Hubmib and AToMMIB Working Groups
INTERNET DRAFT

Mike Ayers
BMC Software, Inc.
John Flick
Hewlett-Packard Company
C. M. Heard
Consultant
Kam Lam
Lucent Technologies
Kerry McDonald
CSU San Bernardino
K. C. Norseth
Enterasys Networks
Kaj Tesink
Telcordia Technologies
November 20, 2001

Definitions of Managed Objects for the Ethernet WAN Interface Sublayer

<draft-ietf-hubmib-wis-mib-01.txt>

Status of this Memo

Copyright Notice

Internet Draft <draft-ietf-hubmib-wis-mib-01.txt> 11/20/2001

1. Abstract

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP based internets. In particular, it defines objects for managing the Ethernet Wide Area Network (WAN) Interface Sublayer (WIS) [P802.3ae].

The MIB module defined in this memo is implemented in conjunction with the Ethernet-like Interface MIB [ETHERIF], the 802.3 Medium Attachment Unit MIB [MAU-MIB], the Interfaces Group MIB [RFC2863], and the Inverted Stack Table MIB [RFC2864]. It also extends the SONET MIB [SONETng] and is implemented in conjunction with that MIB module.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [RFC2571].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [RFC1155], STD 16, RFC 1212 [RFC1212] and 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].

- o 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.

3. Overview

The objects defined in this memo are used in conjunction with objects defined in the Interfaces Group MIB [RFC2863], the SONET MIB [SONETng], and the MAU MIB [MAU-MIB] to manage the WAN Interface Sublayer (WIS) defined in [P802.3ae]. The WIS contains functions to perform OC-192c/VC-4-64c framing and scrambling. It resides between the PCS and PMA sublayers within a 10GBASE-W 10 Gb/s WAN-compatible PHY and may be used in conjunction with any of the PCS, PMA, and PMD sublayers that are defined in [P802.3ae] for 10GBASE-W PHYs. Three types of 10GBASE-W PHYs are defined, distinguished by the type of optics employed: 10GBASE-SW, 10GBASE-LW, and 10GBASE-EW. The objects defined in this memo may be used to manage an Ethernet interface employing any type of 10GBASE-W PHY. They do not apply to any other kind of interface. In particular, they do not apply to

so-called Ethernet Line Terminating Equipment (ELTE) residing within a SONET network element that uses the 10GBASE-W PMA/PMD sublayers but otherwise acts as SONET Line Terminating Equipment (LTE).

The objects presented here -- along with those incorporated by reference from the Interfaces Group MIB, the SONET MIB, and the MAU MIB -- are intended to provide exact representations of the mandatory attributes in the oWIS managed object class (i.e., the members of the pWISBasic package) defined in Clause 30 and Annex 30A of [P802.3ae]. They are also intended to provide approximate representations of the optional attributes (i.e., the members of the pWISOptional package). Some objects with no analogues in oWIS are defined to support WIS testing features required by Clause 50 of [P802.3ae].

3.1. Relationship to the SONET MIB

Since the Ethernet WAN Interface Sublayer was designed to be SONET-compatible, information similar to that provided by most of the members of the oWIS managed object class is available from objects defined in the SONET MIB [SONETng]. Thus, the MIB module defined in this memo is a sparse augmentation of the SONET MIB -- in other words, every table defined here is an extension of some table in the SONET MIB. An agent implementing the objects defined in this memo MUST implement the objects required by the sonetCompliance2 conformance statement in the SONET MIB, and as further detailed in the conformance statement in the MIB module defined in this memo.

It should be noted that some of the objects incorporated by reference from the SONET MIB -- specifically, the threshold objects and interval counter objects -- provide only approximate representations of the corresponding oWIS attributes, as detailed in Section 3.6. An alternative approach would have been to define new objects to exactly match the oWIS definitions. That approach was rejected because the SONET MIB objects are already used in deployed systems to manage the SONET sublayers of ATM over SONET and PPP over SONET interfaces, and it was deemed undesirable to use a different scheme to manage the SONET sublayers of 10 Gb/s WAN-compatible Ethernet interfaces. Note that the approach adopted by this memo requires no hardware support beyond that mandated by [P802.3ae] subclause 50.3.10.

3.2. Relationship to the Ethernet-like Interfaces MIB

An interface which includes the Ethernet WIS is, by definition, an

Ethernet-like interface, and an agent implementing the objects defined in this memo MUST implement the objects required by the dot3Compliance2 compliance statement in the EtherLike-MIB.

3.3. Relationship to the 802.3 MAU MIB

Support for the mauModIfCompl2 compliance statement of the MAU-MIB [MAU-MIB] is REQUIRED for all Ethernet-like interfaces. The MAU-MIB is needed in order to allow applications to control and/or determine the media type in use. That is important for devices than can support both the 10GBASE-R 10 Gb/s LAN format (which does not include the WIS) and the 10GBASE-W 10 Gb/s WAN format (which does include the WIS). The MAU-MIB also provides the means to put a device in standby mode or to reset it; the latter may be used to re-initialize the WIS.

3.4. Use of the ifTable

This section specifies how the ifTable, as defined in [RFC2863], is used for the Ethernet WIS application.

3.4.1. Layering Model

Ethernet interfaces that employ the WIS are layered as defined in [P802.3ae]. The corresponding use of the ifTable [RFC2863] is shown in the figure below.

