifier Stability
: Physical connection endpoint identifiers must be persistent (i.e. stable across device reboots). Dynamic primary key objects like ifIndex and entPhysicalIndex are not suitable for representing physical topology information for remote ports.

Low Polling Impact
: Physical topology polling should be minimized through techniques such as TimeFiltered data tables (from RMON-2 [9]), and last-change notifications.

3.3. Persistent Identifiers

The PTOPO MIB utilizes non-volatile identifiers to distinguish individual chassis and port components. These identifiers are associated with entries in the entPhysicalTable, and identified by a new non-volatile name string.

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

3.4. Relationship to Entity MIB

The Entity MIB [8] allows the physical component inventory and hierarchy to be identified. The physical connection component identifiers defined in this MIB are related to entPhysicalTable entries, and the implementation of the entPhysicalTable and probably entAliasMappingTable are required to implement the PTOPO MIB.

The Entity MIB does not provide persistent component identifiers, which are required for the PTOPO MIB. Therefore, an extension to the Entity MIB is defined in this document to provide that feature. The new table augments the entPhysicalTable, and adds a read-only non-volatile identifier for physical components, suitable for supporting the Chassis ID and Port ID requirements of the PTOPO MIB.

3.5. Relationship to Interfaces MIB

The Interfaces MIB 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 maximum flexibility, the Entity MIB is used to identify ports instead of the Interfaces MIB. However, if a port is represented in both MIBs, then an entAliasMappingEntry must also be present, indicating the relationship. For example, if the port is identified as entPhysicalEntry.33 and ifEntry.6, then the instance entAliasMappingIdentifier.33.0 would contain the value 'ifIndex.6'.

3.6. Relationship to RMON-2 MIB

The RMON-2 MIB ([9],[10]) 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. PTOPO Discovery Protocol

This document defines a discovery protocol, suitable for supporting the data requirements of the PTOPO MIB.

The PTOPO Discovery Protocol (PDP) is a media and protocol independent protocol intended to be run on routers, bridges, access servers, switches, etc., allowing a PDP agent to learn SNMP reachability and physical connection endpoint information from adjacent devices.

PDP runs on various media that support Subnetwork Access Protocol (SNAP), and runs over the data-link layer only, allowing two systems running different network layer protocols can learn about each other.

Each device configured with an active PDP Agent sends periodic messages to a multicast MAC address on all physical interfaces enabled for PDP transmission, and listens for PDP messages on all operational ports. Each PDP message contains information identifying the source port as a PTOPO connection endpoint identifier. It also contains at least one network layer address which can be used by an NMS to reach an SNMP agent on the device. Each PDP message contains a configurable time-to-live value, which tells the recipient PDP agent when to discard each element of learned topology information.

4.1. Frame Encapsulation

A OUI value and HDLC protocol value must be chosen to identify PDP messages [TBD].

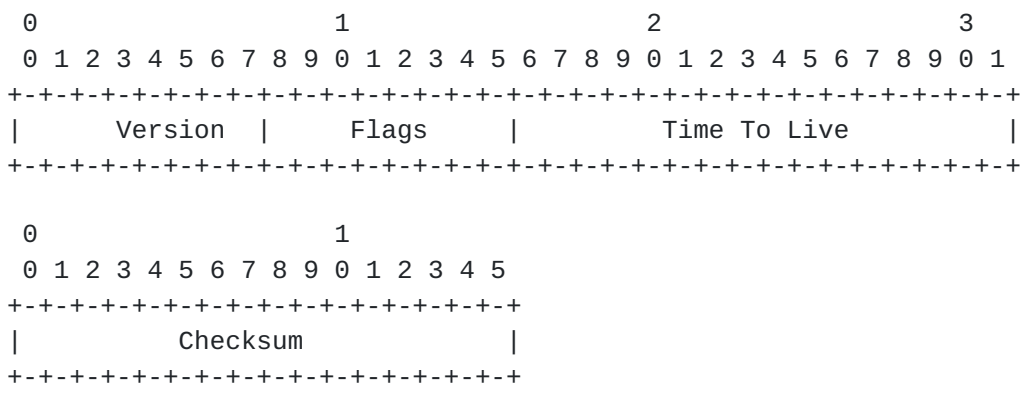
4.2. PDP Message Format

The basic PDP packet consists of a header, followed by a variable number of type/length/value (TLV) triplets, as indicated in Figure 1.

