

Definitions of Managed Objects
for the Ethernet WAN Interface Sublayer

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

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 Synchronous Optical Network/Synchronous Digital Hierarchy (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.

Table of Contents

| | | |
|--------|--|---|
| 1. | Conventions | 2 |
| 2. | The Internet-Standard Management Framework | 2 |
| 3. | Overview | 3 |
| 3.1. | Relationship to the SONET/SDH Interface MIB. | 3 |
| 3.2. | Relationship to the Ethernet-like Interface 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. | 15 |
| 3.8.3. | etherWisPathCurrentTable | 15 |
| 3.8.4. | etherWisFarEndPathCurrentTable | 15 |
| 4. | Object Definitions | 16 |
| 5. | Intellectual Property Statement. | 30 |
| 6. | Acknowledgments. | 30 |
| 7. | Security Considerations. | 31 |
| 8. | References | 32 |
| 8.1. | Normative References | 32 |
| 8.2. | Informative References | 33 |
| Appendix A: Collection of Performance Data Using WIS | | |
| | MDIO Registers | 34 |
| | Contributors | 35 |
| | Editor's Address | 36 |
| | Full Copyright Statement | 37 |

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 BCP 14, RFC 2119 [RFC2119].

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

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 [RFC3592], and the 802.3 MAU MIB [RFC3636] to manage the Ethernet Wide Area Network (WAN) Interface Sublayer (WIS) defined in [802.3ae]. The WIS contains functions to perform OC-192c/VC-4-64c framing and scrambling. It resides between the Physical Coding Sublayer (PCS) and the Physical Medium Attachment (PMA) sublayer within a 10GBASE-W 10 Gb/s WAN-compatible physical layer device (PHY) and may be used in conjunction with any of the PCS, PMA, and Physical Medium Dependent (PMD) sublayers defined in [802.3ae] 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 [802.3ae]. 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 [802.3ae].

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 [RFC3592]. 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 [802.3ae] 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 [RFC3636] 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 [802.3ae]. The corresponding use of the ifTable [RFC2863] is shown in the figure below.

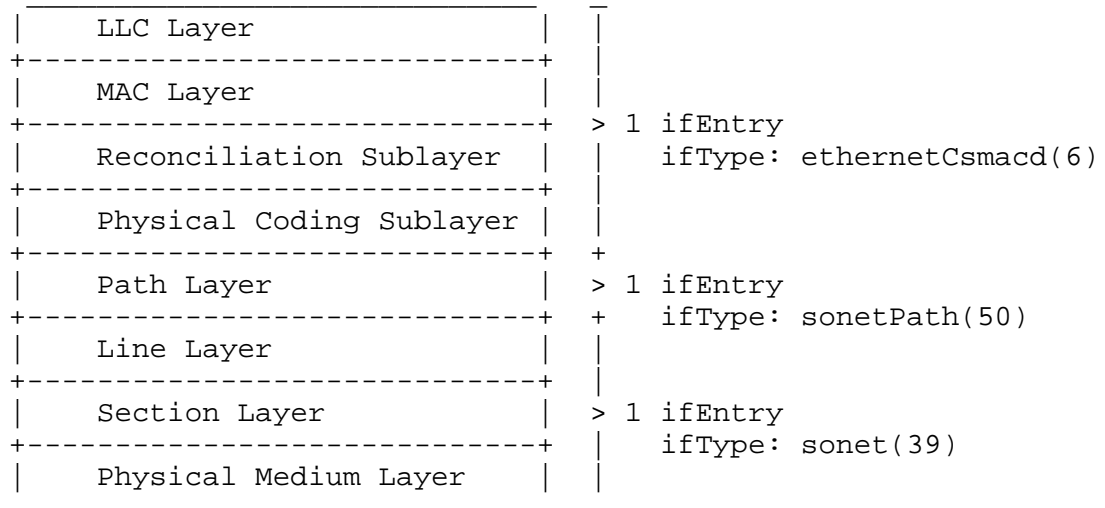


Figure 1 - Use of ifTable for an Ethernet WIS port

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 [RFC3635] and [RFC3636] 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 [RFC3592] 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 [RFC3592] 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 [802.3ae] is mostly the same as in [RFC3592], but there are a few differences. In those cases the definitions in [802.3ae] take precedence. The specific differences are as follows.