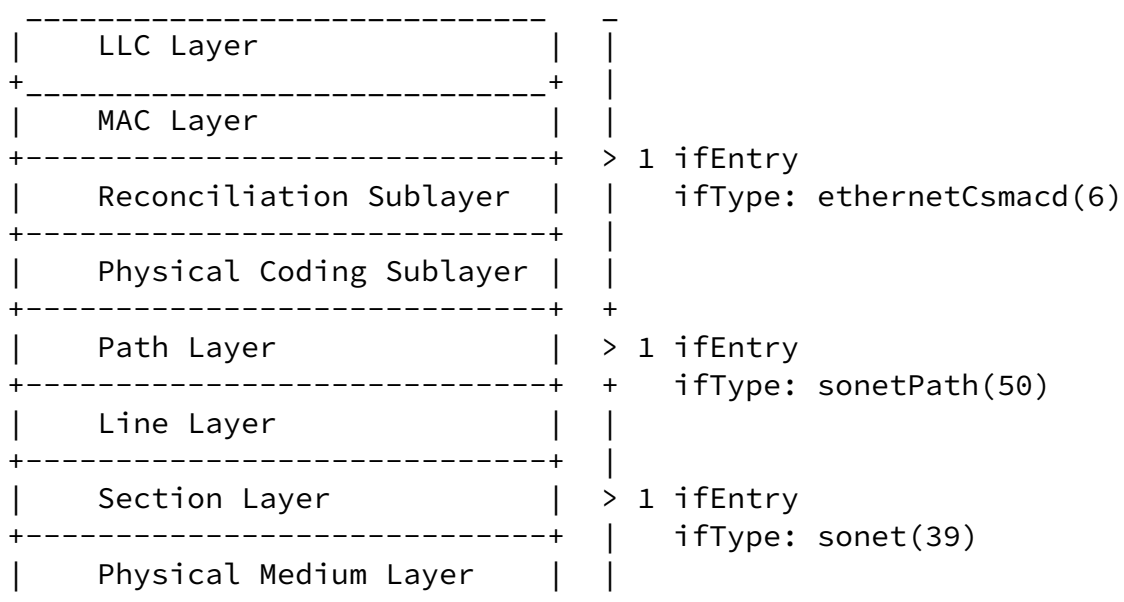

Figure 1 - Use of ifTable for an Ethernet WIS port

The exact configuration and multiplexing of the layers is maintained in the ifStackTable [RFC2863] and in the ifInvStackTable [RFC2864].

3.4.2. Use of ifTable for LLC Layer/MAC Layer/Reconciliation Sublayer/Physical Coding Sublayer

The ifTable MUST be used as specified in [ETHERIF] and [MAU-MIB] for the LLC Layer/MAC Layer/Reconciliation Sublayer/Physical Coding Sublayer.

3.4.3. Use of ifTable for SONET/SDH Path Layer

The ifTable MUST be used as specified in [SONETng] for the SONET/SDH Path Layer. The value of ifHighSpeed is set to 9585. ifSpeed reports a value of 4294967295.

3.4.4. Use of ifTable for SONET/SDH Medium/Section/Line Layer

The ifTable MUST be used as specified in [SONETng] for the SONET/SDH Medium/Section/Line Layer. The value of ifHighSpeed is set to 9953. ifSpeed reports a value of 4294967295.

3.5. SONET/SDH Terminology

The SONET/SDH terminology used in [P802.3ae] is mostly the same as in [SONETng], but there are a few differences. In those cases the definitions in [P802.3ae] take precedence. The specific differences are as follows.

Unequipped
: This defect is not defined by [P802.3ae]. An implementation that supports it SHOULD report it by setting the sonetPathUnequipped bit in the appropriate instance of sonetPathCurrentStatus.

Signal Label Mismatch
: This defect is called Payload Label Mismatch (PLM) in [P802.3ae]. It is reported by setting both the sonetPathSignalLabelMismatch

bit in the appropriate instance of sonetPathCurrentStatus (defined in [SONETng]) and the etherWisPathPLM bit in the corresponding instance of etherWisPathCurrentStatus (defined below).

Loss of Codegroup Delineation
[P802.3ae] defines Loss of Codegroup Delineation (LCD) as occurring when the Physical Coding Sublayer is unable to locate 64B/66B code group boundaries. There is no analogous defect defined in [SONETng]. It is reported by setting the etherWisPathLCD bit in the appropriate instance of the object etherWisPathCurrentStatus defined below.

STS-Path Remote Defect Indication
[P802.3ae] mandates the use of ERDI-P (Enhanced Remote Defect Indication - Path) defined in [T1.231] to signal remote server defects (triggered by path AIS or path LOP) and remote payload defects (triggered by Payload Label Mismatch or Loss of Codegroup Delineation). [SONETng] defines the one-bit RDI-P (Remote Defect Indication - Path), which signals remote server detects (i.e., path AIS and path LOP) only. An implementation of the WIS MUST set the sonetPathSTSRDI bit in the appropriate instance of sonetPathCurrentStatus when it receives an ERDI-P server defect indication from the remote end. Both ERDI-P payload defects and ERDI-P server defects are reported in the object etherWisFarEndPathCurrentStatus defined below.

Path Coding Violations
In [P802.3ae] the path layer CV count is based on block errors and not BIP-8 errors, i.e., it is incremented only once for each B3 byte that indicates incorrect parity, regardless of the number of bits in error. Note that Section 8.4.5.1 of [T1.231] allows either path BIP-8 errors or path block errors to be used for the path layer error count.

3.6. Mapping of IEEE 802.3 Managed Objects

This section contains the mapping between oWIS managed objects defined in [P802.3ae] and managed objects defined in this document and in associated MIB modules, i.e., the IF-MIB [RFC2863], the SONET-MIB [SONETng], and the MAU-MIB [MAU-MIB].

IEEE 802.3 Managed Object Corresponding SNMP Object

```
   oWIS - pWISBasic package
    aWISID                      IF-MIB - ifIndex
    aSectionStatus              SONET-MIB - sonetSectionCurrentStatus
    aLineStatus                 SONET-MIB - sonetLineCurrentStatus
    aPathStatus                 etherWisPathCurrentStatus
    aFarEndPathStatus           etherWisFarEndPathCurrentStatus

   oWIS - pWISOptional package
    aSectionSESThreshold        SONET-MIB - sonetSESthresholdSet
    aSectionSESs                SONET-MIB - sonetSectionCurrentSESs +
                                            sonetSectionIntervalSESs
    aSectionESs                 SONET-MIB - sonetSectionCurrentESs +
                                            sonetSectionIntervalESs
    aSectionSEFSs               SONET-MIB - sonetSectionCurrentSEFSs +
                                            sonetSectionIntervalSEFSs
    aSectionCVs                 SONET-MIB - sonetSectionCurrentCVs +
                                            sonetSectionIntervalCVs
    aJ0ValueTX                  etherWisSectionCurrentJ0Transmitted
    aJ0ValueRX                  etherWisSectionCurrentJ0Received
    aLineSESThreshold           SONET-MIB - sonetSESthresholdSet
    aLineSESs                   SONET-MIB - sonetLineCurrentSESs +
                                            sonetLineIntervalSESs
    aLineESs                    SONET-MIB - sonetLineCurrentESs +
                                            sonetLineIntervalESs
    aLineCVs                    SONET-MIB - sonetLineCurrentCVs +
                                            sonetLineIntervalCVs
    aFarEndLineSESs             SONET-MIB - sonetFarEndLineCurrentSESs +
                                            sonetFarEndLineIntervalSESs
    aFarEndLineESs              SONET-MIB - sonetFarEndLineCurrentESs +
                                            sonetFarEndLineIntervalESs
    aFarEndLineCVs              SONET-MIB - sonetFarEndLineCurrentCVs +
                                            sonetFarEndLineIntervalCVs
    aPathSESThreshold           SONET-MIB - sonetSESthresholdSet
    aPathSESs                   SONET-MIB - sonetPathCurrentSESs +
                                            sonetPathIntervalSESs
    aPathESs                    SONET-MIB - sonetPathCurrentESs +
                                            sonetPathIntervalESs
    aPathCVs                    SONET-MIB - sonetPathCurrentCVs +
                                            sonetPathIntervalCVs
    aJ1ValueTX                  etherWisPathCurrentJ1Transmitted
```



```
    aJ1ValueRX                  etherWisPathCurrentJ1Received
```