[Figure 1 -- Basic PDP Message Format]

4.2.1. PDP Header Format

The PDP header is a 6 byte header containing 4 fields, as shown in figure 2:



[figure 2 -- PDP Message Format]

The PDP header contains the following fields:

Version
: The PDP protocol version number, set to 0x01 for this version of the protocol.

Flags
: The PDP flags field provide for future header extensions and keep the header word-aligned for easier processing. No flag definition bits are defined at this time. This field must be set to zero in all version 1 PDP messages.

Time to live
: The number of seconds the information in this PDP message should be regarded as valid by the recipient. Agents of the PTOPO MIB must not return MIB information based on expired PDP messages. The valid range is 0 to 65535 for this field.

Checksum
: The one's complement of the one's complement checksum of the entire PDP message. PDP messages containing incorrect checksums must be ignored by the recipient.

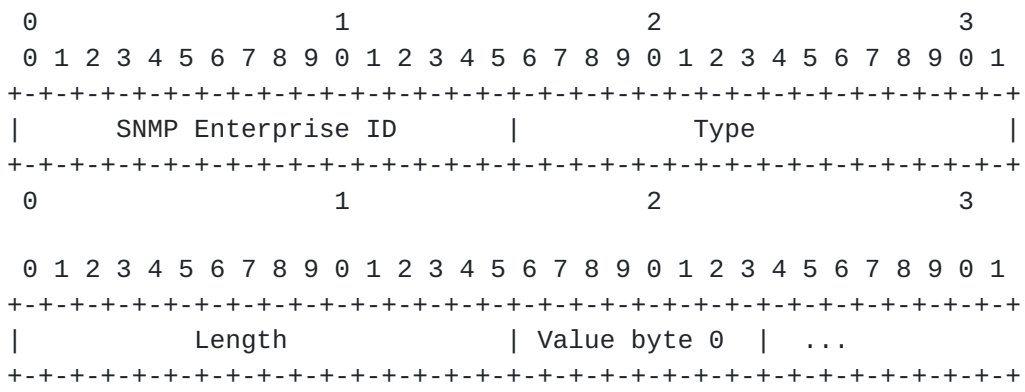
4.2.2. TLV Format

Following the PDP header are a variable number of TLVs, depending on implementation and maximum message size. See figure 3 for TLV field layout.

Type
: This field contains of the 2 byte SNMP Enterprise ID of the naming authority, followed by a 2 byte type identifier.

Length
: This field contains the length, in bytes, of the value field.

Value
: This is a variable-length string [0..65535] bytes, (limited by maximum frame size), of unsigned characters.



[Figure 3 - TLV Format]

The header fields are defined as follows:

SNMP Enterprise ID
: The identifier distinguishing the naming authority defining this TLV, as defined by IANA in the Assigned Numbers Document [11].

Type
: The integer value identifying the type of information contained in the value field.

Length
: The length, in bytes, of the value field to follow.

Value
A variable-length octet-string containing the instance-specific information for this TLV.

4.3. Standard TLV Definitions

The standard PDP TLVs allow for a PDP agent to implement the PTOPO MIB for physical connections terminating on the local chassis.

The following table summarizes the TLVs defined in this document.

Enterprise	Type	TLV Name	Example Usage
IETF	1	Chassis ID	"acme.rg1000-switch.0000c07cf297"
IETF	2	Port ID	"eth0/0/0"
IETF	3	Mgmt Address	{ ipV4(1), 4, '0x01020304' }

[Figure 4 - TLV Summary]

4.3.1. Chassis ID TLV

The Chassis ID is a mandatory TLV which identifies the chassis component of the endpoint identifier associated with the transmitting PDP agent.

It is a DisplayString, length [1..32] bytes, representing the entPhysicalNVName value for the chassis containing the PDP Agent.

