Ethernet Interfaces and Hub MIB Working Group
INTERNET DRAFT

Mike Ayers
Consultant
John Flick
Hewlett-Packard Company
C. M. Heard
Consultant
Kam Lam
Lucent Technologies
Kerry McDonald
CSU San Bernardino
K. C. Norseth
Consultant
Kaj Tesink
Telcordia Technologies
May 2, 2002

Definitions of Managed Objects for the Ethernet WAN Interface Sublayer

<draft-ietf-hubmib-wis-mib-04.txt>

Status of this Memo

Copyright Notice

Internet Draft <draft-ietf-hubmib-wis-mib-04.txt> 05/02/2002

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).

The MIB module defined in this memo is an extension of the SONET/SDH Interface MIB and is implemented in conjunction with it and with the Ethernet-like Interface MIB, the 802.3 Medium Attachment Unit MIB, the Interfaces Group MIB, and the Inverted Stack Table MIB.

1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL", when they appear in this document, are to be interpreted as described in RFC 2119 [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].

- 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/SDH Interface MIB [SONETng], and the 802.3 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/SDH Interface MIB, and the 802.3 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/SDH Interface 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 -- and its compliance statement REQUIRES that an agent implementing the objects defined in this memo also implement the relevant SONET-MIB objects. That includes all objects required by sonetCompliance2 as well as some that it leaves optional.

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 50.3.11.

3.2. Relationship to the Ethernet-like Interface 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

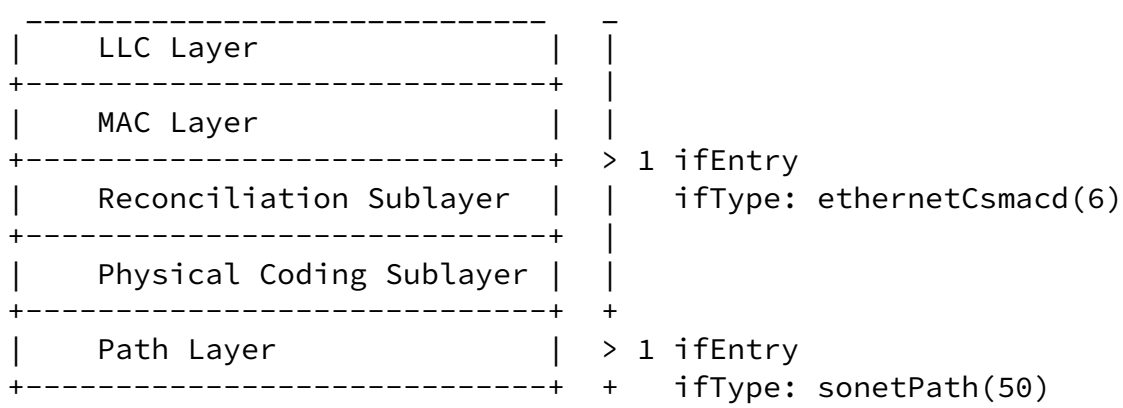
Support for the mauModIfCompl3 compliance statement of the MAU-MIB [MAU-MIB] is REQUIRED for all Ethernet-like interfaces. The MAU-MIB is needed in order to allow applications to control and/or determine the media type in use. That is important for devices than can support both the 10GBASE-R 10 Gb/s 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.



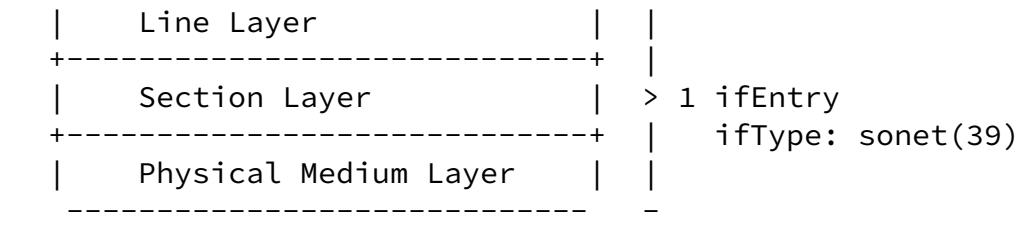


Figure 1 - Use of ifTable for an Ethernet WIS port

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 MIB module defined in this memo 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

