IEEE STANDARDS ASSOCIATION

IEEE Standard for Management Information Base (MIB) Definitions for Ethernet

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

5 July 2011

IEEE Std 802.3.1™-2011

IEEE Std 802.3.1™-2011

IEEE Standard for Management Information Base (MIB) Definitions for Ethernet

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 16 May 2011

IEEE-SA Standards Board

Permissions have been granted as follows:[a]

Portions of this standard reprinted with permission from Avaya, IETF RFC 2108 Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIv2,© 1997.

Portions of this standard reprinted with permission from Microsemi Corporation, IETF RFC 3621 Definitions Power Ethernet MIB, © 2003.

Portions of this standard reprinted with permission from Avaya, IETF RFC 3621 Definitions Power Ethernet MIB, © 2003.

Portions of this standard reprinted with permission from the Hewlett-Packard Company, IETF RFC 3635 Definitions of Managed Objects for the Ethernet-like Interface Types,© 2003.

Material from IETF RFC 4836 Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs) reprinted with permission from the IETF Trust and the persons identified as the document authors,© 2007.

Portions of this standard reprinted with permission from Edward Beili, IETF RFC 4836 Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs),© 2007.

Material from IETF RFC 4837 Managed Objects of Ethernet Passive Optical Networks (EPON) reprinted with permission from the IETF Trust and the persons identified as the document authors,© 2007.

Portions of this standard reprinted with permission from PMC-Sierra, IETF RFC 4837 Managed Objects of Ethernet Passive Optical Networks (EPON),© 2007.

Material from IETF RFC 4878 Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces reprinted with permission from the IETF Trust and the persons identified as the document authors, © 2007.

Material from IETF RFC 5066 Ethernet in the First Mile Copper (EFMCu) Interfaces MIB reprinted with permission from the IETF Trust and the persons identified as the document authors.

Portions of this standard reprinted with permission from Edward Beili, IETF RFC 5066 Ethernet in the First Mile Copper (EFMCu) Interfaces MIB,© 2007.

[a]Every effort has been made to secure permission to reprint borrowed material contained in this document. If omissions have been made, please bring them to our attention.

Abstract: The Management Information Base (MIB) module specifications for IEEE Std 802.3, also known as Ethernet, are contained in this standard. It includes the Structure of Management Information Version 2 (SMIv2) MIB module specifications formerly produced and published by the Internet Engineering Task Force (IETF), and the Guidelines for the Definition of Managed Objects (GDMO) MIB modules formerly specified within IEEE Std 802.3, as well as extensions resulting from amendments to IEEE Std 802.3. The SMIv2 MIB modules are intended for use with the Simple Network Management Protocol (SNMP), commonly used to manage Ethernet.

Keywords: Ethernet, GDMO, IEEE 802.3.1, MIB, network management, SNMP, SMIv2

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-0-7381-6700-8 STD97135
Print: ISBN 978-0-7381-6701-5 STDPD97135

Introduction

This introduction is not part of IEEE Std 802.3.1-2011, IEEE Standard for Management Information Base (MIB) definitions for Ethernet.

The initial version of this standard was based on the managed object definitions provided in IEEE Std 802.3™-2008, which subsumed and superseded IEEE Std 802.3an™-2006, IEEE Std 802.3ap™-2007, IEEE Std 802.3aq™-2006, and IEEE Std 802.3as™-2006. It also includes the Logical Link Discovery Protocol Ethernet extensions provided in IEEE Std 802.1AB™-2009 Annex F. In addition, the initial version of this standard incorporated and updated the MIB module definitions formerly defined in IETF RFC 2108, IETF RFC 3621, IETF RFC 3635, IETF RFC 3637, IETF RFC 4836, IETF RFC 4837, IETF RFC 4878, and IETF RFC 5066.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association website at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this amendment may require use of subject matter covered by patent rights. By publication of this amendment, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this amendment are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

The following individuals were officers and members of the IEEE 802.3 working group at the beginning of the IEEE P802.3.1 working group ballot. Individuals may have not voted, voted for approval, disapproval, or abstained on this standard.

David J. Law, *IEEE 802.3 Working Group Chair*
Wael William Diab, *IEEE 802.3 Working Group Vice-Chair*
Steven B. Carlson, *IEEE 802.3 Working Group Executive Secretary*
Adam Healey, *IEEE 802.3 Working Group Secretary*
Valerie Maguire, *IEEE 802.3 Working Group Treasurer*

Howard M. Frazier, *IEEE P802.3.1 MIB Task Force Chair and Editor*
John Hawkins,[b] *IEEE P802.3.1 MIB Task Force Assistant Editor*

Ghani Abbas
John Abbott
Andrew Ambrose
Jon Anderson
Peter Anslow
Vittal Balasubramanian
Thananya Baldwin
Jim Barnette
Hugh Barrass
Denis Beaudoin
Jonathan Beckwith
Michael Bennett
Ernest Bergmann
Brad Booth
Ralf-Peter Braun
Dirk Breuer
Matthew Brown
Robert Busse
J. Martin Carroll
Mandeep Chadha
David Chalupsky
Frank Chang
Joseph Chou
Terry Cobb
Christopher R. Cole
Doug Coleman
Charles Cook
John Dambrosia
Piers Dawe
Chris Diminico
Thomas Dineen
Dan Dove
Mike Dudek
Frank Effenberger
David Estes
Farzin Firoozmand
Alan Flatman
Ilango S. Ganga
Ali Ghiasi
Dimitrios Giannakopoulos
Larry Green
Michael Grimwood
Robert M. Grow
Mark Gustlin
Paul Gyugyi
Marek Hajduczenia
Hiroshi Hamano
Bernie Hammond
Jeffrey Heath
Rita Horner
Nicholas Ilyadis
James Innis
Osamu Ishida
Hirotake Iwadate
Qiaofeng Jiang
Chad Jones
Sanjay Kasturia
Seung-Hwan Kim
Yong Kim
Mitsunobu Kimura
Jonathan King
Scott Kipp
Shoukei Kobayashi
David Koenen
Paul Kolesar
Seiji Kozaki
Glen Kramer
Hans Lackner
Lowell Lamb
D. Matthew Landry
Jeff Lapak
Ryan Latchman
Kyusang Lee
David Lewis
Mike Peng Li
Sharon Lutz
Eric Lynskey
Anthony Magee
Jeffery J. Maki
Jeff Mandin
Arthur Marris
Phil Mcclay
Michael S. McCormack
John McDonough
Jim McGrath
Greg McSorley
Brian Misek
Merrick Moeller
Shimon Muller
Edward Nakamoto
Gary Nicholl
Paul Nikolich
George Noh
Ahmad Nouri
Mark Nowell
Satoshi Obara
David Ofelt
Gourgen Oganessyan
Raj Ojha
George Oulundsen
Tom Palkert
Sesha Panguluri
Gavin Parnaby
Petar Pepeljugoski
Gerald Pepper
John Petrilla
Velu Pillai
Rick Pimpinella
Scott Powell
Rick Rabinovich
Randy Rannow
Peter Rautenberg
Duane Remein
Michael Ressl
Sam sambasivan
Olindo Savi
Frederick Schindler
Shawn Searles

[b]Not a member of the IEEE 802.3 working group at the beginning of the working group ballot.

Murat Serbay
Farhad Shafai
Hyungsoo Shin
Bryan Sparrowhawk
Olaf Storaasli
Alan Sugg
Ken-Ichi Suzuki
Steve Swanson
Andre Szczepanek
Dimitry Taich
Akio Tajima
Hidenori Takahashi
Motoyuki Takizawa
Sashi Thiagarajan
Geoffrey Thompson
Hidehiro Toyoda
Nathan Tracy
Mario Traeber
Matthew Traverso
Stephen Trowbridge
Shinji Tsuji
Brad Turner
Sterling A. Vaden
Paul Vanderlaan
Anoop Vetteth
George Young
George Zimmerman
Pavel Zivny

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Thomas Alexander
Mark Anderson
Peter Anslow
Lee Armstrong
Jacob Ben Ary
Hugh Barrass
Michael Bennett
Tomo Bogataj
Ralf-Peter Braun
Nancy Bravin
Matthew Brown
William Byrd
James Carlo
Steven B. Carlson
Juan Carreon
Mandeep Chadha
David Chalupsky
Keith Chow
Charles Cook
Fumio Daido
John D'Ambrosia
Wael William Diab
Patrick Diamond
Russell Dietz
Thomas Dineen
Daniel Dove
John Egan
Andrew Fieldsend
C. Fitzgerald
Howard M. Frazier
Yukihiro Fujimoto
Ilango Ganga
Devon Gayle
Eric Gray
Randall Groves
Robert M. Grow
Marek Hajduczenia
Hiroshi Hamano
John Hawkins
Adam Healey
Ryan Hirth
C. Huntley
James Innis
Atsushi Ito
Raj Jain
Junghoon Jee
Shinkyo Kaku
Stuart J. Kerry
Yongbum Kim
Scott Kipp
David Landry
David J. Law
Li Li
Ru Lin
G. Luri
Elvis Maculuba
Valerie Maguire
Jeffery Masters
Phil Mcclay
Brett Mcclellan
Jonathon Mclendon
Jose Morales
Charles Ngethe
Satoshi Obara
Sesha Panguluri
Glenn Parsons
Peter Pepeljugoski
Venkatesha Prasad
Robert Robinson
Randall Safier
Bartien Sayogo
Siddharth Sheth
Gil Shultz
Kapil Sood
Matthew Squire
Manikantan Srinivasan
Thomas Starai
Walter Struppler
Geoffrey Thompson
Edward J. Turner
Prabodh Varshney
George Vlantis
Oren Yuen

Acknowledgements

This document was derived in part from several Request for Comments (RFCs) originally published by the Internet Engineering Task Force (IETF). The individuals listed below contributed to the development of those RFCs, and earlier works on the subject matter.

Brian Arnold
Udi Ashkenazi
Ran Atkinson
Mike Ayers
Hugh Barrass
Edward Beili
Avi Berger
Chuck Black
Brad Booth
Al Braga
Ben Brown
John Cook
Kevin Daines
James Davin
Kathryn de Graaf
Wael Willaim Diab
John Flick
Howard M. Frazier
Floyd Gerhardt
Robert M. Grow
Mike Heard
Alfred Hoenes
Jeff Johnson
Frank Kastenholz
Lior Khermosh
Lynn Kubinec
Kam Lam
David J. Law
Leon Long
Mike Lui
Eric Lynskey
David Martin
Keith McCloghrie
Kerry McDonald
Donna McMaster
Steve McRobert
John Messenger
K. C. Norseth
Dave Perkins
Marina Popilov
Randy Presuhn
Mathias Riess
Anil Rijsinghani
Sam Roberts
Dan Romascanu
Marshall Rose
Scott Simon
Andrew Smith
Matt Squire
Louis Steinberg
Kaj Tesink
Jonathan Thatcher
Geoffrey Thompson
Maurice Turcotte
Bert Wijnen
Paul Woodruff

When the IEEE-SA Standards Board approved this standard on 16 May 2011, it had the following membership:

Contents

IEEE Standard for Management Information Base (MIB) Definitions for Ethernet

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

This document supersedes and makes obsolete IEEE Std 802.3™-2008 Annex 30A and Annex 30B, IEEE Std 802.1AB™-2009 Annex F, IETF RFC 2108, IETF RFC 3621, IETF RFC 3635, IETF RFC 3637, IETF RFC 4836, IETF RFC 4837, IETF RFC 4878, and IETF RFC 5066.[1]

Ethernet technology, as defined by the IEEE 802.3 Working Group of the IEEE, continues to evolve, with scalable increases in speed, new types of cabling and interfaces, and new features. This evolution may require changes in the managed objects in order to reflect this new functionality. This document, as with other documents issued by this working group, reflects a certain stage in the evolution of Ethernet technology. In the future, this document might be revised, or new documents might be issued, in order to reflect the evolution of Ethernet technology.

The term "Ethernet-like interfaces" was historically used because the interfaces defined by the IEEE 802.3 Working Group were not considered "Ethernet" per se, but "Ethernet-like", because "Ethernet" was taken to mean "Ethernet version 2" according to the (DEC, Intel, Xerox) DIX "blue book". Today and in the context of SNMP management and SMIv2 MIB modules, "Ethernet," "Ethernet-like," and "IEEE 802.3" are synonymous and interchangeable in the marketplace. The term "Ethernet-like" is retained in this document because of its common usage in the SNMP-based network management community.

[1]Information on references can be found in Clause 2.

1.1 Scope

This standard contains the Management Information Base (MIB) module specifications for IEEE Std 802.3, also known as Ethernet. It includes the Structure of Management Information Version 2 (SMIv2) MIB module specifications formerly produced and published by the Internet Engineering Task Force (IETF), and the Guidelines for the Definition of Managed Objects (GDMO) MIB modules formerly specified within IEEE Std 802.3, as well as extensions resulting from recent amendments to IEEE Std 802.3. The SMIv2 MIB modules are intended for use with the Simple Network Management Protocol (SNMP), commonly used to manage Ethernet.

1.2 Purpose

The purpose of the standard is to publish the SMIv2 and GDMO MIB module specifications in a single document that is separate from IEEE Std 802.3, and that can be published in a machine-readable format. Future amendments and revisions to IEEE Std 802.3.1 will be performed to update the MIB specifications as required to track future amendments and revisions to IEEE Std 802.3.

1.3 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 IETF RFC 3410.

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 standard specifies MIB modules that are compliant to the SMIv2, which is described in IETF STD 58 (RFC 2578), IETF STD 58 (RFC 2579), and IETF STD 58 (RFC 2580).

1.4 Security considerations

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), 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 a MIB module.

Implementers should consider the security features as provided by the SNMPv3 framework (see IETF RFC 3410, section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

SNMPv3 should be deployed, rather than previous versions of SNMP, and cryptographic security should be enabled. 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.

Throughout this standard, there are a number of accessible management objects that may be considered sensitive or vulnerable in some network environments. The support for some operations in a non-secure environment without proper protection can have a negative effect on network operations. Such management objects are detailed in the clauses that define them.

The user of these MIB modules should 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 these MIB modules (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 GET and NOTIFY access to these objects and possibly encrypt their values when sending them over the network via SNMP.

1.5 Conformance

Specific conformance statements and compliance statements, written in accordance with IETF STD 58, RFC 2580, are included in each MIB module. They can be found by searching for the text strings "Conformance statements" and "Compliance statements".

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

ANSI T1.231-1997, Layer 1 In-Service Digital Transmission Performance Monitoring.[2]

ANSI T1.424-2004, Interface Between Networks and Customer Installation—Very-high-bit-rate Digital Subscriber Lines (VDSL) Metallic Interface (DMT Based).

ETSI TS1 101 270-1, (1999) Transmission and Multiplexing (TM); Access transmission systems on metallic access cables; Very high speed Digital Subscriber Line (VDSL); Part 1: Functional requirements.[3]

IEEE Std 802®, IEEE Standard for Local and Metropolitan Area Nelacktworks: Architecture and Overview.[4, 5]

IEEE Std 802.1D™, IEEE Standard for Local and metropolitan area networks Media Access Control (MAC) Bridges.

IEEE Std 802.1AB-2009 Station and Media Access Control Discovery.

IEEE Std 802.3, 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.

IETF RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, McCloghrie, K., and M. Rose, March 1991.[6]

IETF RFC 1516, *Definitions of Managed Objects for IEEE 802.3 Repeater Devices*, McMaster, D., and K. McCloghrie, September 1993.

IETF RFC 2119, *Keywords for use in RFCs to Indicate Requirement Levels*, Bradner, S., March 1997.

IETF RFC 2434, *Guidelines for Writing an IANA Considerations Section in RFCs*, Narten, T. and H. Alvestrand, October 1998.

IETF STD 58 (RFC 2578), *Structure of Management Information Version 2 (SMIv2)*, McCloghrie, K., Perkins, D., and J. Schoenwaelder, April 1999.

IETF STD 58 (RFC 2579), *Textual Conventions for SMIv2*, McCloghrie, K., Perkins, D., and J. Schoenwaelder, April 1999.

[2]ANSI publications are available from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[3]ETSI standards are available from the European Telecommunications Standards Institute at 650, Route des Lucioles, 06921 Sophia-Antipolis Cedex, France (http://www.etsi.org/).

[4]IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[5]The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[6]Internet Requests for Comments (RFCs) are available on the World Wide Web at the following ftp site: venera.isi.edu; logon: anonymous; password: user's e-mail address; directory: in-inotes.

IETF STD 58 (RFC 2580), *Conformance Statements for SMIv2*, McCloghrie, K., Perkins, D., and J. Schoenwaelder, April 1999.

IETF RFC 2856, *Textual Conventions for Additional High Capacity Data Types*, Bierman, A., McCloghrie, K., and R. Presuhn, June 2000.

IETF RFC 2863, *The Interfaces Group MIB*, McCloghrie, K. and F. Kastenholz, June 2000.

IETF RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*, McCloghrie, K. and G. Hanson, June 2000.

IETF RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework,* Case, J., Mundy R., Partain, D. and B. Stewart, December 2002.

IETF RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*, Harrington, D., Presuhn, R. and B. Wijnen, December 2002.

IETF RFC 3592, *Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type*, Tesink, K., September 2003.

ITU-T Recommendation G.983.1, 1998—Optical line systems for local and access networks—Broadband optical access systems based on Passive Optical Networks (PON).[7]

ITU-T Recommendation G.991.2, 2003—Single-pair High-speed Digital Subscriber Line (SHDSL) transceivers.

ITU-T Recommendation G.993.1, 2004—Very High speed Digital Subscriber Line transceivers.

[7]ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

3. Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary: Glossary of Terms & Definitions* [B6] should be consulted for terms not defined in this clause.[8]

3.1 agent: An entity, typically implemented in software, which provides remote access to management instrumentation, via the Simple Network Management Protocol (SNMP).

3.2 group: Within the context of the repeater management MIB module defined in Clause 7: A recommended, but optional, entity defined in IEEE Std 802.3, Clause 30, in order to support a modular numbering scheme. The classical example allows an implementor to represent field-replaceable units as groups of ports, with the port numbering matching the modular hardware implementation.

3.3 jack type: The jack connector type, as it appears on the outside of the system. The type of mechanical interface to the transmission medium.

3.4 Loss of Codegroup Delineation: *See* IEEE Std 802.3 50.3.5.3.

3.5 managed object: an abstract representation of network resources that are managed. A managed object is defined according to the *Structure of Management Information version 2* (SMIv2) defined in IETF STD 58, RFC 2578.

3.6 managed repeater: A repeater as defined by IEEE Std 802.3 incorporating a management entity that complies with the MIB module definition contained in Clause 7 of this document.

3.7 module: A building block in a modular system. In the context of the MIB definitions, a specification of management capabilities related to the system. In the context of a chassis, it typically maps into one 'slot'; however, the range of configurations may be very large, with several modules entering one slot, or one module covering several slots.

3.8 non-trivial repeater: A repeater as defined by IEEE Std 802.3 having multiple ports.

3.9 Path Coding violations: In IEEE Std 802.3 the path layer coding violations 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 ANSI T1.231-1997 allows either path BIP-8 errors or path block errors to be used for the path layer error count.

3.10 repeater system: A managed entity compliant with this standard, and incorporating at least one managed IEEE Std 802.3 repeater.

3.11 repeater-unit: The portion of a repeater that is inboard of its Physical Medium Attachment (PMA)/ Physical Signaling Sublayer (PLS), or PMA/Physical Coding Sublayer (PCS).

3.12 Signal Label Mismatch: This defect is called Payload Label Mismatch (PLM) in IEEE Std 802.3. It is reported by setting both the sonetPathSignalLabelMismatch bit in the appropriate instance of sonetPathCurrentStatus (defined in IETF RFC 3592) and the etherWisPathPLM bit in the corresponding instance of etherWisPathCurrentStatus.

3.13 stack: A scalable system in which modularity is achieved by interconnecting a number of different systems.

[8]The numbers in brackets correspond to those of the bibliography in Annex A.

3.14 STS-Path Remote Defect Indication: IEEE Std 802.3 mandates the use of ERDI-P (Enhanced Remote Defect Indication-Path) defined in ANSI T1.231-1997 to signal remote server defects (triggered by path AIS or path LOP) and remote payload defects (triggered by Payload Label Mismatch or Loss of Code-group Delineation). IETF RFC 3592 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 Clause 12 sets 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.

3.15 system: An entity compliant with one or more MIB modules of this standard.

3.16 system interconnect segment: An internal segment allowing interconnection of ports belonging to different physical entities into the same logical managed repeater, bridge, or other system. Examples of implementation might be backplane busses in modular hubs, or chaining cables in stacks of bridges/switches. It is not uncommon for such segments to be a proprietary implementation.

3.17 trivial repeater-unit: An isolated port that can gather statistics.

4. Abbreviations

ACK	acknowledge
AIS	Alarm Indication Signal
ARP	address resolution protocol
ASCII	American Standard Code for Information Interchange
Atn	attenuation
BER	bit error ratio
BIP	bit interleaved parity
BW	bandwidth
CO	central office
CPE	customer premises equipment
CRC	cyclic redundancy check
DTE	data terminal equipment
EFM	Ethernet in the First Mile
EFMCu	EFM copper
ELTE	Ethernet line termination equipment
EPON	Ethernet passive optical network
ERDI-P	enhanced remote defect indication - path
FCS	frame check sequence
FEC	forward error correction
GDMO	Guidelines for Definition of Managed Objects
GMII	gigabit media independent interface
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IFG	inter-frame gap
ITU	International Telecommunication Union
LAN	local area network
LCD	Loss of Codegroup Deliniation
LLC	logical link control
LLDP	logical link discovery protocol
LLDPDU	logical link discovery protocol data unit
LLID	logical link identifier
LOP	Loss of Pointer
MAC	media access control
MAU	medium attachment unit
Mb/s	megabit per second
MDI	medium dependent interface
MDIO	management data input/output
MII	media independent interface
MP2PE	multipoint-to-point emulation
MPCP	multipoint control protocol
MPCPDU	multipoint control protocol data unit
MTU	maximum transmission unit
NMS	network management system
OAM	operations, administration and maintenance

OAMPDU	operations, administration, and maintenance protocol data unit
OID	object identifier
OLT	optical line terminal
OMP	optical multipoint
ONU	optical network unit
OSI	Open Systems Interconnection
P2MP	point-to-multipoint
P2PE	point-to-point emulation
PAF	PME aggregation function
PBO	power back-off
PCS	physical coding sublayer
PD	powered device
PDU	protocol data unit
PHY	Physical Layer entity
PLM	Payload Label Mismatch
PMA	physical medium attachment
PMD	physical medium dependent
PME	physical medium entity
PON	passive optical network
PSD	power spectral density
PSE	power sourcing equipment
RFC	Request for Comments
ROM	read-only-memory
RS	reconciliation sublayer
RTT	round-trip time
SDH	Synchronous Digital Hierarchy
SLA	service level agreement
SLD	start of LLID delimiter
SMIv2	structure of management information version 2
SNMP	simple network management protocol
SNR	signal to noise ratio
SONET	Synchronous Optical Network
TCPAM	trellis coded pulse amplitude modulation
TDM	time division multiplexing
TDMA	time division multiple access
TLV	type/length/value
TQ	time quanta
WAN	wide area network
WDM	wavelength division multiplexing
WIS	WAN interface sublayer

5. Ethernet logical link discovery protocol (LLDP) extension MIB module

The logical link discovery protocol (LLDP) is defined in IEEE Std 802.1AB-2009 Station and Media Access Control Discovery. Extensions to this protocol for Ethernet are defined in Clause 79 of IEEE Std 802.3.

5.1 Structure of the IEEE 802.3 LLDP extension MIB

Table 5–1 summarizes the particular object groups that are required for each operating mode. The implemented MIB shall comply with the MIB conformance section for the particular operating mode being supported.

Table 5–1—IEEE 802.3 LLDP extension MIB object group conformance requirements

MIB group	Rx mode	Tx mode	Tx/Rx mode
lldpV2Xdot3ConfigGroup	M[a]	M	M
lldpV2Xdot1LocSysGroup	M	—	M
lldpV2Xdot1RemSysGroup	—	M	M
ifGeneralInformationGroup	M	M	M

[a]M = Mandatory

Table 5–2 shows the structure of the MIB and the relationship of the MIB objects to the LLDP operational status/control variables, LLDP statistics variables, and TLV variables.

5.2 Relationship to other MIBs

Version 1 of the IEEE 802.3 LLDP extension MIB module is deprecated.

Table 5–2—IEEE 802.3/LLDP extension MIB cross reference

MIB table	MIB object	LLDP reference
Configuration group		
lldpV2Xdot3PortConfigTable		Augments lldpV2PortConfigEntry
	lldpV2Xdot3PortConfigTLVsTxEnable	Normal LLDPDUs
Local devices information group		
lldpV2Xdot3LocPortTable		
	lldpV2LocPortIfIndex	(Table index)
	lldpV2Xdot3LocPortAutoNegSupported	Auto-Negotiation support/status
	lldpV2Xdot3LocPortAutoNegEnabled	Auto-Negotiation support/status
	lldpV2Xdot3LocPortAutoNegAdvertisedCap	Auto-Negotiation advertised
	lldpV2Xdot3LocPortOperMauType	Operational MAU type
lldpV2Xdot3LocPowerTable		
	lldpV2LocPortIfIndex	(Table index)
	lldpV2Xdot3LocPowerPortClass	MDI power support
	lldpV2Xdot3LocPowerMDISupported	MDI power support
	lldpV2Xdot3LocPowerMDIEnabled	MDI power support
	lldpV2Xdot3LocPowerPairControlable	MDI power support
	lldpV2Xdot3LocPowerPairs	PSE power pair
	lldpV2Xdot3LocPowerClass	Power class
lldpV2Xdot3LocMaxFrameSizeTable		
	lldpV2LocPortIfIndex	(Table index)
	lldpV2Xdot3LocMaxFrameSize	Maximum frame size
Remote devices information group		
lldpV2Xdot3RemPortTable		
	lldpV2RemTimeMark	(Table index)
	lldpV2RemLocalIfIndex	(Table index)
	lldpV2RemLocalDestMACAddress	(Table index)
	lldpV2RemIndex	(Table index)
	lldpV2Xdot3RemPortAutoNegSupported	Auto-Negotiation support/status
	lldpV2Xdot3RemPortAutoNegEnabled	Auto-Negotiation support/status
	lldpV2Xdot3RemPortAutoNegAdvertisedCap	Auto-Negotiation advertised
	lldpV2Xdot3RemPortOperMauType	Operational MAU type

Table 5–2—IEEE 802.3/LLDP extension MIB cross reference *(continued)*

MIB table	MIB object	LLDP reference
lldpV2Xdot3RemPowerTable		
	lldpV2RemTimeMark	(Table index)
	lldpV2RemLocalIfIndex	(Table index)
	lldpV2RemLocalDestMACAddress	(Table index)
	lldpV2RemIndex	(Table index)
	lldpV2Xdot3RemPowerPortClass	MDI power support
	lldpV2Xdot3RemPowerMDISupported	MDI power support
	lldpV2Xdot3RemPowerMDIEnabled	MDI power support
	lldpV2Xdot3RemPowerPairControlable	MDI power support
	lldpV2Xdot3RemPowerPairs	PSE power pair
	lldpV2Xdot3RemPowerClass	Power class
lldpV2Xdot3RemMaxFrameSizeTable		
	lldpV2RemTimeMark	(Table index)
	lldpV2RemLocalIfIndex	(Table index)
	lldpV2RemLocalDestMACAddress	(Table index)
	lldpV2RemIndex	(Table index)
	lldpV2Xdot3RemMaxFrameSize	Maximum frame size

5.3 Security considerations for IEEE 802.3 LLDP extension MIB module

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write.[9] 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.

Setting the object, lldpXdot3PortConfigTLVsTxEnable, to incorrect values can result in improper operation of LLDP.

The following readable objects in this MIB module may be considered to be sensitive or vulnerable in some network environments:

a) Objects that are associated with the transmit mode are as follows:
   1) lldpV2Xdot3LocPortAutoNegSupported
   2) lldpV2Xdot3LocPortAutoNegEnabled
   3) lldpV2Xdot3LocPortAutoNegAdvertisedCap
   4) lldpV2Xdot3LocPortOperMauType
   5) lldpV2Xdot3LocPowerPortClass
   6) lldpV2Xdot3LocPowerMDISupported
   7) lldpV2Xdot3LocPowerMDIEnabled
   8) lldpV2Xdot3LocPowerPairControlable
   9) lldpV2Xdot3LocPowerPairs
   10) lldpV2Xdot3LocPowerClass
   11) lldpV2Xdot3LocMaxFrameSize
b) Objects that are associated with the receive mode are as follows:
   1) lldpV2Xdot3RemPortAutoNegSupported
   2) lldpV2Xdot3RemPortAutoNegEnabled
   3) lldpV2Xdot3RemPortAutoNegAdvertisedCap
   4) lldpV2Xdot3RemPortOperMauType
   5) lldpV2Xdot3RemPowerPortClass
   6) lldpV2Xdot3RemPowerMDISupported
   7) lldpV2Xdot3RemPowerMDIEnabled
   8) lldpV2Xdot3RemPowerPairControlable
   9) lldpV2Xdot3RemPowerPairs
   10) lldpV2Xdot3RemPowerClass
   11) lldpV2Xdot3RemMaxFrameSize

This concern applies both to objects that describe the configuration of the local host, as well as for objects that describe information from the remote hosts, acquired via LLDP and displayed by the objects in this MIB module. It is thus also important to control GET and/or NOTIFY access to these objects and possibly to encrypt the values of these objects when sending them over the network via SNMP.

[9] In IETF MIB definitions, the MAX-ACCESS clause defines the type of access that is allowed for particular data elements in the MIB. An explanation of the MAX-ACCESS mapping is given in section 7.3 of IETF STD 58, RFC 2578.

5.4 MIB module definition

In the following MIB definition, should any discrepancy between the DESCRIPTION text and the corresponding definition in 5.2 through 5.3 of this clause occur, the definitions in 5.2 through 5.3 shall take precedence.

An ASCII text version of the MIB definition can be found at the following URL:[10]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C5mib.txt

Two additional modules must be imported when compiling the IEEE 802.3 LLDP extension MIB module, and they can be found at the following URLs:

http://www.ieee802.org/1/files/public/MIBs/LLDP-V2-MIB-200906080000Z.txt

http://www.ieee802.org/1/files/public/MIBs/LLDP-V2-TC-MIB-200906080000Z.txt

[10]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
IEEE8023-DOT3-LLDP-EXT-V2-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    org
        FROM SNMPv2-SMI
    TruthValue
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    ifGeneralInformationGroup
        FROM IF-MIB
    lldpV2Extensions,
    lldpV2LocPortIfIndex,
    lldpV2RemTimeMark,
    lldpV2RemLocalIfIndex,
    lldpV2RemLocalDestMACAddress,
    lldpV2RemIndex,
    lldpV2PortConfigEntry
        FROM LLDP-V2-MIB
 -- http://www.ieee802.org/1/files/public/MIBs/LLDP-V2-MIB-200906080000Z.txt
    LldpV2PowerPortClass
        FROM LLDP-V2-TC-MIB
 -- http://www.ieee802.org/1/files/public/MIBs/LLDP-V2-TC-MIB-200906080000Z.txt
;

ieee8023lldpV2Xdot3MIB MODULE-IDENTITY
    LAST-UPDATED "201102020000Z" -- February 2, 2011
    ORGANIZATION "IEEE 802.3 Working Group"
    CONTACT-INFO
            "WG-URL: http://www.ieee802.org/3/index.html
            WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

            Contact: Howard Frazier
            Postal:  3151 Zanker Road
                     San Jose, CA 95134
                     USA
            Tel:     +1.408.922.8164
            E-mail:  hfrazier@broadcom.com"
    DESCRIPTION
            "The LLDP Management Information Base extension module for
            IEEE 802.3 organizationally defined discovery information.

            In order to assure the uniqueness of the LLDP-MIB,
            ieee8023lldpV2Xdot3MIB is branched from lldpV2Extensions using
            OUI value as the node. An OUI/'company_id' is a 24-bit
            globally unique assigned number referenced by various standards."

    REVISION "201102020000Z" -- February 2, 2011

    DESCRIPTION
            "This revision incorporated changes to the MIB to
            support the use of LLDP with multiple destination MAC
            addresses, and to deprecate the Link Aggregation TLV
            (now moved to the 802.1 extension MIB)."
```

```
   ::= { org ieee(111)
         standards-association-numbers-series-standards(2)
         lan-man-stds(802)ieee802dot3(3) ieee802dot3dot1mibs(1) 5 }

-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
--
-- Organizationally Defined Information Extension - IEEE 802.3
--
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------

lldpV2Xdot3Objects    OBJECT IDENTIFIER ::= { ieee8023lldpV2Xdot3MIB 1 }

-- LLDP IEEE 802.3 extension MIB groups
lldpV2Xdot3Config     OBJECT IDENTIFIER ::= { lldpV2Xdot3Objects 1 }
lldpV2Xdot3LocalData  OBJECT IDENTIFIER ::= { lldpV2Xdot3Objects 2 }
lldpV2Xdot3RemoteData OBJECT IDENTIFIER ::= { lldpV2Xdot3Objects 3 }



-------------------------------------------------------------------------------
-- IEEE 802.3 - Configuration
-------------------------------------------------------------------------------


--
-- Version 2 of lldpV2Xdot3PortConfigTable
-- supports use of multiple destination MAC addresses
--


lldpV2Xdot3PortConfigTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot3PortConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table that controls selection of LLDP TLVs to be transmitted
            on individual ports."
    ::= { lldpV2Xdot3Config 1 }

lldpV2Xdot3PortConfigEntry  OBJECT-TYPE
    SYNTAX      LldpV2Xdot3PortConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "LLDP configuration information that controls the
            transmission of IEEE 802.3 organizationally defined TLVs on
            LLDP transmission capable ports.

            This configuration object augments the lldpV2PortConfigEntry of
            the LLDP-MIB, therefore it is only present along with the port
            configuration defined by the associated lldpV2PortConfigEntry
            entry.

            Each active lldpV2Xdot3PortConfigEntry is restored from non-volatile
            storage (along with the corresponding lldpV2PortConfigEntry)
            after a re-initialization of the management system."
```

```
    AUGMENTS { lldpV2PortConfigEntry }
    ::= { lldpV2Xdot3PortConfigTable 1 }

LldpV2Xdot3PortConfigEntry ::= SEQUENCE {
      lldpV2Xdot3PortConfigTLVsTxEnable  BITS
}

lldpV2Xdot3PortConfigTLVsTxEnable  OBJECT-TYPE
    SYNTAX      BITS {
            macPhyConfigStatus(0),
            powerViaMDI(1),
            unused(2), --avoids re-use of the old link agg bit number
            maxFrameSize(3)
    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The lldpV2Xdot3PortConfigTLVsTxEnable, defined as a bitmap,
            includes the IEEE 802.3 organizationally defined set of LLDP
            TLVs whose transmission is allowed by the local LLDP agent by
            the network management. Each bit in the bitmap corresponds
            to an IEEE 802.3 subtype associated with a specific IEEE
            802.3 optional TLV. The bit 0 is not used since there is
            no corresponding subtype.

            The bit 'macPhyConfigStatus(0)' indicates that the LLDP agent
            should transmit 'MAC/PHY configuration/status TLV'.

            The bit 'powerViaMDI(1)' indicates that the LLDP agent should
            transmit 'Power via MDI TLV'.

            The bit 'unused(2)' is no longer used; this was used for
            the 'Link Aggregation TLV' in the previous version.

            The bit 'maxFrameSize(3)' indicates that the LLDP agent should
            transmit 'Maximum-frame-size TLV'.

            The default value for lldpV2Xdot3PortConfigTLVsTxEnable object
            is an empty set, which means no enumerated values are set.

            The value of this object is restored from non-volatile
            storage after a re-initialization of the management system."
    REFERENCE
            "IEEE Std 802.3 30.12.1.1.1"
    DEFVAL  { { } }
    ::= { lldpV2Xdot3PortConfigEntry  1 }



------------------------------------------------------------------------------
-- IEEE 802.3 - Local Device Information
------------------------------------------------------------------------------

---
--- lldpV2Xdot3LocPortTable: Ethernet Port AutoNeg/Speed/Duplex
---                        Information Table
--- V2 modified to be indexed by ifIndex.
---
```



```
lldpV2Xdot3LocPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot3LocPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port of Ethernet port
            information (as a part of the LLDP 802.3 organizational
            extension) on the local system known to this agent."
    ::= { lldpV2Xdot3LocalData 1 }

lldpV2Xdot3LocPortEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot3LocPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular port component."
    INDEX   { lldpV2LocPortIfIndex }
    ::= { lldpV2Xdot3LocPortTable 1 }

LldpV2Xdot3LocPortEntry ::= SEQUENCE {
         lldpV2Xdot3LocPortAutoNegSupported     TruthValue,
         lldpV2Xdot3LocPortAutoNegEnabled       TruthValue,
         lldpV2Xdot3LocPortAutoNegAdvertisedCap OCTET STRING,
         lldpV2Xdot3LocPortOperMauType          Unsigned32
}

lldpV2Xdot3LocPortAutoNegSupported OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The truth value used to indicate whether the given port
            (associated with the local system) supports Auto-negotiation."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.1"
    ::= { lldpV2Xdot3LocPortEntry 1 }

lldpV2Xdot3LocPortAutoNegEnabled  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The truth value used to indicate whether port
            Auto-negotiation is enabled on the given port associated
            with the local system."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.2"
    ::= { lldpV2Xdot3LocPortEntry 2 }

lldpV2Xdot3LocPortAutoNegAdvertisedCap OBJECT-TYPE
    SYNTAX      OCTET STRING(SIZE(2))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the value (bitmap) of the
            ifMauAutoNegCapAdvertisedBits object (defined in IETF RFC
            3636) which is associated with the given port on the
            local system."
    REFERENCE
```

```
            "IEEE Std 802.3 30.12.2.1.3"
    ::= { lldpV2Xdot3LocPortEntry 3 }

lldpV2Xdot3LocPortOperMauType  OBJECT-TYPE
    SYNTAX      Unsigned32(0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "An integer value that indicates the operational MAU type
            of the given port on the local system.

            This object contains the integer value derived from the
            list position of the corresponding dot3MauType as listed
            in Clause 13 and is equal to the last number in the
            respective dot3MauType OID.

            For example, if the ifMauType object is dot3MauType1000BaseTHD
            which corresponds to {dot3MauType 29}, the numerical value of
            this field is 29. For MAU types not listed in Clause 13,
            the value of this field shall be set to zero."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.4"
    ::= { lldpV2Xdot3LocPortEntry 4 }



---
---
--- lldpV2Xdot3LocPowerTable: Power Ethernet Information Table
--- V2 modified to be indexed by ifIndex.
---
---

lldpV2Xdot3LocPowerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot3LocPowerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port of power ethernet
            information (as a part of the LLDP 802.3 organizational
            extension) on the local system known to this agent."
    ::= { lldpV2Xdot3LocalData 2 }

lldpV2Xdot3LocPowerEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot3LocPowerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular port component."
    INDEX   { lldpV2LocPortIfIndex }
    ::= { lldpV2Xdot3LocPowerTable 1 }

LldpV2Xdot3LocPowerEntry ::= SEQUENCE {
         lldpV2Xdot3LocPowerPortClass           LldpV2PowerPortClass,
         lldpV2Xdot3LocPowerMDISupported        TruthValue,
         lldpV2Xdot3LocPowerMDIEnabled           TruthValue,
         lldpV2Xdot3LocPowerPairControlable     TruthValue,
         lldpV2Xdot3LocPowerPairs               Unsigned32,
         lldpV2Xdot3LocPowerClass               Unsigned32
```

```
}

lldpV2Xdot3LocPowerPortClass  OBJECT-TYPE
    SYNTAX      LldpV2PowerPortClass
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The value that identifies the port Class of the given port
            associated with the local system."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.5"
    ::= { lldpV2Xdot3LocPowerEntry 1 }

lldpV2Xdot3LocPowerMDISupported  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The truth value used to indicate whether the MDI power is
            supported on the given port associated with the local system."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.6"
    ::= { lldpV2Xdot3LocPowerEntry 2 }

lldpV2Xdot3LocPowerMDIEnabled  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The truth value used to identify whether MDI power is
            enabled on the given port associated with the local system."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.7"
    ::= { lldpV2Xdot3LocPowerEntry 3 }

lldpV2Xdot3LocPowerPairControlable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The truth value is derived from the value of
            pethPsePortPowerPairsControlAbility object (defined in IETF
            RFC 3621) and is used to indicate whether the pair selection
            can be controlled on the given port associated with the
            local system."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.8"
    ::= { lldpV2Xdot3LocPowerEntry 4 }

lldpV2Xdot3LocPowerPairs  OBJECT-TYPE
    SYNTAX      Unsigned32(1|2)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the value of the pethPsePortPowerPairs
            object (defined in IETF RFC 3621) which is associated with
            the given port on the local system."
    REFERENCE
```



```
            "IEEE Std 802.3 30.12.2.1.9"
    ::= { lldpV2Xdot3LocPowerEntry 5 }

lldpV2Xdot3LocPowerClass  OBJECT-TYPE
    SYNTAX      Unsigned32(1|2|3|4|5)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the value of the
            pethPsePortPowerClassifications object (defined in IETF
            RFC 3621) which is associated with the given port on the
            local system."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.10"
    ::= { lldpV2Xdot3LocPowerEntry 6 }



---
---
--- lldpV2Xdot3LocMaxFrameSizeTable: Maximum Frame Size information
--- V2 modified to be indexed by ifIndex.
---
---
lldpV2Xdot3LocMaxFrameSizeTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot3LocMaxFrameSizeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port of maximum frame
            size information (as a part of the LLDP 802.3 organizational
            extension) on the local system known to this agent."
    ::= { lldpV2Xdot3LocalData 3 }

lldpV2Xdot3LocMaxFrameSizeEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot3LocMaxFrameSizeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Maximum Frame Size information about a particular port
            component."
    INDEX   { lldpV2LocPortIfIndex }
    ::= { lldpV2Xdot3LocMaxFrameSizeTable 1 }

LldpV2Xdot3LocMaxFrameSizeEntry ::= SEQUENCE {
        lldpV2Xdot3LocMaxFrameSize             Unsigned32
}

lldpV2Xdot3LocMaxFrameSize OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "An integer value indicating the maximum supported frame
             size in octets on the given port of the local system."
    REFERENCE
            "IEEE Std 802.3 30.12.2.1.13"
    ::= { lldpV2Xdot3LocMaxFrameSizeEntry 1 }
```

```
-------------------------------------------------------------------------------
-- IEEE 802.3 - Remote Devices Information
-------------------------------------------------------------------------------

---
---
--- lldpV2Xdot3RemPortTable: Ethernet Information Table
--- V2 modified to be indexed by ifIndex and destination MAC address.
---
---

lldpV2Xdot3RemPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot3RemPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains Ethernet port information (as a part
            of the LLDP 802.3 organizational extension) of the remote
            system."
    ::= { lldpV2Xdot3RemoteData 1 }

lldpV2Xdot3RemPortEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot3RemPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular physical network connection."
    INDEX   { lldpV2RemTimeMark,
              lldpV2RemLocalIfIndex,
              lldpV2RemLocalDestMACAddress,
              lldpV2RemIndex }
    ::= { lldpV2Xdot3RemPortTable 1 }

LldpV2Xdot3RemPortEntry ::= SEQUENCE {
              lldpV2Xdot3RemPortAutoNegSupported     TruthValue,
              lldpV2Xdot3RemPortAutoNegEnabled       TruthValue,
              lldpV2Xdot3RemPortAutoNegAdvertisedCap OCTET STRING,
              lldpV2Xdot3RemPortOperMauType          Unsigned32
}


lldpV2Xdot3RemPortAutoNegSupported OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The truth value used to indicate whether the given port
            (associated with remote system) supports Auto-negotiation."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.1"
    ::= { lldpV2Xdot3RemPortEntry 1 }

lldpV2Xdot3RemPortAutoNegEnabled  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```



```
            "The truth value used to indicate whether port
            Auto-negotiation is enabled on the given port associated
            with the remote system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.2"
    ::= { lldpV2Xdot3RemPortEntry 2 }

lldpV2Xdot3RemPortAutoNegAdvertisedCap  OBJECT-TYPE
    SYNTAX      OCTET STRING(SIZE(2))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the value (bitmap) of the
            ifMauAutoNegCapAdvertisedBits object (defined in IETF RFC
            3636) which is associated with the given port on the
            remote system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.3"
    ::= { lldpV2Xdot3RemPortEntry 3 }

lldpV2Xdot3RemPortOperMauType OBJECT-TYPE
    SYNTAX      Unsigned32(0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "An integer value that indicates the operational MAU type
            of the sending device.

            This object contains the integer value derived from the
            list position of the corresponding dot3MauType as listed in
            in Clause 13 and is equal to the last number in
            the respective dot3MauType OID.

            For example, if the ifMauType object is dot3MauType1000BaseTHD
            which corresponds to {dot3MauType 29}, the numerical value of
            this field is 29. For MAU types not listed in Clause 13,
            the value of this field shall be set to zero."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.4"
    ::= { lldpV2Xdot3RemPortEntry 4 }


---
---
--- lldpV2Xdot3RemPowerTable: Power Ethernet Information Table
--- V2 modified to be indexed by ifIndex and destination MAC address.
---
---

lldpV2Xdot3RemPowerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot3RemPowerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains Ethernet power information (as a part
            of the LLDP 802.3 organizational extension) of the remote
            system."
    ::= { lldpV2Xdot3RemoteData 2 }
```

```
lldpV2Xdot3RemPowerEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot3RemPowerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about a particular physical network connection."
    INDEX   { lldpV2RemTimeMark,
              lldpV2RemLocalIfIndex,
              lldpV2RemLocalDestMACAddress,
              lldpV2RemIndex }
    ::= { lldpV2Xdot3RemPowerTable 1 }

LldpV2Xdot3RemPowerEntry ::= SEQUENCE {
              lldpV2Xdot3RemPowerPortClass           LldpV2PowerPortClass,
              lldpV2Xdot3RemPowerMDISupported        TruthValue,
              lldpV2Xdot3RemPowerMDIEnabled          TruthValue,
              lldpV2Xdot3RemPowerPairControlable     TruthValue,
              lldpV2Xdot3RemPowerPairs               Unsigned32,
              lldpV2Xdot3RemPowerClass               Unsigned32
}


lldpV2Xdot3RemPowerPortClass  OBJECT-TYPE
    SYNTAX      LldpV2PowerPortClass
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The value that identifies the port Class of the given port
            associated with the remote system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.5"
    ::= { lldpV2Xdot3RemPowerEntry 1 }

lldpV2Xdot3RemPowerMDISupported  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The truth value used to indicate whether the MDI power
            is supported on the given port associated with the remote
            system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.6"
    ::= { lldpV2Xdot3RemPowerEntry 2 }

lldpV2Xdot3RemPowerMDIEnabled  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The truth value used to identify whether MDI power is
            enabled on the given port associated with the remote system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.7"
    ::= { lldpV2Xdot3RemPowerEntry 3 }

lldpV2Xdot3RemPowerPairControlable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
            "The truth value is derived from the value of
            pethPsePortPowerPairsControlAbility object (defined in IETF
            RFC 3621) and is used to indicate whether the pair selection
            can be controlled on the given port associated with the
            remote system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.8"
    ::= { lldpV2Xdot3RemPowerEntry 4 }

lldpV2Xdot3RemPowerPairs  OBJECT-TYPE
    SYNTAX      Unsigned32(1|2)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the value of the pethPsePortPowerPairs
            object (defined in IETF RFC 3621) which is associated with
            the given port on the remote system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.9"
    ::= { lldpV2Xdot3RemPowerEntry 5 }

lldpV2Xdot3RemPowerClass  OBJECT-TYPE
    SYNTAX      Unsigned32(1|2|3|4|5)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the value of the
            pethPsePortPowerClassifications object (defined in IETF
            RFC 3621) which is associated with the given port on the
            remote system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.10"
    ::= { lldpV2Xdot3RemPowerEntry 6 }



---
--- lldpV2Xdot3RemMaxFrameSizeTable: Maximum Frame Size information
--- V2 modified to be indexed by ifIndex and destination MAC address.
---
---

lldpV2Xdot3RemMaxFrameSizeTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot3RemMaxFrameSizeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port/destination
            address pair of maximum frame
            size information (as a part of the LLDP IEEE 802.3
            organizational extension) of the remote system."
    ::= { lldpV2Xdot3RemoteData 3 }

lldpV2Xdot3RemMaxFrameSizeEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot3RemMaxFrameSizeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
            "Maximum Frame Size information about a particular port
            component."
    INDEX   { lldpV2RemTimeMark,
              lldpV2RemLocalIfIndex,
              lldpV2RemLocalDestMACAddress,
              lldpV2RemIndex }
    ::= { lldpV2Xdot3RemMaxFrameSizeTable 1 }

LldpV2Xdot3RemMaxFrameSizeEntry ::= SEQUENCE {
              lldpV2Xdot3RemMaxFrameSize   Unsigned32
}

lldpV2Xdot3RemMaxFrameSize OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "An integer value indicating the maximum supported frame
             size in octets on the port component associated with the
             remote system."
    REFERENCE
            "IEEE Std 802.3 30.12.3.1.13"
    ::= { lldpV2Xdot3RemMaxFrameSizeEntry 1 }


------------------------------------------------------------------------------
-- Conformance statements
------------------------------------------------------------------------------
lldpV2Xdot3Conformance OBJECT IDENTIFIER ::= { ieee8023lldpV2Xdot3MIB 2 }
lldpV2Xdot3Compliances OBJECT IDENTIFIER ::= { lldpV2Xdot3Conformance 1 }
lldpV2Xdot3Groups      OBJECT IDENTIFIER ::= { lldpV2Xdot3Conformance 2 }

-- Compliance statements


lldpV2Xdot3TxRxCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "A compliance statement for SNMP entities that implement
            the LLDP 802.3 organizational extension MIB.

            This group is mandatory for all agents that implement the
            LLDP 802.3 organizational extension in TX and/or RX mode.

            This version defines compliance requirements for
            V2 of the LLDP MIB."
    MODULE  -- this module
        MANDATORY-GROUPS { lldpV2Xdot3ConfigGroup,
                           ifGeneralInformationGroup
        }
    ::= { lldpV2Xdot3Compliances 1 }

lldpV2Xdot3TxCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "The compliance statement for SNMP entities which implement
            the LLDP 802.3 organizational extension MIB.
```



```
            This group is mandatory for agents which implement the
            LLDP 802.3 organizational extension in the TX mode.

            This version defines compliance requirements for
            V2 of the LLDP MIB."
    MODULE  -- this module
        MANDATORY-GROUPS { lldpV2Xdot3LocSysGroup }
    ::= { lldpV2Xdot3Compliances 2 }

lldpV2Xdot3RxCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "The compliance statement for SNMP entities which implement
            the LLDP 802.3 organizational extension MIB.

            This group is mandatory for agents which implement the
            LLDP 802.3 organizational extension in the RX mode.

            This version defines compliance requirements for
            V2 of the LLDP MIB."
    MODULE  -- this module
        MANDATORY-GROUPS { lldpV2Xdot3RemSysGroup }
    ::= { lldpV2Xdot3Compliances 3 }


-- MIB groupings


lldpV2Xdot3ConfigGroup    OBJECT-GROUP
    OBJECTS {
        lldpV2Xdot3PortConfigTLVsTxEnable
    }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to configure the
            LLDP 802.3 organizational extension implementation behavior. "
    ::= { lldpV2Xdot3Groups 1 }

lldpV2Xdot3LocSysGroup  OBJECT-GROUP
    OBJECTS {
        lldpV2Xdot3LocPortAutoNegSupported,
        lldpV2Xdot3LocPortAutoNegEnabled,
        lldpV2Xdot3LocPortAutoNegAdvertisedCap,
        lldpV2Xdot3LocPortOperMauType,
        lldpV2Xdot3LocPowerPortClass,
        lldpV2Xdot3LocPowerMDISupported,
        lldpV2Xdot3LocPowerMDIEnabled,
        lldpV2Xdot3LocPowerPairControlable,
        lldpV2Xdot3LocPowerPairs,
        lldpV2Xdot3LocPowerClass,
        lldpV2Xdot3LocMaxFrameSize
    }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to represent LLDP
            802.3 organizational extension Local Device Information."
    ::= { lldpV2Xdot3Groups 2 }

lldpV2Xdot3RemSysGroup  OBJECT-GROUP
```

```
    OBJECTS {
        lldpV2Xdot3RemPortAutoNegSupported,
        lldpV2Xdot3RemPortAutoNegEnabled,
        lldpV2Xdot3RemPortAutoNegAdvertisedCap,
        lldpV2Xdot3RemPortOperMauType,
        lldpV2Xdot3RemPowerPortClass,
        lldpV2Xdot3RemPowerMDISupported,
        lldpV2Xdot3RemPowerMDIEnabled,
        lldpV2Xdot3RemPowerPairControlable,
        lldpV2Xdot3RemPowerPairs,
        lldpV2Xdot3RemPowerClass,
        lldpV2Xdot3RemMaxFrameSize
    }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to represent LLDP
            802.3 organizational extension Local Device Information."
    ::= { lldpV2Xdot3Groups 3 }

END
```

6. Ethernet operations, administration, and maintenance (OAM) MIB module

6.1 Introduction

The IEEE 802.3ah Ethernet in the First Mile (EFM) Task Force added management capabilities to Ethernet-like interfaces to provide some basic operations, administration and maintenance (OAM) functions. The defined functionality includes discovery, error signaling, loopback, and link monitoring.

This clause defines a MIB module for use with SNMP to manage these Ethernet-like interface capabilities.

6.2 Overview

Ethernet OAM is composed of a core set of functions and a set of optional functional groups as described in IEEE Std 802.3 Clause 57. The core functions include discovery operations (determining if the other end of the link is OAM capable and what OAM functions it supports), state machine implementation, and some critical event flows. The optional functional groups are for (a) link events, (b) remote loopback, and (c) variable retrieval and response. Each optional functional group is controlled by a separate MIB table(s).

Ethernet OAM is complementary with SNMP management in that it provides some basic management functions at layer 2, rather than using layer 3 and above as required by SNMP over an IP infrastructure. Ethernet OAM provides single-hop functionality in that it works only between two directly-connected Ethernet stations. SNMP can be used to manage the Ethernet OAM interactions of one Ethernet station with another.

Ethernet OAM has three functional objectives, which are detailed in the next three subclauses. The definition of a basic Ethernet OAM protocol data unit is given in 6.2.4.