Unequipped

This defect is not defined by [802.3ae]. 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 [802.3ae]. It is reported by setting both the sonetPathSignalLabelMismatch bit in the appropriate instance of sonetPathCurrentStatus (defined in [RFC3592]) and the etherWisPathPLM bit in the corresponding instance of etherWisPathCurrentStatus (defined below).

Loss of Codegroup Delineation

[802.3ae] 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 [RFC3592]. It is reported by setting the etherWisPathLCD bit in the appropriate instance of the object etherWisPathCurrentStatus defined below.

STS-Path Remote Defect Indication

[802.3ae] 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). [RFC3592] defines the one-bit RDI-P (Remote Defect Indication - Path), which signals remote server defects (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 [802.3ae] 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 [802.3ae] and managed objects defined in this document and in associated MIB modules, i.e., the IF-MIB [RFC2863], the SONET-MIB [RFC3592], and the MAU-MIB [RFC3636].

| 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 - pWISOOptional package | |
| aSectionSESThreshold | SONET-MIB - sonetSESThresholdSet |
| aSectionSESS | SONET-MIB - sonetSectionCurrentSESSs + sonetSectionIntervalSESSs |
| aSectionESS | SONET-MIB - sonetSectionCurrentESSs + sonetSectionIntervalESSs |
| aSectionSEFSS | SONET-MIB - sonetSectionCurrentSEFSSs + sonetSectionIntervalSEFSSs |
| aSectionCVs | SONET-MIB - sonetSectionCurrentCVs + sonetSectionIntervalCVs |
| aJ0ValueTX | etherWisSectionCurrentJ0Transmitted |
| aJ0ValueRX | etherWisSectionCurrentJ0Received |
| aLineSESThreshold | SONET-MIB - sonetSESThresholdSet |
| aLineSESS | SONET-MIB - sonetLineCurrentSESSs + sonetLineIntervalSESSs |
| aLineESS | SONET-MIB - sonetLineCurrentESSs + sonetLineIntervalESSs |
| aLineCVs | SONET-MIB - sonetLineCurrentCVs + sonetLineIntervalCVs |
| aFarEndLineSESS | SONET-MIB - sonetFarEndLineCurrentSESSs + sonetFarEndLineIntervalSESSs |
| aFarEndLineESS | SONET-MIB - sonetFarEndLineCurrentESSs + sonetFarEndLineIntervalESSs |
| aFarEndLineCVs | SONET-MIB - sonetFarEndLineCurrentCVs + |

| | | |
|-------------------|-------------|---|
| | | sonetFarEndLineIntervalCVs |
| aPathSESThreshold | SONET-MIB - | sonetSESthresholdSet |
| aPathSESSs | SONET-MIB - | sonetPathCurrentSESSs + sonetPathIntervalSESSs |
| aPathESSs | SONET-MIB - | sonetPathCurrentESSs + sonetPathIntervalESSs |
| aPathCVs | SONET-MIB - | sonetPathCurrentCVs + sonetPathIntervalCVs |
| aJlValueTX | | etherWisPathCurrentJlTransmitted |
| aJlValueRX | | etherWisPathCurrentJlReceived |
| aFarEndPathSESSs | SONET-MIB - | sonetFarEndPathCurrentSESSs + sonetFarEndPathIntervalSESSs |
| aFarEndPathESSs | SONET-MIB - | sonetFarEndPathCurrentESSs + sonetFarEndPathIntervalESSs |
| 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 [802.3ae] 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. |
| aSectionSESSs | This object is defined in [802.3ae] as a generalized nonresetable counter. The objects sonetSectionCurrentSESSs and sonetSectionIntervalSESSs are 15-minute interval counters. |
| aSectionESSs | This object is defined as a generalized nonresetable counter in [802.3ae]. The objects sonetSectionCurrentESSs and sonetSectionIntervalESSs are 15-minute interval counters. |

| | |
|-------------------|--|
| aSectionSEFSs | This object is defined as a generalized nonresetable counter in [802.3ae]. The objects sonetSectionCurrentSEFSs and sonetSectionIntervalSEFSs are 15-minute interval counters. |
| aSectionCVs | This object is defined as a generalized nonresetable counter in [802.3ae], 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 [802.3ae] 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 [802.3ae], and it is not subject to inhibiting. The objects sonetLineCurrentSESSs and sonetLineIntervalSESSs 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 [802.3ae], and it is not subject to inhibiting. The objects sonetLineCurrentESSs and sonetLineIntervalESSs 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 [802.3ae], 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 [802.3ae], 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 [802.3ae], 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 [802.3ae], 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 [802.3ae] 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 [802.3ae], 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, [802.3ae] includes PLM-P and LCD-P defects in the criteria for declaring path layer severely errored seconds, while [RFC3592] does not.