It should be noted that the threshold and counter objects imported from the SONET-MIB are not completely equivalent to the corresponding IEEE 802.3 objects. The specific differences are as follows:

IEEE 802.3 Managed Object	How Corresponding SNMP Object Differs
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.

Note: despite the semantic differences 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 50.3.11 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 50.3.11 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 50.3.11 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 - etherWisDeviceTxTestPatternMode	10G WIS control 2
ETHER-WIS - etherWisDeviceRxTestPatternMode	10G WIS control 2
ETHER-WIS - etherWisDeviceRxTestPatternErrors	10G WIS test pattern error counter
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 transmit
ETHER-WIS - etherWisSectionCurrentJ0Received	10G WIS J0 receive

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 transmit
ETHER-WIS - etherWisPathCurrentJ1Received       10G WIS J1 receive

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
MAU-MIB - ifMauTypeListBits                     10G WIS status 2
```

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 and optional WIS test features specified in [P802.3ae] subclause 50.3.8.

The etherWisDeviceTable is a sparse augmentation of the sonetMediumTable of the SONET-MIB -- in other words, for each entry in the etherWisDeviceTable there MUST 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 MUST 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 MUST 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 MUST 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 "200205022034Z"  -- May 2, 2002
           ORGANIZATION "IETF Ethernet Interfaces and Hub MIB
                         Working Group"
           CONTACT-INFO
              "WG charter:
                 http://www.ietf.org/html.charters/hubmib-charter.html

               Chair: Dan Romascanu
              Postal: Avaya Inc.
                      Atidim Technology Park, Bldg. 3
                      Tel Aviv 61131
                      Israel
                 Tel: +972 3 645 8414
              E-mail: dromasca@avaya.com

              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 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/D5.0:
             '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', May 1, 2002.

          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    "200205022034Z"  -- May 2, 2002
        DESCRIPTION "Initial version, published as RFC yyyy."
-- RFC Ed.: replace yyyy with actual RFC number & remove this notice

        ::= { transmission XXX }
-- RFC Ed.: replace XXX with IANA-assigned number & remove this notice


   -- 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 MUST be a corresponding
          instance of sonetMediumEntry."
       INDEX  { ifIndex }
        ::= { etherWisDeviceTable 1 }

   EtherWisDeviceEntry ::=
       SEQUENCE {
           etherWisDeviceTxTestPatternMode     INTEGER,
           etherWisDeviceRxTestPatternMode     INTEGER,
           etherWisDeviceRxTestPatternErrors   Gauge32
           }

   etherWisDeviceTxTestPatternMode OBJECT-TYPE
       SYNTAX  INTEGER {
```

```
                none(0),
                squareWave(1),
                prbs31(2),
                mixedFrequency(3)
            }
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
           "This variable controls the transmit test pattern mode.
           The value none(0) puts the the WIS transmit path into
           the normal operating mode.  The value squareWave(1) puts
           the WIS transmit path into the square wave test pattern
           mode described in [IEEE 802.3 Std.] subclause 50.3.8.1.
           The value prbs31(2) puts the WIS transmit path into the
           PRBS31 test pattern mode described in [IEEE 802.3 Std.]
```



```
           subclause 50.3.8.2.  The value mixedFrequency(3) puts the
           WIS transmit path into the mixed frequency test pattern
           mode described in [IEEE 802.3 Std.] subclause 50.3.8.3.
           Any attempt to set this object to a value other than
           none(0) when the corresponding instance of ifAdminState
           has the value up(1) MUST be rejected with the error
           inconsistentValue, and any attempt to set the corresponding
           instance of ifAdminStatus to the value up(1) when an
           instance of this object has a value other than none(0)
           MUST 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
           45.2.2.7.2, PRBS31 pattern testing ability (2.8.1)."
         ::= { etherWisDeviceEntry 1 }

    etherWisDeviceRxTestPatternMode OBJECT-TYPE
        SYNTAX  INTEGER {
                    none(0),
                    prbs31(2),
                    mixedFrequency(3)
                }
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
           "This variable controls the receive test pattern mode.
```

```
            The value none(0) puts the the WIS receive path into the
            normal operating mode.  The value prbs31(2) puts the WIS