6.2.1 Remote fault indication

Remote fault indication provides a mechanism for one end of an Ethernet link to signal the other end that the receive path is non-operational. Some Ethernet physical layers offer mechanisms to signal this condition at the physical layer. Ethernet OAM added a mechanism so that some Ethernet physical layers can operate in unidirectional mode, allowing frames to be transmitted in one direction even when the other direction is non-operational. Traditionally, Ethernet PHYs do not allow frame transmission in one direction if the other direction is not operational. Using this mode, Ethernet OAM allows frame-based signaling of remote fault conditions while still not allowing higher-layer applications to be aware of the unidirectional capability. This clause includes mechanisms for capturing that fault information and reflecting such information in objects and notifications within the SNMP management framework.

6.2.2 Link monitoring

Ethernet OAM includes event signaling capability so that one end of an Ethernet link can indicate the occurrence of certain important events to the other end of the link. This happens via layer 2 protocols. This clause defines methods for incorporating the occurrence of these layer 2 events, both at the local end and far end of the link, into the SNMP management framework.

Ethernet OAM also includes mechanisms for one Ethernet station to query another directly connected Ethernet station about the status of its Ethernet interface variables and status. This clause does not include mechanisms for controlling how one Ethernet endpoint may use this functionality to query the status or statistics of a peer Ethernet entity.

6.2.3 Remote loopback

Remote loopback is a link state where the peer Ethernet entity echoes every received packet (without modifications) back onto the link. Remote loopback is intrusive in that the other end of the link is not forwarding traffic from higher layers out over the link. This clause defines objects controlling loopback operation and reading the status of the loopback state.

6.2.4 Ethernet OAM protocol data units

An Ethernet OAM protocol data unit is a valid Ethernet frame with a destination Media Access Control (MAC) address equal to the reserved MAC address for Slow Protocols (see Annex 57A of IEEE Std 802.3), a lengthOrType field equal to the reserved type for Slow Protocols, and a Slow Protocols subtype equal to that of the subtype reserved for Ethernet OAM.

OAMPDU is used throughout this clause as an abbreviation for Ethernet OAM protocol data unit. OAMPDUs are the mechanism by which two directly connected Ethernet interfaces exchange OAM information.

6.3 Relation to other MIB modules

The definitions presented here are based on Clause 30 and Clause 57 of IEEE Std 802.3. Note that these clauses describe many of these variables and their effects on the MAC sublayer. In some cases, there is a one-to-one relationship between an object in this clause and an object in the Clause 30 MIB. In other cases, the objects of this clause reflect a more complex entity and are reflected by more than one object in the Clause 30 MIB.

The objects defined in this clause manage OAM functionality introduced as part of the 802.3ah project. These objects do not overlap with the Interfaces Group MIB module defined in IETF RFC 2863, the Ethernet-like interface MIB module defined in Clause 10, or any other MIB module currently used to manage various aspects of an Ethernet interface. The objects defined here are defined for Ethernet-like interfaces only and use the same ifIndex as the associated Ethernet interface. Ethernet OAM can be implemented on any Ethernet-like interface.

6.3.1 Relation to other EFM MIB modules

The Ethernet OAM functionality and MIB module is independent of the other functionality and MIB modules derived from IEEE Std 802.3 for copper and EPON. Ethernet OAM may be implemented (or not) on the new EFM interface types, just as it can on any other Ethernet interface.

6.3.2 Mapping of IEEE 802.3 managed objects

Table 6–1 contains the mapping between managed objects defined in IEEE Std 802.3 Clause 30, and managed objects defined in this clause.

All IEEE 802.3 OAM managed objects are reflected in this MIB module.

Table 6–1—Mapping between IEEE 802.3 managed objects and SNMP objects

IEEE 802.3 managed object	Corresponding SNMP object
oOAM	
.aOAMID	IF-MIB ifIndex
.aOAMAdminState	dot3OamAdminState
.aOAMMode	dot3OamMode
.aOAMDiscoveryState	dot3OamOperStatus
.aOAMRemoteMACAddress	dot3OamPeerMacAddress
.aOAMLocalConfiguration	dot3OamFunctionsSupported
.aOAMRemoteConfiguration	dot3OamPeerFunctionsSupported, dot3OamPeerMode
.aOAMLocalPDUConfiguration	dot3OamMaxOamPduSize
.aOAMRemotePDUConfiguration	dot3OamPeerMaxOamPduSize
.aOAMLocalFlagsField	dot3OamOperStatus, dot3OamEventLogEntry
.aOAMRemoteFlagsField	dot3OamOperStatus, dot3OamEventLogEntry
.aOAMLocalRevision	dot3OamConfigRevision
.aOAMRemoteRevision	dot3OamPeerConfigRevision
.aOAMLocalState	dot3OamLoopbackStatus
.aOAMRemoteState	dot3OamLoopbackStatus
.aOAMRemoteVendorOUI	dot3OamPeerVendorOui
.aOAMRemoteVendorSpecificInfo	dot3OamPeerVendorInfo
.aOAMUnsupportedCodesTx	dot3OamUnsupportedCodesTx
.aOAMUnsupportedCodesRx	dot3OamUnsupportedCodesRx
.aOAMInformationTx	dot3OamInformationTx
.aOAMInformationRx	dot3OamInformationRx
.aOAMUniqueEventNotificationTx	dot3OamUniqueEventNotificationTx
.aOAMUniqueEventNotificationRx	dot3OamUniqueEventNotificationRx
.aOAMDuplicateEventNotificationTx	dot3OamDuplicateEventNotificationTx
.aOAMDuplicateEventNotificationRx	dot3OamDuplicateEventNotificationRx
.aOAMLoopbackControlTx	dot3OamLoopbackControlTx

Table 6–1—Mapping between IEEE 802.3 managed objects and SNMP objects *(continued)*

IEEE 802.3 managed object	Corresponding SNMP object
.aOAMLoopbackControlRx	dot3OamLoopbackControlRx
.aOAMVariableRequestTx	dot3OamVariableRequestTx
.aOAMVariableRequestRx	dot3OamVariableRequestRx
.aOAMVariableResponseTx	dot3OamVariableResponseTx
.aOAMVariableResponseRx	dot3OamVariableResponseRx
.aOAMOrganizationSpecificTx	dot3OamOrgSpecificTx
.aOAMOrganizationSpecificRx	dot3OamOrgSpecificTx
.aOAMLocalErrSymPeriodConfig	dot3OamErrSymPeriodWindow, dot3OamErrSymPeriodThreshold
.aOAMLocalErrSymPeriodEvent	dot3OamEventLogEntry
.aOAMLocalErrFrameConfig	dot3OamErrFrameWindow, dot3OamErrFrameThreshold
.aOAMLocalErrFrameEvent	dot3OamEventLogEntry
.aOAMLocalErrFramePeriodConfig	dot3OamErrFramePeriodWindow, dot3OamErrFramePeriodThreshold
.aOAMLocalErrFramePeriodEvent	dot3OamEventLogEntry
.aOAMLocalErrFrameSecsSummaryConfig	dot3OamErrFrameSecsSummaryWindow, dot3OamErrFrameSecssummaryThreshold
.aOAMLocalErrFrameSecsSummaryEvent	dot3OamEventLogEntry
.aOAMRemoteErrSymPeriodEvent	dot3OamEventLogEntry
.aOAMRemoteErrFrameEvent	dot3OamEventLogEntry
.aOAMRemoteErrFramePeriodEvent	dot3OamEventLogEntry
.aOAMRemoteErrFrameSecsSummaryEvent	dot3OamEventLogEntry
.aFramesLostDueToOAmError	dot3OamFramesLostDueToOam
.acOAMAdminControl	dot3OamAdminState

6.4 MIB structure

The Ethernet OAM MIB objects of this clause focus on the OAM capabilities introduced in IEEE Std 802.3. The MIB objects are partitioned into six different MIB groups.

The dot3OamTable group manages the primary OAM objects of the Ethernet interface. This group controls the state and status of OAM as well as the mode in which it operates. The dot3OamPeerTable maintains the current information on the status and configuration of the peer OAM entity on the Ethernet interface. Managed information includes the capabilities and function available on the peer OAM entity.

The dot3OamLoopbackTable manages the loopback function introduced in IEEE Std 802.3. This table controls enabling and disabling loopback, as well as indicating the loopback status of Ethernet OAM on this interface.

The dot3OamStatsTable maintains statistics on the number and type of Ethernet OAM frames being transmitted and received on the Ethernet interface.

The dot3OamEventConfigTable defines the objects for managing the event notification capability available in Ethernet OAM. With Ethernet OAM, one device may send notifications to its peer devices whenever an important event happens on the local device. This table provides management of which events result in notifications via Ethernet OAM notifications and/or via SNMP notifications.

The dot3OamEventLogTable manages the current status of local and remote events detected via Ethernet OAM. This table is updated whenever local events are detected by Ethernet OAM or whenever Ethernet OAM Event Notifications are received from the peer OAM entity.

There are two notifications defined to report Ethernet OAM events (one for threshold crossing events, one for non-threshold crossing events). Both notifications are contained within the same conformance group.

6.5 Security considerations for Ethernet operations, administration, and maintenance (OAM) MIB module

The readable objects in this module can provide information about network traffic, and therefore may be considered sensitive. In particular, OAM provides mechanisms for reading the IEEE Std 802.3 Clause 30 MIB attributes from a link partner via a specialized layer two protocol. Unlike SNMP, IEEE 802.3 OAM does not include encryption or authentication mechanisms. It should not be used in environments where this interface information is considered sensitive, and where the facility terminations are unprotected. By default, OAM is disabled on Ethernet-like interfaces and is therefore not a risk.

IEEE 802.3 OAM is designed to support deployment in access and enterprise networks. In access networks, one end of a link is the CO-side, and the other is the CPE-side, and the facilities are often protected in wiring cages or closets. In such deployments, it is often the case that the CO-side is protected from access from the CPE-side. Within IEEE 802.3 OAM, this protection from remote access is accomplished by configuring the CPE-side in passive mode using the dot3OamMode attribute. This prevents the CPE from accessing functions and information at the CO-side of the connection. In enterprise networks, read-only interface information is often considered non-sensitive.

The frequency of OAM PDUs on an Ethernet interface does not adversely affect data traffic, as OAM is a slow protocol with very limited bandwidth potential, and it is not required for normal link operation. Although there are a number of objects in this module with read-write or read-create MAX-ACCESS, they have limited effects on user data.

The loopback capability of OAM can have potentially disruptive effects; when remote loopback is enabled, the remote station automatically transmits all received traffic back to the local station except for OAM traffic. This completely disrupts all higher layer protocols such as bridging, IP, and SNMP. Therefore, an attribute (dot3OamLoopbackIgnoreRx) was introduced to control whether the local station processes or ignores received loopback commands.

The administrative state and mode are also read-write objects. Disabling OAM can interrupt management activities between peer devices, potentially causing serious problems. Setting the dot3OamMode to an undesired value can allow access to Ethernet monitoring, events, and functions that may not be acceptable in a particular deployment scenario. In addition to loopback functionality, Ethernet interface statistics and events can be accessed via the OAM protocol, which may not be desired in some circumstances.

OAM event configuration also contains read-write objects. These objects control whether events are sent, and at what thresholds. Note that the frequency of event communication is limited by the frequency limits of Slow Protocols on Ethernet interfaces. Also, the information available via OAM events is also available via OAM Variable Requests. Access to this information via either OAM events or Variable Requests is controlled by the dot3OamAdminState and dot3OamMode objects. As mentioned previously, inadequate protection of these variables can result in access to link information and functions.

6.6 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL:[11]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C6mib.txt

[11]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.


```
IEEE8023-DOT3-OAM-MIB DEFINITIONS ::= BEGIN
    IMPORTS
      MODULE-IDENTITY, OBJECT-TYPE, Counter32, Unsigned32,
        Integer32, NOTIFICATION-TYPE, org
        FROM SNMPv2-SMI
        -- from [RFC2578]
      TEXTUAL-CONVENTION, MacAddress, TimeStamp, TruthValue
        FROM SNMPv2-TC
        -- from [RFC2579]
      CounterBasedGauge64
        FROM HCNUM-TC
        -- from [RFC2856]
      ifIndex
        FROM IF-MIB
        -- from [RFC2863]
      MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF;
        -- from [RFC2580]
      ieee8023Dot3OamMIB MODULE-IDENTITY
        LAST-UPDATED "201102020000Z" -- February 2, 2011
        ORGANIZATION
          "IEEE 802.3 working group"
        CONTACT-INFO
            "WG-URL: http://www.ieee802.org/3/index.html
            WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

            Contact: Howard Frazier
            Postal:  3151 Zanker Road
                     San Jose, CA 95134
                     USA
            Tel:     +1.408.922.8164
            E-mail:  hfrazier@broadcom.com"
        DESCRIPTION
          "The MIB module for managing the new Ethernet OAM features
          introduced by the Ethernet in the First Mile Task Force (IEEE
          802.3ah). The functionality presented here is based on IEEE
          Std 802.3ah, released in October, 2004, which was prepared as
          an addendum to IEEE Std 802.3. Since then, IEEE Std 802.3ah
          has been merged into the base IEEE 802.3 standard.

          In particular, this MIB focuses on the new OAM functions
          introduced in Clause 57 of IEEE Std 802.3. The OAM functionality
          of Clause 57 is controlled by new management attributes
          introduced in Clause 30 of IEEE Std 802.3. The OAM functions are
          not specific to any particular Ethernet physical layer, and
          can be generically applied to any Ethernet interface.

          An Ethernet OAM protocol data unit is a valid Ethernet frame
          with a destination MAC address equal to the reserved MAC
          address for Slow Protocols (See 57A of IEEE Std 802.3), a
          lengthOrType field equal to the reserved type for Slow
          Protocols, and a Slow Protocols subtype equal to that of the
          subtype reserved for Ethernet OAM. OAMPDU is used throughout
          this document as an abbreviation for Ethernet OAM protocol
          data unit."

        REVISION    "201102020000Z" -- February 2, 2011
        DESCRIPTION "Initial version, based on an earlier version in RFC 4878."
```

```
   ::= { org ieee(111)
         standards-association-numbers-series-standards(2)
         lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 6 }

--
-- Sections of the Ethernet OAM MIB
--
   dot3OamNotifications OBJECT IDENTIFIER ::= { ieee8023Dot3OamMIB 0 }
   dot3OamObjects       OBJECT IDENTIFIER ::= { ieee8023Dot3OamMIB 1 }
   dot3OamConformance   OBJECT IDENTIFIER ::= { ieee8023Dot3OamMIB 2 }

--
-- Textual conventions for the OAM MIB
--
EightOTwoOui ::= TEXTUAL-CONVENTION
  DISPLAY-HINT "3x:"
  STATUS       current
  DESCRIPTION
    "24-bit Organizationally Unique Identifier. Information on
    OUIs can be found in IEEE 802-2001 [802-2001], Clause 9."
  SYNTAX       OCTET STRING(SIZE(3))

-- ***************************************************************
--
-- Ethernet OAM Control group
--

dot3OamTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF Dot3OamEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table contains the primary controls and status for the
    OAM capabilities of an Ethernet-like interface. There will be
    one row in this table for each Ethernet-like interface in the
    system that supports the OAM functions defined in IEEE Std 802.3."
  ::= { dot3OamObjects 1 }

dot3OamEntry OBJECT-TYPE
  SYNTAX     Dot3OamEntry
  MAX-ACCESS not-accessible
  STATUS     current
  DESCRIPTION
    "An entry in the table that contains information on the
    Ethernet OAM function for a single Ethernet like interface.
    Entries in the table are created automatically for each
    interface supporting Ethernet OAM. The status of the row
    entry can be determined from dot3OamOperStatus.

    A dot3OamEntry is indexed in the dot3OamTable by the ifIndex
    object of the Interfaces Group MIB.
    "
  INDEX       { ifIndex }
  ::= { dot3OamTable 1 }

Dot3OamEntry ::=
  SEQUENCE {
    dot3OamAdminState                  INTEGER,
    dot3OamOperStatus                  INTEGER,
```



```
    dot3OamMode                     INTEGER,
    dot3OamMaxOamPduSize            Unsigned32,
    dot3OamConfigRevision           Unsigned32,
    dot3OamFunctionsSupported       BITS
  }

dot3OamAdminState OBJECT-TYPE
  SYNTAX      INTEGER {
                enabled(1),
                disabled(2)
              }
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "This object is used to provision the default administrative
    OAM mode for this interface. This object represents the
    desired state of OAM for this interface.

    The dot3OamAdminState always starts in the disabled(2) state
    until an explicit management action or configuration
    information retained by the system causes a transition to the
    enabled(1) state. When enabled(1), Ethernet OAM will attempt
    to operate over this interface.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.2"
  ::= { dot3OamEntry 1 }

dot3OamOperStatus OBJECT-TYPE
  SYNTAX      INTEGER {
                disabled(1),
                linkFault(2),
                passiveWait(3),
                activeSendLocal(4),
                sendLocalAndRemote(5),
                sendLocalAndRemoteOk(6),
                oamPeeringLocallyRejected(7),
                oamPeeringRemotelyRejected(8),
                operational(9),
                nonOperHalfDuplex(10)
              }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "At initialization and failure conditions, two OAM entities on
    the same full-duplex Ethernet link begin a discovery phase to
    determine what OAM capabilities may be used on that link. The
    progress of this initialization is controlled by the OAM
    sublayer.

    This value is always disabled(1) if OAM is disabled on this
    interface via the dot3OamAdminState.

    If the link has detected a fault and is transmitting OAMPDUs
    with a link fault indication, the value is linkFault(2).
    Also, if the interface is not operational (ifOperStatus is
    not up(1)), linkFault(2) is returned. Note that the object
    ifOperStatus may not be up(1) as a result of link failure or
    administrative action (ifAdminState being down(2) or
    testing(3)).
```



```
    The passiveWait(3) state is returned only by OAM entities in
    passive mode (dot3OamMode) and reflects the state in which the
    OAM entity is waiting to see if the peer device is OAM
    capable. The activeSendLocal(4) value is used by active mode
    devices (dot3OamMode) and reflects the OAM entity actively
    trying to discover whether the peer has OAM capability but has
    not yet made that determination.

    The state sendLocalAndRemote(5) reflects that the local OAM
    entity has discovered the peer but has not yet accepted or
    rejected the configuration of the peer. The local device can,
    for whatever reason, decide that the peer device is
    unacceptable and decline OAM peering. If the local OAM entity
    rejects the peer OAM entity, the state becomes
    oamPeeringLocallyRejected(7). If the OAM peering is allowed
    by the local device, the state moves to
    sendLocalAndRemoteOk(6). Note that both the
    sendLocalAndRemote(5) and oamPeeringLocallyRejected(7) states
    fall within the state SEND_LOCAL_REMOTE of the Discovery state
    diagram (see IEEE Std 802.3, Figure 57-5), with the difference being
    whether the local OAM client has actively rejected the peering
    or has just not indicated any decision yet. Whether a peering
    decision has been made is indicated via the local flags field
    in the OAMPDU (reflected in the aOAMLocalFlagsField of
    30.3.6.1.10).

    If the remote OAM entity rejects the peering, the state
    becomes oamPeeringRemotelyRejected(8). Note that both the
    sendLocalAndRemoteOk(6) and oamPeeringRemotelyRejected(8)
    states fall within the state SEND_LOCAL_REMOTE_OK of the
    Discovery state diagram (see IEEE Std 802.3, Figure 57-5), with the
    difference being whether the remote OAM client has rejected
    the peering or has just not yet decided. This is indicated
    via the remote flags field in the OAMPDU (reflected in the
    aOAMRemoteFlagsField of 30.3.6.1.11).

    When the local OAM entity learns that both it and the remote
    OAM entity have accepted the peering, the state moves to
    operational(9) corresponding to the SEND_ANY state of the
    Discovery state diagram (see IEEE Std 802.3, Figure 57-5).

    Since Ethernet OAM functions are not designed to work
    completely over half-duplex interfaces, the value
    nonOperHalfDuplex(10) is returned whenever Ethernet OAM is
    enabled (dot3OamAdminState is enabled(1)), but the interface
    is in half-duplex operation.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.4, 30.3.6.1.10, 30.3.6.1.11"
  ::= { dot3OamEntry 2 }

dot3OamMode OBJECT-TYPE
  SYNTAX      INTEGER {
                passive(1),
                active(2)
              }
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
```

```
    "This object configures the mode of OAM operation for this
    Ethernet-like interface. OAM on Ethernet interfaces may be in
    'active' mode or 'passive' mode. These two modes differ in
    that active mode provides additional capabilities to initiate
    monitoring activities with the remote OAM peer entity, while
    passive mode generally waits for the peer to initiate OAM
    actions with it. As an example, an active OAM entity can put
    the remote OAM entity in a loopback state, where a passive OAM
    entity cannot.

    The default value of dot3OamMode is dependent on the type of
    system on which this Ethernet-like interface resides. The
    default value should be 'active(2)' unless it is known that
    this system should take on a subservient role to the other
    device connected over this interface.

    Changing this value results in incrementing the configuration
    revision field of locally generated OAMPDUs (30.3.6.1.12) and
    potentially re-doing the OAM discovery process if the
    dot3OamOperStatus was already operational(9).
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.3"
  ::= { dot3OamEntry 3 }

dot3OamMaxOamPduSize OBJECT-TYPE
  SYNTAX      Unsigned32 (64..1518)
  UNITS       "octets"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The largest OAMPDU that the OAM entity supports. OAM
    entities exchange maximum OAMPDU sizes and negotiate to use
    the smaller of the two maximum OAMPDU sizes between the peers.
    This value is determined by the local implementation.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.8"
  ::= { dot3OamEntry 4 }

dot3OamConfigRevision OBJECT-TYPE
  SYNTAX      Unsigned32(0..65535)
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The configuration revision of the OAM entity as reflected in
    the latest OAMPDU sent by the OAM entity. The config revision
    is used by OAM entities to indicate that configuration changes
    have occurred, which might require the peer OAM entity to
    re-evaluate whether OAM peering is allowed.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.12"
  ::= { dot3OamEntry 5 }

dot3OamFunctionsSupported OBJECT-TYPE
  SYNTAX      BITS {
                unidirectionalSupport (0),
                loopbackSupport(1),
                eventSupport(2),
                variableSupport(3)
              }
```



```
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The OAM functions supported on this Ethernet-like interface.
    OAM consists of separate functional sets beyond the basic
    discovery process that is always required. These functional
    groups can be supported independently by any implementation.
    These values are communicated to the peer via the local
    configuration field of Information OAMPDUs.

    Setting 'unidirectionalSupport(0)' indicates that the OAM
    entity supports the transmission of OAMPDUs on links that are
    operating in unidirectional mode (traffic flowing in one
    direction only). Setting 'loopbackSupport(1)' indicates that
    the OAM entity can initiate and respond to loopback commands.
    Setting 'eventSupport(2)' indicates that the OAM entity can
    send and receive Event Notification OAMPDUs. Setting
    'variableSupport(3)' indicates that the OAM entity can send
    and receive Variable Request and Response OAMPDUs.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.6"
  ::= { dot3OamEntry 6 }

-- *****************************************************************
--
-- Ethernet OAM Peer group
--

dot3OamPeerTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF Dot3OamPeerEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table contains information about the OAM peer for a
    particular Ethernet-like interface. OAM entities communicate
    with a single OAM peer entity on Ethernet links on which OAM
    is enabled and operating properly. There is one entry in this
    table for each entry in the dot3OamTable for which information
    on the peer OAM entity is available.
    "
  ::= { dot3OamObjects 2 }

dot3OamPeerEntry OBJECT-TYPE
  SYNTAX      Dot3OamPeerEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An entry in the table containing information on the peer OAM
    entity for a single Ethernet-like interface.

    Note that there is at most one OAM peer for each Ethernet-like
    interface. Entries are automatically created when information
    about the OAM peer entity becomes available, and automatically
    deleted when the OAM peer entity is no longer in
    communication. Peer information is not available when
    dot3OamOperStatus is disabled(1), linkFault(2),
    passiveWait(3), activeSendLocal(4), or nonOperHalfDuplex(10).
    "
  INDEX       { ifIndex }
```



```
  ::= { dot3OamPeerTable 1 }

Dot3OamPeerEntry ::=
  SEQUENCE {
    dot3OamPeerMacAddress              MacAddress,
    dot3OamPeerVendorOui               EightOTwoOui,
    dot3OamPeerVendorInfo              Unsigned32,
    dot3OamPeerMode                    INTEGER,
    dot3OamPeerMaxOamPduSize           Unsigned32,
    dot3OamPeerConfigRevision          Unsigned32,
    dot3OamPeerFunctionsSupported      BITS
  }

dot3OamPeerMacAddress OBJECT-TYPE
  SYNTAX      MacAddress
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The MAC address of the peer OAM entity. The MAC address is
    derived from the most recently received OAMPDU.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.5."
  ::= { dot3OamPeerEntry 1 }

dot3OamPeerVendorOui OBJECT-TYPE
  SYNTAX      EightOTwoOui
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The OUI of the OAM peer as reflected in the latest
    Information OAMPDU received with a Local Information TLV. The
    OUI can be used to identify the vendor of the remote OAM
    entity. This value is initialized to three octets of zero
    before any Local Information TLV is received.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.16."
  ::= { dot3OamPeerEntry 2 }

dot3OamPeerVendorInfo OBJECT-TYPE
  SYNTAX      Unsigned32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The Vendor Info of the OAM peer as reflected in the latest
    Information OAMPDU received with a Local Information TLV.
    The semantics of the Vendor Information field is proprietary
    and specific to the vendor (identified by the
    dot3OamPeerVendorOui). This information could, for example,
    be used to identify a specific product or product family.
    This value is initialized to zero before any Local
    Information TLV is received.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.17."
  ::= { dot3OamPeerEntry 3 }

dot3OamPeerMode OBJECT-TYPE
  SYNTAX      INTEGER {
                passive(1),
                active(2),
```

```
                unknown(3)
              }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The mode of the OAM peer as reflected in the latest
    Information OAMPDU received with a Local Information TLV. The
    mode of the peer can be determined from the Configuration
    field in the Local Information TLV of the last Information
    OAMPDU received from the peer. The value is unknown(3)
    whenever no Local Information TLV has been received. The
    values of active(2) and passive(1) are returned when a Local
    Information TLV has been received indicating that the peer is
    in active or passive mode, respectively.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.7."
  ::= { dot3OamPeerEntry 4 }

dot3OamPeerMaxOamPduSize OBJECT-TYPE
  SYNTAX      Unsigned32 (0 | 64..1518)
  UNITS       "octets"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The maximum size of OAMPDU supported by the peer as reflected
    in the latest Information OAMPDU received with a Local
    Information TLV. Ethernet OAM on this interface shall not use
    OAMPDUs that exceed this size. The maximum OAMPDU size can be
    determined from the PDU Configuration field of the Local
    Information TLV of the last Information OAMPDU received from
    the peer. A value of zero is returned if no Local Information
    TLV has been received. Otherwise, the value of the OAM peer's
    maximum OAMPDU size is returned in this value.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.9."
  ::= { dot3OamPeerEntry 5 }
dot3OamPeerConfigRevision OBJECT-TYPE
  SYNTAX      Unsigned32(0..65535)
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The configuration revision of the OAM peer as reflected in
    the latest OAMPDU. This attribute is changed by the peer
    whenever it has a local configuration change for Ethernet OAM
    on this interface. The configuration revision can be
    determined from the Revision field of the Local Information
    TLV of the most recently received Information OAMPDU with
    a Local Information TLV. A value of zero is returned if
    no Local Information TLV has been received.
  "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.13."
  ::= { dot3OamPeerEntry 6 }

dot3OamPeerFunctionsSupported OBJECT-TYPE
  SYNTAX      BITS {
                unidirectionalSupport (0),
                loopbackSupport(1),
                eventSupport(2),
                variableSupport(3)
```

```
               }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The OAM functions supported on this Ethernet-like interface.
    OAM consists of separate functionality sets above the basic
    discovery process. This value indicates the capabilities of
    the peer OAM entity with respect to these functions. This
    value is initialized so all bits are clear.

    If unidirectionalSupport(0) is set, then the peer OAM entity
    supports sending OAM frames on Ethernet interfaces when the
    receive path is known to be inoperable. If
    loopbackSupport(1) is set, then the peer OAM entity can send
    and receive OAM loopback commands. If eventSupport(2) is set,
    then the peer OAM entity can send and receive event OAMPDUs to
    signal various error conditions. If variableSupport(3) is
    set, then the peer OAM entity can send and receive variable
    requests to monitor the attribute value as described in Clause
    57 of IEEE Std 802.3.

    The capabilities of the OAM peer can be determined from the
    configuration field of the Local Information TLV of the most
    recently received Information OAMPDU with a Local Information
    TLV. All zeros are returned if no Local Information TLV has
    yet been received.
    "
  REFERENCE   "IEEE Std 802.3, REFERENCE 30.3.6.1.7."
  ::= { dot3OamPeerEntry 7 }

-- ***************************************************************
--
-- Ethernet OAM Loopback group
--

dot3OamLoopbackTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF Dot3OamLoopbackEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table contains controls for the loopback state of the
    local link as well as indicates the status of the loopback
    function. There is one entry in this table for each entry in
    dot3OamTable that supports loopback functionality (where
    dot3OamFunctionsSupported includes the loopbackSupport bit
    set).

    Loopback can be used to place the remote OAM entity in a state
    where every received frame (except OAMPDUs) is echoed back
    over the same interface on which they were received. In this
    state, at the remote entity, 'normal' traffic is disabled as
    only the looped back frames are transmitted on the interface.
    Loopback is thus an intrusive operation that prohibits normal
    data flow and should be used accordingly.
    "
  ::= { dot3OamObjects 3 }

dot3OamLoopbackEntry OBJECT-TYPE
  SYNTAX      Dot3OamLoopbackEntry
```



```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "An entry in the table, containing information on the loopback
      status for a single Ethernet-like interface. Entries in the
      table are automatically created whenever the local OAM entity
      supports loopback capabilities. The loopback status on the
      interface can be determined from the dot3OamLoopbackStatus
      object.
      "
    INDEX       { ifIndex }
    ::= { dot3OamLoopbackTable 1 }

  Dot3OamLoopbackEntry ::=
    SEQUENCE {
      dot3OamLoopbackStatus             INTEGER,
      dot3OamLoopbackIgnoreRx           INTEGER
    }

  dot3OamLoopbackStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                  -- all values, except where noted, can be read
                  -- but cannot be written
                  noLoopback (1),

                  -- initiatingLoopback can be read or written
                  initiatingLoopback (2),
                  remoteLoopback (3),

                  -- terminatingLoopback can be read or written
                  terminatingLoopback (4),
                  localLoopback (5),
                  unknown (6)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "The loopback status of the OAM entity. This status is
      determined by a combination of the local parser and
      multiplexer states, the remote parser and multiplexer states,
      as well as by the actions of the local OAM client. When
      operating in normal mode with no loopback in progress, the
      status reads noLoopback(1).

      The values initiatingLoopback(2) and terminatingLoopback(4)
      can be read or written. The other values can only be read -
      they can never be written. Writing initiatingLoopback causes
      the local OAM entity to start the loopback process with its
      peer. This value can only be written when the status is
      noLoopback(1). Writing the value initiatingLoopback(2) in any
      other state has no effect. When in remoteLoopback(3), writing
      terminatingLoopback(4) causes the local OAM entity to initiate
      the termination of the loopback state. Writing
      terminatingLoopack(4) in any other state has no effect.

      If the OAM client initiates a loopback and has sent a
      Loopback OAMPDU and is waiting for a response, where the local
      parser and multiplexer states are DISCARD (see IEEE Std 802.3,
      57.2.11.1), the status is 'initiatingLoopback'. In this
```

```
    case, the local OAM entity has yet to receive any
    acknowledgment that the remote OAM entity has received its
    loopback command request.
    If the local OAM client knows that the remote OAM entity is in
    loopback mode (via the remote state information as described
    in IEEE Std 802.3, 57.2.11.1, 30.3.6.1.15), the status is
    remoteLoopback(3). If the local OAM client is in the process
    of terminating the remote loopback (see IEEE Std 802.3, 57.2.11.3,
    30.3.6.1.14) with its local multiplexer and parser states in
    DISCARD, the status is terminatingLoopback(4). If the remote
    OAM client has put the local OAM entity in loopback mode as
    indicated by its local parser state, the status is
    localLoopback(5).

    The unknown(6) status indicates that the parser and
    multiplexer combination is unexpected. This status may be
    returned if the OAM loopback is in a transition state but
    should not persist.

    The values of this attribute correspond to the following
    values of the local and remote parser and multiplexer states.

      value            LclPrsr   LclMux    RmtPrsr   RmtMux
      noLoopback         FWD       FWD       FWD       FWD
      initLoopback     DISCARD   DISCARD     FWD       FWD
      rmtLoopback      DISCARD     FWD      LPBK    DISCARD
      tmtngLoopback    DISCARD   DISCARD    LPBK    DISCARD
      lclLoopback        LPBK    DISCARD   DISCARD     FWD
      unknown            ***   any other combination   ***
    "
  REFERENCE   "IEEE Std 802.3, 57.2.11, 30.3.61.14, 30.3.6.1.15"
  ::= { dot3OamLoopbackEntry 1 }

dot3OamLoopbackIgnoreRx OBJECT-TYPE
  SYNTAX      INTEGER { ignore(1), process(2) }
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Since OAM loopback is a disruptive operation (user traffic
    does not pass), this attribute provides a mechanism to provide
    controls over whether received OAM loopback commands are
    processed or ignored. When the value is ignore(1), received
    loopback commands are ignored. When the value is process(2),
    OAM loopback commands are processed. The default value is to
    ignore loopback commands (ignore(1)).
    "
  REFERENCE   "IEEE Std 802.3, 57.2.11, 30.3.61.14, 30.3.6.1.15"
  ::= { dot3OamLoopbackEntry 2 }
-- ***************************************************************
--
-- Ethernet OAM Statistics group
--

dot3OamStatsTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF Dot3OamStatsEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table contains statistics for the OAM function on a
```



```
    particular Ethernet-like interface. There is an entry in the
    table for every entry in the dot3OamTable.

    The counters in this table are defined as 32-bit entries to
    match the counter size as defined in IEEE Std 802.3. Given that
    the OAM protocol is a slow protocol, the counters increment at
    a slow rate.
    "
  ::= { dot3OamObjects 4 }

dot3OamStatsEntry OBJECT-TYPE
  SYNTAX      Dot3OamStatsEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An entry in the table containing statistics information on
    the Ethernet OAM function for a single Ethernet-like
    interface. Entries are automatically created for every entry
    in the dot3OamTable. Counters are maintained across
    transitions in dot3OamOperStatus.
    "
  INDEX       { ifIndex }
  ::= { dot3OamStatsTable 1 }

Dot3OamStatsEntry ::=
  SEQUENCE {
            dot3OamInformationTx                Counter32,
            dot3OamInformationRx                Counter32,
            dot3OamUniqueEventNotificationTx    Counter32,
            dot3OamUniqueEventNotificationRx    Counter32,
            dot3OamDuplicateEventNotificationTx Counter32,
            dot3OamDuplicateEventNotificationRx Counter32,
            dot3OamLoopbackControlTx            Counter32,
            dot3OamLoopbackControlRx            Counter32,
            dot3OamVariableRequestTx            Counter32,
            dot3OamVariableRequestRx            Counter32,
            dot3OamVariableResponseTx           Counter32,
            dot3OamVariableResponseRx           Counter32,
            dot3OamOrgSpecificTx                Counter32,
            dot3OamOrgSpecificRx                Counter32,
            dot3OamUnsupportedCodesTx           Counter32,
            dot3OamUnsupportedCodesRx           Counter32,
            dot3OamFramesLostDueToOam           Counter32
           }

dot3OamInformationTx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Information OAMPDUs transmitted on
    this interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime. "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.20."
  ::= { dot3OamStatsEntry 1 }
```



```
dot3OamInformationRx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Information OAMPDUs received on this
    interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.21."
  ::= { dot3OamStatsEntry 2 }

dot3OamUniqueEventNotificationTx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of unique Event OAMPDUs transmitted on
    this interface. Event Notifications may be sent in duplicate
    to increase the probability of successfully being received,
    given the possibility that a frame may be lost in transit.
    Duplicate Event Notification transmissions are counted by
    dot3OamDuplicateEventNotificationTx.

    A unique Event Notification OAMPDU is indicated as an Event
    Notification OAMPDU with a Sequence Number field that is
    distinct from the previously transmitted Event Notification
    OAMPDU Sequence Number.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.22."
  ::= { dot3OamStatsEntry 3 }

dot3OamUniqueEventNotificationRx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of unique Event OAMPDUs received on
    this interface. Event Notification OAMPDUs may be sent in
    duplicate to increase the probability of successfully being
    received, given the possibility that a frame may be lost in
    transit. Duplicate Event Notification receptions are counted
    by dot3OamDuplicateEventNotificationRx.

    A unique Event Notification OAMPDU is indicated as an Event
    Notification OAMPDU with a Sequence Number field that is
    distinct from the previously received Event Notification
    OAMPDU Sequence Number.
```

```
    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.24."
  ::= { dot3OamStatsEntry 4 }

dot3OamDuplicateEventNotificationTx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of duplicate Event OAMPDUs transmitted
    on this interface. Event Notification OAMPDUs may be sent in
    duplicate to increase the probability of successfully being
    received, given the possibility that a frame may be lost in
    transit.

    A duplicate Event Notification OAMPDU is indicated as an Event
    Notification OAMPDU with a Sequence Number field that is
    identical to the previously transmitted Event Notification
    OAMPDU Sequence Number.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.23."
  ::= { dot3OamStatsEntry 5 }

dot3OamDuplicateEventNotificationRx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of duplicate Event OAMPDUs received on
    this interface. Event Notification OAMPDUs may be sent in
    duplicate to increase the probability of successfully being
    received, given the possibility that a frame may be lost in
    transit.

    A duplicate Event Notification OAMPDU is indicated as an Event
    Notification OAMPDU with a Sequence Number field that is
    identical to the previously received Event Notification OAMPDU
    Sequence Number.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.25."
  ::= { dot3OamStatsEntry 6 }

dot3OamLoopbackControlTx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
```

```
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Loopback Control OAMPDUs transmitted
    on this interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.26."
  ::= { dot3OamStatsEntry 7 }

dot3OamLoopbackControlRx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Loopback Control OAMPDUs received
    on this interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.27."
  ::= { dot3OamStatsEntry 8 }

dot3OamVariableRequestTx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Variable Request OAMPDUs transmitted
    on this interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.28."
  ::= { dot3OamStatsEntry 9 }

dot3OamVariableRequestRx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Variable Request OAMPDUs received on
    this interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.29."
```

```
  ::= { dot3OamStatsEntry 10 }

dot3OamVariableResponseTx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Variable Response OAMPDUs
    transmitted on this interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.30."
  ::= { dot3OamStatsEntry 11 }

dot3OamVariableResponseRx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Variable Response OAMPDUs received
    on this interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.31."
  ::= { dot3OamStatsEntry 12 }

 dot3OamOrgSpecificTx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Organization Specific OAMPDUs
    transmitted on this interface.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.32."
  ::= { dot3OamStatsEntry 13 }

dot3OamOrgSpecificRx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of Organization Specific OAMPDUs
    received on this interface.
```



```
    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.33."
  ::= { dot3OamStatsEntry 14 }

dot3OamUnsupportedCodesTx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of OAMPDUs transmitted on this
    interface with an unsupported op-code.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.18."
  ::= { dot3OamStatsEntry 15 }

dot3OamUnsupportedCodesRx OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of OAMPDUs received on this interface
    with an unsupported op-code.

    Discontinuities of this counter can occur at re-initialization
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.19."
  ::= { dot3OamStatsEntry 16 }

dot3OamFramesLostDueToOam OBJECT-TYPE
  SYNTAX      Counter32
  UNITS       "frames"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "A count of the number of frames that were dropped by the OAM
    multiplexer. Since the OAM multiplexer has multiple inputs
    and a single output, there may be cases where frames are
    dropped due to transmit resource contention. This counter is
    incremented whenever a frame is dropped by the OAM layer.
    Note that any Ethernet frame, not just OAMPDUs, may be dropped
    by the OAM layer. This can occur when an OAMPDU takes
    precedence over a ‘normal’ frame resulting in the ‘normal’
    frame being dropped.

    When this counter is incremented, no other counters in this
    MIB are incremented.

    Discontinuities of this counter can occur at re-initialization
```



```
    of the management system, and at other times as indicated by
    the value of the ifCounterDiscontinuityTime.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.46."
  ::= { dot3OamStatsEntry 17 }

-- ***************************************************************
--
-- Ethernet OAM Event Configuration group
--

dot3OamEventConfigTable OBJECT-TYPE
  SYNTAX     SEQUENCE OF Dot3OamEventConfigEntry
  MAX-ACCESS not-accessible
  STATUS     current
  DESCRIPTION
    "Ethernet OAM includes the ability to generate and receive
    Event Notification OAMPDUs to indicate various link problems.
    This table contains the mechanisms to enable Event
    Notifications and configure the thresholds to generate the
    standard Ethernet OAM events. There is one entry in the table
    for every entry in dot3OamTable that supports OAM events
    (where dot3OamFunctionsSupported includes the eventSupport
    bit set). The values in the table are maintained across
    changes to dot3OamOperStatus.

    The standard threshold crossing events are:
      - Errored Symbol Period Event. Generated when the number of
        symbol errors exceeds a threshold within a given window
        defined by a number of symbols (for example, 1,000 symbols
        out of 1,000,000 had errors).
      - Errored Frame Period Event. Generated when the number of
        frame errors exceeds a threshold within a given window
        defined by a number of frames (for example, 10 frames out
        of 1000 had errors).
      - Errored Frame Event. Generated when the number of frame
        errors exceeds a threshold within a given window defined
        by a period of time (for example, 10 frames in 1 second
        had errors).
      - Errored Frame Seconds Summary Event. Generated when the
        number of errored frame seconds exceeds a threshold within
        a given time period (for example, 10 errored frame seconds
        within the last 100 seconds). An errored frame second is
        defined as a 1 second interval which had >0 frame errors.
    There are other events (dying gasp, critical events) that are
    not threshold crossing events but which can be
    enabled/disabled via this table.
    "
  ::= { dot3OamObjects 5 }

dot3OamEventConfigEntry OBJECT-TYPE
  SYNTAX      Dot3OamEventConfigEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Entries are automatically created and deleted from this
    table, and exist whenever the OAM entity supports Ethernet OAM
    events (as indicated by the eventSupport bit in
    dot3OamFunctionsSuppported). Values in the table are
```



```
      maintained across changes to the value of dot3OamOperStatus.

      Event configuration controls when the local management entity
      sends Event Notification OAMPDUs to its OAM peer, and when
      certain event flags are set or cleared in OAMPDUs.
      "
    INDEX       { ifIndex }
    ::= { dot3OamEventConfigTable 1 }
  Dot3OamEventConfigEntry ::=
    SEQUENCE {
              dot3OamErrSymPeriodWindowHi        Unsigned32,
              dot3OamErrSymPeriodWindowLo        Unsigned32,
              dot3OamErrSymPeriodThresholdHi     Unsigned32,
              dot3OamErrSymPeriodThresholdLo     Unsigned32,
              dot3OamErrSymPeriodEvNotifEnable   TruthValue,
              dot3OamErrFramePeriodWindow        Unsigned32,
              dot3OamErrFramePeriodThreshold     Unsigned32,
              dot3OamErrFramePeriodEvNotifEnable TruthValue,
              dot3OamErrFrameWindow              Unsigned32,
              dot3OamErrFrameThreshold           Unsigned32,
              dot3OamErrFrameEvNotifEnable       TruthValue,
              dot3OamErrFrameSecsSummaryWindow   Integer32,
              dot3OamErrFrameSecsSummaryThreshold Integer32,
              dot3OamErrFrameSecsEvNotifEnable   TruthValue,
              dot3OamDyingGaspEnable             TruthValue,
              dot3OamCriticalEventEnable         TruthValue
            }

  dot3OamErrSymPeriodWindowHi OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "2^32 symbols"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "The two objects dot3OamErrSymPeriodWindowHi and
      dot3OamErrSymPeriodLo together form an unsigned 64-bit
      integer representing the number of symbols over which this
      threshold event is defined. This is defined as
    dot3OamErrSymPeriodWindow = ((2^32)*dot3OamErrSymPeriodWindowHi)
                                  + dot3OamErrSymPeriodWindowLo

      If dot3OamErrSymPeriodThreshold symbol errors occur within a
      window of dot3OamErrSymPeriodWindow symbols, an Event
      Notification OAMPDU should be generated with an Errored Symbol
      Period Event TLV indicating that the threshold has been
      crossed in this window.

      The default value for dot3OamErrSymPeriodWindow is the number
      of symbols in one second for the underlying physical layer.
      "
    REFERENCE   "IEEE Std 802.3, 30.3.6.1.34"
    ::= { dot3OamEventConfigEntry 1 }

  dot3OamErrSymPeriodWindowLo OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "symbols"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
```

```
    "The two objects dot3OamErrSymPeriodWindowHi and
    dot3OamErrSymPeriodWindowLo together form an unsigned 64-bit
    integer representing the number of symbols over which this
    threshold event is defined. This is defined as

  dot3OamErrSymPeriodWindow = ((2^32)*dot3OamErrSymPeriodWindowHi)
                                  + dot3OamErrSymPeriodWindowLo

    If dot3OamErrSymPeriodThreshold symbol errors occur within a
    window of dot3OamErrSymPeriodWindow symbols, an Event
    Notification OAMPDU should be generated with an Errored Symbol
    Period Event TLV indicating that the threshold has been
    crossed in this window.

    The default value for dot3OamErrSymPeriodWindow is the number
    of symbols in one second for the underlying physical layer.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.34"
  ::= { dot3OamEventConfigEntry 2 }

dot3OamErrSymPeriodThresholdHi OBJECT-TYPE
  SYNTAX      Unsigned32
  UNITS       "2^32 symbols"
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "The two objects dot3OamErrSymPeriodThresholdHi and
    dot3OamErrSymPeriodThresholdLo together form an unsigned
    64-bit integer representing the minimum number of symbol errors
    occuring within a given window to cause an Errored Symbol Period Event.

    This is defined as

      dot3OamErrSymPeriodThreshold =
                        ((2^32) * dot3OamErrSymPeriodThresholdHi)
                                + dot3OamErrSymPeriodThresholdLo

    If dot3OamErrSymPeriodThreshold symbol errors occur within a
    window of dot3OamErrSymPeriodWindow symbols, an Event
    Notification OAMPDU is generated with an Errored Symbol
    Period Event TLV indicating that the threshold has been
    crossed in this window.

    The default value for dot3OamErrSymPeriodThreshold is one
    symbol errors. If the threshold value is zero, then an Event
    Notification OAMPDU is sent periodically (at the end of every
    window). This can be used as an asynchronous notification to
    the peer OAM entity of the statistics related to this
    threshold crossing alarm.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.34"
  ::= { dot3OamEventConfigEntry 3 }

dot3OamErrSymPeriodThresholdLo OBJECT-TYPE
  SYNTAX      Unsigned32
  UNITS       "symbols"
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
```

```
      "The two objects dot3OamErrSymPeriodThresholdHi and
      dot3OamErrSymPeriodThresholdLo together form an unsigned
      64-bit integer representing the minimum number of symbol errors
     occuring within a given window to cause an Errored Symbol Period Event.

      This is defined as

        dot3OamErrSymPeriodThreshold =
                         ((2^32) * dot3OamErrSymPeriodThresholdHi)
                                 + dot3OamErrSymPeriodThresholdLo

      If dot3OamErrSymPeriodThreshold symbol errors occur within a
      window of dot3OamErrSymPeriodWindow symbols, an Event
      Notification OAMPDU is generated with an Errored Symbol
      Period Event TLV indicating that the threshold has been
      crossed in this window.

      The default value for dot3OamErrSymPeriodThreshold is one
      symbol error. If the threshold value is zero, then an Event
      Notification OAMPDU is sent periodically (at the end of every
      window). This can be used as an asynchronous notification to
      the peer OAM entity of the statistics related to this
      threshold crossing alarm.
      "
    REFERENCE   "IEEE Std 802.3, 30.3.6.1.34"
    ::= { dot3OamEventConfigEntry 4 }

  dot3OamErrSymPeriodEvNotifEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "If true, the OAM entity sends an Event Notification
      OAMPDU when an Errored Symbol Period Event occurs.

      The default value for this object is true for
      Ethernet-like interfaces that support OAM. If the OAM layer
      does not support Event Notifications (as indicated via the
      dot3OamFunctionsSupported attribute), this value is ignored.
      "
    ::= { dot3OamEventConfigEntry 5 }

  dot3OamErrFramePeriodWindow OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "frames"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "The number of frames over which the threshold is defined.
      The default value of the window is the number of minimum size
      Ethernet frames that can be received over the physical layer
      in one second.

      If dot3OamErrFramePeriodThreshold frame errors occur within a
      window of dot3OamErrFramePeriodWindow frames, an Event
      Notification OAMPDU should be generated with an Errored Frame
      Period Event TLV indicating that the threshold has been
      crossed in this window.
      "
```

```
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.38"
  ::= { dot3OamEventConfigEntry 6 }

dot3OamErrFramePeriodThreshold OBJECT-TYPE
  SYNTAX      Unsigned32
  UNITS       "frames"
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "The minimum number of frame errors that cause an Errored Frame
    Period Event. The default value is one frame error. If the
    threshold value is zero, then an Event Notification OAMPDU is
    sent periodically (at the end of every window). This can be
    used as an asynchronous notification to the peer OAM entity of
    the statistics related to this threshold crossing alarm.

    If dot3OamErrFramePeriodThreshold frame errors occur within a
    window of dot3OamErrFramePeriodWindow frames, an Event
    Notification OAMPDU is generated with an Errored Frame
    Period Event TLV indicating that the threshold has been
    crossed in this window.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.38"
  ::= { dot3OamEventConfigEntry 7 }

dot3OamErrFramePeriodEvNotifEnable OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "If true, the OAM entity should send an Event Notification
    OAMPDU when an Errored Frame Period Event occurs.

    By default, this object should have the value true for
    Ethernet-like interfaces that support OAM. If the OAM layer
    does not support Event Notifications (as indicated via the
    dot3OamFunctionsSupported attribute), this value is ignored.
    "
  ::= { dot3OamEventConfigEntry 8 }

dot3OamErrFrameWindow OBJECT-TYPE
  SYNTAX      Unsigned32
  UNITS       "tenths of a second"
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "The amount of time (in 100ms increments) over which the
    threshold is defined. The default value is 10 (1 second).

    If dot3OamErrFrameThreshold frame errors occur within a window
    of dot3OamErrFrameWindow seconds (measured in tenths of
    seconds), an Event Notification OAMPDU should be generated
    with an Errored Frame Event TLV indicating that the threshold
    has been crossed in this window.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.36"
  DEFVAL { 10 }
  ::= { dot3OamEventConfigEntry 9 }
```

```
dot3OamErrFrameThreshold OBJECT-TYPE
  SYNTAX      Unsigned32
  UNITS       "frames"
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "The minimum number of frame errors that cause an Errored Frame
    Event. The default value is one frame error. If the
    threshold value is zero, then an Event Notification OAMPDU is
    sent periodically (at the end of every window). This can be
    used as an asynchronous notification to the peer OAM entity of
    the statistics related to this threshold crossing alarm.

    If dot3OamErrFrameThreshold frame errors occur within a window
    of dot3OamErrFrameWindow (in tenths of seconds), an Event
    Notification OAMPDU is generated with an Errored Frame
    Event TLV indicating the threshold has been crossed in this
    window.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.36"
  DEFVAL { 1 }
  ::= { dot3OamEventConfigEntry 10 }

dot3OamErrFrameEvNotifEnable OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "If true, the OAM entity should send an Event Notification
    OAMPDU when an Errored Frame Event occurs.

    By default, this object should have the value true for
    Ethernet-like interfaces that support OAM. If the OAM layer
    does not support Event Notifications (as indicated via the
    dot3OamFunctionsSupported attribute), this value is ignored.
    "
  DEFVAL { true }
  ::= { dot3OamEventConfigEntry 11 }

dot3OamErrFrameSecsSummaryWindow OBJECT-TYPE
  SYNTAX      Integer32 (100..9000)
  UNITS       "tenths of a second"
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "The amount of time (in 100 ms intervals) over which the
    threshold is defined. The default value is 100 (10 seconds).

    If dot3OamErrFrameSecsSummaryThreshold frame errors occur
    within a window of dot3OamErrFrameSecsSummaryWindow (in tenths
    of seconds), an Event Notification OAMPDU should be generated
    with an Errored Frame Seconds Summary Event TLV indicating
    that the threshold has been crossed in this window.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.40"
  DEFVAL { 100 }
  ::= { dot3OamEventConfigEntry 12 }

dot3OamErrFrameSecsSummaryThreshold OBJECT-TYPE
```

```
  SYNTAX      Integer32 (1..900)
  UNITS       "errored frame seconds"
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "The minimum number of errored frame seconds that cause an Errored
    Frame Seconds Summary Event. The default value is one errored frame
    second. If the threshold value is zero, then an Event
    Notification OAMPDU is sent periodically (at the end of every
    window). This can be used as an asynchronous notification to
    the peer OAM entity of the statistics related to this
    threshold crossing alarm.

    If dot3OamErrFrameSecsSummaryThreshold frame errors occur
    within a window of dot3OamErrFrameSecsSummaryWindow (in tenths
    of seconds), an Event Notification OAMPDU is generated
    with an Errored Frame Seconds Summary Event TLV indicating
    that the threshold has been crossed in this window.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.40"
  DEFVAL { 1 }
  ::= { dot3OamEventConfigEntry 13 }

dot3OamErrFrameSecsEvNotifEnable OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "If true, the local OAM entity sends an Event Notification
    OAMPDU when an Errored Frame Seconds Event occurs.

    The default value for this object is true for
    Ethernet-like interfaces that support OAM. If the OAM layer
    does not support Event Notifications (as indicated via the
    dot3OamFunctionsSupported attribute), this value is ignored.
    "
  DEFVAL { true }
  ::= { dot3OamEventConfigEntry 14 }

dot3OamDyingGaspEnable OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "If true, the local OAM entity should attempt to indicate a
    dying gasp via the OAMPDU flags field to its peer OAM entity
    when a dying gasp event occurs. The exact definition of a
    dying gasp event is implementation dependent. If the system
    does not support dying gasp capability, setting this object
    has no effect, and reading the object always returns
    'false'.

    The default value for this object is true for
    Ethernet-like interfaces that support OAM. If the OAM layer
    does not support Event Notifications (as indicated via the
    dot3OamFunctionsSupported attribute), this value is ignored.
    "
  DEFVAL { true }
  ::= { dot3OamEventConfigEntry 15 }
```



```
dot3OamCriticalEventEnable OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "If true, the local OAM entity should attempt to indicate a
    critical event via the OAMPDU flags to its peer OAM entity
    when a critical event occurs. The exact definition of a
    critical event is implementation dependent. If the system
    does not support critical event capability, setting this
    object has no effect, and reading the object should always
    result in 'false'.

    By default, this object should have the value true for
    Ethernet-like interfaces that support OAM. If the OAM layer
    does not support Event Notifications (as indicated via the
    dot3OamFunctionsSupported attribute), this value is ignored.
    "
  DEFVAL { true }
  ::= { dot3OamEventConfigEntry 16 }

-- **************************************************************
--
-- Ethernet OAM Event Log group
--

dot3OamEventLogTable OBJECT-TYPE
  SYNTAX       SEQUENCE OF Dot3OamEventLogEntry
  MAX-ACCESS   not-accessible
  STATUS       current
  DESCRIPTION
    "This table records a history of the events that have occurred
    at the Ethernet OAM level. These events can include locally
    detected events, which may result in locally generated
    OAMPDUs, and remotely detected events, which are detected by
    the OAM peer entity and signaled to the local entity via
    Ethernet OAM. Ethernet OAM events can be signaled by Event
    Notification OAMPDUs or by the flags field in any OAMPDU.

    This table contains both threshold crossing events and
    non-threshold crossing events. The parameters for the
    threshold window, threshold value, and actual value
    (dot3OamEventLogWindowXX, dot3OamEventLogThresholdXX,
    dot3OamEventLogValue) are only applicable to threshold
    crossing events, and are returned as all F's (2^32 - 1) for
    non-threshold crossing events.

    Entries in the table are automatically created when such
    events are detected. The size of the table is implementation
    dependent. When the table reaches its maximum size, older
    entries are automatically deleted to make room for newer
    entries.
    "
    ::= { dot3OamObjects 6 }

dot3OamEventLogEntry OBJECT-TYPE
  SYNTAX      Dot3OamEventLogEntry
  MAX-ACCESS  not-accessible
```

```
  STATUS      current
  DESCRIPTION
    "An entry in the dot3OamEventLogTable. Entries are
    automatically created whenever Ethernet OAM events occur at
    the local OAM entity, and when Event Notification OAMPDUs are
    received at the local OAM entity (indicating that events have
    occurred at the peer OAM entity). The size of the table is
    implementation dependent, but when the table becomes full,
    older events are automatically deleted to make room for newer
    events. The table index dot3OamEventLogIndex increments for
    each new entry, and when the maximum value is reached, the
    value restarts at zero.
    "
  INDEX       { ifIndex, dot3OamEventLogIndex }
  ::= { dot3OamEventLogTable 1 }

Dot3OamEventLogEntry ::=
  SEQUENCE {
    dot3OamEventLogIndex               Unsigned32,
    dot3OamEventLogTimestamp           TimeStamp,
    dot3OamEventLogOui                 EightOTwoOui,
    dot3OamEventLogType                Unsigned32,
    dot3OamEventLogLocation            INTEGER,
    dot3OamEventLogWindowHi            Unsigned32,
    dot3OamEventLogWindowLo            Unsigned32,
    dot3OamEventLogThresholdHi         Unsigned32,
    dot3OamEventLogThresholdLo         Unsigned32,
    dot3OamEventLogValue               CounterBasedGauge64,
    dot3OamEventLogRunningTotal        CounterBasedGauge64,
    dot3OamEventLogEventTotal          Unsigned32
  }

dot3OamEventLogIndex       OBJECT-TYPE
  SYNTAX      Unsigned32(1..4294967295)
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An arbitrary integer for identifying individual events
    within the event log. "
  ::= { dot3OamEventLogEntry 1 }

dot3OamEventLogTimestamp  OBJECT-TYPE
  SYNTAX      TimeStamp
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The value of sysUpTime at the time of the logged event. For
    locally generated events, the time of the event can be
    accurately retrieved from sysUpTime. For remotely generated
    events, the time of the event is indicated by the reception of
    the Event Notification OAMPDU indicating that the event
    occurred on the peer. A system may attempt to adjust the
    timestamp value to more accurately reflect the time of the
    event at the peer OAM entity by using other information, such
    as that found in the timestamp found of the Event Notification
    TLVs, which provides an indication of the relative time
    between events at the peer entity. "
  ::= { dot3OamEventLogEntry 2 }
```

```
dot3OamEventLogOui  OBJECT-TYPE
  SYNTAX      EightOTwoOui
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The OUI of the entity defining the object type. All IEEE
    802.3 defined events (as appearing in IEEE Std 802.3 except for the
    Organizationally Unique Event TLVs) use the IEEE 802.3 OUI of
    0x0180C2. Organizations defining their own Event Notification
    TLVs include their OUI in the Event Notification TLV that
    gets reflected here. "
  ::= { dot3OamEventLogEntry 3 }

dot3OamEventLogType      OBJECT-TYPE
  SYNTAX      Unsigned32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The type of event that generated this entry in the event log.
    When the OUI is the IEEE 802.3 OUI of 0x0180C2, the following
    event types are defined:
        erroredSymbolEvent(1),
        erroredFramePeriodEvent(2),
        erroredFrameEvent(3),
        erroredFrameSecondsEvent(4),
        linkFault(256),
        dyingGaspEvent(257),
        criticalLinkEvent(258)
    The first four are considered threshold crossing events, as
    they are generated when a metric exceeds a given value within
    a specified window. The other three are not threshold
    crossing events.

    When the OUI is not 71874 (0x0180C2 in hex), then some other
    organization has defined the event space. If event subtyping
    is known to the implementation, it may be reflected here.
    Otherwise, this value should return all F's (2^32 - 1).
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.10 and 57.5.3."
  ::= { dot3OamEventLogEntry 4 }

dot3OamEventLogLocation OBJECT-TYPE
  SYNTAX      INTEGER { local(1), remote(2) }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "Whether this event occurred locally (local(1)), or was
    received from the OAM peer via Ethernet OAM (remote(2)).
    "
  ::= { dot3OamEventLogEntry 5 }

dot3OamEventLogWindowHi      OBJECT-TYPE
  SYNTAX      Unsigned32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "If the event represents a threshold crossing event, the two
    objects dot3OamEventWindowHi and dot3OamEventWindowLo, form
    an unsigned 64-bit integer yielding the window over which the
```



```
    value was measured for the threshold crossing event (for
    example, 5, when 11 occurrences happened in 5 seconds while
    the threshold was 10). The two objects are combined as:
    dot3OamEventLogWindow = ((2^32) * dot3OamEventLogWindowHi)
                                + dot3OamEventLogWindowLo

    Otherwise, this value is returned as all F's (2^32 - 1) and
    adds no useful information.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
  ::= { dot3OamEventLogEntry 6 }

dot3OamEventLogWindowLo      OBJECT-TYPE
  SYNTAX      Unsigned32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "If the event represents a threshold crossing event, the two
    objects dot3OamEventWindowHi and dot3OamEventWindowLo form an
    unsigned 64-bit integer yielding the window over which the
    value was measured for the threshold crossing event (for
    example, 5, when 11 occurrences happened in 5 seconds while
    the threshold was 10). The two objects are combined as:

    dot3OamEventLogWindow = ((2^32) * dot3OamEventLogWindowHi)
                                + dot3OamEventLogWindowLo

    Otherwise, this value is returned as all F's (2^32 - 1) and
    adds no useful information.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
  ::= { dot3OamEventLogEntry 7 }

dot3OamEventLogThresholdHi      OBJECT-TYPE
  SYNTAX      Unsigned32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "If the event represents a threshold crossing event, the two
    objects dot3OamEventThresholdHi and dot3OamEventThresholdLo
    form an unsigned 64-bit integer yielding the value that was
    crossed for the threshold crossing event (for example, 10,
    when 11 occurrences happened in 5 seconds while the threshold
    was 10). The two objects are combined as:

  dot3OamEventLogThreshold = ((2^32) * dot3OamEventLogThresholdHi)
                                   + dot3OamEventLogThresholdLo

    Otherwise, this value is returned as all F's (2^32 -1) and
    adds no useful information.
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
  ::= { dot3OamEventLogEntry 8 }

dot3OamEventLogThresholdLo      OBJECT-TYPE
  SYNTAX      Unsigned32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
```

```
      "If the event represents a threshold crossing event, the two
      objects dot3OamEventThresholdHi and dot3OamEventThresholdLo
      form an unsigned 64-bit integer yielding the value that was
      crossed for the threshold crossing event (for example, 10,
      when 11 occurrences happened in 5 seconds while the threshold
      was 10). The two objects are combined as:

    dot3OamEventLogThreshold = ((2^32) * dot3OamEventLogThresholdHi)
                                   + dot3OamEventLogThresholdLo

      Otherwise, this value is returned as all F’s (2^32 - 1) and
      adds no useful information.
      "
    REFERENCE   "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
    ::= { dot3OamEventLogEntry 9 }

  dot3OamEventLogValue      OBJECT-TYPE
    SYNTAX      CounterBasedGauge64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "If the event represents a threshold crossing event, this
      value indicates the value of the parameter within the given
      window that generated this event (for example, 11, when 11
      occurrences happened in 5 seconds while the threshold was 10).

      Otherwise, this value is returned as all F’s
      (2^64 - 1) and adds no useful information.
      "
    REFERENCE   "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
    ::= { dot3OamEventLogEntry 10 }

  dot3OamEventLogRunningTotal      OBJECT-TYPE
    SYNTAX      CounterBasedGauge64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "Each Event Notification TLV contains a running total of the
      number of times an event has occurred, as well as the number
      of times an Event Notification for the event has been
      transmitted. For non-threshold crossing events, the number of
      events (dot3OamLogRunningTotal) and the number of resultant
      Event Notifications (dot3OamLogEventTotal) should be
      identical.

      For threshold crossing events, since multiple occurrences may
      be required to cross the threshold, these values are likely
      different. This value represents the total number of times
      this event has happened since the last reset (for example,
      3253, when 3253 symbol errors have occurred since the last
      reset, which has resulted in 51 symbol error threshold
      crossing events since the last reset).
      "
    REFERENCE   "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
    ::= { dot3OamEventLogEntry 11 }

  dot3OamEventLogEventTotal      OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
```

```
  STATUS      current
  DESCRIPTION
    "Each Event Notification TLV contains a running total of the
    number of times an event has occurred, as well as the number
    of times an Event Notification for the event has been
    transmitted. For non-threshold crossing events, the number of
    events (dot3OamLogRunningTotal) and the number of resultant
    Event Notifications (dot3OamLogEventTotal) should be
    identical.

    For threshold crossing events, since multiple occurrences may
    be required to cross the threshold, these values are likely
    different. This value represents the total number of times
    one or more of these occurrences have resulted in an Event
    Notification (for example, 51 when 3253 symbol errors have
    occurred since the last reset, which has resulted in 51 symbol
    error threshold crossing events since the last reset).
    "
  REFERENCE   "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
  ::= { dot3OamEventLogEntry 12 }

-- ***************************************************************
--
-- Ethernet OAM Notifications
--

dot3OamThresholdEvent NOTIFICATION-TYPE
  OBJECTS { dot3OamEventLogTimestamp,
            dot3OamEventLogOui,
            dot3OamEventLogType,
            dot3OamEventLogLocation,
            dot3OamEventLogWindowHi,
            dot3OamEventLogWindowLo,
            dot3OamEventLogThresholdHi,
            dot3OamEventLogThresholdLo,
            dot3OamEventLogValue,
            dot3OamEventLogRunningTotal,
            dot3OamEventLogEventTotal
          }
  STATUS  current
  DESCRIPTION
    "A dot3OamThresholdEvent notification is sent when a local or
    remote threshold crossing event is detected. A local
    threshold crossing event is detected by the local entity,
    while a remote threshold crossing event is detected by the
    reception of an Ethernet OAM Event Notification OAMPDU
    that indicates a threshold event.

    This notification should not be sent more than once per
    second.

    The OAM entity can be derived from extracting the ifIndex from
    the variable bindings. The objects in the notification
    correspond to the values in a row instance in the
    dot3OamEventLogTable.

    The management entity should periodically check
    dot3OamEventLogTable to detect any missed events."
 ::= { dot3OamNotifications 1 }
```

```
dot3OamNonThresholdEvent NOTIFICATION-TYPE
  OBJECTS { dot3OamEventLogTimestamp,
            dot3OamEventLogOui,
            dot3OamEventLogType,
            dot3OamEventLogLocation,
            dot3OamEventLogEventTotal
          }
  STATUS  current
  DESCRIPTION
    "A dot3OamNonThresholdEvent notification is sent when a local
    or remote non-threshold crossing event is detected. A local
    event is detected by the local entity, while a remote event is
    detected by the reception of an Ethernet OAM Event
    Notification OAMPDU that indicates a non-threshold crossing
    event.

    This notification should not be sent more than once per
    second.

    The OAM entity can be derived from extracting the ifIndex from
    the variable bindings. The objects in the notification
    correspond to the values in a row instance of the
    dot3OamEventLogTable.

    The management entity should periodically check
    dot3OamEventLogTable to detect any missed events."
 ::= { dot3OamNotifications 2 }

-- ***************************************************************
--
-- Conformance statements
--

dot3OamGroups OBJECT IDENTIFIER ::= { dot3OamConformance 1 }
dot3OamCompliances OBJECT IDENTIFIER ::= { dot3OamConformance 2 }

-- Compliance statements

dot3OamCompliance MODULE-COMPLIANCE
  STATUS          current
  DESCRIPTION "The compliance statement for managed entities
               supporting OAM on Ethernet-like interfaces.
               "
MODULE   -- this module
  MANDATORY-GROUPS { dot3OamControlGroup,
                     dot3OamPeerGroup,
                     dot3OamStatsBaseGroup
                   }

  GROUP       dot3OamLoopbackGroup
  DESCRIPTION
    "This group is mandatory for all IEEE 802.3 OAM
    implementations that support loopback functionality. "

  GROUP       dot3OamErrSymbolPeriodEventGroup
  DESCRIPTION
    "This group is mandatory for all IEEE 802.3 OAM
    implementations that support event functionality. "
```



```
   GROUP       dot3OamErrFramePeriodEventGroup
   DESCRIPTION
     "This group is mandatory for all IEEE 802.3 OAM
     implementations that support event functionality. "

   GROUP       dot3OamErrFrameEventGroup
   DESCRIPTION
     "This group is mandatory for all IEEE 802.3 OAM
     implementations that support event functionality. "

   GROUP       dot3OamErrFrameSecsSummaryEventGroup
   DESCRIPTION
     "This group is mandatory for all IEEE 802.3 OAM
     implementations that support event functionality. "

   GROUP        dot3OamFlagEventGroup
   DESCRIPTION
     "This group is optional for all IEEE 802.3 OAM
     implementations. The ability to send critical events or dying
     gasp events is not required in any system."

   GROUP       dot3OamEventLogGroup
   DESCRIPTION
     "This group is optional for all IEEE 802.3 OAM
     implementations. Entries in this table are dependent on what
     event functionality is supported in the local OAM
     implementation. At least one type of event shall be supported
     for entries to appear in this table. "

   GROUP       dot3OamNotificationGroup
   DESCRIPTION
     "This group is optional for all IEEE 802.3 OAM
     implementations. Since the information in the notifications
     is dependent on the dot3OamEventLogTable, that table shall be
     implemented for notifications. "

   ::= { dot3OamCompliances 1}

dot3OamControlGroup OBJECT-GROUP
   OBJECTS     {   dot3OamAdminState,
                   dot3OamOperStatus,
                   dot3OamMode,
                   dot3OamMaxOamPduSize,
                   dot3OamConfigRevision,
                   dot3OamFunctionsSupported
               }
   STATUS      current
   DESCRIPTION
     "A collection of objects providing the abilities,
     configuration, and status of an Ethernet OAM entity. "
   ::= { dot3OamGroups 1 }

dot3OamPeerGroup OBJECT-GROUP
   OBJECTS     {   dot3OamPeerMacAddress,
                   dot3OamPeerVendorOui,
                   dot3OamPeerVendorInfo,
                   dot3OamPeerMode,
                   dot3OamPeerFunctionsSupported,
```



```
                 dot3OamPeerMaxOamPduSize,
                 dot3OamPeerConfigRevision
             }
   STATUS      current
   DESCRIPTION
     "A collection of objects providing the abilities,
     configuration, and status of a peer Ethernet OAM entity. "
   ::= { dot3OamGroups 2 }

dot3OamStatsBaseGroup OBJECT-GROUP
   OBJECTS     {   dot3OamInformationTx,
                 dot3OamInformationRx,
                 dot3OamUniqueEventNotificationTx,
                 dot3OamUniqueEventNotificationRx,
                 dot3OamDuplicateEventNotificationTx,
                 dot3OamDuplicateEventNotificationRx,
                 dot3OamLoopbackControlTx,
                 dot3OamLoopbackControlRx,
                 dot3OamVariableRequestTx,
                 dot3OamVariableRequestRx,
                 dot3OamVariableResponseTx,
                 dot3OamVariableResponseRx,
                 dot3OamOrgSpecificTx,
                 dot3OamOrgSpecificRx,
                 dot3OamUnsupportedCodesTx,
                 dot3OamUnsupportedCodesRx,
                 dot3OamFramesLostDueToOam
             }
   STATUS      current
   DESCRIPTION
     "A collection of objects providing the statistics for the
     number of various transmit and receive events for OAM on an
     Ethernet-like interface. Note that all of these counters shall
     be supported even if the related function (as described in
     dot3OamFunctionsSupported) is not supported. "
   ::= { dot3OamGroups 3 }

dot3OamLoopbackGroup OBJECT-GROUP
   OBJECTS     {   dot3OamLoopbackStatus,
                 dot3OamLoopbackIgnoreRx
             }
   STATUS      current
   DESCRIPTION
     "A collection of objects for controlling the OAM remote
     loopback function. "
   ::= { dot3OamGroups 4 }

dot3OamErrSymbolPeriodEventGroup OBJECT-GROUP
   OBJECTS     {   dot3OamErrSymPeriodWindowHi,
                 dot3OamErrSymPeriodWindowLo,
                 dot3OamErrSymPeriodThresholdHi,
                 dot3OamErrSymPeriodThresholdLo,
                 dot3OamErrSymPeriodEvNotifEnable
             }
   STATUS      current
   DESCRIPTION
     "A collection of objects for configuring the thresholds for an
     Errored Symbol Period Event.
```

```
      Each IEEE Std 802.3 defined Event Notification TLV has its own
      conformance group because each event can be implemented
      independently of any other. "
    ::= { dot3OamGroups 5 }

dot3OamErrFramePeriodEventGroup OBJECT-GROUP
    OBJECTS      {   dot3OamErrFramePeriodWindow,
                     dot3OamErrFramePeriodThreshold,
                     dot3OamErrFramePeriodEvNotifEnable
                 }
    STATUS       current
    DESCRIPTION
      "A collection of objects for configuring the thresholds for an
      Errored Frame Period Event.

      Each IEEE Std 802.3 defined Event Notification TLV has its own
      conformance group because each event can be implemented
      independently of any other. "
    ::= { dot3OamGroups 6 }

dot3OamErrFrameEventGroup OBJECT-GROUP
    OBJECTS      {   dot3OamErrFrameWindow,
                     dot3OamErrFrameThreshold,
                     dot3OamErrFrameEvNotifEnable
                 }
    STATUS       current
    DESCRIPTION
      "A collection of objects for configuring the thresholds for an
      Errored Frame Event.

      Each IEEE Std 802.3 defined Event Notification TLV has its own
      conformance group because each event can be implemented
      independently of any other. "
    ::= { dot3OamGroups 7 }

dot3OamErrFrameSecsSummaryEventGroup OBJECT-GROUP
    OBJECTS      {   dot3OamErrFrameSecsSummaryWindow,
                     dot3OamErrFrameSecsSummaryThreshold,
                     dot3OamErrFrameSecsEvNotifEnable
                 }
    STATUS       current
    DESCRIPTION
      "A collection of objects for configuring the thresholds for an
      Errored Frame Seconds Summary Event.

      Each IEEE Std 802.3 defined Event Notification TLV has its own
      conformance group because each event can be implemented
      independently of any other. "
    ::= { dot3OamGroups 8 }

dot3OamFlagEventGroup OBJECT-GROUP
    OBJECTS      {   dot3OamDyingGaspEnable,
                     dot3OamCriticalEventEnable
                 }
    STATUS       current
    DESCRIPTION
      "A collection of objects for configuring the sending OAMPDUs
      with the critical event flag or dying gasp flag enabled. "
    ::= { dot3OamGroups 9 }
```

```
dot3OamEventLogGroup OBJECT-GROUP
  OBJECTS {  dot3OamEventLogTimestamp,
             dot3OamEventLogOui,
             dot3OamEventLogType,
             dot3OamEventLogLocation,
             dot3OamEventLogWindowHi,
             dot3OamEventLogWindowLo,
             dot3OamEventLogThresholdHi,
             dot3OamEventLogThresholdLo,
             dot3OamEventLogValue,
             dot3OamEventLogRunningTotal,
             dot3OamEventLogEventTotal
           }
  STATUS      current
  DESCRIPTION
     "A collection of objects for configuring the thresholds for an
     Errored Frame Seconds Summary Event and maintaining the event
     information. "
   ::= { dot3OamGroups 10 }

dot3OamNotificationGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
              dot3OamThresholdEvent,
              dot3OamNonThresholdEvent
                }
  STATUS      current
  DESCRIPTION
    "A collection of notifications used by Ethernet OAM to signal
    to a management entity that local or remote events have
    occurred on a specified Ethernet link. "
  ::= { dot3OamGroups 11 }

 END
```