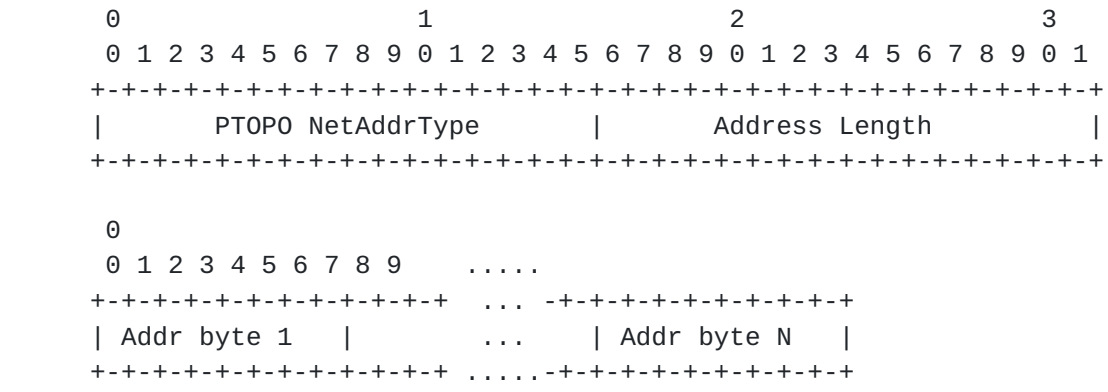
4.3.2. Port ID TLV

The Port ID is a mandatory TLV which identifies the port component of the endpoint identifier associated with the transmitting PDP agent.

It is a DisplayString, length [1..32] bytes, representing the entPhysicalNVName value identifying the source transmission port for a PDP message.

4.3.3. Management Address TLV

The Management Address is a mandatory TLV which identifies a network address associated with the local PDP agent, which can be used to reach an SNMP agent associated with the chassis identified in the Chassis ID TLV.



[Figure 5 -- Management Address TLV Format]

The Management Address fields are defined as follows:

PTOPO NetAddrType
The enumerated value for the network address type identified in this TLV.

Address Length
The number of bytes contained in the address string to follow.

Address
The binary string containing the network management address.

4.4. Protocol Operation

An active PDP Agent must transmit PDP messages, process received PDP messages, and maintain an instance of the PTOPO MIB containing the information learned from received PDP messages.

During processing of received PDP messages, a PDP Agent must skip and ignore TLVs unknown to the agent.

4.4.1. Message Transmission

An active PDP agent must transmit a PDP message out each interface configured for PDP transmission, once each time interval specified in the pdpMessageTxInterval MIB object. Actual transmission intervals are jittered to prevent synchronization effects. Each message is sent with a time-to-live field equal to the value of pdpMessageTxHoldTime MIB object, and must contain at least the three mandatory IETF TLVs supporting the PTOPO MIB. Additional proprietary TLVs may be added, as maximum frame size permits.

4.4.2. Message Processing

Upon reception of a PDP message, and verifying the message checksum to be correct, the TLV information is extracted, and relevant PTOPO MIB information is updated. If an entry is added, deleted, or modified, the appropriate TimeFilter and last change time internal variables are updated to signal the change to an NMS.

PDP messages must not be forwarded by the receiving PDP Agent.

4.4.3. Interface Startup Procedure

In the event an interface becomes operationally enabled and enabled for PDP message transmission, the initial PDP message is generated right away, and it is transmitted three times, at one second intervals. This reduces the convergence delay due to lost packets during system startup.

4.4.4. Interface Shutdown Procedure

In the event an interface becomes administratively disabled, and/or disabled for PDP message transmission, a final PDP message is transmitted with a time to live value of zero, before the interface is disabled.

Upon reception of such a PDP message, a PDP Agent must remove information in the PTOPO MIB associated with the indicated remote physical connection endpoint.

5. Entity MIB Extensions

The Entity MIB is used to identify chassis and port components, and component relationships for one or more chassis 'component-trees'.

This document defines an extension to the Entity MIB, which augments the entPhysicalTable and provides a source for non-volatile string-based component identifiers, suitable for use in an implementation of the PTOPO MIB.

5.1. Entity Physical Extensions Group

This group contains a single table, called the entPhysicalXTable, which augments the entPhysicalTable. Each entPhysicalXEntry provides a DisplayString which can be used by an NMS as a non-volatile alias string for the physical component.

5.2. EntityX MIB Definitions

```
ENTITY-MIB-EXTENSIONS DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, RowStatus, DisplayString
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    entPhysicalEntry
        FROM ENTITY-MIB;

entityXMIB MODULE-IDENTITY
    LAST-UPDATED "9703170000Z"
    ORGANIZATION "Cisco Systems, Inc."
    CONTACT-INFO
                    "Andy Bierman
                    Cisco Systems Inc.
                    170 West Tasman Drive
                    San Jose, CA 95134
                    408-527-3711
                    abierman@cisco.com

                    Keith McCloghrie
```