aPathESS

This object is defined as a generalized nonresetable counter in [802.3ae], 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, [802.3ae] includes PLM-P and LCD-P defects in the criteria for declaring path layer errored seconds, while [RFC3592] does not.

aPathCVs

This object is defined as a generalized nonresetable counter in [802.3ae], 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 [802.3ae], 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, [802.3ae] includes far-end PLM-P and LCD-P defects in the criteria for declaring far-end path layer severely errored seconds, while [RFC3592] does not.

aFarEndPathESS

This object is defined as a generalized nonresetable counter in [802.3ae], 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, [802.3ae] includes far-end PLM-P and LCD-P defects in the criteria for declaring far-end path layer errored seconds, while [RFC3592] does not.

aFarEndPathCVs

This object is defined as a generalized nonresetable counter in [802.3ae], 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 [802.3ae] 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 [RFC3592] or the MAU-MIB [RFC3636] require WIS-specific hardware support. [802.3ae] subclause 50.3.11 specifies WIS management interface requirements, including a required subset of the WIS Management Data Input/Output (MDIO) registers defined in [802.3ae] subclause 45.2.2. The table below provides a cross-reference between those managed objects and the WIS MDIO registers from the subset in [802.3ae] 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 | \ | 10G WIS status 3 + 10G WIS path block error count |
| SONET-MIB - sonetPathCurrentSESS | | |
| SONET-MIB - sonetPathCurrentCVs | | |
| SONET-MIB - sonetPathCurrentUASs | | |
| SONET-MIB - sonetPathIntervalESS | | |
| SONET-MIB - sonetPathIntervalSESS | | |
| SONET-MIB - sonetPathIntervalCVs | | |
| SONET-MIB - sonetPathIntervalUASs | | |
| SONET-MIB - sonetPathIntervalValidData | / | none required |
| ETHER-WIS - etherWisFarEndPathCurrentStatus | | 10G WIS status 3 |
| SONET-MIB - sonetFarEndPathCurrentESS | \ | 10G WIS status 3 + 10G WIS far end path block error count |
| SONET-MIB - sonetFarEndPathCurrentSESS | | |
| SONET-MIB - sonetFarEndPathCurrentCVs | | |
| SONET-MIB - sonetFarEndPathCurrentUASs | | |
| SONET-MIB - sonetFarEndPathIntervalESS | | |
| SONET-MIB - sonetFarEndPathIntervalSESS | | |
| SONET-MIB - sonetFarEndPathIntervalCVs | | |
| 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 [802.3ae] 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 MUST 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 MUST 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 MUST 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 MUST be used for both entries.

4. Object Definitions

```
ETHER-WIS DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE,  
    Gauge32, 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 "200309190000Z" -- September 19, 2003  
    ORGANIZATION "IETF Ethernet Interfaces and Hub MIB  
        Working Group"
```

```
    CONTACT-INFO
```

```
        "WG charter:  
        http://www.ietf.org/html.charters/hubmib-charter.html
```

```
        Mailing Lists:
```

```
            General Discussion: hubmib@ietf.org  
            To Subscribe: hubmib-request@ietf.org  
            In Body: subscribe your_email_address
```

```
        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 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 Std 802.3ae-2002, 'IEEE Standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Media Access Control (MAC) Parameters, Physical Layer and Management Parameters for 10 Gb/s Operation', 30 August 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'.

Copyright (C) The Internet Society (2003). This version of this MIB module is part of RFC 3637; see the RFC itself for full legal notices."

REVISION "200309190000Z" -- September 19, 2003

DESCRIPTION "Initial version, published as RFC 3637."

::= { transmission 134 }

-- 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(1),  
    squareWave(2),  
    prbs31(3),  
    mixedFrequency(4)  
}
```

```
MAX-ACCESS  read-write
```

```
STATUS  current
```