7. Ethernet repeater device MIB module

7.1 Overview

This clause defines a portion of the Management Information Base (MIB) for use with SNMP. In particular, it defines objects for managing IEEE Std 802.3 repeaters.

7.1.1 Repeater management

Instances of the object types defined in this clause represent attributes of an IEEE 802.3 (Ethernet-like) repeater, as defined by Clause 9, “Repeater Unit for 10 Mb/s Baseband Networks” and Clause 27, “Repeater for 100 Mb/s Baseband Networks” of IEEE Std 802.3. Implementors of these MIB objects should note that IEEE Std 802.3 explicitly describes when, where, and how various repeater attributes are measured. IEEE Std 802.3 also describes the effects of repeater actions that may be invoked by manipulating instances of the MIB objects defined here. The definitions presented here are based on 30.4 of IEEE Std 802.3. The counters in this clause are defined to be the same as the counters defined in IEEE Std 802.3, with the intention that the same instrumentation can be used to implement both standards.

These repeater MIB module objects may be used to manage non-standard repeater-like devices; however, defining objects to describe implementation-specific properties of non-standard repeater-like devices is outside the scope of this standard.

7.1.2 Structure of the MIB

Objects in this MIB module are arranged into packages, each of which contains a set of related objects within a broad functional category. Objects within a package are generally defined under the same OID subtree. These packages are intended for organizational convenience only, and have no relation to the conformance groups defined later in the document.

7.1.2.1 Basic definitions

The basic definitions include objects that are applicable to all repeaters: status, parameter, and control objects for each repeater within the managed system, for the port groups within the system, and for the individual ports themselves.

7.1.2.2 Monitor definitions

The monitor definitions include monitoring statistics for each repeater within the system and for individual ports.

7.1.2.3 Address tracking definitions

This collection includes objects for tracking the MAC addresses of the DTEs attached to the ports within the system and for mapping the topology of a network.

7.1.2.4 Top N definitions

These objects may be used for tracking the ports with the most activity within the system or within particular repeaters.

7.1.3 Relationship to MIB-II

It is assumed that a repeater implementing this MIB will also implement (at least) the 'system' group defined in IETF RFC 1213 (MIB-II).

7.1.3.1 Relationship to the 'system' group

In MIB-II, the 'system' group is defined as being mandatory for all systems such that each managed entity contains one instance of each object in the 'system' group. Thus, those objects apply to the entity even if the entity's sole functionality is management of repeaters.

7.1.3.2 Relationship to the 'interfaces' group

In MIB-II, the 'interfaces' group is defined as being mandatory for all systems and contains information on an entity's interfaces, where each interface is thought of as being attached to a 'subnetwork'. (Note that this term is not to be confused with 'subnet', which refers to an addressing partitioning scheme used in the Internet suite of protocols.)

This repeater MIB module uses the notion of ports on a repeater. The concept of a MIB-II interface has no specific relationship to a repeater's port. Therefore, the 'interfaces' group applies only to the one (or more) network interfaces on which the entity managing the repeater sends and receives management protocol operations, and does not apply to the repeater's ports. This is consistent with the physical-layer nature of a repeater. A repeater-unit is a bitwise store-and-forward device. A repeater port has no MAC address, no MAC implementation, and does not pass packets up to higher-level protocol entities for processing.

NOTE—When a network management entity is observing a repeater, it may appear as though the repeater is passing packets to a higher-level protocol entity. However, this is only a means of implementing management, and this passing of management information is not part of the repeater functionality.[12]

7.2 Topology mapping

Network topology mapping is described in section 4 of IETF RFC 2108 [B19].

7.3 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL:[13]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C7mib.txt