Cisco Systems Inc.
170 West Tasman Drive
San Jose, CA 95134
408-526-5260
kzm@cisco.com"

```
    DESCRIPTION
            "The extension MIB module for physical entity information."
    ::= { experimental xx }

--  ********************************************************
--
--     E N T I T Y     P H Y S I C A L    E X T E N S I O N S
--
--  ********************************************************

-- entPhysicalTable extensions

entityXMIBObjects  ::= OBJECT IDENTIFIER { entityXMIB 1 }
entityPhysicalX    ::= OBJECT IDENTIFIER { entityXMIBObjects 1 }

entPhysicalXTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EntPhysicalXEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per physical element
            represented in the entPhysicalTable."
    ::= { entityPhysicalX 1 }

entPhysicalXEntry       OBJECT-TYPE
    SYNTAX      EntPhysicalXEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular physical entity.

            Each entry provides an object (entPhysicalNVName) to help an
            NMS uniquely identify a physical entity with a DisplayString
            stored in non-volatile and re-created after a reboot."
    AUGMENTS   { entPhysicalEntry }
    ::= { entPhysicalXTable 1 }

EntPhysicalXEntry ::= SEQUENCE {
      entPhysicalNVName      DisplayString
}
```

```
entPhysicalNVName    OBJECT-TYPE
    SYNTAX      DisplayString (SIZE(1..32))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object is a non-volatile name for the physical entity.

            On the first instantiation of an physical entity, the value
            of entPhysicalNVName is configured by the agent to a string
            of suitable uniqueness for the indicated component type.

            For components with an associated entPhysicalClass value of
            'chassis(3)', this object should be set to a string that is
            unique within the administrative domain, and preferably
            globally unique.

            For components with an associated entPhysicalClass value
            other than 'chassis(3)', this object should be set to a
            string that is unique within the particular chassis which
            contains the component.

            The value in the entPhysicalNVName instance must be
            associated with the same physical entity for as long as that
            entity remains instantiated, including across all re-
            initializations/reboots of the network management system,
            including those which result in a change of the physical
            entity's entPhysicalIndex value."
    ::= { entPhysicalXEntry 1 }

-- conformance information
entityXConformance OBJECT IDENTIFIER ::= { entityXMIB 2 }

entityXCompliances OBJECT IDENTIFIER ::= { entityXConformance 1 }
entityXGroups      OBJECT IDENTIFIER ::= { entityXConformance 2 }

-- compliance statements

entityXPhysicalCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "The compliance statement for SNMP entities which implement
            the entPhysicalXTable Entity MIB extension."
    MODULE  -- this module
        MANDATORY-GROUPS { entityPhysicalXGroup }
    ::= {  entityXCompliances 1 }
```

```
-- MIB groupings
entityPhysicalXGroup    OBJECT-GROUP
    OBJECTS {
              entPhysicalNVName
            }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to represent
            extended physical component information for which a single
            agent provides management information."
    ::= { entityXGroups 1 }

END
```

6. Physical Topology MIB

The Physical Topology MIB provides information about remote ports (either learned or configured) and physical connections between local ports and remote ports.

Since the PTOPO MIB utilizes the Entity MIB and EntityX MIB, multiple chassis components can be supported by a single SNMP agent, but only one SNMP agent per chassis is supported by the PTOPO MIB.

Physical connections between ports on devices represented by the same Entity MIB implementation should be modeled in the Entity MIB instead of the PTOPO MIB.

For performance reasons, the identifier strings for local components are replaced with the entPhysicalIndex mappings whenever used as an index value. The PTOPO MIB agent and Entity MIB agent represent the same physical resources, and therefore are considered to fate-share (i.e. reset together upon a reinitialization of the management system).

6.1. PTOPO MIB Structure

The PTOPO MIB contains five MIB 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.

ptopoPdpConfig
: Contains configuration variables for the PTOPO Discovery Protocol Agent function.

ptopoNotifications
: Contains trap definitions transmitted on behalf of the PTOPO MIB Agent.

6.1.1. ptopoData Group

This group contains two tables to identity physical topology data.

The ptopoPortTable contains information about the remote physical connection endpoints learned or configured on behalf of the PTOPO MIB SNMP Agent.

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

6.1.2. 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 ptopoData group polling.

6.1.3. ptopoConfig Group

This group contains objects to configure the behavior of the physical topology function. The transmission of ptopoLastChange traps can be configured using the ptopoConfigTrapsEnabled scalar MIB object.

6.1.4. ptopoPdpConfig Group

This group contains objects to configure the behavior of the PTOPO Discovery Protocol (PDP) Agent function. The protocol can be globally enabled or disabled. Transmission of PDP messages can also be enabled or disabled on individual interfaces.

This group is implemented only by SNMP Agents also acting as PDP Agents.

6.1.5. ptopoNotifications Group

This group contains notification definitions relating to the overall status of the PTOPO MIB agent.