aFarEndPathSESs	SONET-MIB - sonetFarEndPathCurrentSESs + sonetFarEndPathIntervalSESs
aFarEndPathESs	SONET-MIB - sonetFarEndPathCurrentESs + sonetFarEndPathIntervalESs
aFarEndPathCVs	SONET-MIB - sonetFarEndPathCurrentCVs + sonetFarEndPathIntervalCVs

Please note that the definitions of the threshold objects and counter objects imported from the SONET-MIB do not exactly match the definitions of the corresponding IEEE 802.3 objects. The specific differences are as follows:

IEEE 802.3 Managed Object	How Corresponding SNMP Object Differs
oWIS - pWISOptional package	
aSectionSESThreshold	This object is defined in [P802.3ae] as an integer with one instance per interface. sonetSESthresholdSet is an enumerated value that has one instance per network element; it controls the thresholds for all layers simultaneously and allows only certain discrete values to be selected.
aSectionSESs	This object is defined in [P802.3ae] as a generalized nonresetable counter. The objects sonetSectionCurrentSESs and sonetSectionIntervalSESs are 15-minute interval counters.
aSectionESs	This object is defined as a generalized nonresetable counter in [P802.3ae]. The objects sonetSectionCurrentESs and sonetSectionIntervalESs are 15-minute interval counters.
aSectionSEFSs	This object is defined as a generalized nonresetable counter in [P802.3ae]. The objects sonetSectionCurrentSEFSs and sonetSectionIntervalSEFSs are 15-minute interval counters.
aSectionCVs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetSectionCurrentCVs and sonetSectionIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as severely errored seconds.

aLineSESThreshold	This object is defined in [P802.3ae] as an integer with one instance per interface. sonetSESthresholdSet is an enumerated value that has one instance per network element; it controls the thresholds for all layers simultaneously and allows only certain discrete values to be selected.
aLineSESs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetLineCurrentSESs and sonetLineIntervalSESs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as unavailable seconds.
aLineESs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetLineCurrentESs and sonetLineIntervalESs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as unavailable seconds.
aLineCVs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetLineCurrentCVs and sonetLineIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify either as severely errored seconds or as unavailable seconds.
aFarEndLineSESs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetFarEndLineCurrentSESs and sonetFarEndLineIntervalSESs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as unavailable seconds.
aFarEndLineESs	This object is defined as a generalized

	nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetFarEndLineCurrentESs and

	sonetFarEndLineIntervalESs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as unavailable seconds.
aFarEndLineCVs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetFarEndLineCurrentCVs and sonetFarEndLineIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify either as severely errored seconds or as unavailable seconds.
aPathSESThreshold	This object is defined in [P802.3ae] as an integer with one instance per interface. sonetSESthresholdSet is an enumerated value that has one instance per network element; it controls the thresholds for all layers simultaneously and allows only certain discrete values to be selected.
aPathSESs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetPathCurrentSESs and sonetPathIntervalSESs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as unavailable seconds. In addition, [P802.3ae] includes PLM-P and LCD-P defects in the criteria for declaring path layer severely errored seconds, while [SONETng] does not.
aPathESs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The

	objects sonetPathCurrentESs and sonetPathIntervalESs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as unavailable seconds. In addition, [P802.3ae] includes PLM-P and LCD-P defects in the criteria for declaring path layer errored seconds, while [SONETng] does not.

aPathCVs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetPathCurrentCVs and sonetPathIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify either as severely errored seconds or as unavailable seconds.
aFarEndPathSESs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetFarEndPathCurrentSESs and sonetFarEndPathIntervalSESs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as unavailable seconds. In addition, [P802.3ae] includes far-end PLM-P and LCD-P defects in the criteria for declaring far-end path layer severely errored seconds, while [SONETng] does not.
aFarEndPathESs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetFarEndPathCurrentESs and sonetFarEndPathIntervalESs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify as

	unavailable seconds. In addition, [P802.3ae] includes far-end PLM-P and LCD-P defects in the criteria for declaring far-end path layer errored seconds, while [SONETng] does not.
aFarEndPathCVs	This object is defined as a generalized nonresetable counter in [P802.3ae], and it is not subject to inhibiting. The objects sonetFarEndPathCurrentCVs and sonetFarEndPathIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during one-second intervals that qualify either as severely errored seconds or as unavailable seconds.

Please note that despite the differences in semantics between the threshold objects and counter objects imported from the SONET-MIB and the corresponding IEEE 802.3 objects, the hardware support mandated by [P802.3ae] subclause 50.3.10 suffices for both. See Appendix A for details.

3.7. Mapping of SNMP Objects to WIS Station Management Registers

Some of the objects defined in this memo or incorporated by reference from the SONET-MIB [SONETng] or the MAU-MIB [MAU-MIB] require WIS-specific hardware support. [P802.3ae] subclause 50.3.10 specifies WIS management interface requirements, including a required subset of the WIS MDIO (Management Data Input/Output) registers defined in [P802.3ae] subclause 45.2.2. The table below provides a cross-reference between those managed objects and the WIS MDIO registers from the subset in [P802.3ae] subclause 50.3.10 required to support them. Note that the MDIO interface is optional; however, if it is not implemented, then the capabilities of the required register subset must be provided by other means.