            receive path into the PRBS31 test pattern mode described
            in [IEEE 802.3 Std.] subclause 50.3.8.2.  The value
            mixedFrequency(3) puts the WIS receive path into the mixed
            frequency test pattern mode described in [IEEE 802.3 Std.]
            subclause 50.3.8.3.  Any attempt to set this object to a
            value other than none(0) when the corresponding instance
            of ifAdminState has the value up(1) MUST be rejected with
            the error inconsistentValue, and any attempt to set the
            corresponding instance of ifAdminStatus to the value up(1)
            when an instance of this object has a value other than
            none(0) MUST 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
            45.2.2.7.2, PRBS31 pattern testing ability (2.8.1)."
         ::= { etherWisDeviceEntry 2 }
```



```
    etherWisDeviceRxTestPatternErrors OBJECT-TYPE
        SYNTAX  Gauge32 ( 0..65535 )
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
            "This object counts the number of errors detected when the
            WIS receive path is operating in the PRBS31 test pattern
            mode.  It is reset to zero when the WIS receive path
            initially enters that mode, and it increments each time
            the PRBS pattern checker detects an error as described in
            [IEEE 802.3 Std.] subclause 50.3.8.2 unless its value is
            65535, in which case it remains unchanged.  This object is
            writeable so that it may be reset upon explicit request
            of a command generator application while the WIS receive
            path continues to operate in PRBS31 test pattern mode."
        REFERENCE
            "[IEEE 802.3 Std.], 50.3.8, WIS test pattern generator and
            checker, 45.2.2.7.2, PRBS31 pattern testing ability
            (2.8.1), and 45.2.2.8, 10G WIS test pattern error counter
```

```
          register (2.9)."
        ::= { 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 MUST be a corresponding
          instance of sonetSectionCurrentEntry."
       INDEX  { ifIndex }
        ::= { etherWisSectionCurrentTable 1 }

   EtherWisSectionCurrentEntry ::=
       SEQUENCE {
           etherWisSectionCurrentJ0Transmitted OCTET STRING,
           etherWisSectionCurrentJ0Received    OCTET STRING
           }

   etherWisSectionCurrentJ0Transmitted OBJECT-TYPE
       SYNTAX  OCTET STRING (SIZE (16))
       MAX-ACCESS  read-write
       STATUS  current
       DESCRIPTION
          "This is the 16-octet section trace message that is
          to be transmitted in the J0 byte.  The value SHOULD
          be '89'h followed by fifteen octets of '00'h
          (or some cyclic shift thereof) when the section
          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.], 30.8.1.1.8, aJ0ValueTX."
        ::= { etherWisSectionCurrentEntry 1 }
```



```
   etherWisSectionCurrentJ0Received OBJECT-TYPE
       SYNTAX  OCTET STRING (SIZE (16))
       MAX-ACCESS  read-only
       STATUS  current
       DESCRIPTION
```

```
            "This is the 16-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 MUST 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-only
    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 trace message that is to
            be transmitted in the J1 byte.  The value SHOULD
            be '89'h followed by fifteen octets of '00'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 MUST 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-only
        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

    etherWisDeviceGroupBasic OBJECT-GROUP
        OBJECTS {
            etherWisDeviceTxTestPatternMode,
            etherWisDeviceRxTestPatternMode
            }
        STATUS  current
        DESCRIPTION
```

```
          "A collection of objects that support test
          features required of all WIS devices."
        ::= { etherWisGroups 1 }

   etherWisDeviceGroupExtra OBJECT-GROUP
       OBJECTS {
           etherWisDeviceRxTestPatternErrors
           }
       STATUS  current
       DESCRIPTION
          "A collection of objects that support
          optional WIS device test features."
        ::= { etherWisGroups 2 }
```



```
   etherWisSectionGroup OBJECT-GROUP
       OBJECTS {
           etherWisSectionCurrentJ0Transmitted,
           etherWisSectionCurrentJ0Received
           }
       STATUS  current
       DESCRIPTION
          "A collection of objects that provide
          required information about a WIS section."
        ::= { etherWisGroups 3 }

   etherWisPathGroup OBJECT-GROUP
       OBJECTS {
           etherWisPathCurrentStatus,
           etherWisPathCurrentJ1Transmitted,
           etherWisPathCurrentJ1Received
           }
       STATUS  current
       DESCRIPTION
          "A collection of objects that provide
          required information about a WIS path."
```

```
        ::= { etherWisGroups 4 }

    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 5 }

    --     Compliance Statements

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

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