[12]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

[13]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.


```
IEEE8023-SNMP-REPEATER-MIB DEFINITIONS ::= BEGIN

IMPORTS
    Counter32, Counter64, Integer32, Gauge32,
    OBJECT-TYPE, MODULE-IDENTITY, NOTIFICATION-TYPE, org
        FROM SNMPv2-SMI
    TimeStamp, MacAddress, TEXTUAL-CONVENTION,
    RowStatus, TestAndIncr
        FROM SNMPv2-TC
    OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
        FROM SNMPv2-CONF
    OwnerString
        FROM RFC1271-MIB;


ieee8023snmpRptrMIB MODULE-IDENTITY
     LAST-UPDATED "201102020000Z" -- February 2, 2011
     ORGANIZATION
       "IEEE 802.3 working group"
     CONTACT-INFO
         "WG-URL: http://www.ieee802.org/3/index.html
         WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

         Contact: Howard Frazier
         Postal:  3151 Zanker Road
                  San Jose, CA 95134
                  USA
         Tel:     +1.408.922.8164
         E-mail:  hfrazier@broadcom.com"
    DESCRIPTION
        "Management information for 802.3 repeaters."
    REVISION "201102020000Z" -- February 2, 2011
    DESCRIPTION
        "Initial revision, based on an earlier version in RFC 2108"

    ::= { org ieee(111) standards-association-numbers-series-standards(2)
          lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 7 }

ieee8023snmpDot3RptrMgt OBJECT IDENTIFIER ::= { ieee8023snmpRptrMIB 1}

OptMacAddr ::= TEXTUAL-CONVENTION
    DISPLAY-HINT    "1x:"
    STATUS          current
    DESCRIPTION
        "Either a 6 octet address in the ‘canonical’
        order defined by IEEE 802.1a, i.e., as if it
        were transmitted least significant bit first
        if a value is available or a zero length string."
    REFERENCE
        "See MacAddress in SNMPv2-TC. The only difference
        is that a zero length string is allowed as a value
        for OptMacAddr and not for MacAddress."
    SYNTAX OCTET STRING (SIZE (0 | 6))



-- Basic information at the repeater, group, and port level.

rptrBasicPackage
```

```
        OBJECT IDENTIFIER ::= { ieee8023snmpDot3RptrMgt 1 }
      rptrGroupInfo
            OBJECT IDENTIFIER ::= { rptrBasicPackage 1 }
      rptrPortInfo
            OBJECT IDENTIFIER ::= { rptrBasicPackage 2 }
      rptrAllRptrInfo
            OBJECT IDENTIFIER ::= { rptrBasicPackage 3 }

    -- Monitoring information at the repeater, group, and port level.
    rptrMonitorPackage
        OBJECT IDENTIFIER ::= { ieee8023snmpDot3RptrMgt 2 }
      rptrMonitorRptrInfo
            OBJECT IDENTIFIER ::= { rptrMonitorPackage 1 }
      rptrMonitorGroupInfo
            OBJECT IDENTIFIER ::= { rptrMonitorPackage 2 }
      rptrMonitorPortInfo
            OBJECT IDENTIFIER ::= { rptrMonitorPackage 3 }
      rptrMonitorAllRptrInfo
            OBJECT IDENTIFIER ::= { rptrMonitorPackage 4 }

    -- Address tracking information at the repeater, group,
    -- and port level.
    rptrAddrTrackPackage
        OBJECT IDENTIFIER ::= { ieee8023snmpDot3RptrMgt 3 }
      rptrAddrTrackRptrInfo
            OBJECT IDENTIFIER ::= { rptrAddrTrackPackage 1 }
      rptrAddrTrackGroupInfo
            -- this subtree is currently unused
            OBJECT IDENTIFIER ::= { rptrAddrTrackPackage 2 }
      rptrAddrTrackPortInfo
            OBJECT IDENTIFIER ::= { rptrAddrTrackPackage 3 }

    -- TopN information.
    rptrTopNPackage
            OBJECT IDENTIFIER ::= { ieee8023snmpDot3RptrMgt 4 }
      rptrTopNRptrInfo
            -- this subtree is currently unused
            OBJECT IDENTIFIER ::= { rptrTopNPackage 1 }
      rptrTopNGroupInfo
            -- this subtree is currently unused
            OBJECT IDENTIFIER ::= { rptrTopNPackage 2 }
      rptrTopNPortInfo
            OBJECT IDENTIFIER ::= { rptrTopNPackage 3 }

    -- Basic information at the group level.
    --
    -- Configuration and status objects for each
    -- managed group in the repeater system, independent
    -- of whether there is one or more managed
    -- repeater-units in the repeater system.

    rptrGroupTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF RptrGroupEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "Table of descriptive and status information about
                the groups of ports."
        ::= { rptrGroupInfo 1 }
```



```
rptrGroupEntry OBJECT-TYPE
    SYNTAX      RptrGroupEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry in the table, containing information
            about a single group of ports."
    INDEX    { rptrGroupIndex }
    ::= { rptrGroupTable 1 }

RptrGroupEntry ::=
    SEQUENCE {
        rptrGroupIndex
            Integer32,
        rptrGroupObjectID
            OBJECT IDENTIFIER,
        rptrGroupOperStatus
            INTEGER,
        rptrGroupPortCapacity
            Integer32
    }

rptrGroupIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the group within the
            repeater system for which this entry contains
            information."
    REFERENCE
            "IEEE Std 802.3, 30.4.2.1.1, aGroupID."
    ::= { rptrGroupEntry 1 }

rptrGroupObjectID OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The vendor's authoritative identification of the
            group. This value may be allocated within the SMI
            enterprises subtree (1.3.6.1.4.1) and provides a
            straight-forward and unambiguous means for
            determining what kind of group is being managed.

            For example, this object could take the value
            1.3.6.1.4.1.4242.1.2.14 if vendor 'Flintstones,
            Inc.' was assigned the subtree 1.3.6.1.4.1.4242,
            and had assigned the identifier
            1.3.6.1.4.1.4242.1.2.14 to its 'Wilma Flintstone
            6-Port FOIRL Plug-in module.'"
    ::= { rptrGroupEntry 2 }

rptrGroupOperStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                  other(1),
                  operational(2),
                  malfunctioning(3),
```

```
                  notPresent(4),
                  underTest(5),
                  resetInProgress(6)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "An object that indicates the operational status
            of the group.

            A status of notPresent(4) indicates that the group
            is temporarily or permanently physically and/or
            logically not a part of the repeater. It is an
            implementation-specific matter as to whether the
            agent effectively removes notPresent entries from
            the table.

            A status of operational(2) indicates that the
            group is functioning, and a status of
            malfunctioning(3) indicates that the group is
            malfunctioning in some way."
    ::= { rptrGroupEntry 3 }

rptrGroupPortCapacity OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The rptrGroupPortCapacity is the number of ports
            that can be contained within the group. Valid
            range is 1-2147483647. Within each group, the
            ports are uniquely numbered in the range from 1 to
            rptrGroupPortCapacity.

            Some ports may not be present in the repeater system, in
            which case the actual number of ports present
            will be less than the value of rptrGroupPortCapacity.
            The number of ports present in the group will never
            be greater than the value of rptrGroupPortCapacity.

            Note:  In practice, this will generally be the
            number of ports on a module, card, or board, and
            the port numbers will correspond to numbers marked
            on the physical embodiment."
    REFERENCE
            "IEEE 802.3 Mgt, 30.4.2.1.2, aGroupPortCapacity."
    ::= { rptrGroupEntry 4 }


-- Basic information at the port level.
--
-- Configuration and status objects for
-- each managed repeater port in the repeater system,
-- independent of whether there is one or more
-- managed repeater-units in the repeater system.

rptrPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrPortEntry
    MAX-ACCESS  not-accessible
```



```
    STATUS      current
    DESCRIPTION
            "Table of descriptive and status information about
            the repeater ports in the repeater system. The number of
            entries is independent of the number of repeaters
            in the managed repeater system."
    ::= { rptrPortInfo 1 }

rptrPortEntry OBJECT-TYPE
    SYNTAX      RptrPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry in the table, containing information
            about a single port."

    INDEX    { rptrPortGroupIndex, rptrPortIndex }
    ::= { rptrPortTable 1 }

RptrPortEntry ::=
    SEQUENCE {
        rptrPortGroupIndex
            Integer32,
        rptrPortIndex
            Integer32,
        rptrPortAdminStatus
            INTEGER,
        rptrPortAutoPartitionState
            INTEGER,
        rptrPortOperStatus
            INTEGER,
        rptrPortRptrId
            Integer32
    }

rptrPortGroupIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the group containing the
            port for which this entry contains information."
    ::= { rptrPortEntry 1 }

rptrPortIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the port within the group
            for which this entry contains information. This
            identifies the port independently from the repeater
            it may be attached to. The numbering scheme for
            ports is implementation specific; however, this
            value can never be greater than
            rptrGroupPortCapacity for the associated group."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.1, aPortID."
    ::= { rptrPortEntry 2 }
```



```
rptrPortAdminStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                  enabled(1),
                  disabled(2)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Setting this object to disabled(2) disables the
            port. A disabled port neither transmits nor
            receives. Once disabled, a port shall be
            explicitly enabled to restore operation. A port
            which is disabled when power is lost or when a
            reset is exerted shall remain disabled when normal
            operation resumes.

            The admin status takes precedence over auto-
            partition and functionally operates between the
            auto-partition mechanism and the AUI/PMA.

            Setting this object to enabled(1) enables the port
            and exerts a BEGIN on the port’s auto-partition
            state machine.

            (In effect, when a port is disabled, the value of
            rptrPortAutoPartitionState for that port is frozen
            until the port is next enabled. When the port
            becomes enabled, the rptrPortAutoPartitionState
            becomes notAutoPartitioned(1), regardless of its
            pre-disabling state.)"
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.2, aPortAdminState
            and 30.4.3.2.1, acPortAdminControl."
    ::= { rptrPortEntry 3 }

rptrPortAutoPartitionState OBJECT-TYPE
    SYNTAX      INTEGER {
                  notAutoPartitioned(1),
                  autoPartitioned(2)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The autoPartitionState flag indicates whether the
            port is currently partitioned by the repeater’s
            auto-partition protection.

            The conditions that cause port partitioning are
            specified in partition state machine in Clauses
            9 and 27 of IEEE Std 802.3. They are not
            differentiated here."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.3, aAutoPartitionState."
    ::= { rptrPortEntry 4 }

rptrPortOperStatus  OBJECT-TYPE
    SYNTAX      INTEGER {
                  operational(1),
```

```
                  notOperational(2),
                  notPresent(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object indicates the port's operational
            status. The notPresent(3) status indicates the
            port is physically removed (note this may or may
            not be possible depending on the type of port.)
            The operational(1) status indicates that the port
            is enabled (see rptrPortAdminStatus) and working,
            even though it might be auto-partitioned (see
            rptrPortAutoPartitionState).

            If this object has the value operational(1) and
            rptrPortAdminStatus is set to disabled(2), it is
            expected that this object's value will soon change
            to notOperational(2)."
    ::= { rptrPortEntry 5 }

rptrPortRptrId OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object identifies the repeater to
            which this port belongs. The repeater
            identified by a particular value of this object
            is the same as that identified by the same
            value of rptrInfoId. A value of zero
            indicates that this port currently is not
            a member of any repeater."
    ::= { rptrPortEntry 6 }


-- New version of basic information at the repeater level.
--
-- Configuration, status, and control objects for
-- each managed repeater in the repeater system.

rptrInfoTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrInfoEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table of information about each
            non-trivial repeater. The number of entries
            depends on the physical configuration of the
            managed repeater system."
    ::= { rptrAllRptrInfo 1 }

rptrInfoEntry OBJECT-TYPE
    SYNTAX      RptrInfoEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry in the table, containing information
            about a single non-trivial repeater."
```



```
    INDEX    { rptrInfoId }
    ::= { rptrInfoTable 1 }

RptrInfoEntry ::=
    SEQUENCE {
        rptrInfoId
            Integer32,
        rptrInfoRptrType
            INTEGER,
        rptrInfoOperStatus
            INTEGER,
        rptrInfoReset
            INTEGER,
        rptrInfoPartitionedPorts
            Gauge32,
        rptrInfoLastChange
            TimeStamp
    }

rptrInfoId OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the repeater for which
            this entry contains information."
    ::= { rptrInfoEntry 1 }

rptrInfoRptrType OBJECT-TYPE
    SYNTAX      INTEGER {
                  other(1),                -- undefined or unknown
                  tenMb(2),
                  onehundredMbClassI(3),
                  onehundredMbClassII(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The rptrInfoRptrType returns a value that identifies
            the CSMA/CD repeater type."
    REFERENCE
            "IEEE Std 802.3, 30.4.1.1.2, aRepeaterType."
    ::= { rptrInfoEntry 2 }

rptrInfoOperStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                  other(1),
                  ok(2),
                  failure(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The rptrInfoOperStatus object indicates the
            operational state of the repeater."
    REFERENCE
            "IEEE Std 802.3, 30.4.1.1.5, aRepeaterHealthState."
    ::= { rptrInfoEntry 3 }
```



```
rptrInfoReset OBJECT-TYPE
    SYNTAX      INTEGER {
                  noReset(1),
                  reset(2)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Setting this object to reset(2) causes a
            transition to the START state of Figure 9-2 in
            Clause 9 IEEE Std 802.3 for a 10 Mb/s repeater,
            and to the START state of Figure 27-2 in Clause 27
            of that standard for a 100 Mb/s repeater.

            Setting this object to noReset(1) has no effect.
            The agent will always return the value noReset(1)
            when this object is read.

            After receiving a request to set this variable to
            reset(2), the agent is allowed to delay the reset
            for a short period. For example, the implementor
            may choose to delay the reset long enough to allow
            the SNMP response to be transmitted. In any
            event, SNMP requires that a response be transmitted.

            This action does not reset the management counters
            defined in this document nor does it affect the
            portAdminStatus parameters. Included in this
            action is the execution of a disruptive Self-Test
            with the following characteristics:  a) The nature
            of the tests is not specified. b) The test resets
            the repeater but without affecting management
            information about the repeater. c) The test does
            not inject packets onto any segment. d) Packets
            received during the test may or may not be
            transferred. e) The test does not interfere with
            management functions.

            After performing this self-test, the agent will
            update the repeater health information (including
            rptrInfoOperStatus), and send a rptrInfoResetEvent
            notification."
    REFERENCE
            "IEEE Std 802.3, 30.4.1.2.1, acResetRepeater."
    ::= { rptrInfoEntry 4 }

rptrInfoPartitionedPorts OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object returns the total number of ports in
            the repeater whose current state meets all three
            of the following criteria:  rptrPortOperStatus
            does not have the value notPresent(3),
            rptrPortAdminStatus is enabled(1), and
            rptrPortAutoPartitionState is autoPartitioned(2)."
    ::= { rptrInfoEntry 5 }
```



```
rptrInfoLastChange OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The value of sysUpTime when any of the following
            conditions occurred:
              1) agent cold- or warm-started;
              2) this instance of repeater was created
                 (such as when a device or module was
                 added to the repeater system);
              3) a change in the value of rptrInfoOperStatus;
              4) ports were added or removed as members of
                 the repeater; or
              5) any of the counters associated with this
                 repeater had a discontinuity."
    ::= { rptrInfoEntry 6 }

-- Statistics at the port level.
--

rptrMonitorPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrMonitorPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Table of performance and error statistics for the
            ports. The number of entries is the same as that
            in the rptrPortTable.

            The columnar object rptrMonitorPortLastChange
            is used to indicate possible discontinuities
            of counter type columnar objects in the table."
    ::= { rptrMonitorPortInfo 1 }

rptrMonitorPortEntry OBJECT-TYPE
    SYNTAX      RptrMonitorPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry in the table, containing performance and
            error statistics for a single port."
    INDEX     { rptrMonitorPortGroupIndex, rptrMonitorPortIndex }
    ::= { rptrMonitorPortTable 1 }

RptrMonitorPortEntry ::=
    SEQUENCE {
        rptrMonitorPortGroupIndex
            Integer32,
        rptrMonitorPortIndex
            Integer32,
        rptrMonitorPortReadableFrames
            Counter32,
        rptrMonitorPortReadableOctets
            Counter32,
        rptrMonitorPortFCSErrors
            Counter32,
        rptrMonitorPortAlignmentErrors
            Counter32,
```

```
        rptrMonitorPortFrameTooLongs
            Counter32,
        rptrMonitorPortShortEvents
            Counter32,
        rptrMonitorPortRunts
            Counter32,
        rptrMonitorPortCollisions
            Counter32,
        rptrMonitorPortLateEvents
            Counter32,
        rptrMonitorPortVeryLongEvents
            Counter32,
        rptrMonitorPortDataRateMismatches
            Counter32,
        rptrMonitorPortAutoPartitions
            Counter32,
        rptrMonitorPortTotalErrors
            Counter32,
        rptrMonitorPortLastChange
            TimeStamp
    }

rptrMonitorPortGroupIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the group containing the
            port for which this entry contains information."
    ::= { rptrMonitorPortEntry 1 }

rptrMonitorPortIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the port within the group
            for which this entry contains information."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.1, aPortID."
    ::= { rptrMonitorPortEntry 2 }

rptrMonitorPortReadableFrames OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object is the number of frames of valid
            frame length that have been received on this port.
            This counter is incremented by one for each frame
            received on this port whose OctetCount is greater
            than or equal to minFrameSize and less than or
            equal to maxFrameSize (Ref: IEEE 802.3 Std,
            4.4.2.1) and for which the FCSError and
            CollisionEvent signals are not asserted.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes.
```

```
            This statistic provides one of the parameters
            necessary for obtaining the packet error rate.
            The approximate minimum time for rollover of this
            counter is 80 hours at 10 Mb/s."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.4, aReadableFrames."
    ::= { rptrMonitorPortEntry 3 }

rptrMonitorPortReadableOctets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object is the number of octets contained in
            valid frames that have been received on this port.
            This counter is incremented by OctetCount for each
            frame received on this port which has been
            determined to be a readable frame (i.e., including
            FCS octets but excluding framing bits and dribble
            bits).

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes.

            This statistic provides an indicator of the total
            data transferred. The approximate minimum time
            for rollover of this counter in a 10 Mb/s repeater
            is 58 minutes.

            For ports receiving traffic at a maximum rate in
            a 100 Mb/s repeater, this counter can roll over
            in less than 6 minutes. Since that amount of time
            could be less than a management station’s poll cycle
            time, in order to avoid a loss of information a
            management station is advised to also poll the
            rptrMonitorPortUpper32Octets object, or to use the
            64-bit counter defined by
            rptrMonitorPortHCReadableOctets instead of the
            two 32-bit counters."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.5, aReadableOctets."
    ::= { rptrMonitorPortEntry 4 }

rptrMonitorPortFCSErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one for each frame
            received on this port with the FCSError signal
            asserted and the FramingError and CollisionEvent
            signals deasserted and whose OctetCount is greater
            than or equal to minFrameSize and less than or
            equal to maxFrameSize (Ref: 4.4.2.1, IEEE 802.3
            Std).

            A discontinuity may occur in the value
```

```
            when the value of object
            rptrMonitorPortLastChange changes.

            The approximate minimum time for rollover of this
            counter is 80 hours at 10 Mb/s."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.6,
            aFrameCheckSequenceErrors."
    ::= { rptrMonitorPortEntry 5 }

rptrMonitorPortAlignmentErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one for each frame
            received on this port with the FCSError and
            FramingError signals asserted and CollisionEvent
            signal deasserted and whose OctetCount is greater
            than or equal to minFrameSize and less than or
            equal to maxFrameSize (Ref: IEEE 802.3 Std,
            4.4.2.1). If rptrMonitorPortAlignmentErrors is
            incremented then the rptrMonitorPortFCSErrors
            Counter shall not be incremented for the same
            frame.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes.

            The approximate minimum time for rollover of this
            counter is 80 hours at 10 Mb/s."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.7, aAlignmentErrors."
    ::= { rptrMonitorPortEntry 6 }

rptrMonitorPortFrameTooLongs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one for each frame
            received on this port whose OctetCount is greater
            than maxFrameSize (Ref: 4.4.2.1, IEEE 802.3 Std).
            If rptrMonitorPortFrameTooLongs is incremented
            then neither the rptrMonitorPortAlignmentErrors
            nor the rptrMonitorPortFCSErrors counter shall be
            incremented for the frame.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes.

            The approximate minimum time for rollover of this
            counter is 61 days in a 10 Mb/s repeater."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.8, aFramesTooLong."
    ::= { rptrMonitorPortEntry 7 }
```

```
rptrMonitorPortShortEvents OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one for each
            CarrierEvent on this port with ActivityDuration
            less than ShortEventMaxTime. ShortEventMaxTime is
            greater than 74 bit times and less than 82 bit
            times. ShortEventMaxTime has tolerances included
            to provide for circuit losses between a
            conformance test point at the AUI and the
            measurement point within the state machine.

            Notes:

            ShortEvents may indicate externally
            generated noise hits which will cause the repeater
            to transmit Runts to its other ports, or propagate
            a collision (which may be late) back to the
            transmitting DTE and damaged frames to the rest of
            the network.

            Implementors may wish to consider selecting the
            ShortEventMaxTime towards the lower end of the
            allowed tolerance range to accommodate bit losses
            suffered through physical channel devices not
            budgeted for within this standard.

            The significance of this attribute is different
            in 10 and 100 Mb/s collision domains. Clause 9
            repeaters perform fragment extension of short
            events which would be counted as runts on the
            interconnect ports of other repeaters. Clause
            27 repeaters do not perform fragment extension.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes.

            The approximate minimum time for rollover of this
            counter is 16 hours in a 10 Mb/s repeater."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.9, aShortEvents."
    ::= { rptrMonitorPortEntry 8 }

rptrMonitorPortRunts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one for each
            CarrierEvent on this port that meets one of the
            following two conditions. Only one test need be
            made. a) The ActivityDuration is greater than
            ShortEventMaxTime and less than ValidPacketMinTime
            and the CollisionEvent signal is deasserted. b)
            The OctetCount is less than 64, the
            ActivityDuration is greater than ShortEventMaxTime
```

```
                and the CollisionEvent signal is deasserted.
                ValidPacketMinTime is greater than or equal to 552
                bit times and less than 565 bit times.

                An event whose length is greater than 74 bit times
                but less than 82 bit times shall increment either
                the shortEvents counter or the runts counter but
                not both. A CarrierEvent greater than or equal to
                552 bit times but less than 565 bit times may or
                may not be counted as a runt.

                ValidPacketMinTime has tolerances included to
                provide for circuit losses between a conformance
                test point at the AUI and the measurement point
                within the state machine.

                Runts usually indicate collision fragments, a
                normal network event. In certain situations
                associated with large diameter networks a
                percentage of collision fragments may exceed
                ValidPacketMinTime.
                A discontinuity may occur in the value
                when the value of object
                rptrMonitorPortLastChange changes.

                The approximate minimum time for rollover of this
                counter is 16 hours in a 10 Mb/s repeater."
        REFERENCE
                "IEEE Std 802.3, 30.4.3.1.10, aRunts."
        ::= { rptrMonitorPortEntry 9 }

    rptrMonitorPortCollisions OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "For a Clause 9 repeater, this counter is
                incremented by one for any CarrierEvent signal
                on any port for which the CollisionEvent signal
                on this port is asserted. For a Clause 27
                repeater port the counter increments on entering
                the Collision Count Increment state of the
                partition state diagram (figure 27-8 of
                IEEE Std 802.3).

                A discontinuity may occur in the value
                when the value of object
                rptrMonitorPortLastChange changes.

                The approximate minimum time for rollover of this
                counter is 16 hours in a 10 Mb/s repeater."
        REFERENCE
                "IEEE Std 802.3, 30.4.3.1.11, aCollisions."
        ::= { rptrMonitorPortEntry 10 }

    rptrMonitorPortLateEvents OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
```



```
    DESCRIPTION
            "For a Clause 9 repeater port, this counter is
            incremented by one for each CarrierEvent
            on this port in which the CollIn(X)
            variable transitions to the value SQE (see
            9.6.6.2, IEEE Std 802.3) while the
            ActivityDuration is greater than the
            LateEventThreshold. For a Clause 27 repeater
            port, this counter is incremented by one on
            entering the Collision Count Increment state
            of the  partition state diagram (Figure 27-8)
            while the  ActivityDuration is greater than
            the LateEvent- Threshold. Such a CarrierEvent
            is counted twice, as both a collision and as a
            lateEvent.

            The LateEventThreshold is greater than 480 bit
            times and less than 565 bit times.
            LateEventThreshold has tolerances included to
            permit an implementation to build a single
            threshold to serve as both the LateEventThreshold
            and ValidPacketMinTime threshold.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes.

            The approximate minimum time for rollover of this
            counter is 81 hours in a 10 Mb/s repeater."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.12, aLateEvents."
    ::= { rptrMonitorPortEntry 11 }

rptrMonitorPortVeryLongEvents OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "For a Clause 9 repeater port, this counter
            is incremented by one for each CarrierEvent
            whose ActivityDuration is greater than the
            MAU Jabber Lockup Protection timer TW3
            (see 9.6.1 & 9.6.5, IEEE Std 802.3).

            For a Clause 27 repeater port, this counter
            is incremented by one on entry to the
            Rx Jabber state of the receiver timer state
            diagram (Figure 27-7). Other counters may
            be incremented as appropriate.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.13, aVeryLongEvents."
    ::= { rptrMonitorPortEntry 12 }

rptrMonitorPortDataRateMismatches OBJECT-TYPE
    SYNTAX      Counter32
```



```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one for each
            frame received by this port that meets all
            of the conditions required by only one of the
            following two measurement methods:

            Measurement method A:  1) The CollisionEvent
            signal is not asserted (10 Mb/s operation) or
            the Collision Count Increment state of the
            partition state diagram (figure 27-8 of
            IEEE Std 802.3) has not been entered
            (100 Mb/s operation). 2) The ActivityDuration
            is greater than ValidPacketMinTime. 3) The
            frequency (data rate) is detectably mismatched
            from the local transmit frequency.

            Measurement method B:  1) The CollisionEvent
            signal is not asserted (10 Mb/s operation)
            or the Collision Count Increment state of the
            partition state diagram (figure 27-8 of
            IEEE Std 802.3) has not been entered
            (100 Mb/s operation). 2) The OctetCount is
            greater than 63. 3) The frequency (data
            rate) is detectably mismatched from the local
            transmit frequency. The exact degree of
            mismatch is vendor specific and is to be
            defined by the vendor for conformance testing.

            When this event occurs, other counters whose
            increment conditions were satisfied may or may not
            also be incremented, at the implementor's
            discretion. Whether or not the repeater was able
            to maintain data integrity is beyond the scope of
            this standard.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.14, aDataRateMismatches."
    ::= { rptrMonitorPortEntry 13 }

rptrMonitorPortAutoPartitions OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one for
            each time the repeater has automatically
            partitioned this port.

            The conditions that cause a Clause 9
            repeater port to partition are specified in
            the partition state diagram in Clause 9 of
            IEEE Std 802.3. They are not differentiated
            here. A Clause 27 repeater port partitions
            on entry to the Partition Wait state of the
```

```
            partition state diagram (Figure 27-8 in
            IEEE Std 802.3).

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.15, aAutoPartitions."
    ::= { rptrMonitorPortEntry 14 }

rptrMonitorPortTotalErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of errors which have occurred on
            this port. This counter is the summation of the
            values of other error counters (for the same
            port), namely:

                rptrMonitorPortFCSErrors,
                rptrMonitorPortAlignmentErrors,
                rptrMonitorPortFrameTooLongs,
                rptrMonitorPortShortEvents,
                rptrMonitorPortLateEvents,
                rptrMonitorPortVeryLongEvents,
                rptrMonitorPortDataRateMismatches, and
                rptrMonitorPortSymbolErrors.

            This counter is redundant in the sense that it is
            the summation of information already available
            through other objects. However, it is included
            specifically because the regular retrieval of this
            object as a means of tracking the health of a port
            provides a considerable optimization of network
            management traffic over the otherwise necessary
            retrieval of the summed counters.

            Note that rptrMonitorPortRunts is not included
            in this total; this is because runts usually
            indicate collision fragments, a normal network
            event.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes."
    ::= { rptrMonitorPortEntry 15 }

rptrMonitorPortLastChange OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The value of sysUpTime when the last of
            the following occurred:
              1) the agent cold- or warm-started;
              2) the row for the port was created
                 (such as when a device or module was added
                  to the repeater system); or
```



```
                3) any condition that would cause one of
                   the counters for the row to experience
                   a discontinuity."
     ::= { rptrMonitorPortEntry 16 }

 rptrMonitor100PortTable OBJECT-TYPE
     SYNTAX      SEQUENCE OF RptrMonitor100PortEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
             "Table of additional performance and error
             statistics for 100 Mb/s ports, above and
             beyond those parameters that apply to both
             10 and 100 Mb/s ports. Entries exist only for
             ports attached to 100 Mb/s repeaters.

             The columnar object rptrMonitorPortLastChange
             is used to indicate possible discontinuities
             of counter type columnar objects in this table."
     ::= { rptrMonitorPortInfo 2 }

 rptrMonitor100PortEntry OBJECT-TYPE
     SYNTAX      RptrMonitor100PortEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
             "An entry in the table, containing performance
             and error statistics for a single 100 Mb/s port."
     INDEX    { rptrMonitorPortGroupIndex, rptrMonitorPortIndex }
     ::= { rptrMonitor100PortTable 1 }

 RptrMonitor100PortEntry ::=
     SEQUENCE {
         rptrMonitorPortIsolates
             Counter32,
         rptrMonitorPortSymbolErrors
             Counter32,
         rptrMonitorPortUpper32Octets
             Counter32,
         rptrMonitorPortHCReadableOctets
             Counter64
     }

 rptrMonitorPortIsolates OBJECT-TYPE
     SYNTAX      Counter32
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
             "This counter is incremented by one each time that
             the repeater port automatically isolates as a
             consequence of false carrier events. The conditions
             which cause a port to automatically isolate are
             defined by the transition from the False Carrier
             state to the Link Unstable state of the carrier
             integrity state diagram (figure 27-9)
             [IEEE 802.3 Standard].

             Note:  Isolates do not affect the value of
             the PortOperStatus object.
```

```
            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.16, aIsolates."
    ::= { rptrMonitor100PortEntry 1 }

rptrMonitorPortSymbolErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one each time when
            valid length packet was received at the port and
            there was at least one occurrence of an invalid
            data symbol. This can increment only once per valid
            carrier event. A collision presence at any port of
            the repeater containing port N, will not cause this
            attribute to increment.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes.

            The approximate minimum time for rollover of this
            counter is 7.4 hours at 100 Mb/s."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.17,
            aSymbolErrorDuringPacket."
    ::= { rptrMonitor100PortEntry 2 }

rptrMonitorPortUpper32Octets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object is the number of octets contained in
            valid frames that have been received on this port,
            modulo 2**32. That is, it contains the upper 32
            bits of a 64-bit octets counter, of which the
            lower 32 bits are contained in the
            rptrMonitorPortReadableOctets object.

            This two-counter mechanism is provided for those
            network management protocols that do not support
            64-bit counters (e.g. SNMP V1) and are used to
            manage a repeater type of 100 Mb/s.

            Conformance clauses for this MIB are defined such
            that implementation of this object is not required
            in a repeater system which does not support 100 Mb/s.
            However, repeater systems with mixed 10 and 100 Mb/s ports
            may implement this object across all ports,
            including 10 Mb/s. If this object is implemented, the
            value shall be a valid count as defined
            in the first paragraph of this description.

            A discontinuity may occur in the value
```



```
            when the value of object
            rptrMonitorPortLastChange changes."
    ::= { rptrMonitor100PortEntry 3 }

rptrMonitorPortHCReadableOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object is the number of octets contained in
            valid frames that have been received on this port.
            This counter is incremented by OctetCount for each
            frame received on this port which has been
            determined to be a readable frame (i.e., including
            FCS octets but excluding framing bits and dribble
            bits).

            This statistic provides an indicator of the total
            data transferred.

            This counter is a 64-bit version of rptrMonitor-
            PortReadableOctets. It should be used by network
            management protocols which suppport 64-bit counters
            (e.g. SNMPv2).

            Conformance clauses for this MIB are defined such
            that implementation of this object is not required
            in a repeater system which does not support 100 Mb/s.
            However, repeater systems with mixed 10 and 100 Mb/s ports
            may implement this object across all ports,
            including 10 Mb/s. If this object is implemented, the
            value shall be a valid count as defined
            in the first paragraph of this description.

            A discontinuity may occur in the value
            when the value of object
            rptrMonitorPortLastChange changes."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.5, aReadableOctets."
    ::= { rptrMonitor100PortEntry 4 }


-- New version of statistics at the repeater level.
--
-- Statistics objects for each managed repeater
-- in the repeater system.

rptrMonTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrMonEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table of information about each
            non-trivial repeater. The number of entries
            in this table is the same as the number of
            entries in the rptrInfoTable.

            The columnar object rptrInfoLastChange is
            used to indicate possible discontinuities of
```

```
            counter type columnar objects in this table."
    ::= { rptrMonitorAllRptrInfo 1 }

rptrMonEntry OBJECT-TYPE
    SYNTAX      RptrMonEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry in the table, containing information
            about a single non-trivial repeater."
    INDEX    { rptrInfoId }
    ::= { rptrMonTable 1 }

RptrMonEntry ::=
    SEQUENCE {
        rptrMonTxCollisions
            Counter32,
        rptrMonTotalFrames
            Counter32,
        rptrMonTotalErrors
            Counter32,
        rptrMonTotalOctets
            Counter32
    }

rptrMonTxCollisions OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "For a Clause 9 (10 Mb/s) repeater, this counter
            is incremented every time the repeater state
            machine enters the TRANSMIT COLLISION state
            from any state other than ONE PORT LEFT
            (see Figure 9-2 IEEE Std 802.3).

            For a Clause 27 repeater, this counter is
            incremented every time the repeater core state
            diagram enters the Jam state as a result of
            Activity(ALL) > 1 (see Figure 27-2 IEEE Std 802.3).

            The approximate minimum time for rollover of this
            counter is 16 hours in a 10 Mb/s repeater and 1.6
            hours in a 100 Mb/s repeater."
    REFERENCE
            "IEEE Std 802.3, 30.4.1.1.8, aTransmitCollisions"
    ::= { rptrMonEntry 1 }

rptrMonTotalFrames OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of frames of valid frame length
            that have been received on the ports in this repeater
            and for which the FCSError and CollisionEvent
            signals were not asserted. If an implementation
            can not obtain a count of frames as seen by
            the repeater itself, this counter may be
```



```
            implemented as the summation of the values of the
            rptrMonitorPortReadableFrames counters for all of
            the ports in the repeater.

            This statistic provides one of the parameters
            necessary for obtaining the packet error rate.
            The approximate minimum time for rollover of this
            counter is 80 hours in a 10 Mb/s repeater."
    ::= { rptrMonEntry 3 }

rptrMonTotalErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of errors which have occurred on
            all of the ports in this repeater. The errors
            included in this count are the same as those listed
            for the rptrMonitorPortTotalErrors counter. If an
            implementation can not obtain a count of these
            errors as seen by the repeater itself, this counter
            may be implemented as the summation of the values of
            the rptrMonitorPortTotalErrors counters for all of
            the ports in the repeater."
    ::= { rptrMonEntry 4 }

rptrMonTotalOctets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of octets contained in the valid
            frames that have been received on the ports in
            this group. If an implementation can not obtain
            a count of octets as seen by the repeater itself,
            this counter may be the summation of the
            values of the rptrMonitorPortReadableOctets
            counters for all of the ports in the group.

            This statistic provides an indicator of the total
            data transferred. The approximate minimum time
            for rollover of this counter in a 10 Mb/s repeater
            is 58 minutes divided by the number of ports in
            the repeater.

            For 100 Mb/s repeaters processing traffic at a
            maximum rate, this counter can roll over in less
            than 6 minutes divided by the number of ports in
            the repeater. Since that amount of time could
            be less than a management station’s poll cycle
            time, in order to avoid a loss of information a
            management station is advised to also poll the
            rptrMonUpper32TotalOctets object, or to use the
            64-bit counter defined by rptrMonHCTotalOctets
            instead of the two 32-bit counters."
    ::= { rptrMonEntry 5 }

rptrMon100Table OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrMon100Entry
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table of additional information about each
            100 Mb/s repeater, augmenting the entries in
            the rptrMonTable. Entries exist in this table
            only for 100 Mb/s repeaters.

            The columnar object rptrInfoLastChange is
            used to indicate possible discontinuities of
            counter type columnar objects in this table."
    ::= { rptrMonitorAllRptrInfo 2 }

rptrMon100Entry OBJECT-TYPE
    SYNTAX      RptrMon100Entry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry in the table, containing information
            about a single 100 Mb/s repeater."
    INDEX    { rptrInfoId }
    ::= { rptrMon100Table 1 }

RptrMon100Entry ::=
    SEQUENCE {
        rptrMonUpper32TotalOctets
            Counter32,
        rptrMonHCTotalOctets
            Counter64
    }

rptrMonUpper32TotalOctets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of octets contained in the valid
            frames that have been received on the ports in
            this repeater, modulo 2**32. That is, it contains
            the upper 32 bits of a 64-bit counter, of which
            the lower 32 bits are contained in the
            rptrMonTotalOctets object. If an implementation
            can not obtain a count of octets as seen
            by the repeater itself, the 64-bit value
            may be the summation of the values of the
            rptrMonitorPortReadableOctets counters combined
            with the corresponding rptrMonitorPortUpper32Octets
            counters for all of the ports in the repeater.

            This statistic provides an indicator of the total
            data transferred within the repeater.

            This two-counter mechanism is provided for those
            network management protocols that do not support
            64-bit counters (e.g. SNMP V1) and are used to
            manage a repeater type of 100 Mb/s.

            Conformance clauses for this MIB are defined such
            that implementation of this object is not required
```



```
            in a repeater system which does not support 100 Mb/s.
            However, repeater systems with mixed 10 and 100 Mb/s ports
            may implement this object across all ports,
            including 10 Mb/s. If this object is implemented, the
            value shall be a valid count as defined
            in the first paragraph of this description."
    ::= { rptrMon100Entry 1 }

rptrMonHCTotalOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of octets contained in the valid
            frames that have been received on the ports in
            this group. If a implementation can not obtain
            a count of octets as seen by the repeater itself,
            this counter may be the summation of the
            values of the rptrMonitorPortReadableOctets
            counters for all of the ports in the group.

            This statistic provides an indicator of the total
            data transferred.

            This counter is a 64-bit (high-capacity) version
            of rptrMonUpper32TotalOctets and rptrMonTotalOctets.
            It should be used by network management protocols
            which support 64-bit counters (e.g. SNMPv2).

            Conformance clauses for this MIB are defined such
            that implementation of this object is not required
            in a repeater system which does not support 100 Mb/s.
            However, repeater systems with mixed 10 and 100 Mb/s ports
            may implement this object across all ports,
            including 10 Mb/s. If this object is implemented, the
            value shall be a valid count as defined
            in the first paragraph of this description."
    ::= { rptrMon100Entry 2 }


--
-- The Repeater Address Search Table
--
-- This table provides an active address tracking
-- capability which can be also used to collect the
-- necessary information for mapping the topology
-- of a network. Note that an NMS is required to have
-- read-write access to the table in order to access
-- this function. Section 4 "Topology Mapping" of
-- IETF RFC 2108 [B19] contains a description of an
-- algorithm which can make use of this table,
-- in combination with the forwarding databases
-- of managed bridges/switches in the network,
-- to map network topology. Devices may also
-- utilise the protocol and a set of managed
-- objects defined in IEEE Std 802.1AB Station
-- and Media Access Control Connectivity
-- Discovery to discover the physical topology
-- from adjacent stations.
```



```
--

rptrAddrSearchTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF RptrAddrSearchEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "This table contains one entry per repeater in the
            repeater system. It defines objects which allow a network
            management application to instruct an agent to watch
            for a given MAC address and report which port it
            was seen on. Only one address search can be in
            progress on each repeater at any one time. Before
            starting an address search, a management application
            should obtain 'ownership' of the entry in
            rptrAddrSearchTable for the repeater that is to
            perform the search. This is accomplished with the
            rptrAddrSearchLock and rptrAddrSearchStatus as
            follows:

            try_again:
                get(rptrAddrSearchLock, rptrAddrSearchStatus)
                while (rptrAddrSearchStatus != notInUse)
                {
                    /* Loop waiting for objects to be available*/
                    short delay
                    get(rptrAddrSearchLock, rptrAddrSearchStatus)
                }

                /* Try to claim map objects */
                lock_value = rptrAddrSearchLock
                if ( set(rptrAddrSearchLock = lock_value,
                         rptrAddrSearchStatus = inUse,
                         rptrAddrSearchOwner = 'my-IP-address)
                      == FAILURE)
                    /* Another manager got the lock */
                    goto try_again

                /* I have the lock */
                set (rptrAddrSearchAddress = <search target>)

                wait for rptrAddrSearchState to change from none

                if (rptrAddrSearchState == single)
                    get (rptrAddrSearchGroup, rptrAddrSearchPort)
                /* release the lock, making sure not to overwrite
                   anyone else's lock */
                set (rptrAddrSearchLock = lock_value+1,
                     rptrAddrSearchStatus = notInUse,
                     rptrAddrSearchOwner = '')

            A management station first retrieves the values of
            the appropriate instances of the rptrAddrSearchLock
            and rptrAddrSearchStatus objects, periodically
            repeating the retrieval if necessary, until the value
            of rptrAddrSearchStatus is 'notInUse'. The
            management station then tries to set the same
            instance of the rptrAddrSearchLock object to the
            value it just retrieved, the same instance of the
```



```
            rptrAddrSearchStatus object to 'inUse', and the
            corresponding instance of rptrAddrSearchOwner to a
            value indicating itself. If the set operation
            succeeds, then the management station has obtained
            ownership of the rptrAddrSearchEntry, and the value
            of rptrAddrSearchLock is incremented by the agent (as
            per the semantics of TestAndIncr). Failure of the
            set operation indicates that some other manager has
            obtained ownership of the rptrAddrSearchEntry.

            Once ownership is obtained, the management station
            can proceed with the search operation. Note that the
            agent will reset rptrAddrSearchStatus to 'notInUse'
            if it has been in the 'inUse' state for an abnormally
            long period of time, to prevent a misbehaving manager
            from permanently locking the entry. It is suggested
            that this timeout period be between one and five
            minutes.

            When the management station has completed its search
            operation, it should free the entry by setting
            the instance of the rptrAddrSearchLock object to the
            previous value + 1, the instance of the
            rptrAddrSearchStatus to 'notInUse', and the instance
            of rptrAddrSearchOwner to a zero length string. This
            is done to prevent overwriting another station's
            lock."
    ::= { rptrAddrTrackRptrInfo 1 }

rptrAddrSearchEntry OBJECT-TYPE
    SYNTAX     RptrAddrSearchEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "An entry containing objects for invoking an address
            search on a repeater."
    INDEX      { rptrInfoId }
    ::= { rptrAddrSearchTable 1 }

RptrAddrSearchEntry ::=
    SEQUENCE {
        rptrAddrSearchLock      TestAndIncr,
        rptrAddrSearchStatus    INTEGER,
        rptrAddrSearchAddress   MacAddress,
        rptrAddrSearchState     INTEGER,
        rptrAddrSearchGroup     Integer32,
        rptrAddrSearchPort      Integer32,
        rptrAddrSearchOwner     OwnerString
    }


rptrAddrSearchLock OBJECT-TYPE
    SYNTAX     TestAndIncr
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
            "This object is used by a management station as an
            advisory lock for this rptrAddrSearchEntry."
    ::= { rptrAddrSearchEntry 1 }
```

```
rptrAddrSearchStatus  OBJECT-TYPE
    SYNTAX     INTEGER {
                    notInUse(1),
                    inUse(2)
               }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
            "This object is used to indicate that some management
            station is currently using this rptrAddrSearchEntry.
            Cooperating managers should set this object to
            'notInUse' when they are finished using this entry.
            The agent will automatically set the value of this
            object to 'notInUse' if it has been set to 'inUse'
            for an unusually long period of time."
    ::= { rptrAddrSearchEntry 2 }

rptrAddrSearchAddress OBJECT-TYPE
    SYNTAX     MacAddress
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
            "This object is used to search for a specified MAC
            address. When this object is set, an address search
            begins. This automatically sets the corresponding
            instance of the rptrAddrSearchState object  to 'none'
            and the corresponding instances of the
            rptrAddrSearchGroup and rptrAddrSearchPort objects to
            0.

            When a valid frame is received by this repeater with
            a source MAC address which matches the current value
            of rptrAddrSearchAddress, the agent will update the
            corresponding instances of rptrAddrSearchState,
            rptrAddrSearchGroup and rptrAddrSearchPort to reflect
            the current status of the search, and the group and
            port on which the frame was seen."
    ::= { rptrAddrSearchEntry 3 }

rptrAddrSearchState OBJECT-TYPE
    SYNTAX     INTEGER {
                     none(1),
                     single(2),
                     multiple(3)
               }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The current state of the MAC address search on this
            repeater. This object is initialized to 'none' when
            the corresponding instance of rptrAddrSearchAddress
            is set. If the agent detects the address on exactly
            one port, it will set this object to 'single', and
            set the corresponding instances of
            rptrAddrSearchGroup and rptrAddrSearchPort to reflect
            the group and port on which the address was heard.
            If the agent detects the address on more than one
            port, it will set this object to 'multiple'."
```



```
    ::= { rptrAddrSearchEntry 4 }

rptrAddrSearchGroup OBJECT-TYPE
    SYNTAX     Integer32 (0..2147483647)
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The group from which an error-free frame whose
            source address is equal to the corresponding instance
            of rptrAddrSearchAddress has been received. The
            value of this object is undefined when the
            corresponding instance of rptrAddrSearchState is
            equal to 'none' or 'multiple'."
    ::= { rptrAddrSearchEntry 5 }

rptrAddrSearchPort OBJECT-TYPE
    SYNTAX     Integer32 (0..2147483647)
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The port rom which an error-free frame whose
            source address is equal to the corresponding instance
            of rptrAddrSearchAddress has been received. The
            value of this object is undefined when the
            corresponding instance of rptrAddrSearchState is
            equal to 'none' or 'multiple'."
    ::= { rptrAddrSearchEntry 6 }

rptrAddrSearchOwner OBJECT-TYPE
    SYNTAX     OwnerString
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
            "The entity which currently has 'ownership' of this
            rptrAddrSearchEntry."
    ::= { rptrAddrSearchEntry 7 }


--
-- The Port Address Tracking Table
--
-- This table provides a way for a network management
-- application to passively gather information (using
-- read-only privileges) about which network addresses
-- are connected to which ports of a repeater.
--

rptrAddrTrackTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrAddrTrackEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Table of address mapping information about the
            ports."
    ::= { rptrAddrTrackPortInfo 1 }

rptrAddrTrackEntry OBJECT-TYPE
    SYNTAX      RptrAddrTrackEntry
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION
            "An entry in the table, containing address mapping
            information about a single port."
    INDEX    { rptrAddrTrackGroupIndex, rptrAddrTrackPortIndex }
    ::= { rptrAddrTrackTable 1 }

RptrAddrTrackEntry ::=
    SEQUENCE {
        rptrAddrTrackGroupIndex
            INTEGER,
        rptrAddrTrackPortIndex
            INTEGER,
        rptrAddrTrackSourceAddrChanges
            Counter32,
        rptrAddrTrackNewLastSrcAddress
            OptMacAddr,
        rptrAddrTrackCapacity
            Integer32
    }

rptrAddrTrackGroupIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the group containing the
            port for which this entry contains information."
    ::= { rptrAddrTrackEntry 1 }

rptrAddrTrackPortIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This object identifies the port within the group
            for which this entry contains information."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.1, aPortID."
    ::= { rptrAddrTrackEntry 2 }

rptrAddrTrackSourceAddrChanges OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This counter is incremented by one for each time
            that the rptrAddrTrackNewLastSrcAddress attribute
            for this port has changed.

            This may indicate whether a link is connected to a
            single DTE or another multi-user segment.

            A discontinuity may occur in the value when the
            value of object rptrMonitorPortLastChange changes.

            The approximate minimum time for rollover of this
            counter is 81 hours in a 10 Mb/s repeater."
    REFERENCE
```



```
            "IEEE Std 802.3, 30.4.3.1.19, aSourceAddressChanges."
    ::= { rptrAddrTrackEntry 3 }

rptrAddrTrackNewLastSrcAddress OBJECT-TYPE
    SYNTAX      OptMacAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object is the SourceAddress of the last
            readable frame (i.e., counted by
            rptrMonitorPortReadableFrames) received by this
            port. If no frames have been received by this
            port since the agent began monitoring the port
            activity, the agent shall return a string of
            length zero."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.18, aLastSourceAddress."
    ::= { rptrAddrTrackEntry 4 }

rptrAddrTrackCapacity OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The maximum number of addresses that can be
            detected on this port. This value indicates
            to the maximum  number of entries in the
            rptrExtAddrTrackTable relative to this port.

            If this object has the value of 1, the agent
            implements only the LastSourceAddress mechanism
            described by RFC 1368 or RFC 1516."
    ::= { rptrAddrTrackEntry 5 }


-- Table for multiple addresses per port

rptrExtAddrTrackTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrExtAddrTrackEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table to extend the address tracking table (i.e.,
            rptrAddrTrackTable) with a list of source MAC
            addresses that were recently received on each port.
            The number of ports is the same as the number
            of entries in table rptrPortTable. The number of
            entries in this table depends on the agent/repeater
            implementation and the number of different
            addresses received on each port.

            The first entry for each port contains
            the same MAC address that is given by the
            rptrAddrTrackNewLastSrcAddress for that port.

            Entries in this table for a particular port are
            retained when that port is switched from one
            repeater to another.
```

```
            The ordering of MAC addresses listed for a
            particular port is implementation dependent."
    ::= { rptrAddrTrackPortInfo 2 }

rptrExtAddrTrackEntry OBJECT-TYPE
    SYNTAX      RptrExtAddrTrackEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A row in the table of extended address tracking
            information for ports. Entries can not be directly
            created or deleted via SNMP operations."
    INDEX       { rptrAddrTrackGroupIndex,
                  rptrAddrTrackPortIndex,
                  rptrExtAddrTrackMacIndex }
    ::= { rptrExtAddrTrackTable 1 }

RptrExtAddrTrackEntry ::= SEQUENCE {
    rptrExtAddrTrackMacIndex Integer32,
    rptrExtAddrTrackSourceAddress MacAddress
    }

rptrExtAddrTrackMacIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The index of a source MAC address seen on
            the port.

            The ordering of MAC addresses listed for a
            particular port is implementation dependent.

            There is no implied relationship between a
            particular index and a particular MAC
            address. The index for a particular MAC
            address may change without notice."
    ::= { rptrExtAddrTrackEntry 1 }

rptrExtAddrTrackSourceAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The source MAC address from a readable frame
            (i.e., counted by rptrMonitorPortReadableFrames)
            recently received by the port."
    REFERENCE
            "IEEE Std 802.3, 30.4.3.1.18, aLastSourceAddress."
    ::= { rptrExtAddrTrackEntry 2 }


-- The Repeater Top "N" Port Group
-- The Repeater Top N Port group is used to prepare reports that
-- describe a list of ports ordered by one of the statistics in the
-- Repeater Monitor Port Table. The statistic chosen by the
-- management station is sampled over a management
-- station-specified time interval, making the report rate based.
-- The management station also specifies the number of ports that
```

```
-- are reported.
--
-- The rptrTopNPortControlTable is used to initiate the generation
-- of a report. The management station may select the parameters
-- of such a report, such as which repeater, which statistic, how
-- many ports, and the start & stop times of the sampling. When
-- the report is prepared, entries are created in the
-- rptrTopNPortTable associated with the relevent
-- rptrTopNControlEntry. These entries are static for
-- each report after it has been prepared.

-- Note that counter discontinuities may appear in some
-- implementations if ports' assignment to repeaters changes
-- during the collection of data for a Top "N" report.
-- A management application could read the corresponding
-- rptrMonitorPortLastChange timestamp in order to check
-- whether a discontinuity occurred.


rptrTopNPortControlTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrTopNPortControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of control records for reports on the top 'N'
        ports for the rate of a selected counter. The number
        of entries depends on the configuration of the agent.
        The maximum number of entries is implementation
        dependent."
    ::= { rptrTopNPortInfo 1 }

rptrTopNPortControlEntry OBJECT-TYPE
    SYNTAX      RptrTopNPortControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A set of parameters that control the creation of a
            report of the top N ports according to several metrics."
    INDEX    { rptrTopNPortControlIndex }
    ::= { rptrTopNPortControlTable 1 }

RptrTopNPortControlEntry ::= SEQUENCE {
    rptrTopNPortControlIndex
        Integer32,
    rptrTopNPortRepeaterId
        Integer32,
    rptrTopNPortRateBase
        INTEGER,
    rptrTopNPortTimeRemaining
        Integer32,
    rptrTopNPortDuration
        Integer32,
    rptrTopNPortRequestedSize
        Integer32,
    rptrTopNPortGrantedSize
        Integer32,
    rptrTopNPortStartTime
        TimeStamp,
    rptrTopNPortOwner
```

```
            OwnerString,
        rptrTopNPortRowStatus
            RowStatus
    }

    rptrTopNPortControlIndex OBJECT-TYPE
        SYNTAX      Integer32 (1 .. 65535)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "An index that uniquely identifies an entry in the
                rptrTopNPortControl table. Each such entry defines
                one top N report prepared for a repeater or repeater system."
        ::= { rptrTopNPortControlEntry 1 }

    rptrTopNPortRepeaterId OBJECT-TYPE
        SYNTAX      Integer32 (0..2147483647)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
                "Identifies the repeater for which a top N report will
                be prepared (see rptrInfoId). If the value of this
                object is positive, only ports assigned to this repeater
                will be used to form the list in which to order the
                Top N table. If this value is zero, all ports will be
                eligible for inclusion on the list.

                The value of this object may not be modified if the
                associated rptrTopNPortRowStatus object is equal to
                active(1).
                If, for a particular row in this table, the repeater
                specified by the value of this object goes away (is
                removed from the rptrInfoTable) while the associated
                rptrTopNPortRowStatus object is equal to active(1),
                the row in this table is preserved by the agent but
                the value of rptrTopNPortRowStatus is changed to
                notInService(2), and the agent may time out the row
                if appropriate. If the specified repeater comes
                back (reappears in the rptrInfoTable) before the row
                has been timed out, the management station sets
                the value of the rptrTopNPortRowStatus object back
                to active(1) if desired (the agent doesn’t do this
                automatically)."
        ::= { rptrTopNPortControlEntry 2 }

    rptrTopNPortRateBase OBJECT-TYPE
        SYNTAX      INTEGER  {
                      readableFrames(1),
                      readableOctets(2),
                      fcsErrors(3),
                      alignmentErrors(4),
                      frameTooLongs(5),
                      shortEvents(6),
                      runts(7),
                      collisions(8),
                      lateEvents(9),
                      veryLongEvents(10),
                      dataRateMismatches(11),
                      autoPartitions(12),
```



```
                    totalErrors(13),
                    isolates(14),
                    symbolErrors(15)
                  }
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
                "The monitored variable, which the rptrTopNPortRate
                variable is based upon.

                The value of this object may not be modified if
                the associated rptrTopNPortRowStatus object has
                a value of active(1)."
        ::= { rptrTopNPortControlEntry 3 }

    rptrTopNPortTimeRemaining OBJECT-TYPE
        SYNTAX      Integer32 (0..2147483647)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
                "The number of seconds left in the report
                currently being collected. When this object
                is modified by the management station, a new
                collection is started, possibly aborting a
                currently running report. The new value is
                used as the requested duration of this report,
                which is loaded into the associated
                rptrTopNPortDuration object.

                When this object is set to a non-zero value,
                any associated rptrTopNPortEntries shall be
                made inaccessible by the agent. While the value
                of this object is non-zero, it decrements by one
                per second until it reaches zero. During this
                time, all associated rptrTopNPortEntries shall
                remain inaccessible. At the time that this object
                decrements to zero, the report is made accessible
                in the rptrTopNPortTable. Thus, the rptrTopNPort
                table needs to be created only at the end of the
                collection interval.

                If the value of this object is set to zero
                while the associated report is running, the
                running report is aborted and no associated
                rptrTopNPortEntries are created."
        DEFVAL { 0 }
        ::= { rptrTopNPortControlEntry 4 }

    rptrTopNPortDuration OBJECT-TYPE
        SYNTAX      Integer32 (0..2147483647)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The number of seconds that this report has
                collected during the last sampling interval,
                or if this report is currently being collected,
                the number of seconds that this report is being
                collected during this sampling interval.
```



```
            When the associated rptrTopNPortTimeRemaining
            object is set, this object shall be set by the
            agent to the same value and shall not be modified
            until the next time the rptrTopNPortTimeRemaining
            is set.

            This value shall be zero if no reports have been
            requested for this rptrTopNPortControlEntry."
      ::= { rptrTopNPortControlEntry 5 }

rptrTopNPortRequestedSize OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The maximum number of repeater ports requested
            for the Top N Table.

            When this object is created or modified, the
            agent should set rptrTopNPortGrantedSize as close
            to this object as is possible for the particular
            implementation and available resources."
    DEFVAL { 10 }
    ::= { rptrTopNPortControlEntry 6 }

rptrTopNPortGrantedSize OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The maximum number of repeater ports in the
            top N table.

            When the associated rptrTopNPortRequestedSize object is
            created or modified, the agent should set this object as
            closely to the requested value as is possible for the
            particular implementation and available resources. The
            agent shall not lower this value except as a result of a
            set to the associated rptrTopNPortRequestedSize object."
    ::= { rptrTopNPortControlEntry 7 }

rptrTopNPortStartTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The value of sysUpTime when this top N report was
            last started. In other words, this is the time that
            the associated rptrTopNPortTimeRemaining object was
            modified to start the requested report.

            If the report has not yet been started, the value
            of this object is zero."
    ::= { rptrTopNPortControlEntry 8 }

rptrTopNPortOwner OBJECT-TYPE
    SYNTAX      OwnerString
    MAX-ACCESS  read-create
    STATUS      current
```



```
    DESCRIPTION
            "The entity that configured this entry and is
            using the resources assigned to it."
    ::= { rptrTopNPortControlEntry 9 }

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

           If the value of this object is not equal to
           active(1), all associated entries in the
           rptrTopNPortTable shall be deleted by the
           agent."
    ::= { rptrTopNPortControlEntry 10 }


-- Top "N" reports

rptrTopNPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RptrTopNPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table of reports for the top 'N' ports based on
            setting of associated control table entries. The
            maximum number of entries depends on the number
            of entries in table rptrTopNPortControlTable and
            the value of object rptrTopNPortGrantedSize for
            each entry.

            For each entry in the rptrTopNPortControlTable,
            repeater ports with the highest value of
            rptrTopNPortRate shall be placed in this table
            in decreasing order of that rate until there is
            no more room or until there are no more ports."
    ::= { rptrTopNPortInfo 2 }

rptrTopNPortEntry OBJECT-TYPE
    SYNTAX      RptrTopNPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A set of statistics for a repeater port that is
            part of a top N report."
    INDEX     { rptrTopNPortControlIndex,
                rptrTopNPortIndex }
    ::= { rptrTopNPortTable 1 }

RptrTopNPortEntry ::= SEQUENCE {
    rptrTopNPortIndex
        Integer32,
    rptrTopNPortGroupIndex
        Integer32,
    rptrTopNPortPortIndex
        Integer32,
    rptrTopNPortRate
```

```
        Gauge32
}

rptrTopNPortIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An index that uniquely identifies an entry in
            the rptrTopNPort table among those in the same
            report. This index is between 1 and N, where N
            is the number of entries in this report. Increasing
            values of rptrTopNPortIndex shall be assigned to
            entries with decreasing values of rptrTopNPortRate
            until index N is assigned to the entry with the
            lowest value of rptrTopNPortRate or there are no
            more rptrTopNPortEntries.

            No ports are included in a report where their
            value of rptrTopNPortRate would be zero."
    ::= { rptrTopNPortEntry 1 }

rptrTopNPortGroupIndex OBJECT-TYPE
    SYNTAX      Integer32  (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object identifes the group containing
            the port for this entry. (See also object
            type rptrGroupIndex.)"
    ::= { rptrTopNPortEntry 2 }

rptrTopNPortPortIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The index of the repeater port.
        (See object type rptrPortIndex.)"
    ::= { rptrTopNPortEntry 3 }

rptrTopNPortRate OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The amount of change in the selected variable
            during this sampling interval for the identified
            port. The selected variable is that port’s
            instance of the object selected by
            rptrTopNPortRateBase."
    ::= { rptrTopNPortEntry 4 }



-- Notifications for use by Repeaters
-- Notifications for repeaters in a multiple-repeater implementation.
-- An implementation may send either the single-repeater OR
-- multiple-repeater version of these notifications (1 or 4; 2 or 5)
```



```
-- but not both.

ieee8023snmpDot3RptrNotifications OBJECT IDENTIFIER
            ::= {ieee8023snmpDot3RptrMgt 0}

rptrInfoHealth NOTIFICATION-TYPE
    OBJECTS     { rptrInfoOperStatus }
    STATUS      current
    DESCRIPTION
            "In a repeater system containing multiple managed repeaters,
            the rptrInfoHealth notification conveys information
            related to the operational status of a repeater.
            It is sent either when the value of rptrInfoOperStatus
            changes, or upon completion of a non-disruptive test.

            The agent shall limit the generation of
            consecutive rptrInfoHealth notifications for
            the same repeater so that there is at least
            a five-second gap between notifications of this type.
            When notifications are throttled, they are dropped,
            not queued for sending at a future time. (Note
            that ‘generating’ a notification means sending
            to all configured recipients.)"
    REFERENCE
            "IEEE Std 802.3, 30.4.1.3.1, nRepeaterHealth
            notification."
    ::= { ieee8023snmpDot3RptrNotifications 4 }

rptrInfoResetEvent NOTIFICATION-TYPE
    OBJECTS     { rptrInfoOperStatus }
    STATUS      current
    DESCRIPTION
            "In a repeater system containing multiple managed
            repeaters, the rptrInfoResetEvent notification
            conveys information related to the operational
            status of a repeater. This notification is sent
            on completion of a repeater reset action. A
            repeater reset action is defined as a transition
            to the START state of Figure 9-2 in Clause 9 of
            IEEE Std 802.3, when triggered by a management
            command (e.g., an SNMP Set on the rptrInfoReset
            object).

            The agent shall limit the generation of
            consecutive rptrInfoResetEvent notifications for
            a single repeater so that there is at least
            a five-second gap between notifications of
            this type. When notifications are throttled,
            they are dropped, not queued for sending at
            a future time. (Note that ‘generating’ a
            notification means sending to all configured
            recipients.)

            The rptrInfoResetEvent is not sent when the
            agent restarts and sends an SNMP coldStart or
            warmStart trap. However, it is recommended that
            a repeater agent send the rptrInfoOperStatus
            object as an optional object with its coldStart
            and warmStart trap PDUs."
```

```
    REFERENCE
            "IEEE Std 802.3, 30.4.1.3.2, nRepeaterReset
            notification."
    ::= { ieee8023snmpDot3RptrNotifications 5 }


-- Conformance statements

snmpRptrModConf
        OBJECT IDENTIFIER ::= { ieee8023snmpRptrMIB 2 }
  snmpRptrModCompls
        OBJECT IDENTIFIER ::= { snmpRptrModConf 1 }
  snmpRptrModObjGrps
        OBJECT IDENTIFIER ::= { snmpRptrModConf 2 }
  snmpRptrModNotGrps
        OBJECT IDENTIFIER ::= { snmpRptrModConf 3 }


-- Object groups

snmpRptrGrpBasic OBJECT-GROUP
    OBJECTS     { rptrGroupObjectID,
                  rptrGroupOperStatus,
                  rptrGroupPortCapacity,
                  rptrPortAdminStatus,
                  rptrPortAutoPartitionState,
                  rptrPortOperStatus,
                  rptrPortRptrId,
                  rptrInfoRptrType,
                  rptrInfoOperStatus,
                  rptrInfoReset,
                  rptrInfoPartitionedPorts,
                  rptrInfoLastChange }
    STATUS      current
    DESCRIPTION
        "Basic group for a repeater system with one or more
        repeater-units in multi-segment (post-RFC 1516)
        version of the MIB module."
    ::= { snmpRptrModObjGrps 1 }

snmpRptrGrpMonitor OBJECT-GROUP
    OBJECTS     { rptrMonitorPortReadableFrames,
                  rptrMonitorPortReadableOctets,
                  rptrMonitorPortFCSErrors,
                  rptrMonitorPortAlignmentErrors,
                  rptrMonitorPortFrameTooLongs,
                  rptrMonitorPortShortEvents,
                  rptrMonitorPortRunts,
                  rptrMonitorPortCollisions,
                  rptrMonitorPortLateEvents,
                  rptrMonitorPortVeryLongEvents,
                  rptrMonitorPortDataRateMismatches,
                  rptrMonitorPortAutoPartitions,
                  rptrMonitorPortTotalErrors,
                  rptrMonitorPortLastChange,

                  rptrMonTxCollisions,
                  rptrMonTotalFrames,
                  rptrMonTotalErrors,
```



```
                 rptrMonTotalOctets }
    STATUS      current
    DESCRIPTION
        "Monitor group for a repeater system with one or more
        repeater-units in multi-segment (post-RFC 1516)
        version of the MIB module."
    ::= { snmpRptrModObjGrps 2 }

snmpRptrGrpMonitor100 OBJECT-GROUP
    OBJECTS     { rptrMonitorPortIsolates,
                  rptrMonitorPortSymbolErrors,
                  rptrMonitorPortUpper32Octets,

                  rptrMonUpper32TotalOctets }
    STATUS      current
    DESCRIPTION
        "Monitor group for 100 Mb/s ports and repeaters
        in a repeater system with one or more repeater-units in
        multi-segment (post-RFC 1516) version of the MIB
        module. Repeater systems which support Counter64 should
        also implement snmpRptrGrpMonitor100w64."
    ::= { snmpRptrModObjGrps 3 }

snmpRptrGrpMonitor100w64 OBJECT-GROUP
    OBJECTS     { rptrMonitorPortHCReadableOctets,
                  rptrMonHCTotalOctets }
    STATUS      current
    DESCRIPTION
        "Monitor group for 100 Mb/s ports and repeaters in a
        repeater system with one or more repeater-units and support
        for Counter64."
    ::= { snmpRptrModObjGrps 4 }

snmpRptrGrpAddrTrack OBJECT-GROUP
    OBJECTS     { rptrAddrTrackSourceAddrChanges,
                  rptrAddrTrackNewLastSrcAddress,
                  rptrAddrTrackCapacity }
    STATUS      current
    DESCRIPTION
        "Passive address tracking group for post-RFC 1516
        version of the MIB module."
    ::= { snmpRptrModObjGrps 5 }

snmpRptrGrpExtAddrTrack OBJECT-GROUP
    OBJECTS     { rptrExtAddrTrackSourceAddress }
    STATUS      current
    DESCRIPTION
        "Extended passive address tracking group for
        a repeater system with one or more repeater-units in
        post-RFC 1516 version of the MIB module."
    ::= { snmpRptrModObjGrps 6 }

snmpRptrGrpRptrAddrSearch OBJECT-GROUP
    OBJECTS     { rptrAddrSearchLock,
                  rptrAddrSearchStatus,
                  rptrAddrSearchAddress,
                  rptrAddrSearchState,
                  rptrAddrSearchGroup,
                  rptrAddrSearchPort,
```

```
                 rptrAddrSearchOwner }
    STATUS       current
    DESCRIPTION
        "Active MAC address search group and topology
        mapping support for repeaters."
    ::= { snmpRptrModObjGrps 7 }

snmpRptrGrpTopNPort OBJECT-GROUP
    OBJECTS      { rptrTopNPortRepeaterId,
                   rptrTopNPortRateBase,
                   rptrTopNPortTimeRemaining,
                   rptrTopNPortDuration,
                   rptrTopNPortRequestedSize,
                   rptrTopNPortGrantedSize,
                   rptrTopNPortStartTime,
                   rptrTopNPortOwner,
                   rptrTopNPortRowStatus,
                   rptrTopNPortGroupIndex,
                   rptrTopNPortPortIndex,
                   rptrTopNPortRate }
    STATUS       current
    DESCRIPTION
        "Top 'N' group for repeater ports."
    ::= { snmpRptrModObjGrps 8 }

ieee8023snmpDot3RptrNotGroup NOTIFICATION-GROUP
    NOTIFICATIONS { rptrInfoHealth,
                    rptrInfoResetEvent }
    STATUS       current
    DESCRIPTION
        "Conformance Group for repeater notifications.
         Formerly an empty group."
    ::= {snmpRptrModNotGrps 1}


-- Compliance statements

snmpRptrModCompl MODULE-COMPLIANCE
    STATUS       current
    DESCRIPTION
        "Compliance for the multi-segment version of the
        MIB module for a repeater system with one or more
        repeater-units."

    MODULE -- this module
        MANDATORY-GROUPS { snmpRptrGrpBasic,
                           snmpRptrGrpMonitor,
                           snmpRptrGrpAddrTrack }

        GROUP snmpRptrGrpMonitor100
        DESCRIPTION
            "Implementation of this group is
            mandatory for managed repeater systems which
            contain 100 Mb/s repeaters."

        GROUP snmpRptrGrpMonitor100w64
        DESCRIPTION
            "Implementation of this group is
            mandatory for managed repeater systems which
```



```
            contain 100 Mb/s repeaters and which
            can support Counter64."

        GROUP snmpRptrGrpExtAddrTrack
        DESCRIPTION
            "Implementation of this group is
            recommended for repeater systems which have
            the necessary instrumentation to track
            MAC addresses of multiple DTEs attached
            to a single repeater port."

        GROUP snmpRptrGrpRptrAddrSearch
        DESCRIPTION
            "Implementation of this group is
            recommended for repeater systems which allow
            read-write access and which have
            the necessary instrumentation to
            search all incoming data streams
            for a particular MAC address."

        GROUP snmpRptrGrpTopNPort
        DESCRIPTION
            "Implementation of this group is
            recommended for repeater systems which have
            the necessary resources to support
            TopN statistics reporting."

        GROUP ieee8023snmpDot3RptrNotGroup
        DESCRIPTION
            "Implementation of this group is
             recommended for repeaters that
             support notifications."

    ::= { snmpRptrModCompls 1 }

END
```

8. Ethernet data terminal equipment (DTE) power via medium dependent interface (MDI) MIB module

8.1 Introduction

This clause defines a portion of the Management Information Base (MIB) for use with SNMP. In particular, it defines a set of MIB objects to manage Power via MDI Power Sourcing Equipment (PSE).

8.2 Overview

IEEE Std 802.3 defines the hardware registers that will allow for management interfaces to be built for a powered Ethernet device. The MIB module defined in this clause extends the Ethernet-like interface MIB defined in Clause 10 with the management objects required for the management of the powered Ethernet devices and ports.

8.3 MIB structure

These MIB objects are categorized into three MIB groups.

The pethPsePortTable defines the objects used for configuring and describing the status of ports on a PSE device. Examples of PSE devices are Ethernet switches that support power Ethernet and mid-span boxes.

The pethMainPseObjects MIB group defines the management objects for a managed main power source in a PSE device. Ethernet switches are one example of boxes that would support these objects.

The pethNotificationControlTable includes objects that control the transmission of notifications from the agent to a management application.

8.4 Security considerations for Ethernet data terminal equipment (DTE) power via medium dependent interface (MDI) MIB module.

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write. 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.

Setting the following object to incorrect values can result in improper operation of the PSE, including the possibility that the PD does not receive power from the PSE port:

— pethPsePortAdminEnable
— pethPsePortPowerPairs
— pethPsePortPowerPriority
— pethPsePortType

Setting the following objects to incorrect values can result in an excessive number of traps being sent to network management stations:

— pethMainPseUsageThreshold
— pethNotificationControlEnable

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. These are as follows:

— pethPsePortPowerPairsControlAbility
— pethPsePortPowerPriority
— pethPsePortPowerClassifications

It is thus important to control GET and/or NOTIFY access to these objects and possibly to encrypt their values when sending them over the network via SNMP.

8.5 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL:[14]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C8mib.txt

[14]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.


```
IEEE8023-POWER-ETHERNET-MIB DEFINITIONS ::= BEGIN

IMPORTS
          MODULE-IDENTITY, OBJECT-TYPE, Integer32,
          Gauge32, Counter32, NOTIFICATION-TYPE, org
                  FROM SNMPv2-SMI
          TruthValue
                  FROM SNMPv2-TC
          MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
                  FROM SNMPv2-CONF

           SnmpAdminString
                   FROM SNMP-FRAMEWORK-MIB;

    ieee8023powerEthernetMIB MODULE-IDENTITY

        LAST-UPDATED "201102020000Z" -- February 2, 2011
        ORGANIZATION
          "IEEE 802.3 working group"
        CONTACT-INFO
            "WG-URL: http://www.ieee802.org/3/index.html
            WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

            Contact: Howard Frazier
            Postal:  3151 Zanker Road
                     San Jose, CA 95134
                     USA
            Tel:     +1.408.922.8164
            E-mail:  hfrazier@broadcom.com"

        DESCRIPTION
              "The MIB module for managing Power Source Equipment
               (PSE) working according to the IEEE 802.af Powered
               Ethernet (DTE Power via MDI) standard."

      REVISION    "201102020000Z" -- February 2, 2011
      DESCRIPTION
            "Initial version, based on an earlier version published as RFC 3621."

          ::= { org ieee(111) standards-association-numbers-series-standards(2)
                lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 8 }

pethNotifications OBJECT IDENTIFIER ::= { ieee8023powerEthernetMIB 0 }
pethObjects       OBJECT IDENTIFIER ::= { ieee8023powerEthernetMIB 1 }
pethConformance   OBJECT IDENTIFIER ::= { ieee8023powerEthernetMIB 2 }

-- PSE Objects

  pethPsePortTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF PethPsePortEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table of objects that display and control the power
            characteristics of power Ethernet ports on a Power Source
            Entity (PSE) device. This group will be implemented in
            managed power Ethernet switches and mid-span devices.
            Values of all read-write objects in this table are
            persistent at restart/reboot."
```

```
      ::= { pethObjects 1 }

  pethPsePortEntry OBJECT-TYPE
      SYNTAX      PethPsePortEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
              "A set of objects that display and control the power
              characteristics of a power Ethernet PSE port."
      INDEX    { pethPsePortGroupIndex , pethPsePortIndex  }
      ::= { pethPsePortTable 1 }

  PethPsePortEntry ::= SEQUENCE {
      pethPsePortGroupIndex
         Integer32,
      pethPsePortIndex
         Integer32,
      pethPsePortAdminEnable
         TruthValue,
      pethPsePortPowerPairsControlAbility
         TruthValue,
      pethPsePortPowerPairs
         INTEGER,
      pethPsePortDetectionStatus
         INTEGER,
      pethPsePortPowerPriority
         INTEGER,

      pethPsePortMPSAbsentCounter
         Counter32,
      pethPsePortType
         SnmpAdminString,
      pethPsePortPowerClassifications
         INTEGER,
      pethPsePortInvalidSignatureCounter
         Counter32,
      pethPsePortPowerDeniedCounter
         Counter32,
      pethPsePortOverLoadCounter
         Counter32,
      pethPsePortShortCounter
         Counter32
 }

    pethPsePortGroupIndex OBJECT-TYPE
      SYNTAX      Integer32 (1..2147483647)
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "This variable uniquely identifies the group
           containing the port to which a power Ethernet PSE is
           connected. Group means box in the stack, module in a
           rack and the value 1 shall be used for non-modular devices.
           Furthermore, the same value shall be used in this variable,
           pethMainPseGroupIndex, and pethNotificationControlGroupIndex
           to refer to a given box in a stack or module in the rack."
      ::= { pethPsePortEntry 1 }

    pethPsePortIndex OBJECT-TYPE
```



```
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This variable uniquely identifies the power Ethernet PSE
         port within group pethPsePortGroupIndex to which the
         power Ethernet PSE entry is connected."
    ::= { pethPsePortEntry 2 }

 pethPsePortAdminEnable OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "true (1) An interface which can provide the PSE functions.
     false(2) The interface will act as it would if it had no PSE
     function."

REFERENCE
  "IEEE Std 802.3, 30.9.1.1.2 aPSEAdminState"
::= { pethPsePortEntry 3 }

 pethPsePortPowerPairsControlAbility OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Describes the capability of controlling the power pairs
     functionality to switch pins for sourcing power.
     The value true indicate that the device has the capability
     to control the power pairs. When false the PSE Pinout
     Alternative used cannot be controlled through the
     PethPsePortAdminEnable attribute."
REFERENCE
  "IEEE Std 802.3, 30.9.1.1.3
   aPSEPowerPairsControlAbility"
::= { pethPsePortEntry 4 }

pethPsePortPowerPairs OBJECT-TYPE
SYNTAX INTEGER   {
           signal(1),
           spare(2)
 }
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "Describes or controls the pairs in use. If the value of
     pethPsePortPowerPairsControl is true, this object is
     writable.
     A value of signal(1) means that the signal pairs
     only are in use.
     A value of spare(2) means that the spare pairs
     only are in use."
REFERENCE
  "IEEE Std 802.3, 30.9.1.1.4 aPSEPowerPairs"
::= { pethPsePortEntry 5 }

 pethPsePortDetectionStatus OBJECT-TYPE
SYNTAX INTEGER   {
```

```
          disabled(1),
          searching(2),
            deliveringPower(3),
            fault(4),
            test(5),
            otherFault(6)
    }

   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
       "Describes the operational status of the port PD detection.
        A value of disabled(1)- indicates that the PSE State diagram
        is in the state DISABLED.
        A value of deliveringPower(3) - indicates that the PSE State
        diagram is in the state POWER_ON for a duration greater than
        tlim max (see IEEE Std 802.3af Table 33-5 tlim).
        A value of fault(4) - indicates that the PSE State diagram is
        in the state TEST_ERROR.
        A value of test(5) - indicates that the PSE State diagram is
        in the state TEST_MODE.
        A value of otherFault(6) - indicates that the PSE State
        diagram is in the state IDLE due to the variable
        error_conditions.
        A value of searching(2)- indicates the PSE State diagram is
        in a state other than those listed above."
   REFERENCE
     "IEEE Std 802.3, 30.9.1.1.5
      aPSEPowerDetectionStatus"
   ::= { pethPsePortEntry 6 }

    pethPsePortPowerPriority OBJECT-TYPE
   SYNTAX INTEGER   {
            critical(1),
            high(2),
            low(3)
    }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
       "This object controls the priority of the port from the point
        of view of a power management algorithm. The priority that
        is set by this variable could be used by a control mechanism
        that prevents over current situations by disconnecting first
        ports with lower power priority. Ports that connect devices
        critical to the operation of the network - like the E911
        telephones ports - should be set to higher priority."
   ::= { pethPsePortEntry 7 }

  pethPsePortMPSAbsentCounter OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
         "This counter is incremented when the PSE state diagram
            transitions directly from the state POWER_ON to the

            state IDLE due to tmpdo_timer_done being asserted."
   REFERENCE
```



```
      "IEEE Std 802.3, 30.9.1.1.11
       aPSEMPSAbsentCounter"
    ::= { pethPsePortEntry 8 }

   pethPsePortType OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "A manager will set the value of this variable to indicate
         the type of powered device that is connected to the port.
         The default value supplied by the agent if no value has
         ever been set should be a zero-length octet string."
    ::= { pethPsePortEntry 9 }

    pethPsePortPowerClassifications OBJECT-TYPE
     SYNTAX INTEGER   {
               class0(1),
               class1(2),
               class2(3),
               class3(4),
               class4(5)
     }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Classification is a way to tag different terminals on the
        Power over LAN network according to their power consumption.
        Devices such as IP telephones, WLAN access points and others,
        will be classified according to their power requirements.

        The meaning of the classification labels is defined in the
        IEEE specification.

       This variable is valid only while a PD is being powered,
        that is, while the attribute pethPsePortDetectionStatus
        is reporting the enumeration deliveringPower."
    REFERENCE
      "IEEE Std 802.3, 30.9.1.1.6
       aPSEPowerClassification"
   ::= { pethPsePortEntry 10 }

   pethPsePortInvalidSignatureCounter OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current

    DESCRIPTION
       "This counter is incremented when the PSE state diagram
         enters the state SIGNATURE_INVALID."
    REFERENCE
          "IEEE Std 802.3, 30.9.1.1.7
           aPSEInvalidSignatureCounter"
    ::= { pethPsePortEntry 11 }

   pethPsePortPowerDeniedCounter OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION
          "This counter is incremented when the PSE state diagram
             enters the state POWER_DENIED."
    REFERENCE
      "IEEE Std 802.3, 30.9.1.1.8
       aPSEPowerDeniedCounter"
    ::= { pethPsePortEntry 12 }

   pethPsePortOverLoadCounter OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
          "This counter is incremented when the PSE state diagram
             enters the state ERROR_DELAY_OVER."
    REFERENCE
      "IEEE Std 802.3, 30.9.1.1.9
       aPSEOverLoadCounter"
    ::= { pethPsePortEntry 13 }

   pethPsePortShortCounter OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
          "This counter is incremented when the PSE state diagram
             enters the state ERROR_DELAY_SHORT."
    REFERENCE
      "IEEE Std 802.3, 30.9.1.1.10
       aPSEShortCounter"
    ::= { pethPsePortEntry 14 }

-- Main PSE Objects

pethMainPseObjects       OBJECT IDENTIFIER ::= { pethObjects 3 }

pethMainPseTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF PethMainPseEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "A table of objects that display and control attributes
            of the main power source in a PSE  device. Ethernet
            switches are one example of boxes that would support
            these objects.
            Values of all read-write objects in this table are
            persistent at restart/reboot."
       ::= { pethMainPseObjects 1 }

   pethMainPseEntry OBJECT-TYPE
       SYNTAX      PethMainPseEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
            "A set of objects that display and control the Main
             power of a PSE. "
       INDEX    { pethMainPseGroupIndex }
       ::= { pethMainPseTable 1 }
```

```
PethMainPseEntry ::= SEQUENCE {
    pethMainPseGroupIndex
        Integer32,
    pethMainPsePower
        Gauge32 ,
    pethMainPseOperStatus
        INTEGER,
    pethMainPseConsumptionPower
        Gauge32,
    pethMainPseUsageThreshold
        Integer32
}
  pethMainPseGroupIndex OBJECT-TYPE
    SYNTAX       Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This variable uniquely identifies the group to which
        power Ethernet PSE is connected. Group means (box in
        the stack, module in a rack) and the value 1 shall be
        used for non-modular devices. Furthermore, the same
        value shall be used in this variable, pethPsePortGroupIndex,
        and pethNotificationControlGroupIndex to refer to a
        given box in a stack or module in a rack."
    ::= { pethMainPseEntry 1 }

  pethMainPsePower OBJECT-TYPE
    SYNTAX      Gauge32  (1..65535)
    UNITS      "Watts"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The nominal power of the PSE expressed in Watts."
    ::= { pethMainPseEntry 2 }

  pethMainPseOperStatus OBJECT-TYPE
    SYNTAX INTEGER   {
            on(1),
            off(2),
            faulty(3)
       }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The operational status of the main PSE."
    ::= { pethMainPseEntry 3 }

  pethMainPseConsumptionPower OBJECT-TYPE
    SYNTAX      Gauge32
    UNITS      "Watts"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "Measured usage power expressed in Watts."
    ::= { pethMainPseEntry 4 }

  pethMainPseUsageThreshold OBJECT-TYPE
    SYNTAX      Integer32  (1..99)
    UNITS      "%"
```

```
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
               "The usage threshold expressed in percents for
                comparing the measured power and initiating
                an alarm if the threshold is exceeded."
       ::= { pethMainPseEntry 5 }

-- Notification  Control  Objects

pethNotificationControl        OBJECT IDENTIFIER ::= { pethObjects 4 }

pethNotificationControlTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF PethNotificationControlEntry
       MAX-ACCESS  not-accessible

       STATUS      current
       DESCRIPTION
           "A table of objects that display and control the
            Notification on a PSE  device.
            Values of all read-write objects in this table are
            persistent at restart/reboot."
       ::= { pethNotificationControl 1 }

   pethNotificationControlEntry OBJECT-TYPE
       SYNTAX      PethNotificationControlEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
            "A set of objects that control the Notification events."
       INDEX    { pethNotificationControlGroupIndex }
       ::= { pethNotificationControlTable 1 }

   PethNotificationControlEntry ::= SEQUENCE {
       pethNotificationControlGroupIndex
           Integer32,
       pethNotificationControlEnable
           TruthValue
   }
     pethNotificationControlGroupIndex OBJECT-TYPE
       SYNTAX      Integer32 (1..2147483647)
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This variable uniquely identifies the group. Group
            means box in the stack, module in a rack and the value
            1 shall be used for non-modular devices. Furthermore,
            the same value shall be used in this variable,
            pethPsePortGroupIndex, and
            pethMainPseGroupIndex to refer to a given box in a
            stack or module in a rack. "
       ::= { pethNotificationControlEntry 1 }

      pethNotificationControlEnable  OBJECT-TYPE
       SYNTAX             TruthValue
       MAX-ACCESS         read-write
       STATUS             current
       DESCRIPTION
          "This object controls, on a per-group basis, whether
```



```
             or not notifications from the agent are enabled. The
             value true(1) means that notifications are enabled; the
             value false(2) means that they are not."
       ::= { pethNotificationControlEntry 2 }
--
-- Notifications Section
--
--

     pethPsePortOnOffNotification NOTIFICATION-TYPE
         OBJECTS     { pethPsePortDetectionStatus }
         STATUS      current
         DESCRIPTION
             " This Notification indicates if Pse Port is delivering or
               not power to the PD. This Notification should be sent on
               every status change except in the searching mode.
               At least 500 msec shall elapse between notifications
               being emitted by the same object instance."
          ::= { pethNotifications 1 }

     pethMainPowerUsageOnNotification NOTIFICATION-TYPE
         OBJECTS     { pethMainPseConsumptionPower  }
         STATUS      current
         DESCRIPTION
           " This Notification indicate PSE Threshold usage
               indication is on, the usage power is above the
               threshold. At least 500 msec shall elapse between
               notifications being emitted by the same object
               instance."
         ::= { pethNotifications 2 }

      pethMainPowerUsageOffNotification NOTIFICATION-TYPE
         OBJECTS     { pethMainPseConsumptionPower }
         STATUS      current
         DESCRIPTION
           " This Notification indicates PSE Threshold usage indication
               off, the usage power is below the threshold.
               At least 500 msec shall elapse between notifications being
               emitted by the same object instance."
         ::= { pethNotifications 3 }

--
-- Conformance statements
--
pethCompliances OBJECT IDENTIFIER ::= { pethConformance 1 }
pethGroups      OBJECT IDENTIFIER ::= { pethConformance 2 }

-- Compliance statements

pethCompliance MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
               "Describes the requirements for conformance to the
               Power Ethernet MIB."

       MODULE  -- this module
           MANDATORY-GROUPS { pethPsePortGroup,
                              pethPsePortNotificationGroup,
                              pethNotificationControlGroup
```



```
                          }
          GROUP   pethMainPseGroup
          DESCRIPTION
              "The pethMainPseGroup is mandatory for PSE systems
               that implement a main power supply."
          GROUP   pethMainPowerNotificationGroup
          DESCRIPTION
              "The pethMainPowerNotificationGroup is mandatory for
              PSE systems that implement a main power supply."
      ::= { pethCompliances 1 }

pethPsePortGroup OBJECT-GROUP
    OBJECTS {
       pethPsePortAdminEnable,
       pethPsePortPowerPairsControlAbility,
       pethPsePortPowerPairs,
       pethPsePortDetectionStatus,
       pethPsePortPowerPriority,
       pethPsePortMPSAbsentCounter,
       pethPsePortInvalidSignatureCounter,
       pethPsePortPowerDeniedCounter,
       pethPsePortOverLoadCounter,
       pethPsePortShortCounter,
       pethPsePortType,
       pethPsePortPowerClassifications
    }
    STATUS  current
    DESCRIPTION
          "PSE Port objects."
    ::= { pethGroups 1 }

pethMainPseGroup OBJECT-GROUP
    OBJECTS {
       pethMainPsePower,
       pethMainPseOperStatus,
       pethMainPseConsumptionPower,
       pethMainPseUsageThreshold
    }
    STATUS  current
    DESCRIPTION
            "Main PSE Objects. "
    ::= { pethGroups 2 }

pethNotificationControlGroup OBJECT-GROUP

    OBJECTS {
       pethNotificationControlEnable
    }
    STATUS  current
    DESCRIPTION
            "Notification Control  Objects. "
    ::= { pethGroups 3 }

pethPsePortNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS { pethPsePortOnOffNotification}
    STATUS           current
    DESCRIPTION  "Pse Port Notifications."
    ::= { pethGroups 4 }
```



```
  pethMainPowerNotificationGroup NOTIFICATION-GROUP
     NOTIFICATIONS { pethMainPowerUsageOnNotification,
                      pethMainPowerUsageOffNotification}
    STATUS           current
    DESCRIPTION  "Main PSE Notifications."
      ::= { pethGroups 5 }

END
```

9. Ethernet passive optical networks (EPON) MIB module

9.1 Overview

This clause defines a MIB module for use with SNMP to manage 1G-EPON interfaces for Ethernet Passive Optical Networks. The clause contains a list of management objects based on the attributes defined in the relevant parts of IEEE Std 802.3 Clause 30, referring to EPON.