SNMP Object	WIS MDIO Register(s)
ETHER-WIS - etherWisDeviceTestPatternType	10G WIS Control 2
ETHER-WIS - etherWisDeviceRxTestPatternMode	10G WIS Control 2
ETHER-WIS - etherWisDeviceTxTestPatternMode	10G WIS Control 2

```
   SONET-MIB - sonetMediumType                       none required
   SONET-MIB - sonetMediumTimeElapsed                none required
   SONET-MIB - sonetMediumValidIntervals             none required
   SONET-MIB - sonetMediumLineCoding                 none required
   SONET-MIB - sonetMediumLineType                   none required
   SONET-MIB - sonetMediumCircuitIdentifier          none required
   SONET-MIB - sonetMediumInvalidIntervals           none required
   SONET-MIB - sonetMediumLoopbackConfig             none required
   SONET-MIB - sonetSESthresholdSet                  none required

   ETHER-WIS - etherWisSectionCurrentJ0Transmitted 10G WIS J0 Tx
   ETHER-WIS - etherWisSectionCurrentJ0Received    10G WIS J0 Rx

   SONET-MIB - sonetSectionCurrentStatus             10G WIS Status 3
   SONET-MIB - sonetSectionCurrentESs                \
   SONET-MIB - sonetSectionCurrentSESs                \
   SONET-MIB - sonetSectionCurrentSEFSs                | 10G WIS Status 3
   SONET-MIB - sonetSectionCurrentCVs                  |        +
   SONET-MIB - sonetSectionIntervalESs                 | 10G WIS Section
   SONET-MIB - sonetSectionIntervalSESs                | BIP Error Count
   SONET-MIB - sonetSectionIntervalSEFSs              /
   SONET-MIB - sonetSectionIntervalCVs               /
   SONET-MIB - sonetSectionIntervalValidData         none required
```



```
   SONET-MIB - sonetLineCurrentStatus                10G WIS Status 3
   SONET-MIB - sonetLineCurrentESs                   \
   SONET-MIB - sonetLineCurrentSESs                   \
   SONET-MIB - sonetLineCurrentCVs                     | 10G WIS Status 3
   SONET-MIB - sonetLineCurrentUASs                    |        +
   SONET-MIB - sonetLineIntervalESs                    | 10G WIS Line
   SONET-MIB - sonetLineIntervalSESs                   | BIP Errors
   SONET-MIB - sonetLineIntervalCVs                   /
   SONET-MIB - sonetLineIntervalUASs                 /
   SONET-MIB - sonetLineIntervalValidData            none required

   SONET-MIB - sonetFarEndLineCurrentESs             \
   SONET-MIB - sonetFarEndLineCurrentSESs             \
   SONET-MIB - sonetFarEndLineCurrentCVs               | 10G WIS Status 3
   SONET-MIB - sonetFarEndLineCurrentUASs              |        +
   SONET-MIB - sonetFarEndLineIntervalESs              | 10G WIS Far End
   SONET-MIB - sonetFarEndLineIntervalSESs             | Line BIP Errors
   SONET-MIB - sonetFarEndLineIntervalCVs             /
```

```
SONET-MIB - sonetFarEndLineIntervalUASs          /
SONET-MIB - sonetFarEndLineIntervalValidData     10G WIS Status 3

ETHER-WIS - etherWisPathCurrentStatus            10G WIS Status 3
ETHER-WIS - etherWisPathCurrentJ1Transmitted     10G WIS J1 Tx
ETHER-WIS - etherWisPathCurrentJ1Received        10G WIS J1 Rx

SONET-MIB - sonetPathCurrentWidth                none required
SONET-MIB - sonetPathCurrentStatus               10G WIS Status 3
SONET-MIB - sonetPathCurrentESs                  \
SONET-MIB - sonetPathCurrentSESs                  \
SONET-MIB - sonetPathCurrentCVs                    | 10G WIS Status 3
SONET-MIB - sonetPathCurrentUASs                   |        +
SONET-MIB - sonetPathIntervalESs                   | 10G WIS
SONET-MIB - sonetPathIntervalSESs                  | Path Block
SONET-MIB - sonetPathIntervalCVs                  /  Error Count
SONET-MIB - sonetPathIntervalUASs                /
SONET-MIB - sonetPathIntervalValidData           none required

ETHER-WIS - etherWisFarEndPathCurrentStatus      10G WIS Status 3

SONET-MIB - sonetFarEndPathCurrentESs            \
SONET-MIB - sonetFarEndPathCurrentSESs            \
SONET-MIB - sonetFarEndPathCurrentCVs              | 10G WIS Status 3
SONET-MIB - sonetFarEndPathCurrentUASs             |        +
SONET-MIB - sonetFarEndPathIntervalESs             | 10G WIS Far End
SONET-MIB - sonetFarEndPathIntervalSESs            | Path Block
SONET-MIB - sonetFarEndPathIntervalCVs            /  Error Count
SONET-MIB - sonetFarEndPathIntervalUASs          /
SONET-MIB - sonetFarEndPathIntervalValidData     10G WIS Status 3
```



```
MAU-MIB - ifMauIfIndex                           none required
MAU-MIB - ifMauIndex                             none required
MAU-MIB - ifMauType                              10G WIS Control 2
MAU-MIB - ifMauStatus                            WIS Control 1
MAU-MIB - ifMauMediaAvailable                    \ WIS Status 1 +
MAU-MIB - ifMauMediaAvailableStateExits          / 10G WIS Status 3
MAU-MIB - ifMauJabberState                       none required
MAU-MIB - ifMauJabberingStateEnters              none required
MAU-MIB - ifMauFalseCarriers                     none required
MAU-MIB - ifMauDefaultType                       10G WIS Control 2
MAU-MIB - ifMauAutoNegSupported                  none required
```

3.8. Structure of the MIB Module

Four tables are defined in this MIB module.

3.8.1. etherWisDeviceTable

The purpose of this table is to define managed objects to control the WIS test pattern mode. These objects are required to support mandatory WIS test features required by Clause 50 of [P802.3ae].

The etherWisDeviceTable is a sparse augmentation of the sonetMediumTable of the SONET MIB -- in other words, for each entry in the etherWisDeviceTable there SHALL be an entry in the sonetMediumTable and the same ifIndex value SHALL be used for both entries.

3.8.2. etherWisSectionCurrentTable

The purpose of this table is to define managed objects for the transmitted and received section trace messages (J0 byte).

The etherWisSectionCurrentTable is a sparse augmentation of the sonetSectionCurrentTable of the SONET MIB -- in other words, for each entry in the etherWisSectionCurrentTable there SHALL be an entry in the sonetSectionCurrentTable and the same ifIndex value SHALL be used for both entries.

3.8.3. etherWisPathCurrentTable

The purpose of this table is to define managed objects for the current WIS path layer status and for the transmitted and received path trace messages (J1 byte). The path layer status object is provided because the WIS supports some near-end path status conditions that are not reported in sonetPathCurrentStatus.