A single trap is defined, the ptopoConfigChange trap, indicating any modification of the ptopoPortTable or ptopoConnTable.

6.2. Physical Topology MIB Definitions

```
PTOPO-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-IDENTITY, OBJECT-TYPE,
    NOTIFICATION-TYPE, Counter32, Integer32
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, RowStatus, DisplayString,
    TimeStamp, TruthValue, AutonomousType
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF
    TimeFilter
        FROM RMON2-MIB;

ptopoMIB MODULE-IDENTITY
    LAST-UPDATED "9703250000Z"
    ORGANIZATION "Cisco Systems, Inc."
    CONTACT-INFO
                    "Andy Bierman
                    Cisco Systems Inc.
                    170 West Tasman Drive
                    San Jose, CA 95134
                    408-527-3711
                    abierman@cisco.com

                    Keith McCloghrie
                    Cisco Systems Inc.
                    170 West Tasman Drive
                    San Jose, CA 95134
                    408-526-5260
                    kzm@cisco.com"
    DESCRIPTION
            "The MIB module for physical topology information."
    ::= { experimental xx }

ptopoMIBObjects   OBJECT IDENTIFIER ::= { ptopoMIB 1 }

-- MIB groups
ptopoData         OBJECT IDENTIFIER ::= { ptopoMIBObjects 1 }
ptopoGeneral      OBJECT IDENTIFIER ::= { ptopoMIBObjects 2 }
ptopoConfig       OBJECT IDENTIFIER ::= { ptopoMIBObjects 3 }
ptopoPdpConfig    OBJECT IDENTIFIER ::= { ptopoMIBObjects 4 }
```

```
-- textual conventions

--
-- NetAddrType TC
--
-- Enumerations distinguishing network-layer address types
-- Eventually, they might be included from a general TC module
--
NetAddrType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
            "An enumeration identifying a network address type."
    SYNTAX      INTEGER {
                                ipV4 (1),
                                decnet (2),
                                pup (3),
                                chaos (4),
                                xns (5),
                                x121 (6),
                                appletalk (7),
                                clns (8),
                                lat (9),
                                vines (10),
                                cons (11),
                                apollo (12),
                                stun (13),
                                novell (14),
                                qllc (15),
                                snapshot (16),
                                atmIlmi (17),
                                bstun (18),
                                x25pvc (19),
                                ipV6(20),
                                unknown (65535)
                        }

NetAddr ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
            "Octet string representing a network layer address.  The
            length and format of the address is protocol dependent as
            follows:
                ipV4            4 octets
                decnet          2 octets
                pup             obsolete
```

```
                chaos           2 octets
                xns             10 octets
                                first 4 octets are the net number
                                last 6 octets are the host number
                x121
                appletalk       3 octets
                                first 2 octets are the net number
                                last octet is the host number
                clns
                lat
                vines           6 octets
                                first 4 octets are the net number
                                last 2 octets are the host number
                cons
                apollo          10 octets
                                first 4 octets are the net number
                                last 6 octets are the host number
                stun            8 octets
                novell          10 octets
                                first 4 octets are the net number
                                last 6 octets are the host number
                qllc            6 octets
                bstun           1 octet - bi-sync serial tunnel
                snapshot        1 octet
                atmIlmi         4 octets
                x25 pvc         2 octets (12 bits)
                ipV6            16 octets"
    SYNTAX      OCTET STRING (SIZE (0..20))

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

-- Port Name Table
ptopoPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PtopoPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port identifier known to
            this agent."
    ::= { ptopoData 1 }
```

```
ptopoPortEntry      OBJECT-TYPE
    SYNTAX      PtopoPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular remote physical port.
            Entries may be created and deleted in this table, either
            manually or by the agent (if a physical topology discovery
            process is active)."
    INDEX   { ptopoPortTimeMark, ptopoChassisID, ptopoPortID }
    ::= { ptopoPortTable 1 }

PtopoPortEntry ::= SEQUENCE {
      ptopoPortTimeMark            TimeFilter,
      ptopoChassisID               DisplayString,
      ptopoPortID                  DisplayString,
      ptopoAgentNetAddrType        NetAddrType,
      ptopoAgentNetAddr            NetAddr,
      ptopoPortDiscAlgorithm       AutonomousType,
      ptopoPortRowStatus           RowStatus
}

ptopoPortTimeMark  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."
    ::= { ptopoPortEntry 1 }

ptopoChassisID  OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The non-volatile identifier string for the indicated
            chassis.

            Note that this string is used to identify the chassis, not a
            particular agent containing management information on behalf