9.1.1 EPON architecture highlights

9.1.1.1 Introduction

The EPON standard, now part of IEEE Std 802.3, defines the Physical Layer and Media Access Control sublayer of EPON interfaces. EPON is a variant of Gigabit Ethernet used in optical access. The Passive Optical Network (PON) is comprised of sections of single-mode fiber connected with passive optical splitter/coupler devices, forming a passive optical tree, as shown in Figure 9–1. Individual branches of the PON are terminated with the Optical Line Terminal (OLT) in the Central Office and Optical Network Units (ONUs) near the subscribers. ONUs can be located either in some remote location (e.g., basement in a multi-dwelling unit) or directly at the subscriber premises. Various types of Customer Premises Equipment (CPE) can be connected to ONUs or even integrated with such devices. Figure 9–1 presents an example PON topology.

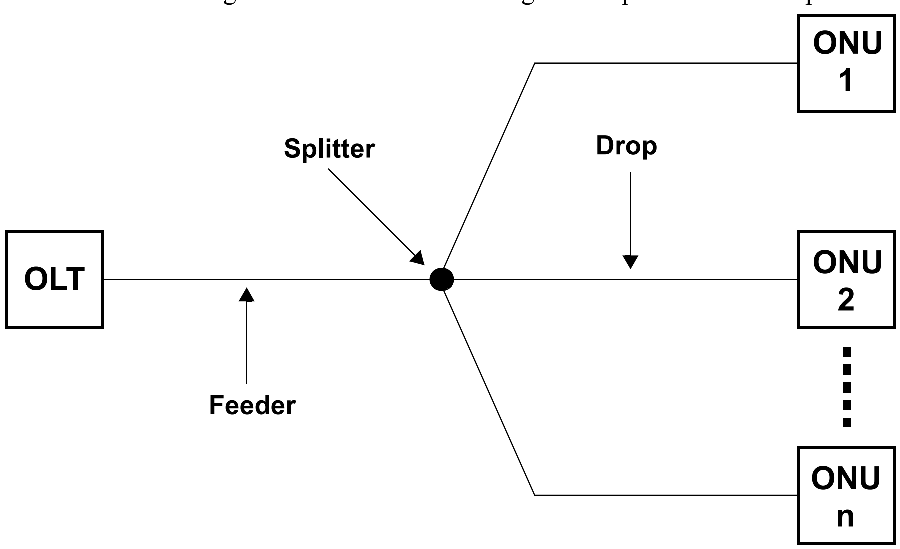


Figure 9–1—PON topology example

The IEEE layering architecture of an EPON interface is defined in the diagram of Figure 56–2 in IEEE Std 802.3. The following clauses in IEEE Std 802.3 define the corresponding layers of an EPON interface:

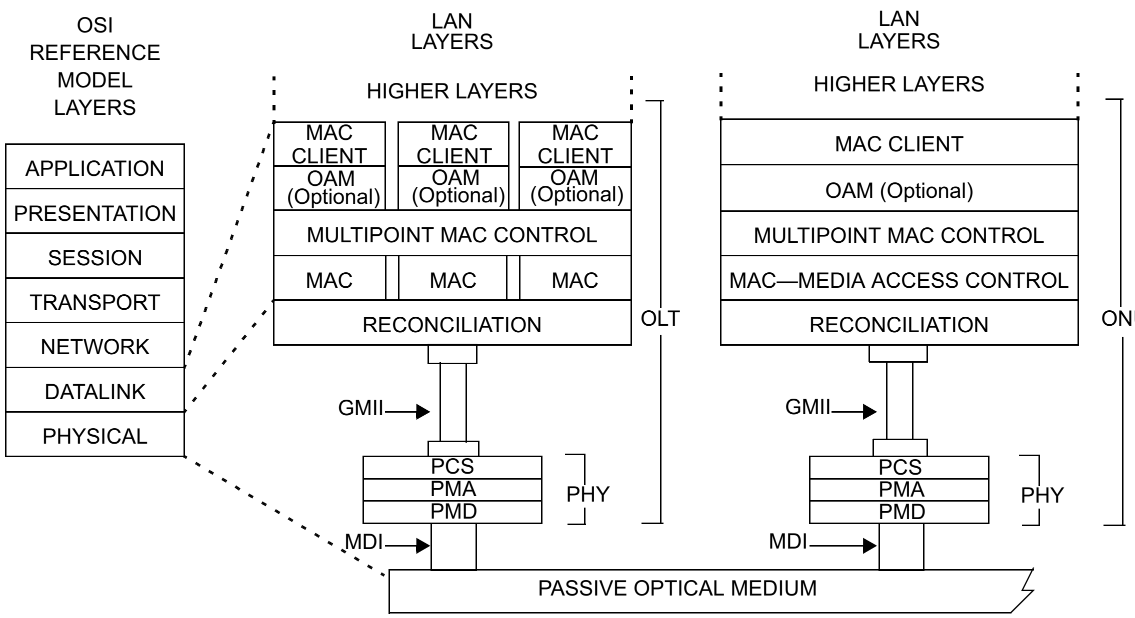
— Clause 30: Management
— Clause 60: PMD for EPON media (burst-mode PMD)
— Clause 64: MPCP (Multipoint Control Protocol), which defines the Multipoint architecture, and control protocol for the media access of EPON.
— Clause 65: Reconciliation Sublayer and Physical Coding Sublayer, which defines a number of extensions to standard Gigabit Ethernet PCS, i.e.,
  a) Definition of Point-to-Point emulation function for EPON
  b) Definition of the optional (frame-based) FEC for EPON
  c) PMA for EPON

9.1.1.2 Principles of operation

The EPON interface specification extends the specification of Gigabit Ethernet as described in IEEE Std 802.3 Clause 35 and Clause 36. The Ethernet MAC operates at the data rate of 1 Gb/s and it is connected to media dependent interface through the GMII interface, as described in Clause 35. The EPON PCS layer extends the Gigabit Ethernet PCS as described in Clause 36. New, EPON-specific layers are added to Gigabit Ethernet layers in the following locations:

- MPCP is placed in the MAC control layer, providing EPON media access, station discovery, and registration protocol.
- Functionality of the reconciliation sublayer (RS) of Gigabit Ethernet was extended, creating logical links over shared passive optical medium, providing private transmission channels to each of the connected ONU.
- (Optional) FEC functionality located between the PCS and PMA layers was added, extending the Gigabit Ethernet PCS layer, enhancing reach and split performance of the EPON optical link.

Figure 9–2 presents the EPON layering model.



GMII = GIGABIT MEDIA INDEPENDENT INTERFACE
MDI = MEDIUM DEPENDENT INTERFACE
OAM = OPERATIONS, ADMINISTRATION & MAINTENANCE
OLT = OPTICAL LINE TERMINAL
ONU = OPTICAL NETWORK UNIT
PCS = PHYSICAL CODING SUBLAYER
PHY = PHYSICAL LAYER DEVICE
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT

Figure 9–2—Relationship of Multipoint MAC control and the OSI protocol stack

9.1.1.3 Physical media

The physical link in EPON comprises single-mode fiber. The OLT and ONUs are connected through a passive optical network comprising sections of single-mode fiber interconnected with passive splitter/coupler devices.

The term *downstream* denotes transmission from the OLT to all connected ONUs, while the term *upstream* denotes transmission from the connected ONUs (one at the time) to the OLT. Upstream and downstream transmissions are wavelength division multiplexed (WDM) into a single strand of single-mode fiber, sharing the same physical link.

The downstream transmission channel is always available to the OLT, thus Time Division Multiplexing (TDM) is used. Transmissions from the OLT arrive at all of the connected ONUs and the individual ONUs filter data from the OLT's transmission based on the logical link identifiers (LLID) assigned to them during the registration and discovery process.

The upstream transmission channel is shared among a number of connected and registered ONUs using Time Division Multiple Access (TDMA). Access to upstream channel is controlled via the Multipoint Control Protocol (MPCP), where the OLT plays the role of the master and ONUs play the role of slave devices. An ONU upon registration remains silent until registered and once registered, it transmits data towards OLT only when granted a transmission opportunity (slot).

9.1.1.4 PMD specifications

The EPON PMD specifications are based on a wavelength plan similar to that used by ITU-T G.983.1. The OLT and ONU optical parameters were derived in part from earlier 1000 Mb/s Ethernet PMD specifications, with the addition of WDM capabilities, and burst mode operation for ONU transmitters and the OLT receiver.

The upstream burst mode operation capability corresponds directly to the TDMA operation in the upstream direction, where queued data is burst from individual ONUs at full data rate for the duration of the allocated transmission period. Once completed, the ONU goes silent and another ONU starts transmitting its data.

9.1.1.5 Point-to-point emulation

The downstream link is a broadcast medium, which means that all data transmitted by the OLT is received by all connected ONUs. In order to guarantee compliance of EPON with Ethernet architecture, the P2PE function was included in the RS, creating a series of logical links between the OLT and connected ONUs. An additional broadcast link is also provided for delivery of any broadcast content. In this way, EPON becomes a collection of logical P2P connections established between the OLT and the ONUs. Therefore, the OLT can be seen as an Ethernet device with N+1 logical ports (N P2P logical interfaces and 1 broadcast interface, where N designates the number of connected ONUs).

Logical links also provide a solution for privacy of data, which otherwise would be shared by all subscribers connected to a single OLT port. In this way, each subscriber is isolated and restricted to accessing data streams addressed only to that particular subscriber.

This concept is illustrated in Figure 9–3, which shows an example of an EPON with a single OLT and three connected ONUs.

The single copy broadcast channel (addressed with a special, reserved LLID, see IEEE Std 802.3 65.1.3.1) was added to take advantage of the broadcast transmission capability of the underlying physical medium. In this way, it is very simple and very bandwidth efficient to deliver broadcast content to all ONUs at the same time, avoiding the need to replicate data into a series of P2P links.

The ONUs filter all downstream data and drop all frames addressed to other devices. Only broadcast frames and frames with correct unicast logical link ID (LLID) are admitted and processed. The LLID replaces two octets of the Ethernet frame preamble, identifying a logical link established between the OLT and the given ONU during the discovery and registration process. The LLID indicates the destination port in the downstream and source port in the upstream. The logical links are used effectively to prevent EPON from violating the IEEE 802.1D bridging rules.

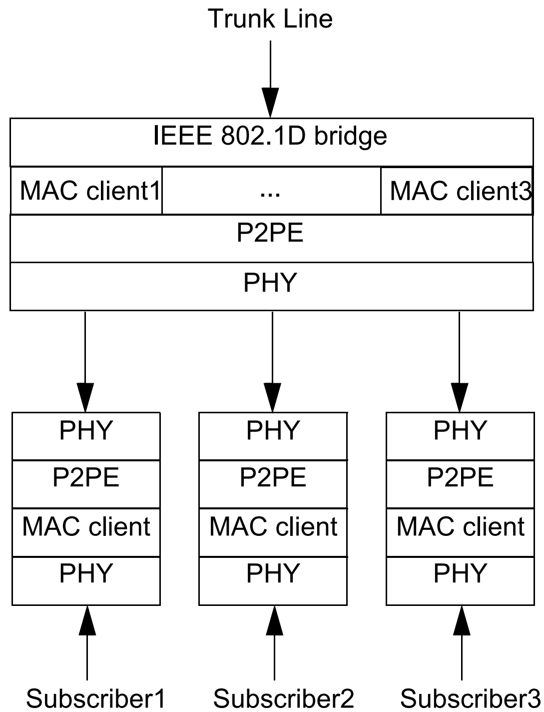


Figure 9–3—Example of point-to-point emulation used between an OLT and three ONUs

9.1.1.6 Principles of the MPCP

The EPON standard comprises a mechanism for media access control, referred to as Multipoint Control Protocol (MPCP). An access network architecture is different from a typical LAN environment, primarily in terms of network provisioning. An access network is an administrated environment, with an operator providing services and subscribers consuming it depending on service provisioning contracts. The operator controls the network, manages traffic and medium access, and enforces the service level agreements (SLAs). For instance, the available bandwidth is controlled and subscribers may be billed for services. In this sense, the access network (and EPON specifically) requires a media access control protocol that provides mechanism for station discovery and registration as well as bandwidth provisioning capabilities.

In the MPCP, the OLT is considered to be the master, controlling a series of connected ONUs (slave devices). The OLT manages the network and controls access to network resources from individual slave devices. The MPCP is also used for provisioning upstream channel access to individual slave devices via a MPCPDU pair i.e., GATE and REPORT. The MPCP is part of the MAC control layer and MPCPDUs are considered MAC control messages, carrying a specific Ethertype of 0x8808. These messages are not forwarded outside of the EPON domain and are used to manage the EPON link only.

A concept of time exists in the MPCP in order to schedule the upstream transmission. A timestamp, which is transmitted in the MPCPDUs downstream by the OLT and received by the connected ONUs, is used to synchronize slave devices to master device clock. This guarantees that upstream transmissions from individual ONUs arrive at the OLT at the precisely anticipated time, which in turn guarantees that data from different ONUs will not overlap.

The MPCP plane is also used to measure the round-trip time (RTT) for each connected ONU. Each MPCPDU carries a generalized timestamp field, which is filled in by the transmitting station with the current value of its MPCP clock at the time when the given MPCPDU is transmitted. The RTT is measured first during the discovery and registration process and then updated regularly upon each exchange of MPCPDUs between the OLT and one of the ONUs. RTT is used by the OLT bandwidth scheduler to schedule upstream transmission slots for individual ONUs in a non-overlapping manner. The IEEE 802.3 EPON standard provides support for the network diameter (distance between the OLT and the farthest ONU) of nominally up to 20 km, which corresponds to the RTT of approximately 200 µs. However, nothing in the standard precludes support for larger network diameters.

The TDMA control is performed using a pair of MPDPUs, namely GATE generated by the OLT to indicate a future transmission opportunity to an ONU and REPORT generated by the ONU with information on the current queue status (bandwidth demand). Internal structure and possible encoding of GATE and REPORT MPCPDUs are defined in Clause 64 of IEEE Std 802.3.

A scheduling algorithm at the OLT, which is not defined in IEEE Std 802.3, is responsible for dividing the bandwidth and controlling the transmission delay of each ONU according to its SLA. The MPCP defines a closed loop operation in order for this algorithm to be efficient. The MPCP allows the ONUs to report on the amount of bandwidth they require for transmission using a special REPORT message. This allows allocating bandwidth to an ONU only when requested, relying on the statistical burst property of the traffic, and allowing different peak bandwidths for different ONUs at different times; hence, allowing oversubscription of the bandwidth. The REPORT message reports the amount of data waiting in the ONU queues.

In addition, the MPCP defines a protocol of auto-discovery and registration of ONUs.

The MPCP registration process is presented in Figure 9–4, while details are described in Clause 64 of IEEE Std 802.3.

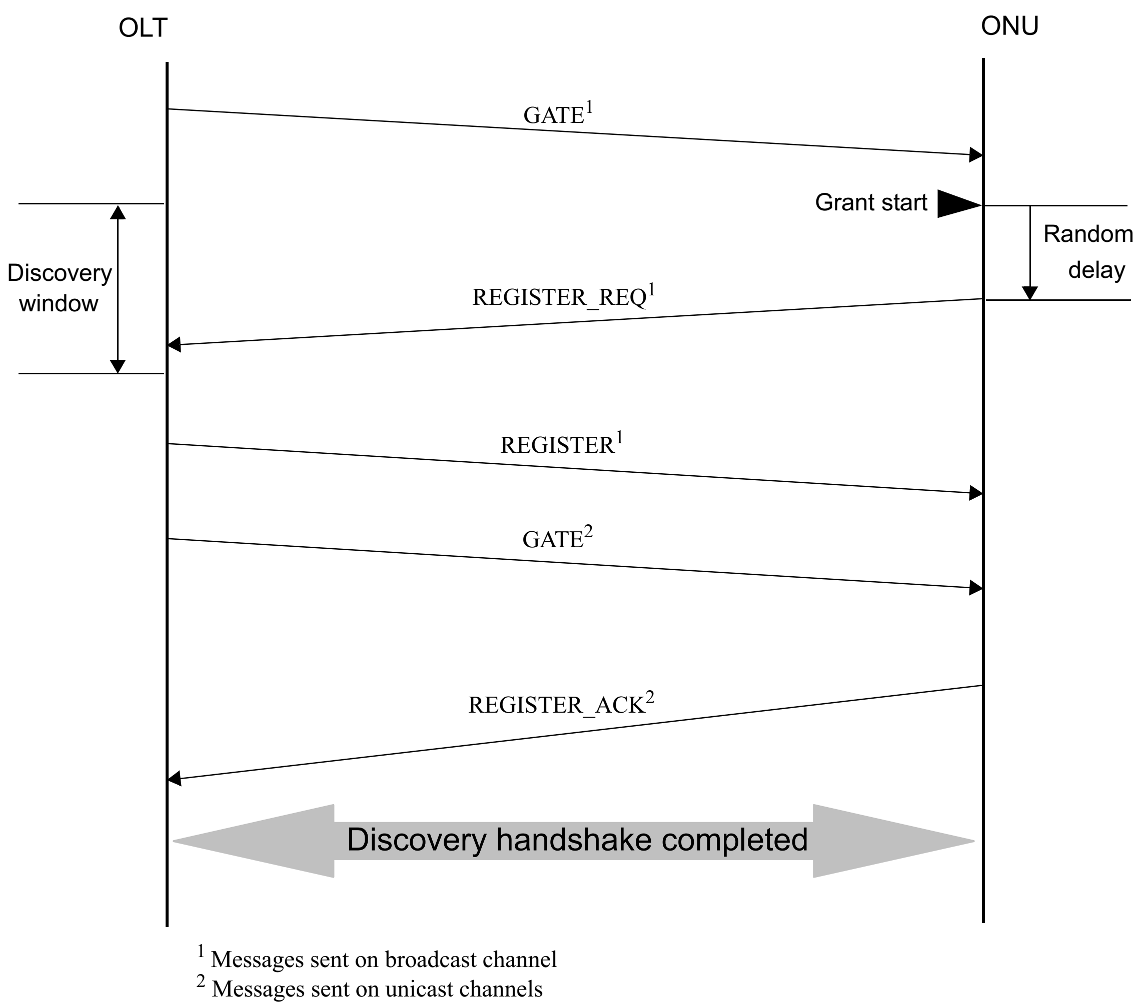


Figure 9–4—Discovery handshake message exchange

A new ONU requests to register during a special upstream window (called Discovery Window), sending the REGISTER_REQ MPCPDU. More than one ONU may attempt registration during that window, which means that their REGISTER_REQ MPCPDUs can potentially collide at the OLT receiver, since the ONU-specific RTT is not yet known and transmissions from individual ONUs cannot be scheduled in an non-overlapping manner. A random backoff mechanism was therefore developed and is used to increase the registration success probability.

When the OLT receives a REGISTER_REQ MPCPDU from an ONU, decision on registration is taken and an LLID is assigned to that ONU. Next, the OLT sends a REGISTER MPCPDU to that ONU, informing the given slave device whether it is admitted to network or not. The registration process is completed with the ONU sending REGISTER_ACK MPCPDU to the OLT, confirming assigned parameters and registration in the network. From that point onward, the OLT can schedule transmissions from that ONU using its LLID, using the measured RTT to guarantee that its transmissions do not collide with other ONUs.

Additional higher layer protocols may be employed to authenticate the ONU and allow it to participate in the network; however, their specification is outside the scope of IEEE Std 802.3.

9.1.1.7 Forward error correction (FEC)

The optional FEC mechanism is defined to enhance the EPON link budget. All the passive components of the fiber plant attenuate the optical signal, thus the target distance (network diameter) and the number of supported splits is limited by the available link budget. The optional FEC mechanism increases the available link budget by improving the link BER from 10^{-4} to 10^{-12} (the target BER at the MAC), effectively increasing the target network diameter and/or split ratio. The target use of the increased power budget remains at the sole discretion of the network architects and is out of the scope of IEEE Std 802.3.

The optional FEC used in EPON is frame-based, meaning that parity information is added at the end of each Ethernet packet. Extra space between individual Ethernet packets is guaranteed by the MAC rate adaptation function, while extra idle symbols were replaced within the FEC function.

The start and end of packet codewords also define the FEC boundaries, and they are outside the FEC protection. They are replaced by a series of symbols to reduce their vulnerability to link errors.

Figure 9–5 presents the structure of an FEC-protected EPON frame.

The optional FEC function is added to the extended Gigabit Ethernet PCS per 65.2 in IEEE Std 802.3. The added, optional FEC function introduces a fixed delay in receive path and transmit path.

S_FEC	Preamble/SFD	Frame	FCS	T_FEC	Parity	T_FEC

Figure 9–5—FEC-protected frame

9.1.2 Management architecture

All of the EPON layers are accompanied by a management interface that is controlled through mechanisms defined in Clause 30 of IEEE Std 802.3. Since IEEE Std 802.3 specifications may be used for different applications (and hence are extensible), and some of the clauses may be used separately, the management clause allocates a separate package for each independent layer. The structure of the MIB modules follows this separation.

Figure 9–6 presents the relation of the MIB module groups to the individual IEEE Std 802.3 layers.

The association is straightforward for the ONU interface. There is one logical and one physical interface, and a single copy of each layer can be remotely queried by the OLT.

The OLT has a single physical interface and N logical interfaces, one for each logical link connected to an ONU. There is also one logical interface for the single copy broadcast link. Per layering diagram in Figure 9–6, the MAC sublayer is virtually replicated. Therefore, in this clause it was elected that management of logical interfaces is performed in the manner identical to management of any physical interfaces—an interface index is allocated for each one of the logical links, and an additional interface index is allocated for the OLT.

For each physical interface, there would be an entry (ifIndex) in the tables of the interface MIB module defined in IETF RFC 2863, the MAU MIB module defined in Clause 13, and the Ethernet-like MIB module defined in Clause 10. Additionally, there would be entries (ifIndexes) for the virtual interfaces of the OLT interface. The justification for the additional allocation of indexes is that the virtual interfaces are quite well

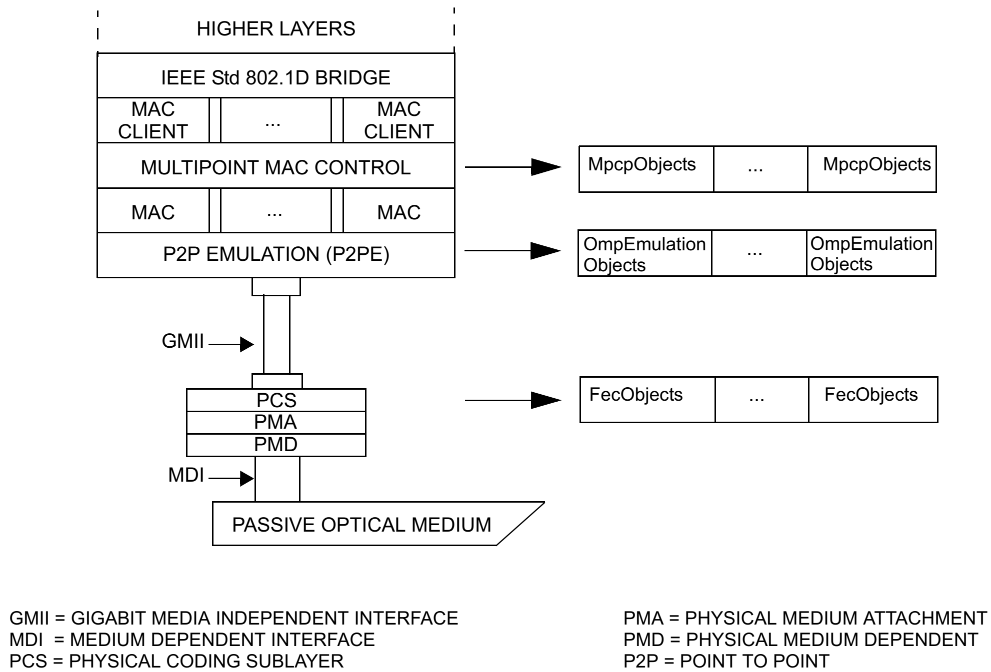


Figure 9–6—Relationship of the MIB groups to the EPON sublayers

distinguished, as they connect different physical ONUs from the OLT side. For instance, there is a meaning for separate bad frames counter or bad octets counter for each virtual link, as the ONUs can be differently distanced. This is quite similar to a case of separate physical interfaces.

The same partition concept exists for the MIB module of this clause. Each row in the tables are indexed according to the ifIndex; specifically, there is a row for each virtual link. There are some control objects that are shared and are the same for the virtual interfaces (and they should have the same value for each ifIndex), but most of the objects have different values for N+1 logical interfaces at the OLT. This is done for each MIB group. It is different from the EPON layering diagram, which presents the P2MP layer as a single layer, while duplicating the MAC and MAC client layers (please see Figure 9–6). However, from a management perspective, it is more convenient to partition the management of the layers for the virtual links, as the atomic managed entity is the virtual link. It is also convenient to use the interface index of the virtual link for that purpose, as it is already used to index the rows of the virtual links at the Interface, MAU, and Ethernet-like interface MIBs.

9.2 MIB structure

This subclause defines the DOT3 EPON MIB module. The DOT3 EPON MIB module defines the objects used for management of the IEEE Std 802.3 EPON interfaces. These MIB objects are included in the following four groups.

a) MPCP MIB objects—MIB objects related to IEEE Std 802.3, Clause 64, Multipoint Control Protocol attributes. The following tables are presented in this group:
   1) The dot3MpcpControlTable defines the objects used for the configuration and status indication, which are per logical link, of MPCP compliant interfaces.

2) The dot3MpcpStatTable defines the statistics objects that are per logical link, of MPCP compliant interfaces.
3) The operational mode of an OLT/ONU for the tables is defined by the dot3MpcpMode object in the dot3MpcpControlTable.

b) The OMPEmulation MIB objects—MIB objects related to IEEE Std 802.3, Clause 65, point-to-point emulation attributes. The following tables are presented in this group:
   1) The dot3OmpEmulationTable defines the objects used for the configuration and status indication, which are per logical links, of OMPEmulation compliant interfaces.
   2) The dot3OmpEmulationStatTable defines the statistics objects that are per logical link, of OMPEmulation compliant interfaces.
   3) The operational mode of an OLT/ONU for the tables is defined by the dot3OmpEmulationType object in the dot3OmpEmulationTable.
c) The FEC MIB objects—MIB objects related to IEEE Std 802.3, Clause 60 and Clause 65, EPON FEC attributes. The following table is presented in this group:
   1) The dot3EponFecTable defines the objects used for the configuration and status indication, which are per logical link, of FEC EPON compliant interfaces.
d) The EPON extended package MIB objects—MIB objects used for configuration and status indication with extended capabilities of the EPON interfaces. The following tables are presented in this group:
   1) The dot3ExtPkgControlTable defines the objects, which are per logical link, used for the configuration and status indication of EPON compliant interfaces.
   2) The dot3ExtPkgQueueTable defines the objects, which are per logical link, and per queue, used for the configuration and status indication of the ONU queues reported in the MPCP REPORT message, of EPON compliant interfaces.
   3) The dot3ExtPkgQueueSetsTable defines the objects, which are per logical link, per queue, and per queue_set, used for the configuration and status indication of the ONU queue_sets reported in the MPCP REPORT message, of EPON compliant interfaces.
   4) The dot3ExtPkgOptIfTable defines the objects, which are per logical link, used for the control and status indication of the optical interface of EPON compliant interfaces.

The interface MIB module defined in IETF RFC 2863 defines the interface index (ifIndex). Interface Index, as specified in IETF RFC 2863, is used in this MIB module as an index to the EPON MIB tables. The ifIndex is used to denote the physical interface and the virtual link interfaces at the OLT. The OLT interface and the virtual link interfaces are stacked using the ifStack table defined in IETF RFC 2863, and the ifInvStack defined in IETF RFC 2864. The OLT interface is the lower layer of all other interfaces associated with the virtual links.

As described in 9.1.2, each row in the tables is indexed according to the ifIndex; specifically, there is a row for each virtual link. There are a few control objects that are shared and have the same value for the virtual interfaces (and they should have the same value for each ifIndex), but most of the objects have different values for N+1 logical interfaces at the OLT. This is done for each MIB group. It is a bit different from the EPON layering diagram, which presents the P2MP layer as a single layer while duplicating the MAC and MAC client layers. However, from a management perspective, it is more convenient to partition the management of the layers for the virtual links, as the atomic managed entity is the virtual link. It is also convenient to use the interface index of the virtual link for that purpose, as it is already used to index the rows of the virtual links at the Interface, MAU, and Ethernet-like interface MIB modules.

The creation of the rows of the ONU interface is done at initialization. Table 9–1 presents the MPCP control table of ONU1 after initialization. A single row exists in the table.

Table 9–2 presents the MPCP control table of ONU1 in working mode. A single row exists in the table.

Table 9–1—MPCP control table of ONU1 after initialization

MPCP control MIB object	Value
ifIndex	100
dot3MpcpOperStatus	true
dot3MpcpAdminState	true
dot3MpcpMode	onu
dot3MpcpSyncTime	0
dot3MpcpLinkID	0
dot3MpcpRemoteMACAddress	00:00:00:00:00:00
dot3MpcpRegistrationState	unregistered
dot3MpcpTransmitElapsed	0
dot3MpcpReceiveElapsed	0
dot3MpcpRoundTripTime	0

Table 9–2—MPCP control table of ONU1 in working mode

MPCP control MIB object	Value
ifIndex	100
dot3MpcpOperStatus	true
dot3MpcpAdminState	true
dot3MpcpMode	onu
dot3MpcpSyncTime	25
dot3MpcpLinkID	1
dot3MpcpRemoteMACAddress	OLT_MAC_Address[a]
dot3MpcpRegistrationState	registered
dot3MpcpTransmitElapsed	10
dot3MpcpReceiveElapsed	10
dot3MpcpRoundTripTime	100

[a]OLT_MAC_Address is the MAC address of the OLT EPON interface.

The creation of the rows of the OLT interface and the broadcast virtual interface is done at initialization.

The creation of rows of the virtual interfaces at the OLT is done when the link is established (ONU registers) and the deletion is done when the link is deleted (ONU deregisters).

Table 9–3 presents the MPCP control table of the OLT after initialization, before the ONUs register. A single row exists in this table associated with the virtual broadcast link.

Table 9–3—MPCP control table of the OLT after initialization

MPCP control MIB object	Value
ifIndex	165535
dot3MpcpOperStatus	true
dot3MpcpAdminState	true
dot3MpcpMode	olt
dot3MpcpSyncTime	25
dot3MpcpLinkID	65535
dot3MpcpRemoteMACAddress	BRCT_MAC_Address[a]
dot3MpcpRegistrationState	registered
dot3MpcpTransmitElapsed	10
dot3MpcpReceiveElapsed	100000
dot3MpcpRoundTripTime	0

[a]BRCT_MAC_Address is the MAC address of the broadcast EPON interface, which is the OLT MAC address.

Table 9–4 presents the MPCP control table of the OLT in working mode. Three rows exist in the table associated with the virtual links.

Table 9–4—MPCP control table of the OLT in working mode

MPCP control MIB object	Value	Value	Value
ifIndex	100001	100002	165535
dot3MpcpOperStatus	true	true	true
dot3MpcpAdminState	true	true	true
dot3MpcpMode	olt	olt	olt
dot3MpcpSyncTime	25	25	25

Table 9–4—MPCP control table of the OLT in working mode *(continued)*

MPCP control MIB object	Value	Value	Value
dot3MpcpLinkID	1	2	65535
dot3MpcpRemote MACAddress	ONU1_MAC_Address[a]	ONU2_MAC_Address[b]	BRCT_MAC_Address[c]
dot3MpcpRegistrationState	registered	registered	registered
dot3MpcpTransmitElapsed	10	10	10
dot3MpcpReceiveElapsed	10	10	10
dot3MpcpRoundTripTime	100	60	0

[a]ONU1_MAC_Address is the MAC address of ONU1 EPON interface.
[b]ONU2_MAC_Address is the MAC address of ONU2 EPON interface.
[c]BRCT_MAC_Address is the MAC address of the broadcast EPON interface, which is the OLT MAC address.

9.3 Relationship to other MIB modules

9.3.1 Relation to the Interfaces Group MIB and Ethernet-like interface MIB

This MIB module extends the objects of the Interfaces Group MIB and the Ethernet-like interface MIB for the EPON type interface. Therefore, if this module is implemented, the Interfaces Group MIB module defined in IETF RFC 2863 and the Ethernet-like interface MIB module defined in Clause 10 shall also be implemented.

Thus, each managed EPON interface would have a corresponding entry in the mandatory tables of the Ethernet-like MIB module found in Clause 10, and likewise in the tables of the Interfaces Group MIB module found in IETF RFC 2863. Also, each managed virtual EPON interface would have a corresponding entry in the mandatory tables of the Ethernet-like MIB module found in Clause 10, and likewise in the tables of the Interfaces Group MIB module found in IETF RFC 2863 with a dedicated ifIndex for this interface.

In this clause, there is no replication of the objects from these MIBs. Therefore, for instance, the clause is defining dot3MpcpRemoteMACAddress only while assuming that the local MAC address object is already defined in Clause 10.

This clause defines the specific EPON objects of an ONU interface and an OLT interface. Information in the tables is per LLID. The rows in the EPON MIB tables referring to the LLIDs are denoted with the corresponding ifIndexes of the virtual link interfaces.

Note that all virtual interfaces have the same physical MAC address at the OLT, since the physical OLT interface used by all virtual interfaces is the same. The value of this physical MAC interface is specified in 64.1.2 of IEEE Std 802.3. The corresponding object of the Ethernet-like interface MIB is replicated for all the virtual interfaces.

For example, the values of the Interfaces Group MIB objects are presented in the following tables, for an OLT with three registered ONUs

Table 9–5 presents the objects of the Interfaces Group MIB of an ONU in working mode.

Table 9–5—Interfaces Group MIB of an ONU in working mode

Interfaces Group MIB object	Value
ifIndex	1
ifDescr	“interface description”
ifType	ethernetCsmacd (6) 1000base-Px
ifMtu	MTU size (1522)
ifSpeed	1000000000
ifPhysAddress	ONU_MAC_Address[a]
ifAdminStatus	up
ifOperStatus	Up
ifLastChange	ONUup_time
ifInOctets	ONU_octets_number
ifInUcastPkts	ONU_unicast_frame_number
ifInNUcastPkts	ONU_non_unicast_frame_number
ifInDiscards	ONU_discard_frame_number
ifInErrors	ONU_error_frame_number
ifInUnknownProtos	ONU_unknown_frame_number
ifOutOctets	ONU_octets_number
ifOutUcastPkts	ONU_unicast_frame_number
ifOutNUcastPkts	ONU_non_unicast_frame_number
ifOutDiscards	ONU_discard_frame_number
ifOutErrors	ONU_error_frame_number
ifOutQLen	ONU_queue_frame_number

[a]ONU_MAC_Address is the MAC address of the ONU EPON interface.

Table 9–6 presents the objects of the Interfaces Group MIB of the ONU interface.

Table 9–6—Interfaces Group MIB of the ONU interface

Interfaces Group MIB object	Value
ifIndex	100
ifDescr	“interface description”
ifType	ethernetCsmacd (6) 1000base-Px
ifMtu	MTU size (1522)
ifSpeed	1000000000
ifPhysAddress	ONU_MAC_Address[a]
ifAdminStatus	up
ifOperStatus	Up
ifLastChange	up_time
ifInOctets	ONU1_octets_number
ifInUcastPkts	ONU1_unicast_frame_number
ifInNUcastPkts	ONU1_non_unicast_frame_number
ifInDiscards	ONU1_discard_frame_number
ifInErrors	ONU1_error_frame_number
ifInUnknownProtos	ONU1_unknown_frame_number
ifOutOctets	ONU1_octets_number
ifOutUcastPkts	ONU1_unicast_frame_number
ifOutNUcastPkts	ONU1_non_unicast_frame_number
ifOutDiscards	ONU1_discard_frame_number
ifOutErrors	ONU1_error_frame_number
ifOutQLen	ONU1_queue_frame_number

[a]ONU_MAC_Address is the MAC address of the ONU EPON interface.

The following values will be set in the ifStack and ifInvStack tables related to this example.

ifStackTable:

— ifStackHigherLayer = 100, ifStackLowerLayer = 1 – map between the physical interface and the ONU

ifInvStackTable:

— ifStackLowerLayer = 1, ifStackHigherLayer = 100 – map between the ONU and the physical interface

Table 9–7 presents the Interfaces Group MIB objects of an OLT interface.

Table 9–7—Interfaces Group MIB objects of an OLT interface

Interfaces Group MIB object	Value
ifIndex	2
ifDescr	“interface description”
ifType	ethernetCsmacd (6) 1000base-Px
ifMtu	MTU size (1522)
ifSpeed	1000000000
ifPhysAddress	OLT_MAC_Address[a]
ifAdminStatus	up
ifOperStatus	Up
ifLastChange	OLTup_time
ifInOctets	OLT_octets_number
ifInUcastPkts	OLT_unicast_frame_number
ifInNUcastPkts	OLT_non_unicast_frame_number
ifInDiscards	OLT_discard_frame_number
ifInErrors	OLT_error_frame_number
ifInUnknownProtos	OLT_unknown_frame_number
ifOutOctets	OLT_octets_number
ifOutUcastPkts	OLT_unicast_frame_number
ifOutNUcastPkts	OLT_non_unicast_frame_number

Table 9–7—Interfaces Group MIB objects of an OLT interface *(continued)*

Interfaces Group MIB object	Value
ifOutDiscards	OLT_discard_frame_number
ifOutErrors	OLT_error_frame_number
ifOutQLen	OLT_queue_frame_number

[a]OLT_MAC_Address is the MAC address of the OLT EPON interface.

Table 9–8 presents the Interfaces Group MIB objects of an OLT interface, associated with the virtual link interfaces**.**

Table 9–8—Interfaces Group MIB objects of an OLT interface, associated with the virtual link interfaces

Interface MIB object	Value	Value	Value
ifIndex	200001	200002	265535
ifDescr	“interface description"	“interface description"	“interface description"
ifType	ethernetCsmacd (6)	ethernetCsmacd (6)	ethernetCsmacd (6)
ifMtu	MTUsize(1522)	MTUsize(1522)	MTUsize(1522)
ifSpeed	1000000000	1000000000	1000000000
ifPhysAddress	OLT_MAC_Address[a]	OLT_MAC_Address	OLT_MAC_Address
ifAdminStatus	up	up	up
ifOperStatus	Up	Up	Up
ifLastChange	ONU1_up_time	ONU2_up_time	up_time
ifInOctets	ONU1_octets_number	ONU2_octets_number	BRCT_octets_number
ifInUcastPkts	ONU1_unic_frame_num	ONU2_unic_frame_num	BRCT_unic_frame_num
ifInNUcastPkts	ONU1_non_unic_frame_num	ONU2_non_unic_frame_num	BRCT_non_unic_frame_num
ifInDiscards	ONU1_disc_frame_num	ONU2_disc_frame_num	BRCT_disc_frame_numr
ifInErrors	ONU1_err_frame_num	ONU2_err_frame_num	BRCT_err_frame_num
ifInUnknownProtos	ONU1_unknw_frame_num	ONU2_unknw_frame_num	BRCT_unknw_frame_num
ifOutOctets	ONU1_octets_number	ONU2_octets_number	BRCT_octets_number
ifOutUcastPkts	ONU1_unic_frame_num	ONU2_unic_frame_num	BRCT_unic_frame_num
ifOutNUcastPkts	ONU1_non_unic_frame_num	ONU2_non_unic_frame_num	BRCT_non_unic_frame_num

Table 9–8—Interfaces Group MIB objects of an OLT interface, associated with the virtual link interfaces *(continued)*

Interface MIB object	Value	Value	Value
ifOutDiscards	ONU1_disc_frame_num	ONU2_disc_frame_num	BRCT_disc_frame_num
ifOutErrors	ONU1_err_frame_num	ONU2_err_frame_num	BRCT_err_frame_num
ifOutQLen	ONU1_queue_frame_num	ONU2_queue_frame_num	BRCt_queue_frame_num

[a]OLT_MAC_Address is the MAC address of the OLT EPON interface.

The following values will be set in the ifStack and ifInvStack tables related to this example:

ifStackTable:

— ifStackHigherLayer = 265535, ifStackLowerLayer = 2 – map between the OLT physical interface and its broadcast virtual interface

— ifStackHigherLayer = 200001, ifStackLowerLayer = 2 – map between the OLT physical interface and its virtual interface of the 1st ONU

— ifStackHigherLayer = 200002, ifStackLowerLayer = 2 – map between the OLT physical interface and its virtual interface of the 2nd ONU

— ifStackHigherLayer = 200003, ifStackLowerLayer = 2 – map between the OLT physical interface and its virtual interface of the 3rd ONU

ifInvStackTable:

— ifStackLowerLayer = 2, ifStackHigherLayer = 265535 – map between the broadcast interface of the OLT and the OLT physical interface

— ifStackLowerLayer = 2, ifStackHigherLayer = 200001 – map between the OLT virtual interface of the 1st ONU and the OLT physical interface

— ifStackLowerLayer = 2, ifStackHigherLayer = 200002 – map between the OLT virtual interface of the 2nd ONU and the OLT physical interface

— ifStackLowerLayer = 2, ifStackHigherLayer = 200003 – map between the OLT virtual interface of the 3rd ONU and the OLT physical interface

The rows for the ONU interface, the OLT interface, and the OLT broadcast interface are created in initialization. The creation of a row for a virtual link is done when the virtual link is established (ONU registers), and deletion is done when the virtual link is deleted (ONU deregisters).

The EPON MIB module also extends the Interfaces Group MIB module with a set of counters, which are specific for the EPON interface. The EPON MIB module implements the same handling of the counters when the operation of the interface starts or stops. The interface MIB clause describes the possible behavior of counters when an interface is re-initialized using the ifCounterDiscontinuityTime indicator, indicating the discontinuity of the counters. See IETF RFC 2863, Section 3.1.5, for more information. The counters of the EPON MIB should be handled in a similar manner.

9.3.2 Relation to the IEEE 802.3 MAU MIBs

The MAU types of the EPON Interface are defined in the Clause 13. This clause assumes the implementation of the MAU MIB for this purpose and does not repeat the EPON MAU types. Therefore, if this module is implemented, the MAU-MIB module defined in Clause 13 shall also be implemented.

The handling of the ifMAU tables for the EPON case is similar to the handling described in the former subclause for the Interface and Ethernet-like interface MIBs. A single row exists for the ONU in the ifMauTable. A row for each virtual link (N+1 rows) exists at the OLT, with a separate value of ifMauIfIndex for each virtual link.

As specified above, the rows for the ONU interface, the OLT interface, and the OLT broadcast interface are created in initialization. The creation of a row for a virtual link is done when the virtual link is established (ONU registers), and deletion is done when the virtual link is deleted (ONU deregisters).

9.3.3 Relation to the Ethernet OAM MIB

The EPON interfaces are intended for use in optical subscriber access networks and most probably will be accompanied with the implementation of the OAM protocol defined in Clause 57 of IEEE Std 802.3. Therefore, the Ethernet OAM MIB module defined in Clause 6 may be implemented when this MIB module is implemented defining managed objects for the OAM protocol that are complementary to the EPON MIB module.

9.3.4 Relation to the bridge MIB

It is very probable that an EPON OLT will implement a bridging functionality above the EPON interface layer, bridging between the EPON users and the network. Bridge functionality is specified in IEEE Std 802.1D. In this scenario, the virtual ports of the EPON are corresponding to the virtual bridge ports. There is a direct mapping between the bridge ports and the LLIDs, which are virtual EPON channels.

Therefore, the bridge MIB modules defined in IEEE Std 802.1Q [B4] may be implemented when the EPON MIB module is implemented for an EPON OLT, defining managed objects for the bridge layer.

The values of dot1dBasePortIfIndex would correspond to the ifIndex of the virtual port (1 for LLID1, 2 for LLID2, etc.).

The broadcast virtual EPON interface of the OLT has no direct mapping to a virtual bridge port as it is not port specific but used for broadcast traffic.

9.4 Mapping of IEEE 802.3 managed objects

This subclause contains the mapping between the managed objects defined in this clause and the attributes defined in IEEE Std 802.3 Clause 30. Table 9–9 provides the mapping between the dot3EPON MIB module MPCP objects and the IEEE Std 802.3 Clause 30 MPCP attributes.

Table 9–10 provides the mapping between the dot3EPON MIB module OMPEmulation objects and IEEE Std 802.3 Clause 30 OMPE attributes.

Table 9–11 provides the mapping between the dot3EPON MIB module FEC objects and IEEE Std 802.3 Clause 30 MAU attributes.

Table 9–9—oMPCP managed object class (30.3.5)

dot3EPON MIB module object	IEEE 802.3 attribute	Reference
ifIndex	aMPCPID	30.3.5.1.1
dot3MpcpOperStatus	aMPCPAdminState	30.3.5.1.2
dot3MpcpMode	aMPCPMode	30.3.5.1.3
dot3MpcpLinkID	aMPCPLinkID	30.3.5.1.4
dot3MpcpRemoteMACAddress	aMPCPRemoteMACAddress	30.3.5.1.5
dot3MpcpRegistrationState	aMPCPRegistrationState	30.3.5.1.6
dot3MpcpMACCtrlFramesTransmitted	aMPCPMACCtrlFramesTransmitted	30.3.5.1.7
dot3MpcpMACCtrlFramesReceived	aMPCPMACCtrlFramesReceived	30.3.5.1.8
dot3MpcpTxGate	aMPCPTxGate	30.3.5.1.9
dot3MpcpTxRegAck	aMPCPTxRegAck	30.3.5.1.10
dot3MpcpTxRegister	aMPCPTxRegister	30.3.5.1.11
dot3MpcpTxRegRequest	aMPCPTxRegRequest	30.3.5.1.12
dot3MpcpTxReport	aMPCPTxReport	30.3.5.1.13
dot3MpcpRxGate	aMPCPRxGate	30.3.5.1.14
dot3MpcpRxRegAck	aMPCPRxRegAck	30.3.5.1.15
dot3MpcpRxRegister	aMPCPRxRegister	30.3.5.1.16
dot3MpcpRxRegRequest	aMPCPRxRegRequest	30.3.5.1.17
dot3MpcpRxReport	aMPCPRxReport	30.3.5.1.18
dot3MpcpTransmitElapsed	aMPCPTransmitElapsed	30.3.5.1.19
dot3MpcpReceiveElapsed	aMPCPReceiveElapsed	30.3.5.1.20
dot3MpcpRoundTripTime	aMPCPRoundTripTime	30.3.5.1.21
dot3MpcpDiscoveryWindowsSent	aMPCPDiscoveryWindowsSent	30.3.5.1.22
dot3MpcpDiscoveryTimeout	aMPCPDiscoveryTimeout	30.3.5.1.23
dot3MpcpMaximumPendingGrants	aMPCPMaximumPendingGrants	30.3.5.1.24
dot3MpcpAdminState	aMPCPAdminControl	30.3.5.2.1
dot3MpcpSyncTime	SyncTime	64.3.3.2

Table 9–10—oOMPEmulation managed object class (30.3.7)

dot3EPON MIB module object	IEEE 802.3 attribute	Reference
ifIndex	aOMPEmulationID	30.3.7.1.1
dot3OmpEmulationType	aOMPEmulationType	30.3.7.1.2
dot3OmpEmulationSLDErrors	aSLDErrors	30.3.7.1.3
dot3OmpEmulationCRC8Errors	aCRC8Errors	30.3.7.1.4
dot3OmpEmulationGoodLLID	aGoodLLID	30.3.7.1.5
dot3OmpEmulationOnuPonCastLLID	aONUPONcastLLID	30.3.7.1.6
dot3OmpEmulationOltPonCastLLID	aOLTPONcastLLID	30.3.7.1.7
dot3OmpEmulationBadLLID	aBadLLID	30.3.7.1.8
dot3OmpEmulationBroadcastBitNotOnuLLid	N/A	
dot3OmpEmulationOnuLLIDNotBroadcast	N/A	
dot3OmpEmulationBroadcastBitPlusOnuLlid	N/A	
dot3OmpEmulationNotBroadcastBitNotOnuLlid	N/A	

Table 9–11—oMAU managed object class (30.5.1)

dot3EPON MIB module object	IEEE 802.3 attribute	Reference
dot3EponFecPCSCodingViolation	aPCSCodingViolation	30.5.1.1.12
dot3EponFecAbility	aFECAbility	30.5.1.1.13
dot3EponFecMode	aFECmode	30.5.1.1.14
dot3EponFecCorrectedBlocks	aFECCorrectedBlocks	30.5.1.1.15
dot3EponFecUncorrectableBlocks	aFECUncorrectableBlocks	30.5.1.1.16
dot3EponFecBufferHeadCodingViolation	N/A	

9.5 Security considerations for Ethernet passive optical networks (EPON) MIB module

There are number of managed objects defined in this MIB module that have a MAX-ACCESS clause of read-write or read-create. Writing to these objects can have potentially disruptive effects on network operation, including those listed in 9.5.1 to 9.5.13.