The etherWisPathCurrentTable is a sparse augmentation of the sonetPathCurrentTable of the SONET MIB -- in other words, for each entry in the etherWisPathCurrentTable there SHALL be an entry in the sonetPathCurrentTable and the same ifIndex value SHALL be used for

both entries.

3.8.4. etherWisFarEndPathCurrentTable

The purpose of this table is to define a managed object for the current status of the far end of the path. This object is provided because the WIS supports some far-end path status conditions that are not reported in sonetPathCurrentStatus.

The etherWisFarEndPathCurrentTable is a sparse augmentation of the sonetFarEndPathCurrentTable of the SONET MIB -- in other words, for each entry in the etherWisFarEndPathCurrentTable there SHALL be an entry in the sonetFarEndPathCurrentTable and the same ifIndex value SHALL be used for both entries.

4. Object Definitions

```
   ETHER-WIS DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE,
       transmission
           FROM SNMPv2-SMI
       ifIndex
           FROM IF-MIB
       MODULE-COMPLIANCE, OBJECT-GROUP
           FROM SNMPv2-CONF
       sonetMediumStuff2, sonetSectionStuff2,
       sonetLineStuff2, sonetFarEndLineStuff2,
       sonetPathStuff2, sonetFarEndPathStuff2,
       sonetMediumType, sonetMediumLineCoding,
       sonetMediumLineType, sonetMediumCircuitIdentifier,
       sonetMediumLoopbackConfig, sonetSESthresholdSet,
       sonetPathCurrentWidth
           FROM SONET-MIB;

   etherWisMIB MODULE-IDENTITY
       LAST-UPDATED "200111202123Z"  -- November 20, 2001
           ORGANIZATION "IETF Hubmib and AToMMIB Working Groups"
           CONTACT-INFO
              "Hubmib WG:
                 http://www.ietf.org/html.charters/hubmib-charter.html

               AToMMIB WG:
                 http://www.ietf.org/html.charters/atommib-charter.html

               Editor: C. M. Heard
               Postal: 600 Rainbow Dr. #141
                       Mountain View, CA 94041-2542
                       USA
                  Tel: +1 650-964-8391
               E-mail: heard@pobox.com"

       DESCRIPTION
         "The objects in this MIB module are used in conjunction
         with objects in the SONET-MIB and the MAU-MIB to manage
         the Ethernet WAN Interface Sublayer (WIS).
```

```
          The following reference is used throughout this MIB module:

          [IEEE 802.3 Std] refers to:
             IEEE Std 802.3, 2000 Edition: 'IEEE Standard
             for Information technology -
             Telecommunications and information exchange
             between systems - Local and metropolitan
             area networks - Specific requirements -
             Part 3: Carrier sense multiple access with
             collision detection (CSMA/CD) access method
             and physical layer specifications', as
             amended by IEEE Draft P802.3ae/D3.3:
             'Supplement to Carrier Sense Multiple Access
             with Collision Detection (CSMA/CD) Access
             Method & Physical Layer Specifications -
             Media Access Control (MAC) Parameters,
             Physical Layer, and Management Parameters
             for 10 Gb/s Operation', October 23, 2001.

          Of particular interest are Clause 50, 'WAN Interface
          Sublayer (WIS), type 10GBASE-W', Clause 30, '10Mb/s,
          100Mb/s, 1000Mb/s, and 10Gb/s MAC Control, and Link
          Aggregation Management', and Clause 45, 'Management
          Data Input/Output (MDIO) Interface'."

        REVISION    "200111202123Z"  -- November 20, 2001
        DESCRIPTION "Initial version."

        ::= { transmission XXX }  -- to be assigned by IANA


    -- The main sections of the module

    etherWisObjects     OBJECT IDENTIFIER ::= { etherWisMIB 1 }

    etherWisObjectsPath OBJECT IDENTIFIER ::= { etherWisMIB 2 }

    etherWisConformance OBJECT IDENTIFIER ::= { etherWisMIB 3 }
```

```
   -- groups in the Ethernet WIS MIB module

   etherWisDevice      OBJECT IDENTIFIER ::= { etherWisObjects 1 }

   etherWisSection     OBJECT IDENTIFIER ::= { etherWisObjects 2 }

   etherWisPath        OBJECT IDENTIFIER ::= { etherWisObjectsPath 1 }

   etherWisFarEndPath  OBJECT IDENTIFIER ::= { etherWisObjectsPath 2 }
```

```
   -- The Device group

   -- These objects provide WIS extensions to
   -- the SONET-MIB Medium Group.

   etherWisDeviceTable OBJECT-TYPE
       SYNTAX  SEQUENCE OF EtherWisDeviceEntry
       MAX-ACCESS  not-accessible
       STATUS  current
       DESCRIPTION
          "The table for Ethernet WIS devices"
        ::= { etherWisDevice 1 }

   etherWisDeviceEntry OBJECT-TYPE
       SYNTAX  EtherWisDeviceEntry
       MAX-ACCESS  not-accessible
       STATUS  current
       DESCRIPTION
          "An entry in the Ethernet WIS device table.  For each
          instance of this object there shall be a corresponding
          instance of sonetMediumEntry."
       INDEX  { ifIndex }
        ::= { etherWisDeviceTable 1 }

   EtherWisDeviceEntry ::=
       SEQUENCE {
           etherWisDeviceTestPatternType       INTEGER,
           etherWisDeviceRxTestPatternMode     INTEGER,
           etherWisDeviceTxTestPatternMode     INTEGER
           }

   etherWisDeviceTestPatternType OBJECT-TYPE
       SYNTAX  INTEGER {
```

```
                mixedFrequency(0),
                squareWave(1)
            }
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
           "This variable controls the type of test pattern.  The value
           mixedFrequency(0) selects the framed mixed frequency test
           pattern specified in [IEEE 802.3 Std.] subclause 50.3.8.2.
           The value squareWave(1) selects the unframed square wave test
           pattern specified in [IEEE 802.3 Std.] subclause 50.3.8.1."
        REFERENCE
           "[IEEE 802.3 Std.], 50.3.8, WIS test pattern generator and
           checker, 45.2.2.6, 10G WIS Control 2 register (2.7), and
           45.2.2.6.1, Test pattern (2.7.3)."
         ::= { etherWisDeviceEntry 1 }
```



```
   etherWisDeviceRxTestPatternMode OBJECT-TYPE
        SYNTAX  INTEGER {
                    disabled(0),
                    enabled(1)
                }
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
           "This variable controls the receive test pattern mode.
           The value disabled(0) sets the WIS receive path into
           normal mode.  The value enabled(1) sets the WIS receive
           path into test pattern mode.  An attempt to set this
           object to enabled(1) when the corresponding instance
           of ifAdminState has the value up(1) SHALL be rejected
           with the error inconsistentValue.  An attempt to set the