            of the chassis. All agents representing the same chassis
            information must identify the chassis with the same value of
            ptopoChassisID.
```

```
            This object refers to the remote entPhysicalEntry with the
            same value of entPhysicalNVName as this ptopoChassisID
            value."
    ::= { ptopoPortEntry 2 }

ptopoPortID     OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The non-volatile identifier string for the indicated port.
            Note that this string must be unique only within the scope
            of a particular chassis. All agents representing the same
            port information must identify the port with the same value
            of ptopoChassisID and ptopoPortID.

            This object refers to the remote entPhysicalEntry with the
            same value of entPhysicalNVName as this ptopoPortID value."
    ::= { ptopoPortEntry 3 }

ptopoAgentNetAddrType  OBJECT-TYPE
    SYNTAX      NetAddrType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "This object identifies the network address type of the
            ptopoAgentNetAddr object.

            This object may not be modified if the associated
            ptopoPortRowStatus object has a value of active(1)."
    ::= { ptopoPortEntry 4 }

ptopoAgentNetAddr  OBJECT-TYPE
    SYNTAX      NetAddr
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "This object identifies a network address which may be used
            to reach an SNMP agent entity on the indicated port.

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

ptopoPortDiscAlgorithm OBJECT-TYPE
```

```
    SYNTAX      AutonomousType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "An indication of the algorithm used to discover this
            information.

            Valid values include the following OBJECT IDENTIFIERs:

            A value of ptopoDiscoveryPDP indicates this entry was
            configured using the PTOPO Discovery Protocol.

            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.

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

ptopoPortRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The status of this conceptual row."
    ::= { ptopoPortEntry 7 }


-- Physical Connection Table
ptopoConnTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PtopoConnEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per physical network connection
            known to this agent. The agent must ensure that duplicate
            connections are not present in the table at any time."
    ::= { ptopoData 2 }

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,
              ptopoConnChassis1,
              ptopoConnPort1,
              ptopoConnChassis2,
              ptopoConnPort2 }
    ::= { ptopoConnTable 1 }

PtopoConnEntry ::= SEQUENCE {
      ptopoConnTimeMark            TimeFilter,
      ptopoConnChassis1            Integer32,
      ptopoConnPort1               Integer32,
      ptopoConnChassis2            DisplayString,
      ptopoConnPort2               DisplayString,
      ptopoConnDiscAlgorithm       AutonomousType,
      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 }

ptopoConnChassis1  OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the value of entPhysicalIndex used
            to represent the particular local chassis component, which
            is associated with the first endpoint in this physical
            connection."
    ::= { ptopoConnEntry 2 }

ptopoConnPort1     OBJECT-TYPE
```

```
    SYNTAX      Integer32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the value of entPhysicalIndex used
            to represent the particular local port component, which is
            associated with the first endpoint in this physical
            connection."
    ::= { ptopoConnEntry 3 }

ptopoConnChassis2  OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The chassis identifier string for the remote chassis
            associated with the second endpoint in this physical
            connection.

            This value will contain the same value as exactly one
            instance of the entPhysicalNVName object on the remote agent
            representing the remote chassis."
    ::= { ptopoConnEntry 4 }

ptopoConnPort2     OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The port ID string for the port associated with the second
            endpoint in this physical connection.

            This value will contain the same value as exactly one
            instance of the entPhysicalNVName object on the remote agent
            representing the remote port, which is contained in the same
            chassis as identified by the ptopoConnChassis2 object."

    ::= { ptopoConnEntry 5 }

ptopoConnDiscAlgorithm OBJECT-TYPE
    SYNTAX      AutonomousType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "An indication of the algorithm used to discover this
```

```
            information.

            A value of ptopoDiscoveryPDP indicates this entry was
            configured using the PTOPO Discovery Protocol.

            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.

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

ptopoConnRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The status of this conceptual row.

            If the agent is capable of non-volatile storage of the
            ptopoConnTable, and the active entry was configured
            manually, then this entry must be restored after a re-
            initialization of the management system."
    ::= { ptopoConnEntry 7 }

--  ********************************************************
--
--           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 was
            last created, modified, or deleted in any of these tables:
                    - ptopoPortTable
                    - ptopoConnTable
```