```
            OBJECT      etherWisDeviceTxTestPatternMode
            SYNTAX      INTEGER {
                none(0),
                squareWave(1),
                mixedFrequency(3)
                }
            DESCRIPTION
                "Support for values other than none(0),
                squareWave(1), and mixedFrequency(3)
                is not required."

            OBJECT      etherWisDeviceRxTestPatternMode
            SYNTAX      INTEGER {
                none(0),
```

```
            mixedFrequency(3)
            }
        DESCRIPTION
            "Support for values other than none(0)
            and mixedFrequency(3) is not required."

        GROUP       etherWisDeviceGroupExtra
        DESCRIPTION
            "Implementation of this group, along with support for
            the value prbs31(2) for etherWisDeviceTxTestPatternMode
            and etherWisDeviceRxTestPatternMode, is necessary if the
            optional PRBS31 test pattern mode is to be supported."

    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 five management objects defined in this MIB that have a MAX-ACCESS clause of read-write: etherWisDeviceTxTestPatternMode, etherWisDeviceRxTestPatternMode, etherWisDeviceRxTestPatternErrors, etherWisSectionCurrentJ0Transmitted, and etherWisPathCurrentJ1Transmitted. Setting these objects can have the following potentially disruptive effects on network operation:

- changing the transmit or receive test pattern mode or modifying the accumulated error count from a PRBS31 pattern test on an administratively disabled 10GBASE-W interface, which can interfere with an in-progress pattern test;

- modifying the transmitted section trace and/or path trace message on an operational 10GBASE-W interface, which can cause connectivity alarms to be raised at the remote of the link.

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) 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

7.1. Normative References

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

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

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

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

[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.

[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", <draft-ietf-atommib-rfc2558bis-00.txt>, 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-v3-01.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-v3-01.txt>, work in progress.

[P802.3ae] Institute of Electrical and Electronic Engineers, IEEE Draft P802.3ae/D5.0, "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", May 1, 2002, work in progress.

7.2. Informative References

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

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

[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.

[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.

8. Authors' Addresses

Mike Ayers
1204 Knox Ave.
San Jose, CA 95122
USA

Phone: +1 408 857 6810
EMail: mike.ayers@earthling.net

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

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
USA

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
934 S. Palos Verdes Dr.
Kaysville, Utah 84037
USA

Phone: +1 801 546 3316
EMail: kcn@norseth.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 50.3.11) 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 also start 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 simply be to increment the generalized non-resetable counters without consideration of any inhibiting rules.

Full Copyright Statement

Revision History

NOTE TO RFC Editor: this section is to be removed prior to publication as an RFC.