           corresponding instance of ifAdminStatus to the value
           up(1) when an instance of this object has the value
           enabled(1) SHALL be rejected with the error
           inconsistentValue."
        REFERENCE
           "[IEEE 802.3 Std.], 50.3.8, WIS test pattern generator and
           checker, 45.2.2.6, 10G WIS Control 2 register (2.7), and
           45.2.2.6.2, Receive test pattern mode (2.7.2)."
         ::= { etherWisDeviceEntry 2 }

   etherWisDeviceTxTestPatternMode OBJECT-TYPE
```

```
        SYNTAX  INTEGER {
                    disabled(0),
                    enabled(1)
                }
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
           "This variable controls the transmit test pattern mode.
           The value disabled(0) set the WIS transmit path into
           normal mode.  The value enabled(1) sets the WIS transmit
           path into test pattern mode.  An attempt to set this
           object to enabled(1) when the corresponding instance
           of ifAdminState has the value up(1) SHALL be rejected
           with the error inconsistentValue.  An attempt to set the
           corresponding instance of ifAdminStatus to the value
           up(1) when an instance of this object has the value
           enabled(1) SHALL be rejected with the error
           inconsistentValue."
        REFERENCE
           "[IEEE 802.3 Std.], 50.3.8, WIS test pattern generator and
           checker, 45.2.2.6, 10G WIS Control 2 register (2.7), and
           45.2.2.6.3, Transmit test pattern mode (2.7.1)."
         ::= { etherWisDeviceEntry 3 }
```



```
   -- The Section group

   -- These objects provide WIS extensions to
   -- the SONET-MIB Section Group.

   etherWisSectionCurrentTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF EtherWisSectionCurrentEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
           "The table for the current state of Ethernet WIS sections."
         ::= { etherWisSection 1 }

   etherWisSectionCurrentEntry OBJECT-TYPE
        SYNTAX  EtherWisSectionCurrentEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
           "An entry in the etherWisSectionCurrentTable.  For each
```

```
          instance of this object there shall be a corresponding
          instance of sonetSectionCurrentEntry."
       INDEX  { ifIndex }
        ::= { etherWisSectionCurrentTable 1 }

   EtherWisSectionCurrentEntry ::=
       SEQUENCE {
           etherWisSectionCurrentJ0Transmitted INTEGER,
           etherWisSectionCurrentJ0Received    INTEGER
           }

   etherWisSectionCurrentJ0Transmitted OBJECT-TYPE
       SYNTAX  INTEGER ( 0..255 )
       MAX-ACCESS  read-write
       STATUS  current
       DESCRIPTION
          "This is the single-octet section trace message
          that is to be transmitted in the J0 byte.  The
          value SHOULD be set to '01'h when the section
          trace function is not used, and the implementation
          SHOULD use that value as a default if no other
          value has been set."
       REFERENCE
          "[IEEE 802.3 Std.], 30.8.1.1.8, aJ0ValueTX."
        ::= { etherWisSectionCurrentEntry 1 }
```



```
   etherWisSectionCurrentJ0Received OBJECT-TYPE
       SYNTAX  INTEGER ( 0..255 )
       MAX-ACCESS  read-only
       STATUS  current
       DESCRIPTION
          "This is the single-octet section trace message
          that was most recently received in the J0 byte."
       REFERENCE
          "[IEEE 802.3 Std.], 30.8.1.1.9, aJ0ValueRX."
        ::= { etherWisSectionCurrentEntry 2 }
```

```
   -- The Path group

   -- These objects provide WIS extensions to
   -- the SONET-MIB Path Group.

   etherWisPathCurrentTable OBJECT-TYPE
       SYNTAX  SEQUENCE OF EtherWisPathCurrentEntry
       MAX-ACCESS  not-accessible
       STATUS  current
       DESCRIPTION
          "The table for the current state of Ethernet WIS paths."
        ::= { etherWisPath 1 }

   etherWisPathCurrentEntry OBJECT-TYPE
       SYNTAX  EtherWisPathCurrentEntry
       MAX-ACCESS  not-accessible
       STATUS  current
       DESCRIPTION
          "An entry in the etherWisPathCurrentTable.  For each
          instance of this object there shall be a corresponding
          instance of sonetPathCurrentEntry."
       INDEX  { ifIndex }
        ::= { etherWisPathCurrentTable 1 }

   EtherWisPathCurrentEntry ::=
       SEQUENCE {
           etherWisPathCurrentStatus           BITS,
           etherWisPathCurrentJ1Transmitted    OCTET STRING,
           etherWisPathCurrentJ1Received       OCTET STRING
           }
```



```
   etherWisPathCurrentStatus OBJECT-TYPE
       SYNTAX  BITS {
                   etherWisPathLOP(0),
                   etherWisPathAIS(1),
                   etherWisPathPLM(2),
```

```
                etherWisPathLCD(3)
            }
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
           "This variable indicates the current status of the
           path payload with a bit map that can indicate multiple
           defects at once.  The bit positions are assigned as
           follows:

           etherWisPathLOP(0)
              This bit is set to indicate that an
              LOP-P (Loss of Pointer - Path) defect
              is being experienced.  Note:  when this
              bit is set, sonetPathSTSLOP MUST be set
              in the corresponding instance of
              sonetPathCurrentStatus.

           etherWisPathAIS(1)
              This bit is set to indicate that an
              AIS-P (Alarm Indication Signal - Path)
              defect is being experienced.  Note:  when
              this bit is set, sonetPathSTSAIS MUST be
              set in the corresponding instance of
              sonetPathCurrentStatus.

           etherWisPathPLM(1)
              This bit is set to indicate that a
              PLM-P (Payload Label Mismatch - Path)
              defect is being experienced.  Note:  when
              this bit is set, sonetPathSignalLabelMismatch
              MUST be set in the corresponding instance of
              sonetPathCurrentStatus.

           etherWisPathLCD(3)
              This bit is set to indicate that an
              LCD-P (Loss of Codegroup Delination - Path)
              defect is being experienced.  Since this
              defect is detected by the PCS and not by
              the path layer itself, there is no
              corresponding bit in sonetPathCurrentStatus."
        REFERENCE
           "[IEEE 802.3 Std.], 30.8.1.1.18, aPathStatus."
         ::= { etherWisPathCurrentEntry 1 }
```

```
etherWisPathCurrentJ1Transmitted OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (16))