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

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

ptopoPortTabDeletes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of times an entry has been deleted from the
            ptopoPortTable."
    ::= { ptopoGeneral 3 }

ptopoConnTabInserts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of times an entry has been inserted into the
            ptopoConnTable."
    ::= { ptopoGeneral 4 }

ptopoConnTabDeletes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of times an entry has been deleted from the
            ptopoConnTable."
    ::= { ptopoGeneral 5 }

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

```
ptopoConfigTrapsEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This object controls the transmission of PTOPO
            notifications.

            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 { true }
    ::= { ptopoConfig 1 }

--
--  ********************************************************
--
--           P T O P O    P D P    C O N F I G
--
--  ********************************************************
--
--           The Physical Topology Discovery Protocol Configuration Group

pdpVersion      OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The version number used in PDP messages transmitted on
            behalf of this PDP Agent."
    ::= { ptopoPdpConfig 1 }

pdpAgentEnabled      OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The current PDP Agent status.

            If this object has a value of 'true(1)', then the PDP Agent
            will transmit PDP messages for the enabled ports, and
            process messages received from other PDP Agents.

            If this object has a value of 'false(2)', then the PDP Agent
```

```
            will not transmit any PDP messages, and will not process
            messages received from other PDP Agents.

            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."
    ::= { ptopoPdpConfig 2 }

pdpMessageTxInterval OBJECT-TYPE
    SYNTAX      Integer32 (5..32768)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The interval at which PDP messages are transmitted on
            behalf of this agent.

            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       { 60 }
    ::= { ptopoPdpConfig 3 }

pdpMessageTxHoldTime OBJECT-TYPE
    SYNTAX      Integer32 (10..65535)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The time-to-live value used in PDP messages transmitted on
            behalf of this agent.

            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       { 180 }
    ::= { ptopoPdpConfig 4 }

pdpTxSuppressTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF PdpTxSuppressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
            "A table for suppressing PDP message transmission on
            individual ports."
    ::= { ptopoPdpConfig 5 }

pdpTxSuppressEntry   OBJECT-TYPE
    SYNTAX      PdpTxSuppressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "PDP message transmission suppression configuration
            information for the indicated port.  The port must be
            contained in the same chassis as the PDP agent. PDP messages
            will not be transmitted on the indicated port, even if the
            port is enabled (e.g., ifOperStatus = 'up(1)').

            If the agent is capable of storing non-volatile
            configuration, then each active pdpTxSuppressEntry must be
            re-created after a re-initialization of the management
            system.

            Only entries pertaining to a local chassis may be created in
            this table."
    INDEX  { pdpTxSuppressPortIndex }
    ::= { pdpTxSuppressTable 1 }

PdpTxSuppressEntry ::= SEQUENCE {
      pdpTxSuppressPortIndex      Integer32,
      pdpTxSuppressRowStatus      RowStatus
}

pdpTxSuppressPortIndex    OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the value of entPhysicalIndex used
            to represent the particular local port component associated
            with this PDP message configuration.

            PDP messages are not to be transmitted on the indicated port
            if this entry is in the active state."
    ::= { pdpTxSuppressEntry 1 }

pdpTxSuppressRowStatus  OBJECT-TYPE
```

```
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
            "The status of this entry."
    ::= { pdpTxSuppressEntry 2 }

--
--  ********************************************************
--
--       P T O P O    D I S C O V E R Y   A L G O R I T M S
--
--  ********************************************************
--
--     The Physical Topology Discovery Types

ptopoDiscoveryTypes OBJECT IDENTIFIER ::= { ptopoMIB 2 }

ptopoDiscoveryPDP OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
            "Indicates the associated PTOPO MIB element was discovered
            using Version 1 of the PTOPO Discovery Protocol."
    ::= { ptopoDiscoveryTypes 1 }

ptopoDiscoveryLocal OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
            "Indicates the associated PTOPO MIB element was not
            discovered, but rather configured using unspecified means by
            the local agent.  This enumeration is not used if the PTOPO
            management element was configured as a result of SNMP Set
            operations."
    ::= { ptopoDiscoveryTypes 2 }

--
--  ********************************************************
--
--              P T O P O    N O T I F I C A T I O N S
--
--  ********************************************************
--
--     The Physical Topology Notification Group

ptopoMIBTraps       OBJECT IDENTIFIER ::= { ptopoMIB 3 }
```

```
ptopoMIBTrapPrefix  OBJECT IDENTIFIER ::= { ptopoMIBTraps 0 }

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

            An agent must not generate more than one ptopoConfigChange
            'trap-event' in a five second period, where a 'trap-event'
            is the transmission of a single trap PDU to a list of trap
            destinations.  If additional configuration changes occur
            within the five second 'throttling' period, then these
            trap-events should 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."
   ::= { ptopoMIBTrapPrefix 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 {
              ptopoAgentNetAddrType,
```

```
              ptopoAgentNetAddr,
              ptopoPortDiscAlgorithm,
              ptopoPortRowStatus,
              ptopoConnLastChangeTime,
              ptopoConnDiscAlgorithm,
              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,
              ptopoPortTabInserts,
              ptopoPortTabDeletes,
              ptopoConnTabInserts,
              ptopoConnTabDeletes
            }
    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 {
              ptopoConfigTrapsEnabled
            }
    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 }

ptopoPdpConfigGroup    OBJECT-GROUP
    OBJECTS {
                pdpVersion,
                pdpAgentEnabled,
                pdpMessageTxInterval,
                pdpMessageTxHoldTime,
                pdpTxSuppressRowStatus
            }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to configure the
            PTOPO Discovery Protocol Agent behavior.

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

ptopoNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { ptopoConfigChange }
    STATUS        current
    DESCRIPTION
            "The collection of notifications used to indicate PTOPO MIB
            data consistency and general status information."
    ::= { ptopoGroups 5 }

END
```