The following changes were made to <draft-ietf-hubmib-wis-mib-01.txt> to produce <draft-ietf-hubmib-wis-mib-02.txt>:

1.) Section 3.1 was updated as agreed in "WIS MIB compliance statement issue" e-mail thread, specifically to explain that the ETHER-WIS compliance statement requires all objects that required by sonetCompliance2 as well as some that it leaves optional.

2.) In Section 3.5 the paragraph dealing with STS-Path Remote Defect Indication was updated to indicate that an implementation of the WIS MIB (not necessarily an arbitrary implementation of the WIS) has to set the SONET-MIB's RDI status bit when a remote server defect is detected.

3.) In Section 3.6 the text introducing the table of semantic differences between IEEE GDMO and SNMP objects was wordsmithed.

4.) In Section 3.7 the mapping between objects in the etherWisDeviceTable and the station management registers was updated to reflect the updated MDIO register definitions in P802.3ae/D4.01 and the corresponding updates to the MIB objects.

5.) In Section 3.8.1 the text was updated to state that the purpose of the etherWisDeviceTable is to support mandatory and optional WIS test features.

6.) In Sections 3.8.1, 3.8.2, 3.8.3, and 3.8.4 certain instances of "SHALL" were changed to "MUST" to improve readability.

7.) The LAST-UPDATED, ORGANIZATION, REVISION, and DESCRIPTION clauses of the MODULE-IDENTITY invocation were updated.

8.) The lower-case "shall" in the DESCRIPTION clause of each table entry was changed to an uppercase "MUST", per RFC 2119, because it describes a requirement of the specification in the draft.

9.) The etherWisDeviceTestPatternType objects was removed, and the objects etherWisDeviceTxTestPatternMode and etherWisDeviceRxTestPatternMode were changed from simple "enable/disable" flags to enumerations that reflect the specific text pattern mode in which the transmitter or receiver is operating. A new optional object called etherWisDeviceRxTestPatternErrors was added to make visible the

error count MDIO register contents when the receiver is operating in the (optional) PRBS31 test pattern mode added in D4.01.

10.) The SYNTAX of etherWisSectionCurrentJ0Transmitted and etherWisSectionCurrentJ0Received was changed from INTEGER (0..255) to OCTET STRING (SIZE (16)) to reflect the change from 1-byte to 16-byte section trace messages in D4.01, and the DESCRIPTION clauses were rewritten along the lines for those of the corresponding path trace objects.

11.) In the DESCRIPTION of etherWisPathCurrentJ0Transmitted "path message" was changes to "path trace message".

12.) etherWisDeviceGroup was split into a mandatory group etherWisDeviceGroupBasic and an optional group etherWisDeviceGroupExtra.

13.) The compliance name was changed from etherWisCurrentCompliance to etherWisCompliance.

14.) OBJECT clauses were added to specify required values for etherWisDeviceTxTestPatternMode and etherWisDeviceRxTestPatternMode.

15.) A GROUP clause was added to state that etherWisDeviceGroupExtra needs to be implemented in order to support the PRBS31 test pattern mode. The DESCRIPTION clause points out that the prbs31 enumeration is needed for etherWisDeviceTxTestPatternMode/etherWisDeviceRxTestPatternMode.

16.) References [SONETng] and [P802.3ae] were updated to draft-ietf-atommib-rfc2558bis-00.txt and P802.3ae/D4.01, respectively.

The following changes were made to <draft-ietf-hubmib-wis-mib-02.txt> to produce <draft-ietf-hubmib-wis-mib-03.txt>:

1.) The page formatting was changed to match that used in recent RFCs.

2.) Heading numbering and bracketed references were removed from the abstract, and "SONET MIB" was expanded to "SONET/SDH Interface MIB".

3.) RFC 2119 terminology conventions were moved from the abstract into a separate numbered "Conventions" section and were wordsmithed to match the style of recent RFCs.