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
       "This is the 16-octet path message that is to
       be transmitted in the J1 byte.  The value SHOULD
       be fifteen octets of '00'h followed by '89'h
       (or some cyclic shift thereof) when the path
       trace function is not used, and the implementation
       SHOULD use that value (or a cyclic shift thereof)
       as a default if no other value has been set."
    REFERENCE
       "[IEEE 802.3 Std.], 30.8.1.1.23, aJ1ValueTX."
     ::= { etherWisPathCurrentEntry 2 }

etherWisPathCurrentJ1Received OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (16))
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
       "This is the 16-octet path trace message that
       was most recently received in the J1 byte."
    REFERENCE
       "[IEEE 802.3 Std.], 30.8.1.1.24, aJ1ValueRX."
     ::= { etherWisPathCurrentEntry 3 }


-- The Far End Path group

-- These objects provide WIS extensions to
-- the SONET-MIB Far End Path Group.

etherWisFarEndPathCurrentTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF EtherWisFarEndPathCurrentEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
       "The table for the current far-end state of Ethernet WIS
       paths."
     ::= { etherWisFarEndPath 1 }
```

```
   etherWisFarEndPathCurrentEntry OBJECT-TYPE
       SYNTAX  EtherWisFarEndPathCurrentEntry
       MAX-ACCESS  not-accessible
       STATUS  current
       DESCRIPTION
          "An entry in the etherWisFarEndPathCurrentTable.  For each
          instance of this object there shall be a corresponding
          instance of sonetFarEndPathCurrentEntry."
       INDEX  { ifIndex }
        ::= { etherWisFarEndPathCurrentTable 1 }

   EtherWisFarEndPathCurrentEntry ::=
       SEQUENCE {
           etherWisFarEndPathCurrentStatus     BITS
           }

   etherWisFarEndPathCurrentStatus OBJECT-TYPE
       SYNTAX  BITS {
                   etherWisFarEndPayloadDefect(0),
                   etherWisFarEndServerDefect(1)
               }
       MAX-ACCESS  read-write
       STATUS  current
       DESCRIPTION
          "This variable indicates the current status at the
          far end of the path using a bit map that can indicate
          multiple defects at once.  The bit positions are
          assigned as follows:

          etherWisFarEndPayloadDefect(0)
             A far end payload defect (i.e., far end
             PLM-P or LCD-P) is currently being signalled
             in G1 bits 5-7.

          etherWisFarEndServerDefect(1)
             A far end server defect (i.e., far end
             LOP-P or AIS-P) is currently being signalled
             in G1 bits 5-7.  Note:  when this bit is set,
             sonetPathSTSRDI MUST be set in the corresponding
             instance of sonetPathCurrentStatus."
       REFERENCE
          "[IEEE 802.3 Std.], 30.8.1.1.25, aFarEndPathStatus."
```

```
        ::= { etherWisFarEndPathCurrentEntry 1 }
```



```
   --
   --     Conformance Statements
   --

   etherWisGroups      OBJECT IDENTIFIER ::= { etherWisConformance 1 }

   etherWisCompliances OBJECT IDENTIFIER ::= { etherWisConformance 2 }

   --     Object Groups

   etherWisDeviceGroup OBJECT-GROUP
       OBJECTS {
           etherWisDeviceTestPatternType,
           etherWisDeviceRxTestPatternMode,
           etherWisDeviceTxTestPatternMode
           }
       STATUS  current
       DESCRIPTION
          "A collection of objects that control test
          features required of all WIS devices."
        ::= { etherWisGroups 1 }

   etherWisSectionGroup OBJECT-GROUP
       OBJECTS {
           etherWisSectionCurrentJ0Transmitted,
           etherWisSectionCurrentJ0Received
           }
       STATUS  current
       DESCRIPTION
          "A collection of objects that provide
          required information about a WIS section."
        ::= { etherWisGroups 2 }

   etherWisPathGroup OBJECT-GROUP
       OBJECTS {
```

```
            etherWisPathCurrentStatus,
            etherWisPathCurrentJ1Transmitted,
            etherWisPathCurrentJ1Received
            }
        STATUS  current
        DESCRIPTION
           "A collection of objects that provide
           required information about a WIS path."
         ::= { etherWisGroups 3 }
```



```
   etherWisFarEndPathGroup OBJECT-GROUP
       OBJECTS {
           etherWisFarEndPathCurrentStatus
           }
       STATUS  current
       DESCRIPTION
          "A collection of objects that provide required
          information about the far end of a WIS path."
        ::= { etherWisGroups 4 }

   --     Compliance Statements

   etherWisCurrentCompliance MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
          "The compliance statement for this module."

       MODULE  -- this module
           MANDATORY-GROUPS {
               etherWisDeviceGroup,
               etherWisSectionGroup,
               etherWisPathGroup,
               etherWisFarEndPathGroup
               }

       MODULE SONET-MIB
           MANDATORY-GROUPS {
               sonetMediumStuff2,
```

```
        sonetSectionStuff2,
        sonetLineStuff2,
        sonetFarEndLineStuff2,
        sonetPathStuff2,
        sonetFarEndPathStuff2
        }

    OBJECT      sonetMediumType
    SYNTAX      INTEGER {
        sonet(1)
        }
    MIN-ACCESS  read-only
    DESCRIPTION
        "Write access is not required, nor is support
        for any value other than sonet(1)."
```



```
    OBJECT      sonetMediumLineCoding
    SYNTAX      INTEGER {
        sonetMediumNRZ(4)
        }
    MIN-ACCESS  read-only
    DESCRIPTION
        "Write access is not required, nor is support
        for any value other than sonetMediumNRZ(4)."

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

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

    OBJECT      sonetMediumLoopbackConfig
    SYNTAX      BITS {
```

```
                sonetNoLoop(0),
                sonetFacilityLoop(1)
                }
            MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required, nor is support for values
                other than sonetNoLoop(0) and sonetFacilityLoop(1)."

            OBJECT      sonetSESthresholdSet
            MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required, and only one
                of the enumerated values need be supported."

            OBJECT      sonetPathCurrentWidth
            SYNTAX      INTEGER {
                sts192cSTM64(6)
                }
            MIN-ACCESS  read-only
            DESCRIPTION
                "Write access is not required, nor is support
                for any value other than sts192cSTM64(6)."

        ::= { etherWisCompliances 1 }

    END
```