9.5.1 dot3MpcpAdminState

Changing dot3MpcpAdminState state can lead to disabling the Multipoint Control Protocol on the respective interface, leading to the interruption of service for the users connected to the respective EPON interface.

9.5.2 dot3EponFecMode

Changing dot3EponFecMode state can lead to disabling the Forward Error Correction on the respective interface, which can lead to a degradation of the optical link, and therefore may lead to an interruption of service for the users connected to the respective EPON interface.

9.5.3 dot3ExtPkgObjectReset

Changing dot3ExtPkgObjectReset state can lead to a reset of the respective interface leading to an interruption of service for the users connected to the respective EPON interface.

9.5.4 dot3ExtPkgObjectPowerDown

Changing dot3ExtPkgObjectPowerDown state can lead to a power down of the respective interface, leading to an interruption of service for the users connected to the respective EPON interface.

9.5.5 dot3ExtPkgObjectFecEnabled

Changing dot3ExtPkgObjectFecEnabled state can lead to disabling the Forward Error Correction on the respective interface, which can lead to a degradation of the optical link, and therefore may lead to an interruption of service for the users connected to the respective EPON interface.

9.5.6 dot3ExtPkgObjectRegisterAction

Changing dot3ExtPkgObjectRegisterAction state can lead to a change in the registration state of the respective interface, leading to a deregistration and an interruption of service for the users connected to the respective EPON interface.

9.5.7 dot3ExtPkgObjectReportNumThreshold

Changing dot3ExtPkgObjectReportNumThreshold can lead to a change in the reporting of the ONU interface and therefore to a change in the bandwidth allocation of the respective interface. This change may lead to a degradation or an interruption of service for the users connected to the respective EPON interface.

9.5.8 dot3ExtPkgObjectReportThreshold

Changing dot3ExtPkgObjectReportThreshold can lead to a change in the reporting of the ONU interface and therefore to a change in the bandwidth allocation of the respective interface. This change may lead to a degradation or an interruption of service for the users connected to the respective EPON interface.

9.5.9 dot3ExtPkgOptIfLowerInputPowerThreshold

Changing dot3ExtPkgOptIfLowerInputPowerThreshold can lead to a Threshold Crossing Alert (TCA) being sent for the respective interface. This alert may be leading to an interruption of service for the users connected to the respective EPON interface, depending on the system action on such an alert.

9.5.10 dot3ExtPkgOptIfUpperInputPowerThreshold

Changing dot3ExtPkgOptIfUpperInputPowerThreshold can lead to a Threshold Crossing Alert (TCA) being sent for the respective interface. This alert may be leading to an interruption of service for the users connected to the respective EPON interface, depending on the system action on such an alert.

9.5.11 dot3ExtPkgOptIfLowerOutputPowerThreshold

Changing dot3ExtPkgOptIfLowerOutputPowerThreshold can lead to a Threshold Crossing Alert (TCA) being sent for the respective interface. This alert may be leading to an interruption of service for the users connected to the respective EPON interface, depending on the system action on such an alert.

9.5.12 dot3ExtPkgOptIfUpperOutputPowerThreshold

Changing dot3ExtPkgOptIfUpperOutputPowerThreshold can lead to a Threshold Crossing Alert (TCA) being sent for the respective interface. This alert may be leading to an interruption of service for the users connected to the respective EPON interface, depending on the system action on such an alert.

9.5.13 dot3ExtPkgOptIfTransmitEnable

9.6 Changing dot3ExtPkgOptIfTransmitEnable state can lead to a halt in the optical transmission of the respective interface, leading to an interruption of service for the users connected to the respective EPON interface.**MIB module definition**

An ASCII text version of the MIB definition can be found at the following URL:[15]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C9mib.txt

[15]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
IEEE8023-DOT3-EPON-MIB DEFINITIONS ::= BEGIN

     IMPORTS
         MODULE-IDENTITY, OBJECT-TYPE, Counter32,
         Integer32, Unsigned32, Counter64, org
             FROM SNMPv2-SMI
         TruthValue, MacAddress
             FROM SNMPv2-TC
         ifIndex
             FROM IF-MIB
         MODULE-COMPLIANCE, OBJECT-GROUP
             FROM SNMPv2-CONF
         ;

ieee8023dot3EponMIB MODULE-IDENTITY
        LAST-UPDATED "201102020000Z" -- February 2, 2011
        ORGANIZATION
          "IEEE 802.3 working group"
        CONTACT-INFO
            "WG-URL: http://www.ieee802.org/3/index.html
            WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

            Contact: Howard Frazier
            Postal:  3151 Zanker Road
                     San Jose, CA 95134
                     USA
            Tel:     +1.408.922.8164
            E-mail:  hfrazier@broadcom.com"
    DESCRIPTION
            "The objects in this MIB module are used to manage the
             Ethernet in the First Mile (EFM) Ethernet Passive Optical
             Network (EPON) Interfaces as defined in IEEE Std 802.3
             Clauses 60, 64, and 65.

             Of particular interest are Clause 64 (Multi-Point Control
             Protocol - MPCP), Clause 65 (Point-to-Multipoint
             Reconciliation Sublayer - P2MP RS), Clause 60 (Ethernet
             Passive Optical Network Physical Medium Dependent - EPON
             PMDs), Clause 30, ‘Management’, and Clause 45, ‘Management
             Data Input/Output (MDIO) Interface’."

    REVISION     "201102020000Z" -- February 2, 2011
    DESCRIPTION
           "Initial version, based on an earlier version published as RFC 4837."

          ::= { org ieee(111) standards-association-numbers-series-standards(2)
                lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 9 }

dot3EponObjects OBJECT IDENTIFIER ::= { ieee8023dot3EponMIB 1}

dot3EponConformance OBJECT IDENTIFIER ::= { ieee8023dot3EponMIB 2}

-- MPCP MIB modules definitions (IEEE Std 802.3, Clause 30.3.5)

dot3EponMpcpObjects
     OBJECT IDENTIFIER ::= { dot3EponObjects 1 }

dot3MpcpControlTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dot3MpcpControlEntry
```



```
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "A Table of dot3 Multi-Point Control Protocol (MPCP)
             MIB objects. The entries in the table are control and
             status objects of the MPCP.
             Each object has a row for every virtual link denoted by
             the corresponding ifIndex.
             The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
             register (15-bit field and a broadcast bit) limiting the
             number of virtual links to 32768. Typically the number
             of expected virtual links in a PON is like the number of
             ONUs, which is 32-64, plus an additional entry for
             broadcast LLID."
    ::= { dot3EponMpcpObjects 1 }

dot3MpcpControlEntry OBJECT-TYPE
    SYNTAX  Dot3MpcpControlEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "An entry in the dot3 MPCP Control table.
             Rows exist for an OLT interface and an ONU interface.
             A row in the table is denoted by the ifIndex of the link
             and it is created when the ifIndex is created.
             The rows in the table for an ONU interface are created
             at system initialization.
             The row in the table corresponding to the OLT ifIndex
             and the row corresponding to the broadcast virtual link
             are created at system initialization.
             A row in the table corresponding to the ifIndex of a
             virtual links is created when a virtual link is
             established (ONU registers) and deleted when the virtual
             link is deleted (ONU deregisters)."
    INDEX  { ifIndex }
    ::= { dot3MpcpControlTable 1}

Dot3MpcpControlEntry ::=
    SEQUENCE {
        dot3MpcpOperStatus                   TruthValue,
        dot3MpcpAdminState                   TruthValue,
        dot3MpcpMode                         INTEGER,
        dot3MpcpSyncTime                     Unsigned32,
        dot3MpcpLinkID                       Unsigned32,
        dot3MpcpRemoteMACAddress             MacAddress,
        dot3MpcpRegistrationState            INTEGER,
        dot3MpcpTransmitElapsed              Unsigned32,
        dot3MpcpReceiveElapsed               Unsigned32,
        dot3MpcpRoundTripTime                Unsigned32,
        dot3MpcpMaximumPendingGrants         Unsigned32
    }

dot3MpcpOperStatus OBJECT-TYPE
    SYNTAX  TruthValue
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "This object reflects the operational state of the
             Multi-Point MAC Control sublayer as defined in
```

```
           IEEE Std 802.3, Clause 64. When the value is true(1), the
           interface will act as if the Multi-Point Control Protocol
           is enabled. When the value is false(2), the interface
           will act as if the Multi-Point Control Protocol is
           disabled. The operational state can be changed using the
           dot3MpcpAdminState object.
           This object is applicable for an OLT, with the same
           value for all virtual interfaces, and for an ONU."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.2."
    ::= { dot3MpcpControlEntry 1 }

dot3MpcpAdminState OBJECT-TYPE
    SYNTAX  TruthValue
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
            "This object is used to define the admin state of the
           Multi-Point MAC Control sublayer, as defined in
           IEEE Std 802.3, Clause 64, and to reflect its state.
           When selecting the value as true(1), the Multi-Point
           Control Protocol of the interface is enabled.
           When selecting the value as false(2), the Multi-Point
           Control Protocol of the interface is disabled.
           This object reflects the administrative state of the
           Multi-Point Control Protocol of the interface.
           The write operation is not restricted in this document
           and can be done at any time. Changing
           dot3MpcpAdminState state can lead to disabling the
           Multi-Point Control Protocol on the respective interface,
           leading to the interruption of service for the users
           connected to the respective EPON interface.
           This object is applicable for an OLT, with the same
           value for all virtual interfaces, and for an ONU."
    REFERENCE   "IEEE Std 802.3, 30.3.5.2.1."
    DEFVAL { false }
    ::= { dot3MpcpControlEntry 2 }

dot3MpcpMode  OBJECT-TYPE
    SYNTAX  INTEGER {
            olt(1),
            onu(2)
    }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "This object is used to identify the operational
           state of the Multi-Point MAC Control sublayer as
           defined in IEEE Std 802.3, Clause 64. Reading olt(1) for an
           OLT (server) mode and onu(2) for an ONU (client) mode.
           This object is used to identify the operational mode
           for the MPCP tables.
           This object is applicable for an OLT, with the same
           value for all virtual interfaces, and for an ONU."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.3."
    DEFVAL { olt }
    ::= { dot3MpcpControlEntry 3 }

dot3MpcpSyncTime OBJECT-TYPE
    SYNTAX  Unsigned32
```

```
    UNITS       "TQ (16 ns)"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that reports the 'sync lock time' of the
             OLT receiver in increments of Time Quanta (TQ)-16ns
             as defined in IEEE Std 802.3, Clauses 60, 64, and 65. The
             value returned shall be (sync lock time ns)/16, rounded up
             to the nearest TQ. If this value exceeds (2^32-1), the
             value (2^32-1) shall be returned. This object is applicable
             for an OLT, with distinct values for all virtual interfaces,
             and for an ONU."
    REFERENCE   "IEEE Std 802.3, 64.3.3.2."
   ::= { dot3MpcpControlEntry 4 }

dot3MpcpLinkID OBJECT-TYPE
    SYNTAX  Unsigned32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that identifies the Logical Link
             Identifier (LLID) associated with the MAC of the virtual
             link as specified in IEEE Std 802.3, 65.1.3.2.2.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             The ONU and the corresponding virtual MAC of the OLT,
             for the same virtual link, have the same value.
             Value is assigned when the ONU registers.
             Value is freed when the ONU deregisters."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.4."
    ::= { dot3MpcpControlEntry 5 }

dot3MpcpRemoteMACAddress OBJECT-TYPE
    SYNTAX  MacAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that identifies the source_address
             parameter of the last MPCPDUs passed to the MAC Control.
             This value is updated on reception of a valid frame with
             1) a destination Field equal to the reserved multicast
             address for MAC Control as specified in [802.3], Annex
             31A; 2) the lengthOrType field value equal to the reserved
             Type for MAC Control as specified in [802.3], Annex
             31A; 3) an MPCP subtype value equal to the subtype
             reserved for MPCP as specified in IEEE Std 802.3, Annex 31A.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             The value reflects the MAC address of the remote entity
             and therefore the OLT holds a value for each LLID, which
             is the MAC address of the ONU; the ONU has a single
             value that is the OLT MAC address."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.5."
    ::= { dot3MpcpControlEntry 6 }

dot3MpcpRegistrationState OBJECT-TYPE
    SYNTAX  INTEGER {
            unregistered(1),
            registering(2),
```



```
            registered(3)
    }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that identifies the registration state
             of the Multi-Point MAC Control sublayer as defined in
             IEEE Std 802.3, Clause 64. When this object has the
             enumeration unregistered(1), the interface is
             unregistered and may be used for registering a link
             partner. When this object has the enumeration
             registering(2), the interface is in the process of
             registering a link-partner. When this object has the
             enumeration registered(3), the interface has an
             established link-partner.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.6."
    ::= { dot3MpcpControlEntry 7 }

dot3MpcpTransmitElapsed OBJECT-TYPE
    SYNTAX  Unsigned32
    UNITS       "TQ (16 ns)"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that reports the interval from the last
             MPCP frame transmission in increments of Time Quanta
             (TQ)-16ns. The value returned shall be (interval from
             last MPCP frame transmission in ns)/16. If this value
             exceeds (2^32-1), the value (2^32-1) shall be returned.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.19."
    ::= { dot3MpcpControlEntry 8 }

dot3MpcpReceiveElapsed OBJECT-TYPE
    SYNTAX  Unsigned32
    UNITS       "TQ (16 ns)"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that reports the interval from last MPCP frame
             reception in increments of Time Quanta (TQ)-16ns. The
             value returned shall be (interval from last MPCP frame
             reception in ns)/16. If this value exceeds (2^32-1), the
             value (2^32-1) shall be returned.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.20."
    ::= { dot3MpcpControlEntry 9 }

dot3MpcpRoundTripTime OBJECT-TYPE
    SYNTAX  Unsigned32 (0..'ffff'h)
    UNITS       "TQ (16 ns)"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that reports the MPCP round trip time in
```

```
             increments of Time Quanta (TQ)-16ns. The value returned
             shall be (round trip time in ns)/16. If this value
             exceeds (2^16-1), the value (2^16-1) shall be returned.
             This object is applicable for an OLT. At the
             OLT, it has a distinct value for each virtual interface."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.21."
    ::= { dot3MpcpControlEntry 10 }

dot3MpcpMaximumPendingGrants OBJECT-TYPE
    SYNTAX  Unsigned32 (0..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that reports the maximum number of grants
             that an ONU can store for handling. The maximum number
             of grants that an ONU can store for handling has a
             range of 0 to 255.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             At the OLT, the value should be zero."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.24."
    ::= { dot3MpcpControlEntry 11 }

dot3MpcpStatTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF Dot3MpcpStatEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "This table defines the list of statistics counters of
             an interface implementing the IEEE Std 802.3, Clause 64 MPCP.
             Each object has a row for every virtual link denoted by
             the corresponding ifIndex.
             The LLID field, as defined in IEEE Std 802.3, is a 2-byte
             register (15-bit field and a broadcast bit) limiting the
             number of virtual links to 32768. Typically the number
             of expected virtual links in a PON is like the number of
             ONUs, which is 32-64, plus an additional entry for
             broadcast LLID."
::= { dot3EponMpcpObjects 2 }

dot3MpcpStatEntry OBJECT-TYPE
    SYNTAX     Dot3MpcpStatEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "An entry in the table of statistics counters of the
             IEEE Std 802.3, Clause 64, MPCP interface.
             Rows exist for an OLT interface and an ONU interface.
             A row in the table is denoted by the ifIndex of the link
             and it is created when the ifIndex is created.
             The rows in the table for an ONU interface are created
             at system initialization.
             The row in the table corresponding to the OLT ifIndex
             and the row corresponding to the broadcast virtual link
             are created at system initialization.
             A row in the table corresponding to the ifIndex of a
             virtual link is created when a virtual link is
             established (ONU registers) and deleted when the virtual
             link is deleted (ONU deregisters)."
```

```
    INDEX  { ifIndex}
    ::= { dot3MpcpStatTable 1 }

Dot3MpcpStatEntry ::=
    SEQUENCE {
            dot3MpcpMACCtrlFramesTransmitted       Counter64,
            dot3MpcpMACCtrlFramesReceived          Counter64,
            dot3MpcpDiscoveryWindowsSent           Counter32,
            dot3MpcpDiscoveryTimeout               Counter32,
            dot3MpcpTxRegRequest                   Counter64,
            dot3MpcpRxRegRequest                   Counter64,
            dot3MpcpTxRegAck                       Counter64,
            dot3MpcpRxRegAck                       Counter64,
            dot3MpcpTxReport                       Counter64,
            dot3MpcpRxReport                       Counter64,
            dot3MpcpTxGate                         Counter64,
            dot3MpcpRxGate                         Counter64,
            dot3MpcpTxRegister                     Counter64,
            dot3MpcpRxRegister                     Counter64
    }

dot3MpcpMACCtrlFramesTransmitted OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of MPCP frames passed to the MAC sublayer for
             transmission. This counter is incremented when a
             MA_CONTROL.request service primitive is generated within
             the MAC control sublayer with an opcode indicating an
             MPCP frame.
             This object is applicable for an OLT and an ONU. At the
             OLT it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system, and at other
             times as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.7."
    ::= { dot3MpcpStatEntry 1 }

dot3MpcpMACCtrlFramesReceived OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of MPCP frames passed by the MAC sublayer to the
             MAC Control sublayer. This counter is incremented when a
             ReceiveFrame function call returns a valid frame with
             1) a lengthOrType field value equal to the reserved
             Type for 802.3_MAC_Control as specified in IEEE Std 802.3
             31.4.1.3, and
             2) an opcode indicating an MPCP frame.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
```

```
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
   REFERENCE   "IEEE Std 802.3, 30.3.5.1.8."
    ::= { dot3MpcpStatEntry 2}

dot3MpcpDiscoveryWindowsSent OBJECT-TYPE
    SYNTAX  Counter32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of discovery windows generated. The counter is
             incremented by one for each generated discovery window.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             At the ONU, the value should be zero.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
   REFERENCE   "IEEE Std 802.3, 30.3.5.1.22."
    ::= { dot3MpcpStatEntry 3}

dot3MpcpDiscoveryTimeout OBJECT-TYPE
    SYNTAX  Counter32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of the number of times a discovery timeout
             occurs. Increment the counter by one for each discovery
             processing state-machine reset resulting from timeout
             waiting for message arrival.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
   REFERENCE   "IEEE Std 802.3, 30.3.5.1.23."
    ::= { dot3MpcpStatEntry 4}

dot3MpcpTxRegRequest OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of the number of times a REGISTER_REQ MPCP
             frame transmission occurs. Increment the counter by one
             for each REGISTER_REQ MPCP frame transmitted as defined
             in IEEE Std 802.3, Clause 64.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             At the OLT, the value should be zero.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
```

```
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.12."
    ::= { dot3MpcpStatEntry 5}

dot3MpcpRxRegRequest OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a REGISTER_REQ MPCP
            frame reception occurs.
            Increment the counter by one for each REGISTER_REQ MPCP
            frame received as defined in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the ONU, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
  REFERENCE   "IEEE Std 802.3, 30.3.5.1.17."
    ::= { dot3MpcpStatEntry 6}

dot3MpcpTxRegAck OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a REGISTER_ACK MPCP
            frame transmission occurs. Increment the counter by one
            for each REGISTER_ACK MPCP frame transmitted as defined
            in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the OLT, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
 REFERENCE   "IEEE Std 802.3, 30.3.5.1.10."
    ::= { dot3MpcpStatEntry 7}

dot3MpcpRxRegAck OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a REGISTER_ACK MPCP
            frame reception occurs.
            Increment the counter by one for each REGISTER_ACK MPCP
            frame received as defined in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
```



```
            At the ONU, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.15."
    ::= { dot3MpcpStatEntry 8}

dot3MpcpTxReport OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a REPORT MPCP frame
            transmission occurs. Increment the counter by one for
            each REPORT MPCP frame transmitted as defined in
            IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the OLT, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.13."
    ::= { dot3MpcpStatEntry 9}

dot3MpcpRxReport OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a REPORT MPCP frame
            reception occurs.
            Increment the counter by one for each REPORT MPCP frame
            received as defined in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the ONU, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.18."
    ::= { dot3MpcpStatEntry 10}

dot3MpcpTxGate OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a GATE MPCP frame
            transmission occurs.
```

```
            Increment the counter by one for each GATE MPCP frame
            transmitted as defined in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the ONU, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.9."
    ::= { dot3MpcpStatEntry 11}

dot3MpcpRxGate OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a GATE MPCP frame
            reception occurs.
            Increment the counter by one for each GATE MPCP frame
            received as defined in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the OLT, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.14."
    ::= { dot3MpcpStatEntry 12}

dot3MpcpTxRegister OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a REGISTER MPCP frame
            transmission occurs.
            Increment the counter by one for each REGISTER MPCP
            frame transmitted as defined in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the ONU, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
   REFERENCE   "IEEE Std 802.3, 30.3.5.1.11."
    ::= { dot3MpcpStatEntry 13}

dot3MpcpRxRegister OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
```

```
    STATUS  current
    DESCRIPTION
            "A count of the number of times a REGISTER MPCP frame
             reception occurs.
             Increment the counter by one for each REGISTER MPCP
             frame received as defined in IEEE Std 802.3, Clause 64.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             At the OLT, the value should be zero.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    REFERENCE   "IEEE Std 802.3, 30.3.5.1.16."
    ::= { dot3MpcpStatEntry 14}

-- Optical Multi Point Emulation (OMPEmulation)
-- managed object definitions

dot3OmpEmulationObjects OBJECT IDENTIFIER ::={dot3EponObjects 2}

dot3OmpEmulationTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dot3OmpEmulationEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "A table of dot3 OmpEmulation MIB objects. The table
             contain objects for the management of the OMPEmulation
             sublayer.
             Each object has a row for every virtual link denoted by
             the corresponding ifIndex.
             The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
             register (15-bit field and a broadcast bit) limiting the
             number of virtual links to 32768. Typically the number
             of expected virtual links in a PON is like the number of
             ONUs, which is 32-64, plus an additional entry for
             broadcast LLID."
    ::= { dot3OmpEmulationObjects 1 }

dot3OmpEmulationEntry OBJECT-TYPE
    SYNTAX  Dot3OmpEmulationEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "An entry in the dot3 OmpEmulation table.
             Rows exist for an OLT interface and an ONU interface.
             A row in the table is denoted by the ifIndex of the link
             and it is created when the ifIndex is created.
             The rows in the table for an ONU interface are created
             at system initialization.
             The row in the table corresponding to the OLT ifIndex
             and the row corresponding to the broadcast virtual link
             are created at system initialization.
             A row in the table corresponding to the ifIndex of a
             virtual links is created when a virtual link is
             established (ONU registers) and deleted when the virtual
             link is deleted (ONU deregisters)."
    INDEX  { ifIndex }
```

```
    ::= { dot3OmpEmulationTable 1 }

    Dot3OmpEmulationEntry ::=
    SEQUENCE {
            dot3OmpEmulationType               INTEGER
    }

dot3OmpEmulationType OBJECT-TYPE
    SYNTAX  INTEGER {
            unknown(1),
            olt(2),
            onu(3)
    }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object that indicates the mode of operation
             of the Reconciliation Sublayer for Point-to-Point
             Emulation (see IEEE Std 802.3, 65.1). unknown(1) value
             is assigned in initialization; true state or type is not
             yet known. olt(2) value is assigned when the sublayer is
             operating in OLT mode. onu(3) value is assigned when the
             sublayer is operating in ONU mode.
             This object is applicable for an OLT, with the same
             value for all virtual interfaces, and for an ONU."
    REFERENCE   "IEEE Std 802.3, 30.3.7.1.2."
    ::= { dot3OmpEmulationEntry 1}

dot3OmpEmulationStatTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF Dot3OmpEmulationStatEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "This table defines the list of statistics counters of
             IEEE Std 802.3, Clause 65, OMPEmulation sublayer.
             Each object has a row for every virtual link denoted by
             the corresponding ifIndex.
             The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
             register (15-bit field and a broadcast bit) limiting the
             number of virtual links to 32768. Typically the number
             of expected virtual links in a PON is like the number of
             ONUs, which is 32-64, plus an additional entry for
             broadcast LLID."
    ::= { dot3OmpEmulationObjects 2}

dot3OmpEmulationStatEntry OBJECT-TYPE
    SYNTAX     Dot3OmpEmulationStatEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "An entry in the table of statistics counters of
             IEEE Std 802.3, Clause 65, OMPEmulation sublayer.
             Rows exist for an OLT interface and an ONU interface.
             A row in the table is denoted by the ifIndex of the link
             and it is created when the ifIndex is created.
             The rows in the table for an ONU interface are created
             at system initialization.
             The row in the table corresponding to the OLT ifIndex
             and the row corresponding to the broadcast virtual link
```



```
            are created at system initialization.
            A row in the table corresponding to the ifIndex of a
            virtual links is created when a virtual link is
            established (ONU registers) and deleted when the virtual
            link is deleted (ONU deregisters)."
    INDEX  { ifIndex}
    ::= { dot3OmpEmulationStatTable 1 }

Dot3OmpEmulationStatEntry::=
    SEQUENCE {
            dot3OmpEmulationSLDErrors                   Counter64,
            dot3OmpEmulationCRC8Errors                  Counter64,
            dot3OmpEmulationBadLLID                     Counter64,
            dot3OmpEmulationGoodLLID                    Counter64,
            dot3OmpEmulationOnuPonCastLLID              Counter64,
            dot3OmpEmulationOltPonCastLLID              Counter64,
            dot3OmpEmulationBroadcastBitNotOnuLlid      Counter64,
            dot3OmpEmulationOnuLLIDNotBroadcast         Counter64,
            dot3OmpEmulationBroadcastBitPlusOnuLlid      Counter64,
            dot3OmpEmulationNotBroadcastBitNotOnuLlid   Counter64
    }

dot3OmpEmulationSLDErrors OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of frames received that do not contain a valid
             SLD field as defined in IEEE Std 802.3, 65.1.3.3.1.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    REFERENCE   "IEEE Std 802.3, 30.3.7.1.3."
    ::= { dot3OmpEmulationStatEntry 1}

dot3OmpEmulationCRC8Errors OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of frames received that contain a valid SLD
             field, as defined in IEEE Std 802.3, 65.1.3.3.1, but do
             not pass the CRC-8 check as defined in IEEE Std 802.3,
             65.1.3.3.3.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    REFERENCE   "IEEE Std 802.3, 30.3.7.1.4."
    ::= { dot3OmpEmulationStatEntry 2}
```



```
dot3OmpEmulationBadLLID OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of frames received that contain a valid SLD
             field, as defined in IEEE Std 802.3, 65.1.3.3.1, and
             pass the CRC-8 check, as defined in IEEE Std 802.3,
             65.1.3.3.3, but are discarded due to the LLID check as
             defined in IEEE Std 802.3, 65.1.3.3.2.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    REFERENCE   "IEEE Std 802.3, 30.3.7.1.8."
    ::= { dot3OmpEmulationStatEntry 3}

dot3OmpEmulationGoodLLID OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of frames received that contain a valid SLD
             field, as defined in IEEE Std 802.3, 65.1.3.3.1, and
             pass the CRC-8 check as defined in IEEE Std 802.3,
             65.1.3.3.3.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    REFERENCE   "IEEE Std 802.3, 30.3.7.1.5."
    ::= { dot3OmpEmulationStatEntry 4}

dot3OmpEmulationOnuPonCastLLID OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of frames received that contain a valid SLD
             field, as defined in IEEE Std 802.3, 65.1.3.3.1,
             pass the CRC-8 check, as defined in IEEE Std 802.3,
             65.1.3.3.3, and meet the rules of acceptance for an
             ONU defined in IEEE Std 802.3, 65.1.3.3.2.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             At the OLT, the value should be zero.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
```

```
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.3.7.1.6."
    ::= { dot3OmpEmulationStatEntry 5}

dot3OmpEmulationOltPonCastLLID OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of frames received that contain a valid SLD
            field, as defined in IEEE Std 802.3, 65.1.3.3.1,
            pass the CRC-8 check, as defined in IEEE Std 802.3,
            65.1.3.3.3, and meet the rules of acceptance for an
            OLT defined in IEEE Std 802.3, 65.1.3.3.2.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the ONU, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
  REFERENCE   "IEEE Std 802.3, 30.3.7.1.7."
    ::= { dot3OmpEmulationStatEntry 6}

dot3OmpEmulationBroadcastBitNotOnuLlid OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of frames received that contain a valid SLD
            field, as defined in IEEE Std 802.3,
            65.1.3.3.1, pass the CRC-8 check, as defined in
            IEEE Std 802.3, 65.1.3.3.3, and contain the broadcast
            bit in the LLID and not the ONU's LLID (frame accepted)
            as defined in IEEE Std 802.3, Clause 65.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            At the OLT, the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
   ::= { dot3OmpEmulationStatEntry 7}

dot3OmpEmulationOnuLLIDNotBroadcast OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of frames received that contain a valid SLD
            field, as defined in IEEE Std 802.3,
            65.1.3.3.1, pass the CRC-8 check, as defined in
            IEEE Std 802.3, 65.1.3.3.3, and contain the ONU's LLID
```



```
             as defined in IEEE Std 802.3, Clause 65.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             At the OLT, the value should be zero.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    ::= { dot3OmpEmulationStatEntry 8}

dot3OmpEmulationBroadcastBitPlusOnuLlid OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of frames received that contain a valid SLD
             field, as defined in IEEE Std 802.3,
             65.1.3.3.1, pass the CRC-8 check, as defined in
             IEEE Std 802.3, 65.1.3.3.3, and contain the broadcast
             bit in the LLID and match the ONU's LLID (frame
             reflected) as defined in IEEE Std 802.3, Clause 65.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             At the OLT, the value should be zero.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    ::= { dot3OmpEmulationStatEntry 9}

dot3OmpEmulationNotBroadcastBitNotOnuLlid OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A count of frames received that contain a valid SLD
             field, as defined in IEEE Std 802.3,
             65.1.3.3.1, pass the CRC-8 check, as defined in
             IEEE Std 802.3, 65.1.3.3.3, and do not contain
             the ONU's LLID as defined in IEEE Std 802.3, Clause 65.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             At the OLT, the value should be zero.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    ::= { dot3OmpEmulationStatEntry 10}

--  FEC managed object definitions (30.5.1)

dot3EponFecObjects OBJECT IDENTIFIER ::={dot3EponObjects 3}

dot3EponFecTable OBJECT-TYPE
```

```
    SYNTAX  SEQUENCE OF Dot3EponFecEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "A table of dot3 EPON FEC management objects.
             The entries in the table are control and status objects
             and statistic counters for the FEC layer.
             Each object has a row for every virtual link denoted by
             the corresponding ifIndex.
             The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
             register (15-bit field and a broadcast bit) limiting the
             number of virtual links to 32768. Typically the number
             of expected virtual links in a PON is like the number of
             ONUs, which is 32-64, plus an additional entry for
             broadcast LLID."
    ::= { dot3EponFecObjects 1 }

dot3EponFecEntry OBJECT-TYPE
    SYNTAX  Dot3EponFecEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "An entry in the dot3 EPON FEC table.
             Rows exist for an OLT interface and an ONU interface.
             A row in the table is denoted by the ifIndex of the link
             and it is created when the ifIndex is created.
             The rows in the table for an ONU interface are created
             at system initialization.
             The row in the table corresponding to the OLT ifIndex
             and the row corresponding to the broadcast virtual link
             are created at system initialization.
             A row in the table corresponding to the ifIndex of a
             virtual links is created when a virtual link is
             established (ONU registers) and deleted when the virtual
             link is deleted (ONU deregisters)."
    INDEX  { ifIndex}
    ::= { dot3EponFecTable 1 }

Dot3EponFecEntry ::=
    SEQUENCE {
            dot3EponFecPCSCodingViolation           Counter64,
            dot3EponFecAbility                      INTEGER,
            dot3EponFecMode                         INTEGER,
            dot3EponFecCorrectedBlocks              Counter64,
            dot3EponFecUncorrectableBlocks          Counter64,
            dot3EponFecBufferHeadCodingViolation    Counter64
    }

dot3EponFecPCSCodingViolation OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "octets"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "For a 100 Mb/s operation, it is a count of the number of
             times an invalid code-group is received, other than the
             /H/ code-group. For a 1000 Mb/s operation, it is a count
             of the number of times an invalid codegroup is received,
             other than the /V/ code-group. /H/ denotes a special
```

```
            4b5b codeword of the IEEE Std 802.3 Clause 24 100 Mb/s PCS layer,
            and /V/ denotes a special 8b10b codeword of the IEEE Std 802.3
            Clause 36 1000 Mb/s PCS layer.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.12."
    ::= { dot3EponFecEntry 1}

dot3EponFecAbility OBJECT-TYPE
    SYNTAX  INTEGER {
            unknown(1),
            supported(2),
            unsupported(3)
    }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "An object that indicates the support of operation of the
            optional FEC sublayer of the 1000BASE-PX PHY specified
            in IEEE Std 802.3, 65.2.
            unknown(1) value is assigned in the initialization, for non
            FEC support state or type not yet known. unsupported(3)
            value is assigned when the sublayer is not supported.
            supported(2) value is assigned when the sublayer is
            supported.
            This object is applicable for an OLT, with the same
            value for all virtual interfaces, and for an ONU.
            The FEC counters will have a zero value when the
            interface is not supporting FEC.
            The counters:
             dot3EponFecPCSCodingViolation - not affected by FEC
             ability.
             dot3EponFecCorrectedBlocks    - has a zero value when
              dot3EponFecAbility is unknown(1) and unsupported(3).
             dot3EponFecUncorrectableBlocks  - has a zero value when
              dot3EponFecAbility is unknown(1) and unsupported(3).
             dot3EponFecBufferHeadCodingViolation - has a zero value
              when dot3EponFecAbility is unknown(1) and
              unsupported(3)."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.13."
    ::= { dot3EponFecEntry 2}

dot3EponFecMode OBJECT-TYPE
    SYNTAX  INTEGER {
            unknown(1),
            disabled(2),
            enabled(3)
    }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
           "An object that defines the mode of operation of the
            optional FEC sublayer of the 1000BASE-PX PHY, specified
            in IEEE Std 802.3, 65.2, and reflects its state.
```

```
            A GET operation returns the current mode of operation
            of the PHY. A SET operation changes the mode of
            operation of the PHY to the indicated value.
            unknown(1) value is assigned in the initialization for non
              FEC support state or type not yet known.
            disabled(2) value is assigned when the FEC sublayer is
              operating in disabled mode.
            enabled(3) value is assigned when the FEC sublayer is
              operating in FEC mode.
            The write operation is not restricted in this document
            and can be done at any time. Changing dot3EponFecMode
            state can lead to disabling the Forward Error Correction
            on the respective interface, which can lead to a
            degradation of the optical link, and therefore may lead
            to an interruption of service for the users connected to
            the respective EPON interface.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            The counting of
            the FEC counters will stop when the FEC of the interface
            is disabled.
            The counters:
             dot3EponFecPCSCodingViolation - not affected by FEC
             mode.
             dot3EponFecCorrectedBlocks - stops counting when
             Rx_FEC is not enabled. (unknown(1) and disabled(2)).
             dot3EponFecUncorrectableBlocks - stops counting when
             Rx_FEC is not enabled (unknown(1) and disabled(2)).
             dot3EponFecBufferHeadCodingViolation - stops counting
             when Rx_FEC is not enabled (unknown(1) and
             disabled(2)).
            The object:
             dot3EponFecAbility - indicates the FEC ability and
             is not affected by the dot3EponFecMode object."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.14."
    DEFVAL { unknown }
    ::= { dot3EponFecEntry 3}

dot3EponFecCorrectedBlocks OBJECT-TYPE
    SYNTAX  Counter64
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "For 10PASS-TS, 2BASE-TL, and 1000BASE-PX PHYs, it is a
            count of corrected FEC blocks. This counter will not
            increment for other PHY Types. Increment the counter by
            one for each received block that is corrected by the FEC
            function in the PHY.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.15."
    ::= { dot3EponFecEntry 4}

dot3EponFecUncorrectableBlocks OBJECT-TYPE
```



```
    SYNTAX  Counter64
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "For 10PASS-TS, 2BASE-TL, and 1000BASE-PX PHYs, it is a
             count of uncorrectable FEC blocks. This counter will not
             increment for other PHY Types. Increment the counter by
             one for each FEC block that is determined to be
             uncorrectable by the FEC function in the PHY.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.16."
    ::= { dot3EponFecEntry 5}

dot3EponFecBufferHeadCodingViolation OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "octets"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "For a 1000 Mb/s operation, it is a count of the number of
             invalid code-group received directly from the link. The
             value has a meaning only in 1000 Mb/s mode and it is
             zero otherwise.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
    ::= { dot3EponFecEntry 6}

-- ExtendedPackage managed object definitions

dot3ExtPkgObjects OBJECT IDENTIFIER ::={dot3EponObjects 4}

dot3ExtPkgControlObjects OBJECT IDENTIFIER ::= { dot3ExtPkgObjects 1}

dot3ExtPkgControlTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dot3ExtPkgControlEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "A table of Extended package Control management
             objects. Entries in the table are control and status
             indication objects of an EPON interface, which are
             gathered in an extended package as an addition to the
             objects based on the IEEE Std 802.3, Clause 30, attributes.
             Each object has a row for every virtual link denoted by
             the corresponding ifIndex.
             The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
             register (15-bit field and a broadcast bit) limiting the
             number of virtual links to 32768. Typically the number
```



```
              of expected virtual links in a PON is like the number of
              ONUs, which is 32-64, plus an additional entry for
              broadcast LLID."
    ::= { dot3ExtPkgControlObjects 1 }

dot3ExtPkgControlEntry OBJECT-TYPE
    SYNTAX  Dot3ExtPkgControlEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "An entry in the Extended package Control table.
             Rows exist for an OLT interface and an ONU interface.
             A row in the table is denoted by the ifIndex of the link
             and it is created when the ifIndex is created.
             The rows in the table for an ONU interface are created
             at system initialization.
             The row in the table corresponding to the OLT ifIndex
             and the row corresponding to the broadcast virtual link
             are created at system initialization.
             A row in the table corresponding to the ifIndex of a
             virtual links is created when a virtual link is
             established (ONU registers) and deleted when the virtual
             link is deleted (ONU deregisters)."
    INDEX  { ifIndex}
    ::= { dot3ExtPkgControlTable 1 }

Dot3ExtPkgControlEntry ::=
    SEQUENCE {
     dot3ExtPkgObjectReset                       INTEGER,
     dot3ExtPkgObjectPowerDown                   TruthValue,
     dot3ExtPkgObjectNumberOfLLIDs               Unsigned32,
     dot3ExtPkgObjectFecEnabled                  INTEGER,
     dot3ExtPkgObjectReportMaximumNumQueues      Unsigned32,
     dot3ExtPkgObjectRegisterAction              INTEGER
    }

dot3ExtPkgObjectReset OBJECT-TYPE
    SYNTAX  INTEGER {
            running(1),
            reset(2)
    }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
            "This object is used to reset the EPON interface. The
             interface may be unavailable while the reset occurs and
             data may be lost.
             Setting this object to running(1) will cause the
             interface to enter into running mode. Setting this
             object to reset(2) will cause the interface to go into
             reset mode. When getting running(1), the interface is in
             running mode. When getting reset(2), the interface is in
             reset mode.
             The write operation is not restricted in this document
             and can be done at any time. Changing
             dot3ExtPkgObjectReset state can lead to a reset of the
             respective interface, leading to an interruption of
             service for the users connected to the respective EPON
             interface.
```

```
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             A reset for a specific virtual interface resets only
             this virtual interface and not the physical interface.
             Thus, a virtual link that is malfunctioning can be
             reset without affecting the operation of other virtual
             interfaces.
             The reset can cause Discontinuities in the values of the
             counters of the interface, similar to re-initialization
             of the management system. Discontinuity should be
             indicated by the ifCounterDiscontinuityTime object of
             the Interfaces Group MIB module."
    DEFVAL { running }
    ::= { dot3ExtPkgControlEntry 1 }

dot3ExtPkgObjectPowerDown OBJECT-TYPE
    SYNTAX  TruthValue
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
            "This object is used to power down the EPON interface.
             The interface may be unavailable while the power down
             occurs and data may be lost.
             Setting this object to true(1) will cause the interface
             to enter into power down mode. Setting this object to
             false(2) will cause the interface to go out of power
             down mode. When getting true(1), the interface is in
             power down mode. When getting false(2), the interface is
             not in power down mode.
             The write operation is not restricted in this document
             and can be done at any time. Changing
             dot3ExtPkgObjectPowerDown state can lead to a power down
             of the respective interface, leading to an interruption
             of service of the users connected to the respective EPON
             interface.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface.
             A power down/up of a specific virtual interface affects
             only the virtual interface and not the physical
             interface. Hence a virtual link, which needs a certain
             handling, can be powered down and then powered up without
             disrupting the operation of other virtual interfaces.
             The object is relevant when the admin state of the
             interface is active as set by the dot3MpcpAdminState."
    DEFVAL { false }
    ::= { dot3ExtPkgControlEntry 2 }

dot3ExtPkgObjectNumberOfLLIDs OBJECT-TYPE
    SYNTAX  Unsigned32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "A read only object that indicates the number of
             registered LLIDs. The initialization value is 0.
             This object is applicable for an OLT with the same
             value for all virtual interfaces and for an ONU.
             The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
             register (15-bit field and a broadcast bit) limiting the
             number of virtual links to 32768. Typically the number
```

```
             of expected virtual links in a PON is like the number of
             ONUs, which is 32-64, plus an additional entry for
             broadcast LLID. At the ONU the
             number of LLIDs for an interface is one."
    ::= { dot3ExtPkgControlEntry 3 }

dot3ExtPkgObjectFecEnabled OBJECT-TYPE
    SYNTAX  INTEGER {
            noFecEnabled(1),
            fecTxEnabled(2),
            fecRxEnabled(3),
            fecTxRxEnabled(4)
    }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
           "An object defining the FEC mode of operation of the
            interface, and indicating its state. The modes defined in
            this object are extensions to the FEC modes defined in
            the dot3EponFecMode object.
            When noFECEnabled(1), the interface does not enable FEC
            mode.
            When fecTxEnabled(2), the interface enables the FEC
            transmit mode.
            When fecRxEnabled(3), the interface enables the FEC
            receive mode.
            When fecTxRxEnabled(4), the interface enables the FEC
            transmit and receive mode.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface.
            The FEC counters are referring to the receive path. The
            FEC counters will stop when the FEC receive mode of the
            interface is disabled, as defined by fecRxEnabled(3)
            and fecTxRxEnabled(4) values.
            The counters:
             dot3EponFecPCSCodingViolation - not affected by FEC
             mode.
             dot3EponFecCorrectedBlocks - stops counting when
             Rx_FEC is not enabled (noFecEnabled(1) and
             fecTxEnabled(2)).
             dot3EponFecUncorrectableBlocks - stops counting when
             Rx_FEC is not enabled (noFecEnabled(1) and
             fecTxEnabled(2)).
             dot3EponFecBufferHeadCodingViolation - stops counting
             when Rx_FEC is not enabled (noFecEnabled(1) and
             fecTxEnabled(2)).
            The objects:
             dot3EponFecAbility - indicates the FEC ability and is
             not affected by the FEC mode.
             dot3EponFecMode - indicates the FEC mode for combined RX
             and TX.
            The write operation is not restricted in this document
            and can be done at any time. Changing
            dot3ExtPkgObjectFecEnabled state can lead to disabling
            the Forward Error Correction on the respective interface,
            which can lead to a degradation of the optical link, and
            therefore may lead to an interruption of service for the
            users connected to the respective EPON interface."
    DEFVAL { noFecEnabled }
```



```
    ::= { dot3ExtPkgControlEntry 4 }

dot3ExtPkgObjectReportMaximumNumQueues OBJECT-TYPE
    SYNTAX  Unsigned32 (0..7)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "An object, that defines the maximal number of queues in
             the REPORT message as defined in IEEE Std 802.3, Clause 64. For
             further information please see the description of the
             queue table.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface."
    DEFVAL { 0 }
    ::= { dot3ExtPkgControlEntry 5 }

dot3ExtPkgObjectRegisterAction OBJECT-TYPE
    SYNTAX  INTEGER {
            none(1),
            register(2),
            deregister(3),
            reregister(4)
    }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
           "An object configuring the registration state of an
           interface, and indicating its registration state.
            Write operation changes the registration state to its new
            value.
            Read operation returns the value of the state.
            The registration state is reflected in this object and in
            the dot3MpcpRegistrationState object.
            none(1) indicates an unknown state,
            register(2) indicates a registered LLID,
            deregister(3) indicates a deregistered LLID,
            reregister(4) indicates an LLID that is reregistering.
            The following list describes the operation of the
            interface, as specified in the IEEE Std 802.3, when a write
            operation is setting a value.
             none(1) - not doing any action.
             register(2) - registering an LLID that has been requested
             for registration (The LLID is in registering mode.
               dot3MpcpRegistrationState - registering(2) ).
               deregister(3) - deregisters an LLID that is registered
                 (dot3MpcpRegistrationState - registered(3) ).
               reregister(4) - reregister an LLID that is registered
                 (dot3MpcpRegistrationState - registered(3) ).
               The behavior of an ONU and OLT interfaces, at each one
               of the detailed operation at each state, is described in
               the registration state machine of figure 64-22,
               IEEE Std 802.3.
              This object is applicable for an OLT and an ONU. At the
              OLT, it has a distinct value for each virtual interface.
              The write operation is not restricted in this document
              and can be done at any time. Changing
              dot3ExtPkgObjectRegisterAction state can lead to a change
              in the registration state of the respective interface
              leading to a deregistration and an interruption of
```

```
               service of the users connected to the respective EPON
               interface."
       DEFVAL { none }
       ::= { dot3ExtPkgControlEntry 6 }

   dot3ExtPkgQueueTable OBJECT-TYPE
       SYNTAX  SEQUENCE OF Dot3ExtPkgQueueEntry
       MAX-ACCESS  not-accessible
       STATUS  current
       DESCRIPTION
               "A table of the extended package objects for queue
                management. The IEEE Std 802.3 MPCP defines a report message
                of the occupancy of the transmit queues for the feedback
                BW request from the ONUs. These queues serve the uplink
                transmission of the ONU and data is gathered there until
                the ONU is granted for transmission.
                The management table of the queues is added here mainly
                to control the reporting and to gather some statistics
                of their operation. This table is not duplicating
                existing management objects of bridging queues,
                specified in IEEE Std 802.1D, since the existence of a
                dedicated transmit queuing mechanism is implied in the
                IEEE Std 802.3, and the ONU may be a device that is not a
                bridge with embedded bridging queues.
                The format of the REPORT message, as specified
                in IEEE Std 802.3, is presented below:
                +------------------------------------+
                |          Destination Address       |
                +------------------------------------+
                |          Source Address            |
                +------------------------------------+
                |          Length/Type               |
                +------------------------------------+
                |          OpCode                    |
                +------------------------------------+
                |          TimeStamp                 |
                +------------------------------------+
                |          Number of queue Sets      |
                +------------------------------------+   /|\
                |          Report bitmap             |    |
                +------------------------------------+    |
                |          Queue 0 report            |    |
                +------------------------------------+    | repeated for
                |          Queue 1 report            |    | every
                +------------------------------------+    | queue_set
                |          Queue 2 report            |    |
                +------------------------------------+    |
                |          Queue 3 report            |    |
                +------------------------------------+    |
                |          Queue 4 report            |    |
                +------------------------------------+    |
                |          Queue 5 report            |    |
                +------------------------------------+    |
                |          Queue 6 report            |    |
                +------------------------------------+    |
                |          Queue 7 report            |    |
                +------------------------------------+   \|/
                |          Pad/reserved              |
                +------------------------------------+
```



```
            |          FCS                    |
            +---------------------------------+

            The 'Queue report' field reports the occupancy of each
            uplink transmission queue.
            The number of queue sets defines the number of the
            reported sets, as would be explained in the description
            of the dot3ExtPkgQueueSetsTable table. For each set the
            report bitmap defines which queue is present in the
            report, meaning that although the MPCP REPORT message
            can report up to 8 queues in a REPORT message, the
            actual number is flexible. The Queue table has a
            variable size that is limited by the
            dot3ExtPkgObjectReportMaximumNumQueues object, as an
            ONU can have fewer queues to report.
            The entries in the table are control and status
            indication objects for managing the queues of an EPON
            interface that are gathered in an extended package as
            an addition to the objects that are based on the
            IEEE Std 802.3 attributes.
            Each object has a row for every virtual link and for
            every queue in the report.
            The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
            register (15-bit field and a broadcast bit) limiting the
            number of virtual links to 32768. Typically the number
            of expected virtual links in a PON is like the number of
            ONUs, which is 32-64, plus an additional entry for
            broadcast LLID.
            The number of queues is between 0 and 7 and limited by
            dot3ExtPkgObjectReportMaximumNumQueues."
    ::= { dot3ExtPkgControlObjects 2 }

dot3ExtPkgQueueEntry OBJECT-TYPE
    SYNTAX  Dot3ExtPkgQueueEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
           "An entry in the Extended package Queue table. At the
            OLT, the rows exist for each ifIndex and dot3QueueIndex.
            At the ONU, rows exist for the single ifIndex for each
            dot3QueueIndex.
            Rows in the table are created when the ifIndex of the
            link is created. A set of rows per queue are added for
            each ifIndex, denoted by the dot3QueueIndex.
            A set of rows per queue in the table, for an ONU
            interface, are created at the system initialization.
            A set of rows per queue in the table, corresponding to
            the OLT ifIndex and a set of rows per queue
            corresponding to the broadcast virtual link, are
            created at the system initialization.
            A set of rows per queue in the table, corresponding to
            the ifIndex of a virtual link, are created when the
            virtual link is established (ONU registers), and deleted
            when the virtual link is deleted (ONU deregisters)."
    INDEX  { ifIndex, dot3QueueIndex }
    ::= { dot3ExtPkgQueueTable 1 }

Dot3ExtPkgQueueEntry ::=
    SEQUENCE {
```

```
      dot3QueueIndex                                Unsigned32,
      dot3ExtPkgObjectReportNumThreshold             Unsigned32,
      dot3ExtPkgObjectReportMaximumNumThreshold      Unsigned32,
      dot3ExtPkgStatTxFramesQueue                    Counter64,
      dot3ExtPkgStatRxFramesQueue                    Counter64,
      dot3ExtPkgStatDroppedFramesQueue               Counter64
     }

 dot3QueueIndex OBJECT-TYPE
     SYNTAX  Unsigned32 (0..7)
     MAX-ACCESS  not-accessible
     STATUS  current
     DESCRIPTION
             "An object that identifies an index for the queue table
              reflecting the queue index of the queues that are
              reported in the MPCP REPORT message as defined in
              IEEE Std 802.3, Clause 64.
              The number of queues is between 0 and 7, and limited by
              dot3ExtPkgObjectReportMaximumNumQueues."
     ::= { dot3ExtPkgQueueEntry 1 }

 dot3ExtPkgObjectReportNumThreshold OBJECT-TYPE
     SYNTAX  Unsigned32 (0..7)
     MAX-ACCESS  read-write
     STATUS  current
     DESCRIPTION
             "An object that defines the number of thresholds for each
              queue in the REPORT message as defined in IEEE Std 802.3,
              Clause 64.
              Each queue_set reporting will provide information on the
              queue occupancy of frames below the matching Threshold.
              Read operation reflects the number of thresholds.
              Write operation sets the number of thresholds for each
              queue.
              The write operation is not restricted in this document
              and can be done at any time. Value cannot exceed the
              maximal value defined by the
              dot3ExtPkgObjectReportMaximumNumThreshold object.
              Changing dot3ExtPkgObjectReportNumThreshold can lead to
              a change in the reporting of the ONU interface and
              therefore to a change in the bandwidth allocation of the
              respective interface. This change may lead a degradation
              or an interruption of service of the users connected to
              the respective EPON interface.
              This object is applicable for an OLT and an ONU. At the
              OLT, it has a distinct value for each virtual interface
              and for each queue. At the ONU, it has a distinct value
              for each queue."
     DEFVAL { 0 }
     ::= { dot3ExtPkgQueueEntry 2 }

 dot3ExtPkgObjectReportMaximumNumThreshold OBJECT-TYPE
     SYNTAX  Unsigned32 (0..7)
     MAX-ACCESS  read-only
     STATUS  current
     DESCRIPTION
             "An object, that defines the maximal number of thresholds
              for each queue in the REPORT message as defined in
              IEEE Std 802.3, Clause 64. Each queue_set reporting will
```



```
            provide information on the queue occupancy of frames
            below the matching Threshold.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface
            and for each queue. At the ONU, it has a distinct value
            for each queue."
    DEFVAL { 0 }
    ::= { dot3ExtPkgQueueEntry 3 }

 dot3ExtPkgStatTxFramesQueue OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a frame transmission
            occurs from the corresponding ‘Queue’.
            Increment the counter by one for each frame transmitted,
            which is an output of the ‘Queue’.
            The ‘Queue’ marking matches the REPORT MPCP message
            Queue field as defined in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface
            and for each queue. At the ONU, it has a distinct value
            for each queue.
            At the OLT the value should be zero.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    ::= { dot3ExtPkgQueueEntry 4}

dot3ExtPkgStatRxFramesQueue OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
           "A count of the number of times a frame reception
            occurs from the corresponding ‘Queue’.
            Increment the counter by one for each frame received,
            which is an input to the corresponding ‘Queue’.
            The ‘Queue’ marking matches the REPORT MPCP message
            Queue field as defined in IEEE Std 802.3, Clause 64.
            This object is applicable for an OLT and an ONU. At the
            OLT, it has a distinct value for each virtual interface
            and for each queue. At the ONU, it has a distinct value
            for each queue.
            Discontinuities of this counter can occur at
            re-initialization of the management system and at other
            times, as indicated by the value of the
            ifCounterDiscontinuityTime object of the Interfaces Group MIB
            module."
    ::= { dot3ExtPkgQueueEntry 5}

dot3ExtPkgStatDroppedFramesQueue OBJECT-TYPE
    SYNTAX  Counter64
    UNITS      "frames"
```