DESCRIPTION

"This variable controls the transmit test pattern mode. The value none(1) puts the the WIS transmit path into the normal operating mode. The value squareWave(2) 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(3) 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(4) 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(1) when the corresponding instance of ifAdminStatus 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(1) 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(1),  
    prbs31(3),  
    mixedFrequency(4)  
}
```

```
MAX-ACCESS  read-write
```

```
STATUS  current
```

DESCRIPTION

"This variable controls the receive test pattern mode. The value none(1) puts the the WIS receive path into the normal operating mode. The value prbs31(3) 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(4) 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(1) when the corresponding instance of ifAdminStatus 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(1) 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 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 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 signaled
            in G1 bits 5-7.
```

```

    etherWisFarEndServerDefect(1)
        A far end server defect (i.e., far end
        LOP-P or AIS-P) is currently being signaled
        in G1 bits 5-7. Note: when this bit is set,
        sonetPathSTSREDI 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 interfaces that include
        the Ethernet WIS.  Compliance with the following
        external compliance statements is prerequisite:

        MIB Module                      Compliance Statement
        -----
        IF-MIB                          ifCompliance3
        IF-INVERTED-STACK-MIB          ifInvCompliance
        EtherLike-MIB                  dot3Compliance2
        MAU-MIB                        mauModIfCompl3"

```

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

  OBJECT          etherWisDeviceTxTestPatternMode
  SYNTAX          INTEGER {
    none(1),
    squareWave(2),
    mixedFrequency(4)
  }
  DESCRIPTION
    "Support for values other than none(1),
    squareWave(2), and mixedFrequency(4)
    is not required."

  OBJECT          etherWisDeviceRxTestPatternMode
  SYNTAX          INTEGER {
    none(1),
    mixedFrequency(4)
  }
  DESCRIPTION
    "Support for values other than none(1)
    and mixedFrequency(4) is not required."

  GROUP          etherWisDeviceGroupExtra
  DESCRIPTION
    "Implementation of this group, along with support for
    the value prbs31(3) for etherWisDeviceTxTestPatternMode
    and etherWisDeviceRxTestPatternMode, is necessary if the
    optional PRBS31 test pattern mode is to be supported."

  OBJECT          etherWisDeviceRxTestPatternErrors
  WRITE-SYNTAX Gauge32 ( 0 )
  DESCRIPTION
    "An implementation is not required to
    allow values other than zero to be
    written to this object."
```

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. Intellectual Property Statement

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.

6. Acknowledgments

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

7. Security Considerations

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

- o 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;
- o 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.

The user of this MIB module must therefore be aware that support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

The readable objects in this MIB module (i.e., those with MAX-ACCESS other than not-accessible) may be considered sensitive in some environments since, collectively, they provide information about the performance of network interfaces and can reveal some aspects of their configuration. In such environments it is important to control even GET and NOTIFY access to these objects and possibly even to encrypt their values when sending them over the network via SNMP.

SNMP versions prior to SNMPv3 did not include adequate security. 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 module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module 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.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirements Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [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.
- [RFC3592] Tesink, K., "Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type", RFC 3592, September 2003.
- [T1.231] American National Standard for Telecommunications - Digital Hierarchy - Layer 1 In-Service Digital Transmission Performance Monitoring, ANSI T1.231-1997, September 1997.
- [RFC3635] Flick, J., "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 3635, September 2003.
- [RFC3636] Flick, J., "Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)", RFC 3636, September 2003.

- [802.3ae] Institute of Electrical and Electronic Engineers, IEEE Std 802.3ae-2002, "IEEE Standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Media Access Control (MAC) Parameters, Physical Layer and Management Parameters for 10 Gb/s Operation", August 2002.

8.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

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 [802.3ae] subclause 45.2.2 (and more specifically the subset required by [802.3ae] 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 [802.3ae] 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 [RFC3592]. 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 [802.3ae] 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-resettable counters without consideration of any inhibiting rules.

Contributors

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

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
L-3 Communications
640 N. 2200 West.
Salt Lake City, Utah 84116-0850
USA

Phone: +1 801 594 2809
EMail: kenyon.c.norseth@L-3com.com
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

Editor's Address

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

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

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "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.

Acknowledgement

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