5. Acknowledgments

This document is a product of the IETF Hubmib and AToMMIB Working Groups. It builds upon the work of the IEEE P802.3ae 10 Gigabit Ethernet Task Force.

6. 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.

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.

7. References

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

[RFC1155] Rose, M., and K. McCloghrie, "Structure and Identification

of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

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

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

[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.

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

[RFC1901] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, 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.

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

[RFC2574] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

[RFC1905] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

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

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

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirements Levels", BCP 14, RFC 2119, March 1997.

[RFC2863] McCloghrie, K., and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC2864] McCloghrie, K., and G. Hanson, "The Inverted Stack Table Extension to the Interfaces Group MIB", RFC 2864, June 2000.

[SONETng] Tesink, K., "Definitions of Managed Objects for the SONET/SDH Interface Type", rfc2558 update (forthcoming), work in progress.

[T1.231] American National Standard for Telecommunications - Digital Hierarchy - Layer 1 In-Service Digital Transmission Performance Monitoring, ANSI T1.231-1997, September 1997.

[ETHERIF] Flick, J., "Definitions of Managed Objects for the Ethernet-like Interface Types", <draft-ietf-hubmib-etherif-mib-v3-00.txt>, work in progress.

[MAU-MIB] Flick, J., "Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)", <draft-ietf-hubmib-mau-mib-v3-00.txt>, work in progress.

[P802.3ae] Law, D., Editor, Draft Supplement to IEEE Std. 802.3, IEEE Draft P802.3ae/D3.3, October 23, 2001, work in progress.

8. Authors' Addresses

Mike Ayers
BMC Software, Inc.
2141 North First Street
San Jose, CA 95131
USA

Phone: +1 408 546 0947
Fax: +1 408 965 0359
Email: mayers@bmc.com

John Flick
Hewlett-Packard Company
8000 Foothills Blvd. M/S 5557
Roseville, CA 95747-5557

Phone: +1 916 785 4018
Fax: +1 916 785 1199
Email: johnf@rose.hp.com

C. M. Heard
600 Rainbow Dr. #141
Mountain View, CA 94041-2542
USA

Phone: +1 650 964 8391
EMail: heard@pobox.com

Kam Lam
Lucent Technologies
101 Crawfords Corner Road, Room 4C-616A
Holmdel, NJ 07733

Phone: +1 732 949 8338
EMail: hklam@lucent.com

Kerry McDonald
Institute for Applied Supercomputing

California State University San Bernardino

Email: kerry_mcd@hotmail.com
kmcdonal@csci.csusb.edu

K. C. Norseth
Enterasys Networks
2691 South Decker Lake Lane
Salt Lake City, Utah 84119

Phone: +1 801 887 9823
Email: knorseth@enterasys.com

Kaj Tesink
Telcordia Technologies
331 Newman Springs Road
P.O. Box 7020
Red Bank, NJ 07701-7020
USA

Phone: +1 732 758 5254
EMail: kaj@research.telcordia.com

9. 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.

Appendix A: Collection of Performance Data Using WIS MDIO Registers

The purpose of this appendix is to illustrate how the WIS MDIO registers specified in [P802.3ae] subclause 45.2.2 (and more specifically the subset required by [P802.3ae] subclause 50.3.10) can be used to collect performance data either according to the conventions adopted by this document or according to the conventions specified in [P802.3ae] Clause 30.

For an agent implementing the SNMP managed objects required by this document the first step in collecting WIS performance data would be to poll the 10G WIS Status 3 register and the various error count registers (10G WIS Section BIP Error Count, 10G WIS Line BIP Errors, 10G WIS Far End Line BIP Errors, 10G WIS Path Block Error Count, and 10G WIS Far End Path Block Error Count) once per second. The 10G WIS Status 3 register bits are all latched until read and so would indicate whether a given defect occurred any time during the previous second. The error count registers roll over modulo 2^16 or 2^32, and so to find the number of errors within the previous second the agent would need to subtract (modulo 2^16 or 2^32) the current reading from the reading taken one second ago. Armed with that information, the agent could determine for any layer whether the one second interval was an errored second, a severely errored second (that requires comparison with a threshold unless a defect is present), or a severely errored frame second. Determining whether a given second is or is not part of unavailable time requires additional logic; the most straightforward and accurate method is the delay-line approach outlined in Appendix A of [SONETng]. With that information available the agent would be able to determine by how much each current count

should be incremented (including effects of inhibiting). Implementations that conform to [T1.231] would end each 15-minute interval on time-of-day clock 1/4 hour boundaries; if the delay-line approach is used then a time-of-day timestamp would accompany the one-second statistics. At the end of each interval the current registers would be pushed onto the history stack and then would be cleared. The xyxIntervalValidData flags would be set to False(2) if the number of samples was not between 890 and 910 or, in the case of far-end counts, if a near-end defect occurred during the just-completed interval (see [T1.231] Section 9.1.2.2 for details).

An agent implementing the [P802.3ae] Clause 30 oWIS objects could start in much the same way, i.e., by polling the 10G WIS Status 3 register and the various error count registers to find the defects and error counts for the previous second, and it could determine the number of errors and whether the second was an errored second, a severely errored second, or a severely errored frame second in the same manner as above. The rest of the process would be simply to increment the generalized non-resetable counters without consideration of any inhibiting rules.

Full Copyright Statement

"AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

1 Abstract 2
2 The SNMP Management Framework 2
3 Overview 3
3.1 Relationship to the SONET MIB 4
3.2 Relationship to the Ethernet-like Interfaces MIB 4
3.3 Relationship to the 802.3 MAU MIB 4
3.4 Use of the ifTable 4
3.4.1 Layering Model 5
3.4.2 Use of ifTable for LLC Layer/MAC Layer/Reconciliation Sublayer/Physical Coding Sublayer 5
3.4.3 Use of ifTable for SONET/SDH Path Layer 5
3.4.4 Use of ifTable for SONET/SDH Medium/Section/Line Layer

..... 5
3.5 SONET/SDH Terminology 6
3.6 Mapping of IEEE 802.3 Managed Objects 7
3.7 Mapping of SNMP Objects to WIS Station Management Registers 12
3.8 Structure of the MIB Module 14
3.8.1 etherWisDeviceTable 14
3.8.2 etherWisSectionCurrentTable 14
3.8.3 etherWisPathCurrentTable 14
3.8.4 etherWisFarEndPathCurrentTable 15
4 Object Definitions 16
5 Acknowledgments 28
6 Security Considerations 28
7 References 29
8 Authors' Addresses 31
9 Intellectual Property 32
Appendix A: Collection of Performance Data Using WIS MDIO Registers 33
Full Copyright Statement 34