```
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
             "A count of the number of times a frame drop
             occurs from the corresponding ‘Queue’.
             Increment the counter by one for each frame dropped
             from the corresponding ‘Queue’.
             The ‘Queue’ marking matches the REPORT MPCP message
             Queue field as defined in IEEE Std 802.3, Clause 64.
             This object is applicable for an OLT and an ONU. At the
             OLT, it has a distinct value for each virtual interface
             and for each queue. At the ONU, it has a distinct value
             for each queue.
             At the OLT, the value should be zero.
             Discontinuities of this counter can occur at
             re-initialization of the management system and at other
             times, as indicated by the value of the
             ifCounterDiscontinuityTime object of the Interfaces Group MIB
             module."
   ::= { dot3ExtPkgQueueEntry 6}

dot3ExtPkgQueueSetsTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dot3ExtPkgQueueSetsEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
            "A table of Extended package objects used for the
             management of the queue_sets. Entries are control and
             status indication objects of an EPON interface, which
             are gathered in an extended package as an addition to
             the objects based on the IEEE Std 802.3 attributes. The
             objects in this table are specific for the queue_sets,
             which are reported in the MPCP REPORT message as defined
             in IEEE Std 802.3, Clause 64.
             The IEEE Std 802.3 MPCP defines a report message of the
             occupancy of the transmit queues for the feedback BW
             request from the ONUs. These queues serve the uplink
             transmission of the ONU and data is gathered there until
             the ONU is granted for transmission.
             The management table of the queues_sets is added here
             mainly to control the reporting and to gather some
             statistics of their operation. This table is not
             duplicating existing management objects of bridging
             queues, specified in IEEE Std 802.1D, since the existence of a
             dedicated transmit queuing mechanism is implied in the
             IEEE Std 802.3, and the ONU may be a device that is not a
             bridge with embedded bridging queues.
             The format of the REPORT message, as specified
             in IEEE Std 802.3, is presented below:
             +-----------------------------------+
             |          Destination Address      |
             +-----------------------------------+
             |          Source Address           |
             +-----------------------------------+
             |          Length/Type              |
             +-----------------------------------+
             |          OpCode                   |
             +-----------------------------------+
             |          TimeStamp                |
```

```
+----------------------------------+
|          Number of queue Sets    |
+----------------------------------+   /|\
|          Report bitmap           |    |
+----------------------------------+    |
|          Queue 0 report          |    |
+----------------------------------+    | repeated for
|          Queue 1 report          |    | every
+----------------------------------+    | queue_set
|          Queue 2 report          |    |
+----------------------------------+    |
|          Queue 3 report          |    |
+----------------------------------+    |
|          Queue 4 report          |    |
+----------------------------------+    |
|          Queue 5 report          |    |
+----------------------------------+    |
|          Queue 6 report          |    |
+----------------------------------+    |
|          Queue 7 report          |    |
+----------------------------------+   \|/
|          Pad/reserved            |
+----------------------------------+
|          FCS                     |
+----------------------------------+
```

```
As can be seen from the message format, the ONU
interface reports of the status of up to 8 queues
and it can report in a single MPCP REPORT message
of a few sets of queues.
The number of queue_sets defines the number of the
reported sets, and it can reach a value of up to 8.
It means that an ONU can hold a variable number of
sets between 0 and 7.
The dot3ExtPkgQueueSetsTable table has a variable
queue_set size that is limited by the
dot3ExtPkgObjectReportMaximumNumThreshold object as an
ONU can have fewer queue_sets to report.
The 'Queue report' field reports the occupancy of each
uplink transmission queue. The queue_sets can be used to
report the occupancy of the queues in a few levels as to
allow granting, in an accurate manner, of only part of
the data available in the queues. A Threshold is
defined for each queue_set to define the level of the
queue that is counted for the report of the occupancy.
The threshold is reflected in the queue_set table by the
dot3ExtPkgObjectReportThreshold object.
For each queue set, the report bitmap defines which
queues are present in the report, meaning that
although the MPCP REPORT message can report of up to 8
queues in a REPORT message, the actual number is
flexible.
The dot3ExtPkgQueueSetsTable table has a variable queue
size that is limited by the
dot3ExtPkgObjectReportMaximumNumQueues object as an ONU
can have fewer queues to report.
Each object has a row for every virtual link, for each
queue in the report and for each queue_set in the queue.
The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
```

```
            register (15-bit field and a broadcast bit) limiting the
            number of virtual links to 32768. Typically the number
            of expected virtual links in a PON is like the number of
            ONUs, which is 32-64, plus an additional entry for
            broadcast LLID.
            The number of queues is between 0 and 7 and limited by
            dot3ExtPkgObjectReportMaximumNumQueues.
            The number of queues_sets is between 0 and 7 and limited
            by dot3ExtPkgObjectReportMaximumNumThreshold."
    ::= { dot3ExtPkgControlObjects 3 }

dot3ExtPkgQueueSetsEntry OBJECT-TYPE
    SYNTAX  Dot3ExtPkgQueueSetsEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
           "An entry in the Extended package queue_set table. At
            the OLT, the rows exist for each ifIndex,
            dot3QueueSetQueueIndex and dot3QueueSetIndex. At the
            ONU, rows exist for the single ifIndex, for each
            dot3QueueSetQueueIndex and dot3QueueSetIndex.
            Rows in the table are created when the ifIndex of the
            link is created. A set of rows per queue and per
            queue_set are added for each ifIndex, denoted by
            dot3QueueSetIndex and dot3QueueSetQueueIndex.
            A set of rows per queue and per queue_set in the table,
            for an ONU interface are created at system
            initialization.
            A set of rows per queue and per queue_Set in the table,
            corresponding to the OLT ifIndex and a set of rows per
            queue and per queue_set, corresponding to the broadcast
            virtual link, are created at system initialization.
            A set of rows per queue and per queue_set in the table,
            corresponding to the ifIndex of a virtual link are
            created when the virtual link is established (ONU
            registers) and deleted when the virtual link is deleted
            (ONU deregisters)."
            INDEX { ifIndex,
           dot3QueueSetQueueIndex,dot3QueueSetIndex}
    ::= { dot3ExtPkgQueueSetsTable 1 }

Dot3ExtPkgQueueSetsEntry ::=
    SEQUENCE {
     dot3QueueSetQueueIndex                      Unsigned32,
     dot3QueueSetIndex                           Unsigned32,
     dot3ExtPkgObjectReportThreshold             Unsigned32
    }

dot3QueueSetQueueIndex OBJECT-TYPE
    SYNTAX  Unsigned32 (0..7)
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
           "An object that identifies the queue index for the
            dot3ExtPkgQueueSetsTable table. The queues are reported
            in the MPCP REPORT message as defined in IEEE Std 802.3,
            Clause 64.
            The number of queues is between 0 and 7, and limited by
            dot3ExtPkgObjectReportMaximumNumQueues.
```



```
              Value corresponds to the dot3QueueIndex of the queue
              table."
     ::= { dot3ExtPkgQueueSetsEntry 1 }

 dot3QueueSetIndex OBJECT-TYPE
     SYNTAX  Unsigned32 (0..7)
     MAX-ACCESS  not-accessible
     STATUS  current
     DESCRIPTION
             "An object that identifies the queue_set index for the
              dot3ExtPkgQueueSetsTable table. The queues are reported
              in the MPCP REPORT message as defined in IEEE Std 802.3,
              Clause 64.
              The number of queues_sets is between 0 and 7, and
              limited by dot3ExtPkgObjectReportMaximumNumThreshold."
     ::= { dot3ExtPkgQueueSetsEntry 2 }

     dot3ExtPkgObjectReportThreshold OBJECT-TYPE
     SYNTAX  Unsigned32
     UNITS       "TQ (16 ns)"
     MAX-ACCESS  read-write
     STATUS  current
     DESCRIPTION
             "An object that defines the value of a threshold report
              for each queue_set in the REPORT message as defined in
              IEEE Std 802.3, Clause 64. The number of sets for each queue
              is dot3ExtPkgObjectReportNumThreshold.
              In the REPORT message, each queue_set reporting will
              provide information on the occupancy of the queues for
              frames below the matching Threshold.
              The value returned shall be in Time quanta (TQ), which
              is 16 ns or 2 octets increments.
              Read operation provides the threshold value. Write
              operation sets the value of the threshold.
              The write operation is not restricted in this document
              and can be done at any time. Changing
              dot3ExtPkgObjectReportThreshold can lead to a change in
              the reporting of the ONU interface and therefore to a
              change in the bandwidth allocation of the respective
              interface. This change may lead a degradation or an
              interruption of service for the users connected to the
              respective EPON interface.
              This object is applicable for an OLT and an ONU. At the
              OLT, it has a distinct value for each virtual interface,
              for each queue and for each queue_set. At the ONU, it has
              a distinct value for each queue and for each queue_set."
     DEFVAL { 0 }
     ::= { dot3ExtPkgQueueSetsEntry 3 }

 --Optical Interface status tables

 dot3ExtPkgOptIfTable OBJECT-TYPE
     SYNTAX     SEQUENCE OF Dot3ExtPkgOptIfEntry
     MAX-ACCESS not-accessible
     STATUS     current
     DESCRIPTION
             "This table defines the control and status indication
              objects for the optical interface of the EPON interface.
              Each object has a row for every virtual link denoted by
```



```
            the corresponding ifIndex.
            The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
            register (15-bit field and a broadcast bit) limiting the
            number of virtual links to 32768. Typically the number
            of expected virtual links in a PON is like the number of
            ONUs, which is 32-64, plus an additional entry for
            broadcast LLID.
            Although the optical interface is a physical interface,
            there is a row in the table for each virtual interface.
            The reason for having a separate row for each virtual
            link is that the OLT has a separate link for each one of
            the ONUs. For instance, ONUs could be in different
            distances with different link budgets and different
            receive powers, therefore having different power alarms.
            It is quite similar to a case of different physical
            interfaces."
    ::= { dot3ExtPkgControlObjects 5}

dot3ExtPkgOptIfEntry OBJECT-TYPE
    SYNTAX     Dot3ExtPkgOptIfEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "An entry in the optical interface table of the EPON
            interface.
            Rows exist for an OLT interface and an ONU interface.
            A row in the table is denoted by the ifIndex of the link
            and it is created when the ifIndex is created.
            The rows in the table for an ONU interface are created
            at system initialization.
            The row in the table corresponding to the OLT ifIndex
            and the row corresponding to the broadcast virtual link
            are created at system initialization.
            A row in the table corresponding to the ifIndex of a
            virtual links is created when a virtual link is
            established (ONU registers) and deleted when the virtual
            link is deleted (ONU deregisters)."
    INDEX       { ifIndex }
    ::= { dot3ExtPkgOptIfTable 1 }

 Dot3ExtPkgOptIfEntry ::=
   SEQUENCE {
     dot3ExtPkgOptIfSuspectedFlag             TruthValue,
     dot3ExtPkgOptIfInputPower                Integer32,
     dot3ExtPkgOptIfLowInputPower             Integer32,
     dot3ExtPkgOptIfHighInputPower            Integer32,
     dot3ExtPkgOptIfLowerInputPowerThreshold  Integer32,
     dot3ExtPkgOptIfUpperInputPowerThreshold  Integer32,
     dot3ExtPkgOptIfOutputPower               Integer32,
     dot3ExtPkgOptIfLowOutputPower            Integer32,
     dot3ExtPkgOptIfHighOutputPower           Integer32,
     dot3ExtPkgOptIfLowerOutputPowerThreshold Integer32,
     dot3ExtPkgOptIfUpperOutputPowerThreshold Integer32,
     dot3ExtPkgOptIfSignalDetect              TruthValue,
     dot3ExtPkgOptIfTransmitAlarm             TruthValue,
     dot3ExtPkgOptIfTransmitEnable            TruthValue
     }

dot3ExtPkgOptIfSuspectedFlag OBJECT-TYPE
```

```
    SYNTAX  TruthValue
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
      "This object is a reliability indication.
       If true, the data in this entry may be unreliable.
       This object is applicable for an OLT and an ONU. At the
       OLT, it has a distinct value for each virtual interface."
    ::= { dot3ExtPkgOptIfEntry 1 }

  dot3ExtPkgOptIfInputPower OBJECT-TYPE
    SYNTAX  Integer32
    UNITS  "0.1 dbm"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
      "The optical power monitored at the input.
       This object is applicable for an OLT and an ONU. At the
       OLT, it has a distinct value for each virtual interface."
  ::= { dot3ExtPkgOptIfEntry 2 }

  dot3ExtPkgOptIfLowInputPower OBJECT-TYPE
    SYNTAX  Integer32
    UNITS  "0.1 dbm"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
      "The lowest optical power monitored at the input during the
       current 15-minute interval.
       This object is applicable for an OLT and an ONU. At the
       OLT, it has a distinct value for each virtual interface."
    ::= { dot3ExtPkgOptIfEntry 3 }
  dot3ExtPkgOptIfHighInputPower OBJECT-TYPE
    SYNTAX  Integer32
    UNITS  "0.1 dbm"
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
      "The highest optical power monitored at the input during the
       current 15-minute interval.
       This object is applicable for an OLT and an ONU. At the
       OLT, it has a distinct value for each virtual interface."
    ::= { dot3ExtPkgOptIfEntry 4 }

  dot3ExtPkgOptIfLowerInputPowerThreshold OBJECT-TYPE
    SYNTAX  Integer32
    UNITS  "0.1 dbm"
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
      "The lower limit threshold on input power. If
       dot3ExtPkgOptIfInputPower drops to this value or below,
       a Threshold Crossing Alert (TCA) should be sent.
       Reading will present the threshold value. Writing will
       set the value of the threshold.
       The write operation is not restricted in this document
       and can be done at any time. Changing
       dot3ExtPkgOptIfLowerInputPowerThreshold can lead to a Threshold
       Crossing Alert (TCA) being sent for the respective interface.
```



```
      This alert may be leading to an interruption of service for the
      users connected to the respective EPON interface, depending on
      the system action on such an alert.
      This object is applicable for an OLT and an ONU. At the
      OLT, it has a distinct value for each virtual interface."
  ::= { dot3ExtPkgOptIfEntry 5 }

dot3ExtPkgOptIfUpperInputPowerThreshold OBJECT-TYPE
  SYNTAX  Integer32
  UNITS  "0.1 dbm"
  MAX-ACCESS  read-write
  STATUS  current
  DESCRIPTION
    "The upper limit threshold on input power. If
     dot3ExtPkgOptIfInputPower reaches or exceeds this value,
     a Threshold Crossing Alert (TCA) should be sent.
     Reading will present the threshold value. Writing will
     set the value of the threshold.
     The write operation is not restricted in this document
     and can be done at any time. Changing
     dot3ExtPkgOptIfUpperInputPowerThreshold can lead to a Threshold
     Crossing Alert (TCA) being sent for the respective interface.
     This alert may be leading to an interruption of service for the
     users connected to the respective EPON interface, depending on
     the system action on such an alert.
     This object is applicable for an OLT and an ONU. At the
     OLT, it has a distinct value for each virtual interface."
 ::= { dot3ExtPkgOptIfEntry 6 }

dot3ExtPkgOptIfOutputPower OBJECT-TYPE
  SYNTAX  Integer32
  UNITS  "0.1 dbm"
  MAX-ACCESS  read-only
  STATUS  current
  DESCRIPTION
    "The optical power monitored at the output.
     This object is applicable for an OLT and an ONU. At the
     OLT, it has a distinct value for each virtual interface."
  ::= { dot3ExtPkgOptIfEntry 7 }

dot3ExtPkgOptIfLowOutputPower OBJECT-TYPE
  SYNTAX  Integer32
  UNITS  "0.1 dbm"
  MAX-ACCESS  read-only
  STATUS  current
  DESCRIPTION
    "The lowest optical power monitored at the output during the
     current 15-minute interval.
     This object is applicable for an OLT and an ONU. At the
     OLT, it has a distinct value for each virtual interface."
  ::= { dot3ExtPkgOptIfEntry 8 }

dot3ExtPkgOptIfHighOutputPower OBJECT-TYPE
  SYNTAX  Integer32
  UNITS  "0.1 dbm"
  MAX-ACCESS  read-only
  STATUS  current
  DESCRIPTION
    "The highest optical power monitored at the output during the
```



```
      current 15-minute interval.
      This object is applicable for an OLT and an ONU. At the
      OLT, it has a distinct value for each virtual interface."
  ::= { dot3ExtPkgOptIfEntry 9 }

 dot3ExtPkgOptIfLowerOutputPowerThreshold OBJECT-TYPE
   SYNTAX  Integer32
   UNITS  "0.1 dbm"
   MAX-ACCESS  read-write
   STATUS  current
   DESCRIPTION
     "The lower limit threshold on output power. If
      dot3ExtPkgOptIfOutputPower drops to this value or below,
      a Threshold Crossing Alert (TCA) should be sent.
      Reading will present the threshold value. Writing will
      set the value of the threshold.
      The write operation is not restricted in this document
      and can be done at any time. Changing
      dot3ExtPkgOptIfLowerOutputPowerThreshold can lead to a Threshold
      Crossing Alert (TCA) being sent for the respective interface.
      This alert may be leading to an interruption of service for the
      users connected to the respective EPON interface, depending on
      the system action on such an alert.
      This object is applicable for an OLT and an ONU. At the
      OLT, it has a distinct value for each virtual interface."
 ::= { dot3ExtPkgOptIfEntry 10 }

 dot3ExtPkgOptIfUpperOutputPowerThreshold OBJECT-TYPE
   SYNTAX  Integer32
   UNITS  "0.1 dbm"
   MAX-ACCESS  read-write
   STATUS  current
   DESCRIPTION
     "The upper limit threshold on output power. If
      dot3ExtPkgOptIfOutputPower reaches or exceeds this value,
      a Threshold Crossing Alert (TCA) should be sent.
      Reading will present the threshold value. Writing will
      set the value of the threshold.
      The write operation is not restricted in this document
      and can be done at any time. Changing
      dot3ExtPkgOptIfUpperOutputPowerThreshold can lead to a Threshold
      Crossing Alert (TCA) being sent for the respective interface.
      This alert may be leading to an interruption of service of the
      users connected to the respective EPON interface, depending on
      the system action on such an alert.
      This object is applicable for an OLT and an ONU. At the
      OLT, it has a distinct value for each virtual interface."
   ::= { dot3ExtPkgOptIfEntry 11 }

 dot3ExtPkgOptIfSignalDetect OBJECT-TYPE
     SYNTAX  TruthValue
     MAX-ACCESS  read-only
     STATUS  current
     DESCRIPTION
             "When getting true(1), there is a valid optical signal at
              the receive that is above the optical power level for
              signal detection. When getting false(2) the optical
              signal at the receive is below the optical power level
              for signal detection.
```

```
              This object is applicable for an OLT and an ONU. At the
              OLT, it has a distinct value for each virtual interface."
     DEFVAL { false }
     ::= { dot3ExtPkgOptIfEntry 12 }

 dot3ExtPkgOptIfTransmitAlarm OBJECT-TYPE
     SYNTAX  TruthValue
     MAX-ACCESS  read-only
     STATUS  current
     DESCRIPTION
             "When getting true(1) there is a non-valid optical signal
              at the transmit of the interface, either a higher level
              or lower level than expected. When getting false(2) the
              optical signal at the transmit is valid and in the
              required range.
              This object is applicable for an OLT and an ONU. At the
              OLT, it has a distinct value for each virtual interface."
     DEFVAL { false }
     ::= { dot3ExtPkgOptIfEntry 13 }

 dot3ExtPkgOptIfTransmitEnable OBJECT-TYPE
     SYNTAX  TruthValue
     MAX-ACCESS  read-write
     STATUS  current
     DESCRIPTION
             "Setting this object to true(1) will cause the optical
              interface to start transmission (according to the
              control protocol specified for the logical interface).
              Setting this object to false(2) will cause the
              interface to stop the optical transmission.
              When getting true(1), the optical interface is in
              transmitting mode (obeying to the logical control
              protocol).
              When getting false(2), the optical interface is not in
              transmitting mode.
              The write operation is not restricted in this document
              and can be done at any time. Changing
              dot3ExtPkgOptIfTransmitEnable state can lead to a halt
              in the optical transmission of the respective interface
              leading to an interruption of service of the users
              connected to the respective EPON interface.
              The object is relevant when the admin state of the
              interface is active as set by the dot3MpcpAdminState.
              This object is applicable for an OLT and an ONU. At the
              OLT it, has a distinct value for each virtual interface."
     DEFVAL { false }
     ::= { dot3ExtPkgOptIfEntry 14 }

 -- Conformance statements

 -- Conformance Groups

 dot3EponGroups      OBJECT IDENTIFIER ::= { dot3EponConformance 1 }

 dot3MpcpGroupBase OBJECT-GROUP
     OBJECTS {
             dot3MpcpOperStatus,
             dot3MpcpAdminState,
             dot3MpcpMode,
```

```
            dot3MpcpSyncTime,
            dot3MpcpLinkID,
            dot3MpcpRemoteMACAddress,
            dot3MpcpRegistrationState,
            dot3MpcpMaximumPendingGrants,
            dot3MpcpTransmitElapsed,
            dot3MpcpReceiveElapsed,
            dot3MpcpRoundTripTime
    }
    STATUS  current
    DESCRIPTION
           "A collection of objects of dot3 Mpcp Control entity state
            definition. Objects are per LLID."
    ::= { dot3EponGroups 1 }

dot3MpcpGroupStat OBJECT-GROUP
    OBJECTS {
            dot3MpcpMACCtrlFramesTransmitted,
            dot3MpcpMACCtrlFramesReceived,
            dot3MpcpDiscoveryWindowsSent,
            dot3MpcpDiscoveryTimeout,
            dot3MpcpTxRegRequest,
            dot3MpcpRxRegRequest,
            dot3MpcpTxRegAck,
            dot3MpcpRxRegAck,
            dot3MpcpTxReport,
            dot3MpcpRxReport,
            dot3MpcpTxGate,
            dot3MpcpRxGate,
            dot3MpcpTxRegister,
            dot3MpcpRxRegister
    }
    STATUS  current
    DESCRIPTION
            "A collection of objects of dot3 Mpcp Statistics.
             Objects are per LLID."
    ::= { dot3EponGroups 2 }
dot3OmpeGroupID OBJECT-GROUP
    OBJECTS {
            dot3OmpEmulationType

    }
    STATUS  current
    DESCRIPTION
            "A collection of objects of dot3 OMP emulation entity
             state definition. Objects are per LLID."
    ::= { dot3EponGroups 3 }

dot3OmpeGroupStat OBJECT-GROUP
    OBJECTS {
            dot3OmpEmulationSLDErrors,
            dot3OmpEmulationCRC8Errors,
            dot3OmpEmulationBadLLID,
            dot3OmpEmulationGoodLLID,
            dot3OmpEmulationOnuPonCastLLID,
            dot3OmpEmulationOltPonCastLLID,
            dot3OmpEmulationBroadcastBitNotOnuLlid,
            dot3OmpEmulationOnuLLIDNotBroadcast,
            dot3OmpEmulationBroadcastBitPlusOnuLlid,
```

```
            dot3OmpEmulationNotBroadcastBitNotOnuLlid
    }
    STATUS  current
    DESCRIPTION
            "A collection of objects of dot3 OMP emulation
             Statistics. Objects are per LLID."
    ::= { dot3EponGroups 4 }

dot3EponFecGroupAll OBJECT-GROUP
    OBJECTS {
            dot3EponFecPCSCodingViolation,
            dot3EponFecAbility,
            dot3EponFecMode,
            dot3EponFecCorrectedBlocks,
            dot3EponFecUncorrectableBlocks,
            dot3EponFecBufferHeadCodingViolation
    }
    STATUS  current
    DESCRIPTION
            "A collection of objects of dot3 FEC group control and
            statistics. Objects are per LLID."
    ::= { dot3EponGroups 5 }

dot3ExtPkgGroupControl OBJECT-GROUP
    OBJECTS {
            dot3ExtPkgObjectReset,
            dot3ExtPkgObjectPowerDown,
            dot3ExtPkgObjectNumberOfLLIDs,
            dot3ExtPkgObjectFecEnabled,
            dot3ExtPkgObjectReportMaximumNumQueues,
            dot3ExtPkgObjectRegisterAction
    }
    STATUS  current
    DESCRIPTION
            "A collection of objects of dot3ExtPkg control
             definition. Objects are per LLID."
    ::= { dot3EponGroups 6 }

dot3ExtPkgGroupQueue OBJECT-GROUP
    OBJECTS {
     dot3ExtPkgObjectReportNumThreshold,
     dot3ExtPkgObjectReportMaximumNumThreshold,
     dot3ExtPkgStatTxFramesQueue,
     dot3ExtPkgStatRxFramesQueue,
     dot3ExtPkgStatDroppedFramesQueue
    }
    STATUS  current
    DESCRIPTION
            "A collection of objects of dot3ExtPkg Queue
             control. Objects are per LLID, per queue."
    ::= { dot3EponGroups 7 }

dot3ExtPkgGroupQueueSets OBJECT-GROUP
    OBJECTS {
     dot3ExtPkgObjectReportThreshold
    }
    STATUS  current
    DESCRIPTION
            "A collection of objects of dot3ExtPkg queue_set
```

```
               control. Objects are per LLID, per queue, per
               queue_set."
        ::= { dot3EponGroups 8 }

    dot3ExtPkgGroupOptIf OBJECT-GROUP
        OBJECTS {
       dot3ExtPkgOptIfSuspectedFlag,
         dot3ExtPkgOptIfInputPower,
         dot3ExtPkgOptIfLowInputPower,
         dot3ExtPkgOptIfHighInputPower,
         dot3ExtPkgOptIfLowerInputPowerThreshold,
         dot3ExtPkgOptIfUpperInputPowerThreshold,
         dot3ExtPkgOptIfOutputPower,
         dot3ExtPkgOptIfLowOutputPower,
         dot3ExtPkgOptIfHighOutputPower,
         dot3ExtPkgOptIfLowerOutputPowerThreshold,
         dot3ExtPkgOptIfUpperOutputPowerThreshold,
         dot3ExtPkgOptIfSignalDetect,
         dot3ExtPkgOptIfTransmitAlarm,
         dot3ExtPkgOptIfTransmitEnable
         }
        STATUS  current
        DESCRIPTION
               "A collection of objects of control and status indication
                of the optical interface.
                Objects are per LLID."
        ::= { dot3EponGroups 9 }

    -- Compliance statements

       dot3EponCompliances
           OBJECT IDENTIFIER ::= { dot3EponConformance 2 }

    dot3MPCPCompliance MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION "The compliance statement for Multi-Point
                     Control Protocol interfaces."

        MODULE  -- this module
        MANDATORY-GROUPS { dot3MpcpGroupBase}

        GROUP       dot3MpcpGroupStat
       DESCRIPTION "This group is mandatory for all MPCP supporting
                   interfaces for statistics collection."
       ::= { dot3EponCompliances 1}

    dot3OmpeCompliance MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION "The compliance statement for OMPEmulation
                     interfaces."

        MODULE  -- this module
        MANDATORY-GROUPS { dot3OmpeGroupID}

        GROUP       dot3OmpeGroupStat
        DESCRIPTION "This group is mandatory for all OMPemulation
                     supporting interfaces for statistics collection."

        ::= { dot3EponCompliances 2}
```

```
dot3EponFecCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION "The compliance statement for FEC EPON interfaces.
                 This group is mandatory for all FEC supporting
                 interfaces for control and statistics collection."

    MODULE  -- this module
    MANDATORY-GROUPS { dot3EponFecGroupAll }

    ::= { dot3EponCompliances 3}

dot3ExtPkgCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION "The compliance statement for EPON Interfaces
                 using the extended package."
    MODULE  -- this module
    MANDATORY-GROUPS { dot3ExtPkgGroupControl }

    GROUP       dot3ExtPkgGroupQueue
    DESCRIPTION " This group is mandatory for all EPON interfaces
                 supporting REPORT queue management of the extended
                 package."

    GROUP       dot3ExtPkgGroupQueueSets
    DESCRIPTION " This group is mandatory for all EPON interfaces
                 supporting REPORT queue_sets management of the
                 extended package."

    GROUP       dot3ExtPkgGroupOptIf
    DESCRIPTION "This group is mandatory for all EPON interfaces
                 supporting optical interfaces management,
                 of the extended package."

    ::= { dot3EponCompliances 4}

END
```

10. Ethernet-like interface MIB module

10.1 Introduction

This clause defines a portion of the Management Information Base (MIB) for use with SNMP. In particular, it defines objects for managing Ethernet-like interfaces.

10.2 Overview

Instances of these object types represent attributes of an interface to an Ethernet-like communications medium.

The definitions presented here are based on Clause 30 of IEEE Std 802.3. Implementors of these MIB objects should note that IEEE Std 802.3 explicitly describes (in the form of Pascal pseudocode) when, where, and how various MAC attributes are measured. IEEE Std 802.3 also describes the effects of MAC actions that may be invoked by manipulating instances of the MIB objects defined here.

To the extent that some of the attributes defined in IEEE Std 802.3 are represented by previously defined objects in MIB-2 from IETF RFC 1213 or in the Interfaces Group MIB defined in IETF RFC 2863, such attributes are not redundantly represented by objects defined in this clause. Among the attributes represented by objects defined in other MIB module specifications are the number of octets transmitted or received on a particular interface, the number of frames transmitted or received on a particular interface, the promiscuous status of an interface, the MAC address of an interface, and multicast information associated with an interface.

10.2.1 Relation to MIB-2

This subclause applies only when this MIB is used in conjunction with the IETF RFC 1213 interface group.

The relationship between an Ethernet-like interface and an interface in the context of MIB-2 is one-to-one. As such, the value of an ifIndex object instance can be directly used to identify corresponding instances of the objects defined herein.

10.2.2 Relation to the Interfaces Group MIB

The Interfaces Group MIB defined in IETF RFC 2863 requires that any MIB that is an adjunct of the Interfaces Group MIB clarify specific areas within the Interfaces Group MIB. These areas were intentionally left vague in the Interfaces Group MIB to avoid over constraining the MIB, thereby precluding management of certain media-types.

Section 4 of IETF RFC 2863 enumerates several areas which a media-specific MIB must (wherein the word “must” is used in accordance with the requirements of IETF RFC 2119) clarify. Each of these areas is addressed in a following subclause. The implementor is referred to IETF RFC 2863 in order to understand the general intent of these areas.

10.2.2.1 ifRcvAddressTable

This table contains all IEEE 802.3 addresses, unicast, multicast, and broadcast, for which this interface will receive packets and forward them up to a higher layer entity for local consumption. The format of the address, contained in ifRcvAddressAddress, is the same as for ifPhysAddress.

In the event that the interface is part of a MAC bridge, this table does not include unicast addresses that are accepted for possible forwarding out some other port. This table is explicitly not intended to provide a bridge address filtering mechanism.

10.2.2.2 ifType

All Ethernet-like interfaces shall return ethernetCsmacd(6) for ifType. Information on the particular port type and operating speed is available from ifSpeed in the Interfaces Group MIB, and ifMauType in the MAU-MIB module defined in Clause 13. All Ethernet-like interfaces shall also implement the MAU-MIB module defined in Clause 13.[16]

10.2.2.3 ifXxxOctets

The Interfaces Group MIB octet counters, ifInOctets, ifOutOctets, ifHCInOctets, and ifHCOutOctets include all octets in valid frames sent or received on the interface, including the MAC header and FCS, but not the preamble, start of frame delimiter, or extension octets. This corresponds to the definition of frameSize/8 in 4.2.7.1 of IEEE Std 802.3 (frameSize is defined in bits rather than octets, and is defined as 2 × addressSize + lengthOrTypeSize + dataSize + crcSize). They do not include the number of octets in collided or failed transmit attempts, since the MAC layer driver typically does not have visibility to count these octets. They also do not include octets in received invalid frames, since this information is normally not passed to the MAC layer, and since non-promiscuous MAC implementations cannot reliably determine whether an invalid frame was actually addressed to this station.

Note that these counters do include octets in valid MAC control frames sent or received on the interface, as well as octets in otherwise valid received MAC frames that are discarded by the MAC layer for some reason (insufficient buffer space, unknown protocol, etc.).

Note that the octet counters in IF-MIB do not exactly match the definition of the octet counters in IEEE Std 802.3. aOctetsTransmittedOK and aOctetsReceivedOK count only the octets in the clientData and Pad fields, whereas ifInOctets and ifOutOctets include the entire MAC frame, including MAC header and FCS. However, the IF-MIB counters can be derived from the IEEE 802.3 counters as follows in Equation (1) and Equation (2):

$$\text{ifInOctets} = \text{aOctetsReceivedOK} + (18 \times \text{aFramesReceivedOK}) \tag{1}$$

$$\text{ifOutOctets} = \text{aOctetsTransmittedOK} + (18 \times \text{aFramesTransmittedOK}) \tag{2}$$

Another difference to keep in mind between the IF-MIB counters and IEEE 802.3 counters is that, in IEEE Std 802.3, the frame counters and octet counters are always incremented together. aOctetsTransmittedOK counts the number of octets in frames that were counted by aFramesTransmittedOK. aOctetsReceivedOK counts the number of octets in frames that were counted by aFramesReceivedOK. This is not the case with the IF-MIB counters. The IF-MIB octet counters count the number of octets sent to or received from the layer below this interface, whereas the packet counters count the number of packets sent to or received from the layer above. Therefore, received MAC Control frames, ifInDiscards, and ifInUnknownProtos are counted by ifInOctets, but not ifInXcastPkts. Transmitted MAC Control frames are counted by ifOutOctets, but not ifOutXcastPkts. ifOutDiscards and ifOutErrors are counted by ifOutXcastPkts, but not ifOutOctets.

10.2.2.4 ifXxxXcastPkts

The packet counters in the IF-MIB do not exactly match the definition of the frame counters in IEEE Std 802.3. aFramesTransmittedOK counts the number of frames successfully transmitted on the interface,

[16]There are three other interface types defined in IANAifType-MIB for Ethernet, namely fastEther(62), fastEtherFX(69), and gigabitEthernet(117). Management applications should be prepared to receive these obsolete ifType values from older implementations.

whereas ifOutUcastPkts, ifOutMulticastPkts, and ifOutBroadcastPkts count the number of transmit requests made from a higher layer, whether or not the transmit attempt was successful. This means that packets counted by ifOutErrors or ifOutDiscards are also counted by ifOutXcastPkts, but are not counted by aFramesTransmittedOK. This also means that, since MAC Control frames are generated by a sublayer internal to the interface layer rather than by a higher layer, they are not counted by ifOutXcastPkts, but are counted by aFramesTransmittedOK.

$$\begin{aligned}\text{aFramesTransmittedOK} = {} & \text{ifOutUcastPkts} + \text{ifOutMulticastPkts} \\ & + \text{ifOutBroadcastPkts} + \text{dot3OutPauseFrames} \\ & - (\text{ifOutErrors} + \text{ifOutDiscards})\end{aligned} \tag{3}$$

Similarly, aFramesReceivedOK counts the number of frames received successfully by the interface, whether or not they are passed to a higher layer, whereas ifInUcastPkts, ifInMulticastPkts, and ifInBroadcastPkts count only the number of packets passed to a higher layer. This means that packets counted by ifInDiscards or ifInUnknownProtos are also counted by aFramesReceivedOK, but are not counted by ifInXcastPkts. This also means that, since MAC Control frames are consumed by a sublayer internal to the interface layer and not passed to a higher layer, they are not counted by ifInXcastPkts, but are counted by aFramesReceivedOK.

$$\begin{aligned}\text{aFramesReceivedOK} = {} & \text{ifInUcastPkts} + \text{ifInMulticastPkts} \\ & + \text{ifInBroadcastPkts} + \text{dot3InPauseFrames} \\ & + \text{ifInDiscards} + \text{ifInUnknownProtos}\end{aligned} \tag{4}$$

This specification chooses to treat MAC control frames as being originated and consumed within the interface and not counted by the IF-MIB packet counters. MAC control frames are normally sent as multicast packets. In many network environments, MAC control frames can greatly outnumber multicast frames carrying actual data. If MAC control frames were included in the ifInMulticastPkts and ifOutMulticastPkts, the count of data-carrying multicast packets would tend to be drowned out by the count of MAC control frames, rendering those counters considerably less useful.

To better understand the issues surrounding the mapping of the IF-MIB packet and octet counters to an Ethernet interface, it is useful to refer to a Case diagram (Case and Partridge [B1]) for the IF-MIB counters, with modifications to show the proper interpretation for the Ethernet interface. This is depicted in Figure 10–1.

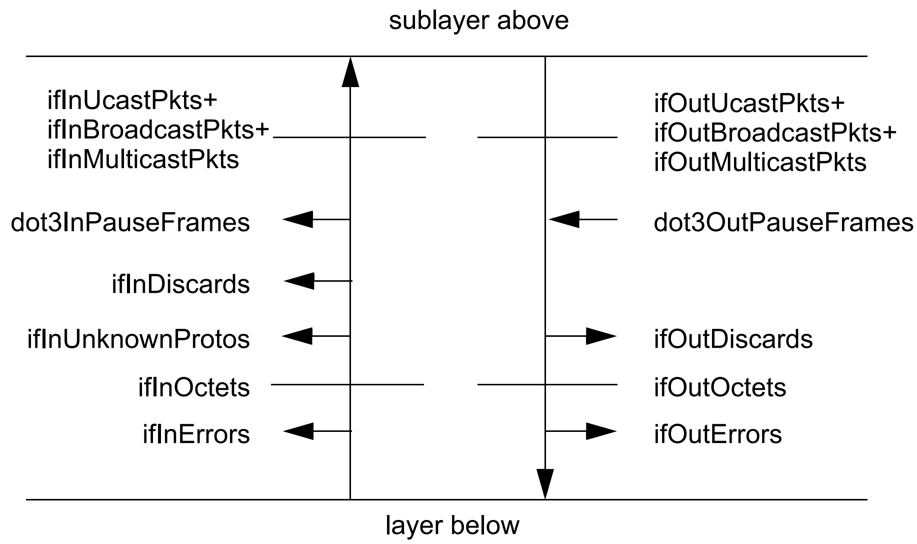


Figure 10–1—Case diagram for the IF-MIB counters

10.2.2.5 ifMtu

The defined standard MTU for Ethernet-like interfaces is 1500 octets. However, many implementations today support larger packet sizes than IEEE Std 802.3. The value of this object shall reflect the actual MTU in use on the interface, whether it matches the standard MTU or not.

This value should reflect the value seen by the MAC client interface. When a higher layer protocol, like IP, is running over Ethernet framing, this is the MTU that will be seen by that higher layer protocol. However, most Ethernet-like interfaces today run multiple protocols that use a mix of different framing types. For example, an IEEE 802.2 LLC type 1 client protocol will see an MTU of 1497 octets on an interface using the IEEE standard maximum packet size, and a protocol running over SNAP will see an MTU of 1492 octets on an interface using the IEEE standard maximum packet size. However, since specification mandates using the MTU as seen at the MAC client interface, the value of ifMtu would be reported as 1500 octets in these cases.

10.2.2.6 ifSpeed and ifHighSpeed

For Ethernet-like interfaces operating at 1000 Megabits per second (Mb/s) or less, ifSpeed will represent the current operational speed of the interface in bits per second. For such interface types, this will be equal to 1 000 000 (1 million), 10 000 000 (10 million), 100 000 000 (100 million), or 1 000 000 000 (1 billion). ifHighSpeed will represent the current operational speed in millions of bits per second. For such Ethernet-like interfaces, this will be equal to 1, 10, 100, or 1000. If the interface implements Auto-Negotiation, Auto-Negotiation is enabled for this interface, and the interface has not yet negotiated to an operational speed, these objects should reflect the maximum speed supported by the interface.

For Ethernet-like interfaces operating at greater than 1000 Mb/s, ifHighSpeed will represent the current operational speed of the interface in millions of bits per second. Note that for WAN implementations, this will be the payload data rate over the WAN interface sublayer. For current implementations, this will be equal to 10 000 for LAN implementations of 10 Gb/s, and 9294 for WAN implementations of the 10 Gb/s MAC over an OC-192 PHY. For these speeds, ifSpeed should report a maximum unsigned 32-bit value of 4 294 967 295 as specified in IETF RFC 2863.

These objects shall indicate the correct line speed regardless of the current duplex mode. They shall not indicate a doubled value when operating in full-duplex mode. The duplex mode of the interface may be determined by examining either the dot3StatsDuplexStatus object in this MIB module, or the ifMauType MAU-MIB module object defined in Clause 13.

10.2.2.7 ifPhysAddress

This object contains the IEEE 802.3 address that is placed in the source-address field of any Ethernet, Starlan, or IEEE 802.3 frames that originate at this interface. Usually this will be kept in ROM on the interface hardware. Some systems may set this address via software.

In a system where there are several such addresses the designer has a tougher choice. The address chosen should be the one most likely to be of use to network management (e.g., the address placed in ARP responses for systems that are primarily IP systems).

If the designer truly cannot choose, use of the factory-provided ROM address is suggested.

If the address cannot be determined, an octet string of zero length should be returned.

The address is stored in binary in this object. The address is stored in “canonical” bit order, that is, the Group Bit is positioned as the low-order bit of the first octet. Thus, the first byte of a multicast address would have the bit 0x01 set.

10.2.2.8 Specific Interfaces Group MIB objects

Table 10–1 provides specific implementation guidelines for applying the Interfaces Group objects to Ethernet-like interfaces.

Table 10–1—Implementation guidelines

Object	Guidelines
ifIndex	Each Ethernet-like interface is represented by an ifEntry. The dot3StatsTable in this MIB module is indexed by dot3StatsIndex. The interface identified by a particular value of dot3StatsIndex is the same interface as identified by the same value of ifIndex.
ifDescr	Refer to IETF RFC 2863.
ifType	Refer to 10.2.2.2.
ifMtu	Refer to 10.2.2.5.
ifSpeed	Refer to 10.2.2.6.
ifPhysAddress	Refer to 10.2.2.7.
ifAdminStatus	Write access is not required. Support for ‘testing’ is not required.
ifOperStatus	The operational state of the interface. Support for ‘testing’ is not required. The value ‘dormant’ has no meaning for an Ethernet-like interface.
ifLastChange	Refer to IETF RFC 2863.

Table 10–1—Implementation guidelines *(continued)*

Object	Guidelines
ifInOctets	The number of octets in valid MAC frames received on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames received on this interface. See 10.2.2.3.
ifInUcastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer and are not passed to any higher layer protocol. See 10.2.2.4.
ifInDiscards	Refer to IETF RFC 2863.
ifInErrors	The sum for this interface of dot3StatsAlignmentErrors, dot3StatsFCSErrors, dot3StatsFrameTooLongs, and dot3StatsInternalMacReceiveErrors.
ifInUnknownProtos	Refer to IETF RFC 2863.
ifOutOctets	The number of octets transmitted in valid MAC frames on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames transmitted on this interface. See 10.2.2.3.
ifOutUcastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol. See 10.2.2.4.
ifOutDiscards	Refer to IETF RFC 2863.
ifOutErrors	The sum for this interface of: dot3StatsSQETestErrors, dot3StatsLateCollisions, dot3StatsExcessiveCollisions, dot3StatsInternalMacTransmitErrors and dot3StatsCarrierSenseErrors.
ifName	Locally-significant textual name for the interface (e.g., lan0).
ifInMulticastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer and are not passed to any higher layer protocol. See 10.2.2.4.
ifInBroadcastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer, and are not passed to ny higher layer protocol. See 10.2.2.4.
ifOutMulticastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol. See 10.2.2.4.
ifOutBroadcastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol. See 10.2.2.4.
ifHCInOctets, ifHCOutOctets	64-bit versions of counters. Required for Ethernet-like interfaces that are capable of operating at 20 Mb/s or faster, even if the interface is currently operating at less than 20 Mb/s.

Table 10–1—Implementation guidelines *(continued)*

Object	Guidelines
ifHCInUcastPkts, ifHCInMulticastPkts, ifHCInBroadcastPkts, ifHCOutUcastPkts, ifHCOutMulticastPkts, ifHCOutBroadcastPkts	64-bit versions of packet counters. Required for Ethernet-like interfaces that are capable of operating at 640 Mb/s or faster, even if the interface is currently operating at less than 640 Mb/s.
ifLinkUpDownTrapEnable	Refer to IETF RFC 2863. Default is ‘enabled’.
ifHighSpeed	Refer to 10.2.2.6.
ifPromiscuousMode	Refer to IETF RFC 2863.
ifConnectorPresent	This will normally be ‘true’. It will be ‘false’ in the case where this interface uses the WAN Interface Sublayer. See Clause 12 for details.
ifAlias	Refer to IETF RFC 2863.
ifCounterDiscontinuityTime	Refer to IETF RFC 2863. Note that a discontinuity in the Interfaces Group MIB counters may also indicate a discontinuity in some or all of the counters in this MIB that are associated with that interface.
ifStackHigherLayer, ifStackLowerLayer, ifStackStatus	Refer to 11.2.1.1.
ifRcvAddressAddress, ifRcvAddressStatus, ifRcvAddressType	Refer to 10.2.2.1.

10.2.3 Relation to the IEEE 802.3 MAU-MIB module

Support for the mauModIfCompl3 compliance statement of the MAU-MIB module defined in Clause 13 is required for Ethernet-like interfaces. This MIB module is needed in order to allow applications to determine the current MAU type in use by the interface, and to control autonegotiation and duplex mode for the interface. Implementing this MIB module without implementing the MAU-MIB module would leave applications with no standard way to determine the media type in use, and no standard way to control the duplex mode of the interface.

10.2.4 Mapping of IEEE 802.3 managed objects

The mapping of IEEE 802.3 managed objects to SNMP objects is shown in Table 10–2.

Table 10–2—Mapping of IEEE 802.3 managed objects

IEEE 802.3 managed object		Corresponding SNMP object
oMacEntity	.aMACID	dot3StatsIndex or IF-MIB – ifIndex
	.aFramesTransmittedOK	IF-MIB – ifOutUCastPkts + ifOutMulticastPkts + ifOutBroadcastPkts[a]
	.aSingleCollisionFrames	dot3StatsSingleCollisionFrames
	.aMultipleCollisionFrames	dot3StatsMultipleCollisionFrames
	.aFramesReceivedOK	IF-MIB – ifInUcastPkts + ifInMulticastPkts + ifInBroadcastPkts[a]
	.aFrameCheckSequenceErrors	dot3StatsFCSErrors
	.aAlignmentErrors	dot3StatsAlignmentErrors
	.aOctetsTransmittedOK	IF-MIB – ifOutOctets[a]
	.aFramesWithDeferredXmissions	dot3StatsDeferredTransmissions
	.aLateCollisions	dot3StatsLateCollisions
	.aFramesAbortedDueToXSColls	dot3StatsExcessiveCollisions
	.aFramesLostDueToIntMACXmitError	dot3StatsInternalMacTransmitErrors
	.aCarrierSenseErrors	dot3StatsCarrierSenseErrors
	.aOctetsReceivedOK	IF-MIB – ifInOctets[a]
	.aFramesLostDueToIntMACRcvError	dot3StatsInternalMacReceiveErrors
	.aPromiscuousStatus	IF-MIB – ifPromiscuousMode
	.aReadMulticastAddressList	IF-MIB – ifRcvAddressTable
	.aMulticastFramesXmittedOK	IF-MIB – ifOutMulticastPkts[a]
	.aBroadcastFramesXmittedOK	IF-MIB – ifOutBroadcastPkts[a]
	.aMulticastFramesReceivedOK	IF-MIB – ifInMulticastPkts[a]
	.aBroadcastFramesReceivedOK	IF-MIB – ifInBroadcastPkts[a]
	.aFrameTooLongErrors	dot3StatsFrameTooLongs
	.aReadWriteMACAddress	IF-MIB – ifPhysAddress
	.aCollisionFrames	dot3CollFrequencies
	.aDuplexStatus	dot3StatsDuplexStatus
	.aRateControlAbility	dot3StatsRateControlAbility
	.aMaxFrameLength	dot3StatsMaxFrameLength
	.aRateControlStatus	dot3StatsRateControlStatus

Table 10–2—Mapping of IEEE 802.3 managed objects *(continued)*

IEEE 802.3 managed object		Corresponding SNMP object
oPHYEntity	.aPHYID	dot3StatsIndex or IF-MIB – ifIndex
	.aSQETestErrors	dot3StatsSQETestErrors
	.aSymbolErrorDuringCarrier	dot3StatsSymbolErrors
oMACControlEntity	.aMACControlID	dot3StatsIndex or IF-MIB – ifIndex
	.aMACControlFunctionsSupported	dot3ControlFunctionsSupported and dot3ControlFunctionsEnabled
	.aUnsupportedOpcodesReceived	dot3ControlInUnknownOpcodes
oPAUSEEntity	.aPAUSEMACCtrlFramesTransmitted	dot3OutPauseFrames
	.aPAUSEMACCtrlFramesReceived	dot3InPauseFrames

[a]Note that the octet counters in IF-MIB do not exactly match the definition of the octet counters in IEEE Std 802.3. See 10.2.2.3 for details.
Also note that the packet counters in the IF-MIB do not exactly match the definition of the frame counters in IEEE Std 802.3. See 10.2.2.4 for details.

10.3 Security considerations for Ethernet-like interface MIB module

There is one management object defined in this MIB that has a MAX-ACCESS clause of read-write. That object, dot3PauseAdminMode, may be used to change the flow control configuration on a network interface, which may result in dropped packets, or sending flow control packets on links where the link partner will not understand them. Either action could be detrimental to network performance.

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.

Most of the objects in this MIB module contain statistical information about particular network links. In some network environments, this information may be considered sensitive. It is thus important to control GET and/or NOTIFY access to these objects and possibly to encrypt the values of these objects when sending them over the network via SNMP.

10.4 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL:[17]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C10mib.txt

[17]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.


```
IEEE8023-EtherLike-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE,
        Integer32, Counter32, Counter64, org
            FROM SNMPv2-SMI
        MODULE-COMPLIANCE, OBJECT-GROUP
            FROM SNMPv2-CONF
        TruthValue
            FROM SNMPv2-TC
        ifIndex, InterfaceIndex
            FROM IF-MIB;

    ieee8023etherMIB MODULE-IDENTITY
     LAST-UPDATED "201102020000Z" -- February 2, 2011
     ORGANIZATION
       "IEEE 802.3 working group"
     CONTACT-INFO
         "WG-URL: http://www.ieee802.org/3/index.html
         WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

         Contact: Howard Frazier
         Postal:  3151 Zanker Road
                  San Jose, CA 95134
                  USA
         Tel:     +1.408.922.8164
         E-mail:  hfrazier@broadcom.com"

        DESCRIPTION "The MIB module to describe generic objects for
                    Ethernet-like network interfaces."

        REVISION    "201102020000Z" -- February 2, 2011
        DESCRIPTION
         "Initial version, based on an earlier version published in RFC 3635."

         ::= { org ieee(111) standards-association-numbers-series-standards(2)
               lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 10 }


    ieee8023etherMIBObjects OBJECT IDENTIFIER ::= { ieee8023etherMIB 1 }

    -- the Ethernet-like Statistics group

    dot3StatsTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF Dot3StatsEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION "Statistics for a collection of Ethernet-like
                    interfaces attached to a particular system.
                    There will be one row in this table for each
                    Ethernet-like interface in the system."
        ::= { ieee8023etherMIBObjects 2 }

    dot3StatsEntry OBJECT-TYPE
        SYNTAX     Dot3StatsEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION "Statistics for a particular interface to an
                    Ethernet-like medium."
```

```
    INDEX       { dot3StatsIndex }
    ::= { dot3StatsTable 1 }

Dot3StatsEntry ::=
    SEQUENCE {

        dot3StatsIndex                      InterfaceIndex,
        dot3StatsAlignmentErrors            Counter32,
        dot3StatsFCSErrors                  Counter32,
        dot3StatsSingleCollisionFrames      Counter32,
        dot3StatsMultipleCollisionFrames    Counter32,
        dot3StatsSQETestErrors              Counter32,
        dot3StatsDeferredTransmissions      Counter32,
        dot3StatsLateCollisions             Counter32,
        dot3StatsExcessiveCollisions        Counter32,
        dot3StatsInternalMacTransmitErrors  Counter32,
        dot3StatsCarrierSenseErrors         Counter32,
        dot3StatsFrameTooLongs              Counter32,
        dot3StatsInternalMacReceiveErrors   Counter32,
        dot3StatsSymbolErrors               Counter32,
        dot3StatsDuplexStatus               INTEGER,
        dot3StatsRateControlAbility         TruthValue,
        dot3StatsRateControlStatus          INTEGER,
        dot3StatsMaxFrameLength             INTEGER
    }

dot3StatsIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An index value that uniquely identifies an
                interface to an Ethernet-like medium. The
                interface identified by a particular value of
                this index is the same interface as identified
                by the same value of ifIndex."
    REFERENCE   "RFC 2863, ifIndex"
    ::= { dot3StatsEntry 1 }

dot3StatsAlignmentErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames received on a particular
                interface that are not an integral number of
                octets in length and do not pass the FCS check.

                The count represented by an instance of this
                object is incremented when the alignmentError
                status is returned by the MAC service to the
                LLC (or other MAC user). Received frames for
                which multiple error conditions pertain are,
                according to the conventions of IEEE 802.3
                Layer Management, counted exclusively according

                to the error status presented to the LLC.

                This counter does not increment for group
                encoding schemes greater than 4 bits per group.
```



```
                 For interfaces operating at 10 Gb/s, this
                 counter can roll over in less than 5 minutes if
                 it is incrementing at its maximum rate. Since
                 that amount of time could be less than a
                 management station's poll cycle time, in order
                 to avoid a loss of information, a management
                 station is advised to poll the
                 dot3HCStatsAlignmentErrors object for 10 Gb/s
                 or faster interfaces.

                 Discontinuities in the value of this counter can
                 occur at re-initialization of the management
                 system, and at other times as indicated by the
                 value of ifCounterDiscontinuityTime."
    REFERENCE    "IEEE Std 802.3, 30.3.1.1.7,
                 aAlignmentErrors"
    ::= { dot3StatsEntry 2 }

dot3StatsFCSErrors OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION  "A count of frames received on a particular
                 interface that are an integral number of octets
                 in length but do not pass the FCS check. This
                 count does not include frames received with
                 frame-too-long or frame-too-short error.

                 The count represented by an instance of this
                 object is incremented when the frameCheckError
                 status is returned by the MAC service to the
                 LLC (or other MAC user). Received frames for
                 which multiple error conditions pertain are,
                 according to the conventions of IEEE 802.3
                 Layer Management, counted exclusively according
                 to the error status presented to the LLC.

                 Note:  Coding errors detected by the physical
                 layer for speeds above 10 Mb/s will cause the
                 frame to fail the FCS check.