7. Acknowledgements

This document is based on the Cisco Discovery Protocol (CDP) [12], developed at Cisco Systems by Dino Farinacci and Keith McCloghrie.

8. References

[1] SNMPv2 Working Group, 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.

[2] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17,

RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.

[3] SNMPv2 Working Group, 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.

[4] SNMPv2 Working Group, 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.

[5] SNMPv2 Working Group, 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.

[6] Case, J., M. Fedor, M. Schoffstall, J. Davin, "Simple Network Management Protocol", RFC 1157, SNMP Research, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[7] McCloghrie, K., and Kastenholtz, F., "Interfaces Group Evolution", RFC 1573, Hughes LAN Systems, FTP Software, January 1994.

[8] McCloghrie, K., and Bierman, A., "Entity MIB Using SMIv2", RFC 2037, Cisco Systems, October 1996.

[9] Waldbusser S., "Remote Network Monitoring MIB Version 2 using SMIv2", RFC 2021, INS, January 1997.

[10] Bierman A., and Iddon, R., "Remote Network Monitoring MIB Protocol Identifiers", RFC 2074, Cisco Systems, 3Com/AXON Networks, January 1997.

[11] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.

[12] Farinacci, D. & McCloghrie, K., "Cisco Discovery Protocol (CDP)", Internal Cisco Document, Cisco Systems, August 1996.

9. Security Considerations

This document defines mechanisms which can potentially expose physical topology and connectivity information pertaining to particular networks.

A network administrator should take care to restrict PTOPO Discovery

Protocol message transmission and PTOPO MIB trap transmission to interfaces deemed appropriate to carry packets containing such information.

A network administrator should also utilize access control to prevent inappropriate manual configuration of the writable objects defined in this document.

10. Authors' Addresses

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

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
Phone: 408-526-5260
Email: kzm@cisco.com

Table of Contents

1 Introduction 1
2 The SNMP Network Management Framework 2
2.1 Object Definitions 2
3 Overview 2
3.1 Terms 3
3.2 Design Goals 4
3.3 Persistent Identifiers 5
3.4 Relationship to Entity MIB 5
3.5 Relationship to Interfaces MIB 6
3.6 Relationship to RMON-2 MIB 6
4 PTOPO Discovery Protocol 7
4.1 Frame Encapsulation 7
4.2 PDP Message Format 7
4.2.1 PDP Header Format 8
4.2.2 TLV Format 9
4.3 Standard TLV Definitions 10
4.3.1 Chassis ID TLV 10
4.3.2 Port ID TLV 10
4.3.3 Management Address TLV 11
4.4 Protocol Operation 11
4.4.1 Message Transmission 12
4.4.2 Message Processing 12
4.4.3 Interface Startup Procedure 12
4.4.4 Interface Shutdown Procedure 12
5 Entity MIB Extensions 13
5.1 Entity Physical Extensions Group 13
5.2 EntityX MIB Definitions 13
6 Physical Topology MIB 17
6.1 PTOPO MIB Structure 17
6.1.1 ptopoData Group 18
6.1.2 ptopoGeneral Group 18
6.1.3 ptopoConfig Group 18
6.1.4 ptopoPdpConfig Group 18
6.1.5 ptopoNotifications Group 18
6.2 Physical Topology MIB Definitions 19
7 Acknowledgements 35
8 References 35
9 Security Considerations 36
10 Authors' Addresses 37