4.) Each occurrence of "SONET MIB" in Section 3, "Overview", was expanded to "SONET/SDH Interface MIB".

5.) Each occurrence of "MAU-MIB" in Section 3, "Overview", was expanded to "802.3 MAU MIB".

6.) Each occurrence of "SONET MIB" in Sections 3.1, 3.8.1, 3.8.2, 3.8.3, and 3.8.4 was changed to "SONET-MIB" (this matches the usage elsewhere, specifically Sections 3.2 and 3.3).

7.) All occurrences of "[P802.3ae] subclause 50.3.10" were updated to "[P802.3ae] subclause 50.3.11" in order to match the subclause numbering in P802.3ae/D4.2 (affected sections are 3.1, 3.6, 3.7, and Appendix A).

8.) "Ethernet-like Interfaces MIB" was changed to "Ethernet-like Interface MIB" in Section 3.2 in order to match the usage in the abstract.

9.) "mauModIfCompl2" was updated to "mauModIfCompl3" in Section 3.3 in order to match <draft-ietf-hubmib-mau-mib-v3-01.txt>.

10.) An unnecessary heading was removed the table in Section 3.6 that compares the semantics of IEEE threshold and counter objects with those of the corresponding objects imported from the SONET-

MIB.

11.) The text at the end of Section 3.6 was wordsmithed to remove unnecessary verbiage.

12.) The MDIO register name spellings in Section 3.7, Section 4, and Appendix A were updated to match those in P802.3ae/D4.2 subclause 45.2.2.

13.) A missing comma was added to the etherWisDeviceRxTestPatternErrors REFERENCE clause.

14.) DESCRIPTION clauses of etherWisSectionCurrentJ0Transmitted and etherWisPathCurrentJ1Transmitted were updates to state that the default value should be '89'h followed by 16 octets of '00'h, rather than 16 octets of '00'h followed by '89'h (this matches the resolution in P802.3ae/D4.2 of a byte ordering inconsistency).

15.) The MAX-ACCESS clauses of etherWisPathCurrentStatus and etherWisPathCurrentStatus were changed from 'read-write' to

16.) In Section 6, "Security Considerations", all read-write objects have been explicitly listed and mention of read-create objects has been removed.

17.) References [ETHERIF], [MAU-MIB], and [P802.3ae] were updated to <draft-ietf-hubmib-etherif-mib-v3-01.txt>, <draft-ietf-hubmib-mau-mib-v3-01.txt>, and P802.3ae/D4.2, respectively.

18.) Normative and informative references were moved into separate sub-sections.

19.) A "To-Do List" section was added.

The following changes were made to <draft-ietf-hubmib-wis-mib-03.txt> to produce <draft-ietf-hubmib-wis-mib-04.txt>:

1.) "tx" and "rx" were spelled out in the MDIO register names in Section 3.7 in order to match the usage in P802.3ae/D4.3 and P802.3ae/D5.0.

2.) References [IEEE 802.3 Std] in the MIB module and [P802.3ae] in the text were updated to point to P802.3ae/D5.0.

3.) Author contact information was updated.

To-Do List

NOTE TO RFC Editor: prior to publishing this document please take care of any open items listed below and then remove this section.

1.) All occurrences of "IEEE Draft P802.3ae/D5.0" must be changed to "IEEE Std 802.3ae" once the standard has been approved, the approval date must replace the draft publication date, and all occurrences of the reference tag [P802.3ae] must be changed to [802.3ae]. Note that this draft is referenced both in Section 4, "Object Definitions", and in Section 7.1, "Normative References".

2.) Normative references [SONETng], [ETHERIF], and [MAU-MIB] must be updated to point to the appropriate RFCs when the respective Internet Drafts are published, and all occurrences of the reference tags must be changed to [RFCnnnn] where nnnn is the assigned RFC number.

Table of Contents