                 For interfaces operating at 10 Gb/s, this
                 counter can roll over in less than 5 minutes if

                 it is incrementing at its maximum rate. Since
                 that amount of time could be less than a
                 management station's poll cycle time, in order
                 to avoid a loss of information, a management
                 station is advised to poll the
                 dot3HCStatsFCSErrors object for 10 Gb/s or
                 faster interfaces.

                 Discontinuities in the value of this counter can
                 occur at re-initialization of the management
                 system, and at other times as indicated by the
                 value of ifCounterDiscontinuityTime."
    REFERENCE    "IEEE Std 802.3, 30.3.1.1.6,
                 aFrameCheckSequenceErrors."
    ::= { dot3StatsEntry 3 }
```

```
dot3StatsSingleCollisionFrames OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames that are involved in a single
                collision, and are subsequently transmitted
                successfully.

                A frame that is counted by an instance of this
                object is also counted by the corresponding
                instance of either the ifOutUcastPkts,
                ifOutMulticastPkts, or ifOutBroadcastPkts,
                and is not counted by the corresponding
                instance of the dot3StatsMultipleCollisionFrames
                object.

                This counter does not increment when the
                interface is operating in full-duplex mode.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.3,
                aSingleCollisionFrames."
    ::= { dot3StatsEntry 4 }

dot3StatsMultipleCollisionFrames OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames that are involved in more

                than one collision and are subsequently
                transmitted successfully.

                A frame that is counted by an instance of this
                object is also counted by the corresponding
                instance of either the ifOutUcastPkts,
                ifOutMulticastPkts, or ifOutBroadcastPkts,
                and is not counted by the corresponding
                instance of the dot3StatsSingleCollisionFrames
                object.

                This counter does not increment when the
                interface is operating in full-duplex mode.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.4,
                aMultipleCollisionFrames."
    ::= { dot3StatsEntry 5 }

dot3StatsSQETestErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION "A count of times that the SQE TEST ERROR
                is received on a particular interface. The
                SQE TEST ERROR is set in accordance with the
                rules for verification of the SQE detection
                mechanism in the PLS Carrier Sense Function as
                described in IEEE Std 802.3, 7.2.4.6.

                This counter does not increment on interfaces
                operating at speeds greater than 10 Mb/s, or on
                interfaces operating in full-duplex mode.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 7.2.4.6, also 30.3.2.1.4,
                aSQETestErrors."
    ::= { dot3StatsEntry 6 }

dot3StatsDeferredTransmissions OBJECT-TYPE
    SYNTAX      Counter32

    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames for which the first
                transmission attempt on a particular interface
                is delayed because the medium is busy.

                The count represented by an instance of this
                object does not include frames involved in
                collisions.

                This counter does not increment when the
                interface is operating in full-duplex mode.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.9,
                aFramesWithDeferredXmissions."
    ::= { dot3StatsEntry 7 }

dot3StatsLateCollisions OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of times that a collision is
                detected on a particular interface later than
                one slotTime into the transmission of a packet.

                A (late) collision included in a count
                represented by an instance of this object is
                also considered as a (generic) collision for
                purposes of other collision-related
                statistics.

                This counter does not increment when the
```

```
                interface is operating in full-duplex mode.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.10,
                aLateCollisions."
    ::= { dot3StatsEntry 8 }

dot3StatsExcessiveCollisions OBJECT-TYPE
    SYNTAX      Counter32

    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames for which transmission on a
                particular interface fails due to excessive
                collisions.

                This counter does not increment when the
                interface is operating in full-duplex mode.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.11,
                aFramesAbortedDueToXSColls."
    ::= { dot3StatsEntry 9 }

dot3StatsInternalMacTransmitErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames for which transmission on a
                particular interface fails due to an internal
                MAC sublayer transmit error. A frame is only
                counted by an instance of this object if it is
                not counted by the corresponding instance of
                either the dot3StatsLateCollisions object, the
                dot3StatsExcessiveCollisions object, or the
                dot3StatsCarrierSenseErrors object.

                The precise meaning of the count represented by
                an instance of this object is implementation-
                specific. In particular, an instance of this
                object may represent a count of transmission
                errors on a particular interface that are not
                otherwise counted.

                For interfaces operating at 10 Gb/s, this
                counter can roll over in less than 5 minutes if
                it is incrementing at its maximum rate. Since
                that amount of time could be less than a
                management station’s poll cycle time, in order
                to avoid a loss of information, a management
                station is advised to poll the
                dot3HCStatsInternalMacTransmitErrors object for
                10 Gb/s or faster interfaces.
```

```
                Discontinuities in the value of this counter can

                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.12,
                aFramesLostDueToIntMACXmitError."
    ::= { dot3StatsEntry 10 }

dot3StatsCarrierSenseErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of times that the carrier sense
                condition was lost or never asserted when
                attempting to transmit a frame on a particular
                interface.

                The count represented by an instance of this
                object is incremented at most once per
                transmission attempt, even if the carrier sense
                condition fluctuates during a transmission
                attempt.

                This counter does not increment when the
                interface is operating in full-duplex mode.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.13,
                aCarrierSenseErrors."
    ::= { dot3StatsEntry 11 }

-- { dot3StatsEntry 12 } is not assigned

dot3StatsFrameTooLongs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames received on a particular
                interface that exceed the maximum permitted
                frame size.

                The count represented by an instance of this
                object is incremented when the frameTooLong
                status is returned by the MAC service to the
                LLC (or other MAC user). Received frames for
                which multiple error conditions pertain are,

                according to the conventions of IEEE 802.3
                Layer Management, counted exclusively according
                to the error status presented to the LLC.

                For interfaces operating at 10 Gb/s, this
                counter can roll over in less than 80 minutes if
                it is incrementing at its maximum rate. Since
```

```
                that amount of time could be less than a
                management station's poll cycle time, in order
                to avoid a loss of information, a management
                station is advised to poll the
                dot3HCStatsFrameTooLongs object for 10 Gb/s
                or faster interfaces.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.25,
                aFrameTooLongErrors."
    ::= { dot3StatsEntry 13 }

-- { dot3StatsEntry 14 } is not assigned

-- { dot3StatsEntry 15 } is not assigned

dot3StatsInternalMacReceiveErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames for which reception on a
                particular interface fails due to an internal
                MAC sublayer receive error. A frame is only
                counted by an instance of this object if it is
                not counted by the corresponding instance of
                either the dot3StatsFrameTooLongs object, the
                dot3StatsAlignmentErrors object, or the
                dot3StatsFCSErrors object.

                The precise meaning of the count represented by
                an instance of this object is implementation-
                specific. In particular, an instance of this
                object may represent a count of receive errors
                on a particular interface that are not
                otherwise counted.

                For interfaces operating at 10 Gb/s, this
                counter can roll over in less than 5 minutes if

                it is incrementing at its maximum rate. Since
                that amount of time could be less than a
                management station's poll cycle time, in order
                to avoid a loss of information, a management
                station is advised to poll the
                dot3HCStatsInternalMacReceiveErrors object for
                10 Gb/s or faster interfaces.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.15,
                aFramesLostDueToIntMACRcvError."
    ::= { dot3StatsEntry 16 }

dot3StatsSymbolErrors OBJECT-TYPE
```



```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "For an interface operating at 100 Mb/s, the
            number of times there was an invalid data symbol
            when a valid carrier was present.

            For an interface operating in half-duplex mode
            at 1000 Mb/s, the number of times the receiving
            media is non-idle (a carrier event) for a period
            of time equal to or greater than slotTime, and
            during which there was at least one occurrence
            of an event that causes the PHY to indicate
            'Data reception error' or 'carrier extend error'
            on the GMII.

            For an interface operating in full-duplex mode
            at 1000 Mb/s, the number of times the receiving
            media is non-idle (a carrier event) for a period
            of time equal to or greater than minFrameSize,
            and during which there was at least one
            occurrence of an event that causes the PHY to
            indicate 'Data reception error' on the GMII.

            For an interface operating at 10 Gb/s, the
            number of times the receiving media is non-idle
            (a carrier event) for a period of time equal to
            or greater than minFrameSize, and during which
            there was at least one occurrence of an event
            that causes the PHY to indicate 'Receive Error'
            on the XGMII.

            The count represented by an instance of this
            object is incremented at most once per carrier
            event, even if multiple symbol errors occur
            during the carrier event. This count does
            not increment if a collision is present.

            This counter does not increment when the
            interface is operating at 10 Mb/s.

            For interfaces operating at 10 Gb/s, this
            counter can roll over in less than 5 minutes if
            it is incrementing at its maximum rate. Since
            that amount of time could be less than a

            management station's poll cycle time, in order
            to avoid a loss of information, a management
            station is advised to poll the
            dot3HCStatsSymbolErrors object for 10 Gb/s
            or faster interfaces.

            Discontinuities in the value of this counter can
            occur at re-initialization of the management
            system, and at other times as indicated by the
            value of ifCounterDiscontinuityTime."
REFERENCE   "IEEE Std 802.3, 30.3.2.1.5,
            aSymbolErrorDuringCarrier."
::= { dot3StatsEntry 17 }
```



```
dot3StatsDuplexStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    unknown(1),
                    halfDuplex(2),
                    fullDuplex(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current mode of operation of the MAC
                entity. ‘unknown’ indicates that the current
                duplex mode could not be determined.

                Management control of the duplex mode is
                accomplished through the MAU MIB. When
                an interface does not support autonegotiation,
                or when autonegotiation is not enabled, the
                duplex mode is controlled using
                ifMauDefaultType. When autonegotiation is
                supported and enabled, duplex mode is controlled
                using ifMauAutoNegAdvertisedBits. In either
                case, the currently operating duplex mode is
                reflected both in this object and in ifMauType.

                Note that this object provides redundant
                information with ifMauType. Normally, redundant
                objects are discouraged. However, in this
                instance, it allows a management application to
                determine the duplex status of an interface
                without having to know every possible value of
                ifMauType. This was felt to be sufficiently
                valuable to justify the redundancy."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.32,
                aDuplexStatus."
    ::= { dot3StatsEntry 18 }

dot3StatsRateControlAbility OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "‘true’ for interfaces operating at speeds above
                1000 Mb/s that support Rate Control through
                lowering the average data rate of the MAC
                sublayer, with frame granularity, and ‘false’
                otherwise."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.33,
                aRateControlAbility."
    ::= { dot3StatsEntry 19 }

dot3StatsRateControlStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    rateControlOff(1),
                    rateControlOn(2),
                    unknown(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current Rate Control mode of operation of
                the MAC sublayer of this interface."
```

```
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.34,
                aRateControlStatus."
    ::= { dot3StatsEntry 20 }

dot3StatsMaxFrameLength OBJECT-TYPE
    SYNTAX      INTEGER {
                    unknown(1),
                    baseFrame(2),
                    qTaggedFrame(3),
                    envelopeFrame(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "This indicates the MAC frame length at
                 which the dot3StatsFrameTooLongs counter is
                 incremented."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.37, aMaxFrameLength."
    ::= { dot3StatsEntry 21 }

-- the Ethernet-like Collision Statistics group

-- Implementation of this group is optional; it is appropriate
-- for all systems which have the necessary metering

dot3CollTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot3CollEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A collection of collision histograms for a
                particular set of interfaces."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.30,
                aCollisionFrames."
    ::= { ieee8023etherMIBObjects 5 }

dot3CollEntry OBJECT-TYPE
    SYNTAX      Dot3CollEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A cell in the histogram of per-frame
                collisions for a particular interface. An

                instance of this object represents the
                frequency of individual MAC frames for which
                the transmission (successful or otherwise) on a
                particular interface is accompanied by a
                particular number of media collisions."
    INDEX       { ifIndex, dot3CollCount }
    ::= { dot3CollTable 1 }

Dot3CollEntry ::=
    SEQUENCE {
        dot3CollCount        Integer32,
        dot3CollFrequencies  Counter32
    }

-- { dot3CollEntry 1 } is no longer in use

dot3CollCount OBJECT-TYPE
    SYNTAX      Integer32 (1..16)
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The number of per-frame media collisions for
                which a particular collision histogram cell
                represents the frequency on a particular
                interface."
    ::= { dot3CollEntry 2 }

dot3CollFrequencies OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of individual MAC frames for which the
                transmission (successful or otherwise) on a
                particular interface occurs after the
                frame has experienced exactly the number
                of collisions in the associated
                dot3CollCount object.

                For example, a frame which is transmitted
                on interface 77 after experiencing
                exactly 4 collisions would be indicated
                by incrementing only dot3CollFrequencies.77.4.
                No other instance of dot3CollFrequencies would
                be incremented in this example.

                This counter does not increment when the
                interface is operating in full-duplex mode.

                Discontinuities in the value of this counter can

                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    ::= { dot3CollEntry 3 }

dot3ControlTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot3ControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of descriptive and status information
                about the MAC Control sublayer on the
                Ethernet-like interfaces attached to a
                particular system. There will be one row in
                this table for each Ethernet-like interface in
                the system which implements the MAC Control
                sublayer. If some, but not all, of the
                Ethernet-like interfaces in the system implement
                the MAC Control sublayer, there will be fewer
                rows in this table than in the dot3StatsTable."
    ::= { ieee8023etherMIBObjects 9 }

dot3ControlEntry OBJECT-TYPE
    SYNTAX      Dot3ControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the table, containing information
                about the MAC Control sublayer on a single
                Ethernet-like interface."
```

```
    INDEX       { dot3StatsIndex }
    ::= { dot3ControlTable 1 }

Dot3ControlEntry ::=
    SEQUENCE {
        dot3ControlFunctionsSupported       BITS,
        dot3ControlInUnknownOpcodes         Counter32,
        dot3HCControlInUnknownOpcodes       Counter64
    }

dot3ControlFunctionsSupported OBJECT-TYPE
    SYNTAX      BITS {
                    pause(0)   -- 802.3 flow control
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A list of the possible MAC Control functions
                implemented for this interface."
    REFERENCE   "IEEE Std 802.3, 30.3.3.2,
                aMACControlFunctionsSupported."

    ::= { dot3ControlEntry 1 }

dot3ControlInUnknownOpcodes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of MAC Control frames received on this
                interface that contain an opcode that is not
                supported by this device.

                For interfaces operating at 10 Gb/s, this
                counter can roll over in less than 5 minutes if
                it is incrementing at its maximum rate. Since
                that amount of time could be less than a
                management station's poll cycle time, in order
                to avoid a loss of information, a management
                station is advised to poll the
                dot3HCControlInUnknownOpcodes object for 10 Gb/s
                or faster interfaces.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.3.5,
                aUnsupportedOpcodesReceived"
    ::= { dot3ControlEntry 2 }

dot3HCControlInUnknownOpcodes OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of MAC Control frames received on this
                interface that contain an opcode that is not
                supported by this device.

                This counter is a 64-bit version of
                dot3ControlInUnknownOpcodes. It should be used
```



```
                on interfaces operating at 10 Gb/s or faster.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.3.5,
                aUnsupportedOpcodesReceived"
    ::= { dot3ControlEntry 3 }


dot3PauseTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot3PauseEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of descriptive and status information
                about the MAC Control PAUSE function on the
                Ethernet-like interfaces attached to a
                particular system. There will be one row in
                this table for each Ethernet-like interface in
                the system which supports the MAC Control PAUSE
                function (i.e., the 'pause' bit in the
                corresponding instance of
                dot3ControlFunctionsSupported is set). If some,
                but not all, of the Ethernet-like interfaces in
                the system implement the MAC Control PAUSE
                function (for example, if some interfaces only
                support half-duplex), there will be fewer rows
                in this table than in the dot3StatsTable."
    ::= { ieee8023etherMIBObjects 10 }

dot3PauseEntry OBJECT-TYPE
    SYNTAX      Dot3PauseEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the table, containing information
                about the MAC Control PAUSE function on a single
                Ethernet-like interface."
    INDEX       { dot3StatsIndex }
    ::= { dot3PauseTable 1 }

Dot3PauseEntry ::=

    SEQUENCE {
        dot3PauseAdminMode                  INTEGER,
        dot3PauseOperMode                   INTEGER,
        dot3InPauseFrames                   Counter32,
        dot3OutPauseFrames                  Counter32,
        dot3HCInPauseFrames                 Counter64,
        dot3HCOutPauseFrames                Counter64
    }

dot3PauseAdminMode OBJECT-TYPE
    SYNTAX      INTEGER {
                    disabled(1),
                    enabledXmit(2),
                    enabledRcv(3),
                    enabledXmitAndRcv(4)
                }
```



```
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "This object is used to configure the default
                administrative PAUSE mode for this interface.

                This object represents the
                administratively-configured PAUSE mode for this
                interface. If Auto-Negotiation is not enabled
                or is not implemented for the active MAU
                attached to this interface, the value of this
                object determines the operational PAUSE mode
                of the interface whenever it is operating in
                full-duplex mode. In this case, a set to this
                object will force the interface into the
                specified mode.

                If Auto-Negotiation is implemented and enabled
                for the MAU attached to this interface, the
                PAUSE mode for this interface is determined by
                Auto-Negotiation, and the value of this object
                denotes the mode to which the interface will
                automatically revert if/when Auto-Negotiation is
                later disabled. Note that when Auto-Negotiation
                is running, administrative control of the PAUSE
                mode may be accomplished using the
                ifMauAutoNegCapAdvertisedBits object in the
                MAU-MIB module.

                Note that the value of this object is ignored
                when the interface is not operating in
                full-duplex mode.

                An attempt to set this object to
                ‘enabledXmit(2)’ or ‘enabledRcv(3)’ will fail
                on interfaces that do not support operation
                at greater than 100 Mb/s."
    ::= { dot3PauseEntry 1 }

dot3PauseOperMode OBJECT-TYPE
    SYNTAX      INTEGER {
                    disabled(1),
                    enabledXmit(2),
                    enabledRcv(3),
                    enabledXmitAndRcv(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "This object reflects the PAUSE mode currently

                in use on this interface, as determined by
                either (1) the result of the Auto-Negotiation
                function or (2) if Auto-Negotiation is not
                enabled or is not implemented for the active MAU
                attached to this interface, by the value of
                dot3PauseAdminMode. Interfaces operating at
                100 Mb/s or less will never return
                ‘enabledXmit(2)’ or ‘enabledRcv(3)’. Interfaces
                operating in half-duplex mode will always return
```

```
                'disabled(1)'. Interfaces on which
                Auto-Negotiation is enabled but not yet
                completed should return the value
                'disabled(1)'."
    ::= { dot3PauseEntry 2 }

dot3InPauseFrames OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of MAC Control frames received on this
                interface with an opcode indicating the PAUSE
                operation.

                This counter does not increment when the
                interface is operating in half-duplex mode.

                For interfaces operating at 10 Gb/s, this
                counter can roll over in less than 5 minutes if
                it is incrementing at its maximum rate. Since
                that amount of time could be less than a
                management station's poll cycle time, in order
                to avoid a loss of information, a management
                station is advised to poll the
                dot3HCInPauseFrames object for 10 Gb/s or
                faster interfaces.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.4.3,
                aPAUSEMACCtrlFramesReceived."
    ::= { dot3PauseEntry 3 }

dot3OutPauseFrames OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current

    DESCRIPTION "A count of MAC Control frames transmitted on
                this interface with an opcode indicating the
                PAUSE operation.

                This counter does not increment when the
                interface is operating in half-duplex mode.

                For interfaces operating at 10 Gb/s, this
                counter can roll over in less than 5 minutes if
                it is incrementing at its maximum rate. Since
                that amount of time could be less than a
                management station's poll cycle time, in order
                to avoid a loss of information, a management
                station is advised to poll the
                dot3HCOutPauseFrames object for 10 Gb/s or
                faster interfaces.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
```

```
                 system, and at other times as indicated by the
                 value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.4.2,
                 aPAUSEMACCtrlFramesTransmitted."
    ::= { dot3PauseEntry 4 }

dot3HCInPauseFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of MAC Control frames received on this
                 interface with an opcode indicating the PAUSE
                 operation.

                 This counter does not increment when the
                 interface is operating in half-duplex mode.

                 This counter is a 64-bit version of
                 dot3InPauseFrames. It should be used on
                 interfaces operating at 10 Gb/s or faster.

                 Discontinuities in the value of this counter can
                 occur at re-initialization of the management
                 system, and at other times as indicated by the
                 value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.4.3,
                 aPAUSEMACCtrlFramesReceived."
    ::= { dot3PauseEntry 5 }


dot3HCOutPauseFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of MAC Control frames transmitted on
                 this interface with an opcode indicating the
                 PAUSE operation.

                 This counter does not increment when the
                 interface is operating in half-duplex mode.

                 This counter is a 64-bit version of
                 dot3OutPauseFrames. It should be used on
                 interfaces operating at 10 Gb/s or faster.

                 Discontinuities in the value of this counter can
                 occur at re-initialization of the management
                 system, and at other times as indicated by the
                 value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.4.2,
                 aPAUSEMACCtrlFramesTransmitted."
    ::= { dot3PauseEntry 6 }

dot3HCStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot3HCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table containing 64-bit versions of error
                 counters from the dot3StatsTable. The 32-bit
```

```
                versions of these counters may roll over quite
                quickly on higher speed ethernet interfaces.
                The counters that have 64-bit versions in this
                table are the counters that apply to full-duplex
                interfaces, since 10 Gb/s and faster
                Ethernet-like interfaces do not support
                half-duplex, and very few 1000 Mb/s
                Ethernet-like interfaces support half-duplex.

                Entries in this table are recommended for
                interfaces capable of operating at 1000 Mb/s or
                faster, and are required for interfaces capable
                of operating at 10 Gb/s or faster. Lower speed
                Ethernet-like interfaces do not need entries in
                this table, in which case there may be fewer
                entries in this table than in the
                dot3StatsTable. However, implementations
                containing interfaces with a mix of speeds may
                choose to implement entries in this table for
                all Ethernet-like interfaces."
    ::= { ieee8023etherMIBObjects 11 }

dot3HCStatsEntry OBJECT-TYPE
    SYNTAX      Dot3HCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry containing 64-bit statistics for a
                single Ethernet-like interface."
    INDEX       { dot3StatsIndex }
    ::= { dot3HCStatsTable 1 }

Dot3HCStatsEntry ::=
    SEQUENCE {
        dot3HCStatsAlignmentErrors           Counter64,
        dot3HCStatsFCSErrors                 Counter64,
        dot3HCStatsInternalMacTransmitErrors Counter64,
        dot3HCStatsFrameTooLongs             Counter64,
        dot3HCStatsInternalMacReceiveErrors  Counter64,
        dot3HCStatsSymbolErrors              Counter64
    }

dot3HCStatsAlignmentErrors OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames received on a particular
                interface that are not an integral number of
                octets in length and do not pass the FCS check.

                The count represented by an instance of this
                object is incremented when the alignmentError
                status is returned by the MAC service to the
                LLC (or other MAC user). Received frames for
                which multiple error conditions pertain are,
                according to the conventions of IEEE 802.3
                Layer Management, counted exclusively according
                to the error status presented to the LLC.

                This counter does not increment for group
```

```
                encoding schemes greater than 4 bits per group.

                This counter is a 64-bit version of
                dot3StatsAlignmentErrors. It should be used
                on interfaces operating at 10 Gb/s or faster.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management

                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.7,
                aAlignmentErrors"
    ::= { dot3HCStatsEntry 1 }

dot3HCStatsFCSErrors OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames received on a particular
                interface that are an integral number of octets
                in length but do not pass the FCS check. This
                count does not include frames received with
                frame-too-long or frame-too-short error.

                The count represented by an instance of this
                object is incremented when the frameCheckError
                status is returned by the MAC service to the
                LLC (or other MAC user). Received frames for
                which multiple error conditions pertain are,
                according to the conventions of IEEE 802.3
                Layer Management, counted exclusively according
                to the error status presented to the LLC.

                Note:  Coding errors detected by the physical
                layer for speeds above 10 Mb/s will cause the
                frame to fail the FCS check.

                This counter is a 64-bit version of
                dot3StatsFCSErrors. It should be used on
                interfaces operating at 10 Gb/s or faster.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.6,
                aFrameCheckSequenceErrors."
    ::= { dot3HCStatsEntry 2 }

dot3HCStatsInternalMacTransmitErrors OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of frames for which transmission on a
                particular interface fails due to an internal
                MAC sublayer transmit error. A frame is only

                counted by an instance of this object if it is
```



```
                 not counted by the corresponding instance of
                 either the dot3StatsLateCollisions object, the
                 dot3StatsExcessiveCollisions object, or the
                 dot3StatsCarrierSenseErrors object.

                 The precise meaning of the count represented by
                 an instance of this object is implementation-
                 specific. In particular, an instance of this
                 object may represent a count of transmission
                 errors on a particular interface that are not
                 otherwise counted.

                 This counter is a 64-bit version of
                 dot3StatsInternalMacTransmitErrors. It should
                 be used on interfaces operating at 10 Gb/s or
                 faster.

                 Discontinuities in the value of this counter can
                 occur at re-initialization of the management
                 system, and at other times as indicated by the
                 value of ifCounterDiscontinuityTime."
    REFERENCE    "IEEE Std 802.3, 30.3.1.1.12,
                 aFramesLostDueToIntMACXmitError."
    ::= { dot3HCStatsEntry 3 }

dot3HCStatsFrameTooLongs OBJECT-TYPE
    SYNTAX       Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION "A count of frames received on a particular
                 interface that exceed the maximum permitted
                 frame size.

                 The count represented by an instance of this
                 object is incremented when the frameTooLong
                 status is returned by the MAC service to the
                 LLC (or other MAC user). Received frames for
                 which multiple error conditions pertain are,
                 according to the conventions of IEEE 802.3
                 Layer Management, counted exclusively according
                 to the error status presented to the LLC.

                 This counter is a 64-bit version of
                 dot3StatsFrameTooLongs. It should be used on
                 interfaces operating at 10 Gb/s or faster.

                 Discontinuities in the value of this counter can

                 occur at re-initialization of the management
                 system, and at other times as indicated by the
                 value of ifCounterDiscontinuityTime."
    REFERENCE    "IEEE Std 802.3, 30.3.1.1.25,
                 aFrameTooLongErrors."
    ::= { dot3HCStatsEntry 4 }

dot3HCStatsInternalMacReceiveErrors OBJECT-TYPE
    SYNTAX       Counter64
    MAX-ACCESS   read-only
    STATUS       current
```



```
    DESCRIPTION "A count of frames for which reception on a
                particular interface fails due to an internal
                MAC sublayer receive error. A frame is only
                counted by an instance of this object if it is
                not counted by the corresponding instance of
                either the dot3StatsFrameTooLongs object, the
                dot3StatsAlignmentErrors object, or the
                dot3StatsFCSErrors object.

                The precise meaning of the count represented by
                an instance of this object is implementation-
                specific. In particular, an instance of this
                object may represent a count of receive errors
                on a particular interface that are not
                otherwise counted.

                This counter is a 64-bit version of
                dot3StatsInternalMacReceiveErrors. It should be
                used on interfaces operating at 10 Gb/s or
                faster.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.1.1.15,
                aFramesLostDueToIntMACRcvError."
    ::= { dot3HCStatsEntry 5 }

dot3HCStatsSymbolErrors OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "For an interface operating at 100 Mb/s, the
                number of times there was an invalid data symbol
                when a valid carrier was present.


                For an interface operating in half-duplex mode
                at 1000 Mb/s, the number of times the receiving
                media is non-idle (a carrier event) for a period
                of time equal to or greater than slotTime, and
                during which there was at least one occurrence
                of an event that causes the PHY to indicate
                'Data reception error' or 'carrier extend error'
                on the GMII.

                For an interface operating in full-duplex mode
                at 1000 Mb/s, the number of times the receiving
                media is non-idle (a carrier event) for a period
                of time equal to or greater than minFrameSize,
                and during which there was at least one
                occurrence of an event that causes the PHY to
                indicate 'Data reception error' on the GMII.

                For an interface operating at 10 Gb/s, the
                number of times the receiving media is non-idle
                (a carrier event) for a period of time equal to
                or greater than minFrameSize, and during which
```



```
                there was at least one occurrence of an event
                that causes the PHY to indicate 'Receive Error'
                on the XGMII.

                The count represented by an instance of this
                object is incremented at most once per carrier
                event, even if multiple symbol errors occur
                during the carrier event. This count does
                not increment if a collision is present.

                This counter is a 64-bit version of
                dot3StatsSymbolErrors. It should be used on
                interfaces operating at 10 Gb/s or faster.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system, and at other times as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.3.2.1.5,
                aSymbolErrorDuringCarrier."
    ::= { dot3HCStatsEntry 6 }


-- { ieee8023etherMIBObjects 6 }, the dot3ChipSets tree,
--   is defined in [RFC2666]

-- Conformance statements

etherConformance OBJECT IDENTIFIER ::= { ieee8023etherMIB 2 }

etherGroups      OBJECT IDENTIFIER ::= { etherConformance 1 }
etherCompliances OBJECT IDENTIFIER ::= { etherConformance 2 }

-- Compliance statements

    dot3Compliance2 MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION "The compliance statement for managed network
                    entities which have Ethernet-like network
                    interfaces.

                    Note that compliance with this MIB module
                    requires compliance with the ifCompliance3
                    MODULE-COMPLIANCE statement of the IF-MIB
                    (IETF RFC 2863). In addition, compliance with this
                    MIB module requires compliance  with the
                    mauModIfCompl3 MODULE-COMPLIANCE statement of
                    the MAU-MIB module defined in Clause 13."

    MODULE  -- this module
        MANDATORY-GROUPS { etherStatsBaseGroup2 }

        GROUP       etherDuplexGroup
        DESCRIPTION "This group is mandatory for all
                    Ethernet-like network interfaces which are
                    capable of operating in full-duplex mode.
                    It is highly recommended for all
                    Ethernet-like network interfaces."
```

```
GROUP       etherRateControlGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces which are
            capable of operating at speeds faster than
            1000 Mb/s. It is highly recommended for all
            Ethernet-like network interfaces."

GROUP       etherStatsLowSpeedGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces which are
            capable of operating at 10 Mb/s or slower in
            half-duplex mode."

GROUP       etherStatsHighSpeedGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces which are
            capable of operating at 100 Mb/s or faster."

GROUP       etherStatsHalfDuplexGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces which are

            capable of operating in half-duplex mode."

GROUP       etherHCStatsGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces which are
            capable of operating at 10 Gb/s or faster.
            It is recommended for all Ethernet-like
            network interfaces which are capable of
            operating at 1000 Mb/s or faster."

GROUP       etherControlGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces that
            support the MAC Control sublayer."

GROUP       etherHCControlGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces that
            support the MAC Control sublayer and are
            capable of operating at 10 Gb/s or faster."

GROUP       etherControlPauseGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces that
            support the MAC Control PAUSE function."

GROUP       etherHCControlPauseGroup
DESCRIPTION "This group is mandatory for all
            Ethernet-like network interfaces that
            support the MAC Control PAUSE function and
            are capable of operating at 10 Gb/s or
            faster."

GROUP       etherCollisionTableGroup
DESCRIPTION "This group is optional. It is appropriate
            for all Ethernet-like network interfaces
            which are capable of operating in
```

```
                    half-duplex mode and have the necessary
                    metering. Implementation in systems with
                    such interfaces is highly recommended."
    ::= { etherCompliances 1 }

-- units of conformance

etherCollisionTableGroup OBJECT-GROUP
    OBJECTS     { dot3CollFrequencies
                }
    STATUS      current
    DESCRIPTION "A collection of objects providing a histogram
                of packets successfully transmitted after
                experiencing exactly N collisions."
    ::= { etherGroups 1 }

etherStatsLowSpeedGroup OBJECT-GROUP
    OBJECTS     { dot3StatsSQETestErrors }
    STATUS      current
    DESCRIPTION "A collection of objects providing information
                applicable to Ethernet-like network interfaces
                capable of operating at 10 Mb/s or slower in
                half-duplex mode."
    ::= { etherGroups 2 }

etherStatsHighSpeedGroup OBJECT-GROUP
    OBJECTS     { dot3StatsSymbolErrors }
    STATUS      current
    DESCRIPTION "A collection of objects providing information
                applicable to Ethernet-like network interfaces
                capable of operating at 100 Mb/s or faster."
    ::= { etherGroups 3 }

etherDuplexGroup OBJECT-GROUP
    OBJECTS     { dot3StatsDuplexStatus }
    STATUS      current
    DESCRIPTION "A collection of objects providing information
                about the duplex mode of an Ethernet-like
                network interface."
    ::= { etherGroups 4 }

etherControlGroup OBJECT-GROUP
    OBJECTS     { dot3ControlFunctionsSupported,
                  dot3ControlInUnknownOpcodes
                }
    STATUS      current
    DESCRIPTION "A collection of objects providing information
                about the MAC Control sublayer on Ethernet-like
                network interfaces."
    ::= { etherGroups 5 }

etherControlPauseGroup OBJECT-GROUP
    OBJECTS     { dot3PauseAdminMode,
                  dot3PauseOperMode,
                  dot3InPauseFrames,
                  dot3OutPauseFrames
                }
    STATUS      current
    DESCRIPTION "A collection of objects providing information
```

```
                 about and control of the MAC Control PAUSE
                 function on Ethernet-like network interfaces."
    ::= { etherGroups 6 }

etherStatsBaseGroup2 OBJECT-GROUP
    OBJECTS     { dot3StatsAlignmentErrors,
                  dot3StatsFCSErrors,
                  dot3StatsInternalMacTransmitErrors,
                  dot3StatsFrameTooLongs,
                  dot3StatsInternalMacReceiveErrors,
                  dot3StatsMaxFrameLength
                }
    STATUS      current
    DESCRIPTION "A collection of objects providing information
                applicable to all Ethernet-like network
                interfaces."
    ::= { etherGroups 7 }

etherStatsHalfDuplexGroup OBJECT-GROUP
    OBJECTS     { dot3StatsSingleCollisionFrames,
                  dot3StatsMultipleCollisionFrames,
                  dot3StatsDeferredTransmissions,
                  dot3StatsLateCollisions,
                  dot3StatsExcessiveCollisions,
                  dot3StatsCarrierSenseErrors
                }
    STATUS      current
    DESCRIPTION "A collection of objects providing information
                applicable only to half-duplex Ethernet-like
                network interfaces."
    ::= { etherGroups 8 }

etherHCStatsGroup OBJECT-GROUP
    OBJECTS     { dot3HCStatsAlignmentErrors,
                  dot3HCStatsFCSErrors,
                  dot3HCStatsInternalMacTransmitErrors,
                  dot3HCStatsFrameTooLongs,
                  dot3HCStatsInternalMacReceiveErrors,
                  dot3HCStatsSymbolErrors
                }
    STATUS      current
    DESCRIPTION "A collection of objects providing high-capacity
                statistics applicable to higher-speed
                Ethernet-like network interfaces."
    ::= { etherGroups 9 }

etherHCControlGroup OBJECT-GROUP
    OBJECTS     { dot3HCControlInUnknownOpcodes }
    STATUS      current
    DESCRIPTION "A collection of objects providing high-capacity
                statistics for the MAC Control sublayer on
                higher-speed Ethernet-like network interfaces."
    ::= { etherGroups 10 }

etherHCControlPauseGroup OBJECT-GROUP
    OBJECTS     { dot3HCInPauseFrames,
                  dot3HCOutPauseFrames

                }
```

```
        STATUS      current
        DESCRIPTION "A collection of objects providing high-capacity
                    statistics for the MAC Control PAUSE function on
                    higher-speed Ethernet-like network interfaces."
        ::= { etherGroups 11 }

    etherRateControlGroup OBJECT-GROUP
        OBJECTS     { dot3StatsRateControlAbility,
                      dot3StatsRateControlStatus
                    }
        STATUS      current
        DESCRIPTION "A collection of objects providing information
                    about the Rate Control function on Ethernet-like
                    interfaces."
        ::= { etherGroups 12 }

END
```


11. Ethernet in the First Mile copper (EFMCu) interfaces MIB module

11.1 Introduction

Ethernet-like interfaces have been defined in IEEE Std 802.3 known as Ethernet in the First Mile (EFM). In particular, 2BASE-TL and 10PASS-TS physical interfaces (PHYs), defined over voice-grade copper pairs, have been specified for the long and short reach, respectively. These interfaces, collectively called EFM Copper (EFMCu), are based on single-pair high-speed digital subscriber line (SHDSL, see ITU-T G.991.2) and very high-speed digital subscriber line (VDSL, see ITU-T G.993.1) technology, supporting optional physical medium entity (PME) aggregation (a.k.a. multi-pair bonding) with variable rates.

The 2BASE-TL PHY is capable of providing at least 2 Mb/s over a 2700 m long single copper pair with a mean bit error ratio (BER) of 10^{-7} (using 5 dB target noise margin).

The 10PASS-TS PHY is capable of providing at least 10 Mb/s over a 750 m long single copper pair with a mean BER of 10^{-7} (using 6 dB target noise margin). This clause defines a Management Information Base (MIB) module for use with SNMP to manage EFMCu interfaces. In addition, a MIB module is defined describing the cross-connect capability of a stacked interface.

11.2 Relation to other MIB modules

This subclause outlines the relationship of the MIB modules defined in this clause with other MIB modules described in other clauses of this standard, or the relevant RFCs. Specifically, the Interfaces Group MIB (IF-MIB), Ethernet-Like (IEEE8023-EtherLike-MIB), MAU (MAU-MIB), SHDSL (HDSL2-SHDSL-LINE-MIB), and VDSL (VDSL-LINE-EXT-MCM-MIB) modules are discussed.

11.2.1 Relation to Interfaces Group MIB module

2BASE-TL and 10PASS-TS PHYs specified in the EFM-CU-MIB module are stacked (a.k.a. aggregated or bonded) Ethernet interfaces and as such are managed using generic interface management objects defined in the IF-MIB defined in IETF RFC 2863.

The stack management (i.e., actual connection of the sublayers to the top-layer interface) is done via the ifStackTable, as defined in the IF-MIB defined in IETF RFC 2863, and its inverse ifInvStackTable, as defined in the IF-INVERTED-STACK-MIB defined in IETF RFC 2864.

The new tables ifCapStackTable and its inverse ifInvCapStackTable defined in the IF-CAP-STACK-MIB module below, extend the stack management with an ability to describe possible connections or cross-connect capability, when a flexible cross-connect matrix is present between the interface layers.

11.2.1.1 Layering model

An EFMCu interface can aggregate up to 32 physical medium entity (PME) sublayer devices (modems), using the so-called PME aggregation function (PAF).

A generic EFMCu device can have a number of physical coding sublayer (PCS) ports, each connected to a media access controller (MAC) via a media independent interface (MII) at the upper layer, and cross-connected to a number of underlying PMEs, with a single PCS per PME relationship. See 61.1 of IEEE Std 802.3 for more details.

Each PME in the aggregated EFMCu port is represented in the Interface table (ifTable) as a separate interface with ifType of shdsl(169) for 2BASE-TL or vdsl(97) for 10PASS-TS. The ifType values are defined in [IANAifType-MIB].

ifSpeed for each PME shall return the actual data bitrate of the active PME (e.g., for 2BaseTL PMEs it is a multiple of 64 kb/s). A zero value shall be returned when the PME is Initializing or Down.

The ifSpeed of the PCS is the sum of the current operating data rates of all PMEs in the aggregation group, without the 64/65-octet encapsulation overhead and PAF overhead, but accounting for the inter-frame gaps (IFGs).

When using the stated definition of ifSpeed for the PCS, there would be no frame loss in the configuration shown in Figure 11–1 (the test-sets are configured to generate 100% of back-to-back traffic, i.e., minimal IFG, at 10 Mb/s or 100 Mb/s, with min and max frame sizes; the EFM interfaces are aggregated to achieve the shown speed).

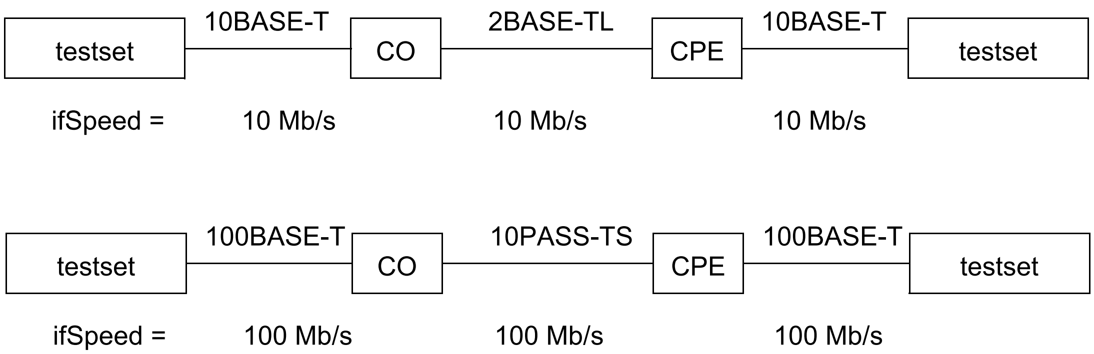


Figure 11–1—Example configuration with no frame loss

Figure 11–2 shows the IEEE 802.3 layering diagram and corresponding use of ifTable and ifMauTable.

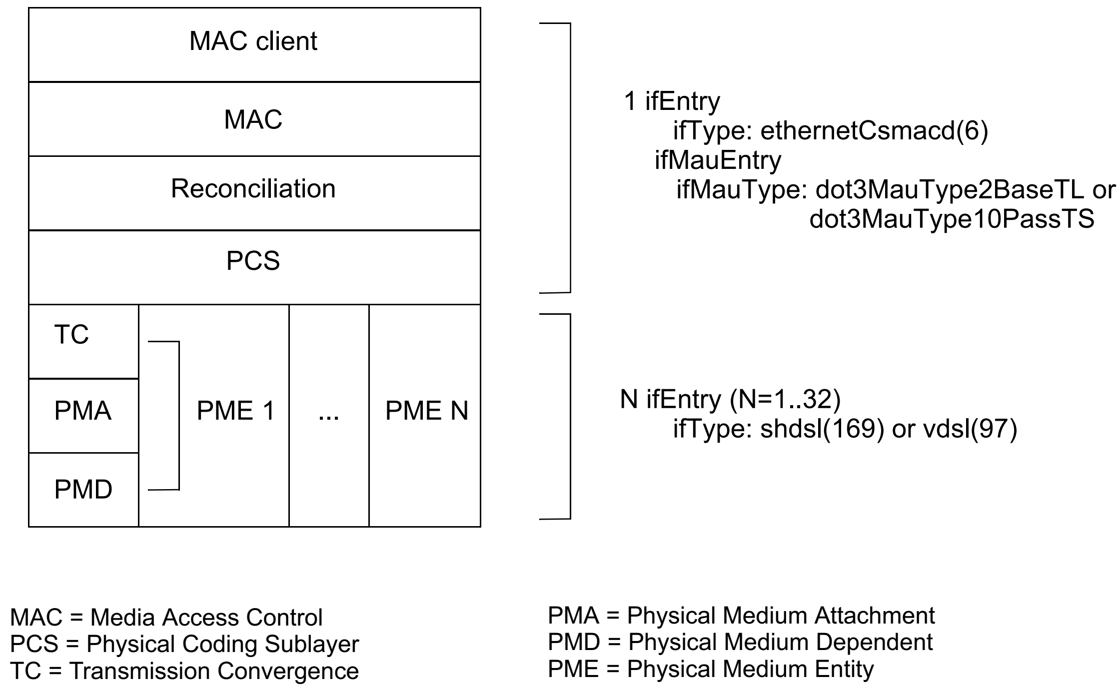


Figure 11–2—Use of ifTable and ifMauTable for EFMCu ports

The ifStackTable is indexed by the ifIndex values of the aggregated EFMCu port (PCS) and the PMEs connected to it. ifStackTable allows a Network Management application to determine which PMEs are connected to a particular PCS and change connections (if supported by the application). The ifInvStackTable, being an inverted version of the ifStackTable, provides an efficient means for a Network Management application to read a subset of the ifStackTable and thereby determine which PCS runs on top of a particular PME.

A new table ifCapStackTable, defined in the IF-CAP-STACK-MIB module, specifies for each higher-layer interface (e.g., PCS port) a list of lower-layer interfaces (e.g., PMEs), which can possibly be cross-connected to that higher-layer interface, determined by the cross- connect capability of the device. This table, modeled after ifStackTable, is read-only, reflecting current cross-connect capability of stacked interface, which can be dynamic in some implementations (e.g., if PMEs are located on a pluggable module and the module is pulled out). Note that PME availability per PCS, described by ifCapStackTable, can be constrained by other parameters, for example, by aggregation capacity of a PCS or by the PME in question being already connected to another PCS. So, in order to ensure that a particular PME can be connected to the PCS, all respective parameters (e.g., ifCapStackTable, ifStackTable, and efmCuPAFCapacity) shall be inspected.

The ifInvCapStackTable, also defined in the IF-CAP-STACK-MIB module, describes which higher-layer interfaces (e.g., PCS ports) can possibly be connected to a particular lower-layer interface (e.g., PME), providing an inverted mapping of the ifCapStackTable. While it contains no additional information beyond that already contained in the ifCapStackTable, the ifInvCapStackTable has the ifIndex values in its INDEX clause in the reverse order, i.e., the lower-layer interface first, and the higher-layer interface second, providing an efficient means for a Network Management application to read a subset of the ifCapStackTable and thereby determine which interfaces can be connected to run on top of a particular interface.

11.2.1.2 PME aggregation function (PAF)

The PME aggregation function (PAF) allows a number of PMEs to be aggregated onto a PCS port, by fragmenting the Ethernet frames, transmitting the fragments over multiple PMEs, and assembling the original frames at the remote port. PAF is optional, meaning that a device with a single PME may perform fragmentation and re-assembly if this function is supported by the device. Note that the agent is required to report on the PAF capability for all EFMCu ports (2BASE-TL and 10PASS-TS).

The EFM-CU-MIB module allows a network management application to query the PAF capability and enable/disable it if supported. Note that enabling PAF effectively turns on fragmentation and re-assembly, even on a single-PME port.

11.2.1.3 Discovery operation

The EFMCu ports may optionally support discovery operation, whereby PMEs, during initialization, exchange information about their respective aggregation groups (PCS). This information can then be used to detect copper misconnections or for an automatic assignment of the local PMEs into aggregation groups instead of a fixed pre-configuration.

The MIB modules defined in this clause allow a network management application to control the EFM discovery mechanism and query its results. Note that the discovery mechanism can work only if PAF is supported and enabled.

Two tables are used by the EFM discovery mechanism: ifStackTable and ifCapStackTable. The following pseudo-code gives an example of the discovery and automatic PME assignment for a generic PAF-enabled multi-PCS EFMCu device, located at central office (CO), using objects defined in these MIB modules and in the IF-MIB. (Note that automatic PME assignment is only shown here for the purposes of the example. Fixed PME pre-assignment, manual assignment, or auto-assignment using an alternative internal algorithm may be chosen by a particular implementation.)

```
// Go over all PCS ports in the CO device
FOREACH pcs[i] IN CO_device
{ // Perform discovery and auto-assignment only on PAF enabled ports
  // with room for more PMEs
  IF ( pcs[i].PAFSupported AND pcs[i].NumPMEs &lt; pcs[i].PAFCapacity )
  { // Assign a unique 6-octet local discovery code to the PCS
    // e.g., MAC address
    dc = pcs[i].DiscoveryCode = MAC[i];
    // Go over all disconnected PMEs, which can
    // potentially be connected to the PCS
    FOREACH pme[j] IN ifCapStackTable[pcs[i]] AND
                   NOT IN ifStackTable[pcs[i]]  // not connected
    { // Try to grab the remote RT_device, by writing the value
      // of the local 6-octet discovery code to the remote
      // discovery code register (via handshake mechanism).
      // This operation is atomic Set-if-Clear action, i.e., it
      // would succeed only if the remote discovery register was
      // zero. Read the remote discovery code register via Get
      // operation to see if the RT_device, attached via the PME
      // is indeed marked as being the CO_device peer.
      pme[j].RemoteDiscoveryCode = dc;          // Set-if-Clear
      r = pme[j].RemoteDiscoveryCode;           // Get
      IF ( r == dc AND pcs[i].NumPMEs &lt; pcs[i].PAFCapacity)
      { // Remote RT_device connected via PME[j] is/was a peer
```



```
         // for PCS[i] and there is room for another PME in the
         // PCS[i] aggregation group (max. PAF capacity is not
         // reached yet).
         // Connect this PME to the PCS (via ifStackTable,
         // ifInvStackTable being inverse of ifStackTable is
         // updated automatically, i.e., pcs[i] is auto-added
         // to ifInvStackTable[pme[j]])
         ADD pme[j] TO ifStackTable[pcs[i]];
         pcs[i].NumPMEs = pcs[i].NumPMEs + 1;
         // Discover all other disconnected PMEs,
         // attached to the same RT_device and connect them to
         // the PCS provided there is enough room for more PMEs.
         FOREACH pme[k] IN ifCapStackTable[pcs[i]] AND
                        NOT IN ifStackTable[pcs[i]]
{ // Get Remote Discovery Code from the PME to see if
           // it belongs to a connected RT_device "grabbed" by
           // the CO_device.
           r = pme[k].RemoteDiscoveryCode;
           IF ( r == dc AND pcs[i].NumPMEs &lt; pcs[i].PAFCapacity)
           { // Physically connect the PME to the PCS
             // (pcs[i] is auto-added TO ifInvStackTable[pme[k]])
             ADD pme[k] TO ifStackTable[pcs[i]];
             pcs[i].NumPMEs = pcs[i].NumPMEs + 1;
           }
         }
       }
       // At this point we have discovered all local PMEs which
       // are physically connected to the same remote RT_device
       // and connected them to PCS[i]. Go to the next PCS.
       BREAK;
     }
   }
 }
```

An SNMP Agent for an EFMCu device builds the ifCapStackTable and its inverse ifInvCapStackTable according to the information contained in the Clause 45 PME_Available_register (see IEEE Std 802.3 61.1.5.3 and 45.2.3.20).

Adding a PME to the ifStackTable row for a specific PCS involves actual connection of the PME to the PCS, which can be done by modifying Clause 45 PME_Aggregate_register (see IEEE Std 802.3 61.1.5.3 and 45.2.3.21).

Note that the PCS port does not have to be operationally ‘down’ for the connection to succeed. In fact, a dynamic PME addition (and removal) may be implemented with an available PME being initialized first (by setting its ifAdminStatus to ‘up’) and then added to an operationally ‘up’ PCS port, by modifying a respective ifStackTable (and respective ifInvStackTable) entry.

It is recommended that a removal of the last operationally ‘up’ PME from an operationally ‘up’ PCS would be rejected by the implementation, as this action would completely drop the link.

11.2.1.4 EFMCu ports initialization

EFMCu ports being built on top of xDSL technology require a lengthy initialization or ‘training’ process, before any data can pass. During this initialization, both ends of a link (peers) work cooperatively to achieve the required data rate on a particular copper pair. Sometimes, when the copper line is too long or the noise on the line is too high, that ‘training’ process may fail to achieve a specific target rate with required characteristics.

The ifAdminStatus object from the IF-MIB controls the desired state of a PCS with all the PMEs connected to it or of an individual PME port. Setting this object to ‘up’ instructs a particular PCS or PME to start the initialization process, which may take tens of seconds for EFMCu ports, especially if PAF is involved. The ifOperStatus object shows the operational state of an interface (extended by the ifMauMediaAvailable object from the MAU-MIB module for PCS and efmCuPmeOperStatus defined in the EFM-CU-MIB module for PME interfaces).

A disconnected PME may be initialized by changing the ifAdminState from ‘down’ to ‘up’. Changing the ifAdminState to ‘up’ on the PCS initializes all PMEs connected to that particular PCS. Note that in case of PAF some interfaces may fail to initialize while others succeed. The PCS is considered operationally ‘up’ if at least one PME aggregated by its PAF is operationally ‘up’. When all PMEs connected to the PCS are ‘down’, the PCS shall be considered operationally ‘lowerLayerDown’. The PCS shall be considered operationally ‘notPresent’ if it is not connected to any PME. The PCS/PME interface shall remain operationally ‘down’ during initialization.

The efmCuPmeOperStatus defined in the EFM-CU-MIB module expands PME’s ifOperStatus value of ‘down’ to ‘downReady’, ‘downNotReady’, and ‘init’ values, indicating various EFMCu PME-specific states.

11.2.1.5 Usage of ifTable

Both PME and PCS interfaces of the EFMCu PHY are managed using interface-specific management objects defined in the EFM-CU-MIB module and generic interface objects from the ifTable of IF-MIB, with all management table entries referenced by the interface index ifIndex.

Table 11–1 summarizes EFMCu-specific interpretations for some of the ifTable objects specified in the mandatory ifGeneralInformationGroup.

Table 11–1—EFMCu interpretation of IF-MIB objects

IF-MIB object	EFMCu interpretation
ifIndex	Interface index. Each PME and each PCS in the EFMCu PHY shall have a unique index, as there are some PCS and PME-specific attributes accessible only on the PCS or PME level.
ifType	ethernetCsmacd(6) for PCS, shdsl(169) for 2BASE-TL PME, vdsl(97) for 10PASS-TS PME. Operating data rate for the PME. For the PCS, it is the sum of the current operating data rates of all PMEs in the aggregation group, without the 64/65-octet encapsulation overhead and PAF overhead, but accounting for the Inter-Frame Gaps (IFGs).
ifSpeed	Setting this object to ‘up’ instructs a particular PCS (with all PMEs connected to it) or PME to start initialization process.
ifAdminStatus	Setting this object to ‘up’ instructs a particular PCS (with all PMEs connected to it) or PME to start initialization process.
ifOperStatus	efmCuPmeOperStatus supplements the ‘down’ value of ifOperStatus for PMEs.

11.2.2 Relation to SHDSL MIB module

G.SHDSL.bis modems, similar to PMEs comprising a 2BASE-TL port, are described in the HDSL2-SHDSL-LINE-MIB module defined in IETF RFC 4319. Note that not all attributes of G.SHDSL modems reflected in the HDSL2-SHDSL-LINE-MIB module have adequate management objects (Clause 30 attributes and Clause 45 registers) in IEEE Std 802.3.

Because of these differences and for the purposes of simplicity, unification of attributes common to both 2BASE-TL and 10PASS-TS PMEs, and name consistency (e.g., prefixing the 2BASE-TL PME related objects with 'efmCuPme2B' instead of 'hdsl2shdsl'), it was decided not to reference HDSL2-SHDSL-LINE-MIB objects, but define all the relevant objects in the EFM-CU-MIB module.

However, if some functionality not available in the EFM-CU-MIB module is required and supported by the PME, e.g., performance monitoring, relevant HDSL2-SHDSL-LINE-MIB groups may be included and applied for PMEs of 2BASE-TL subtype.

11.2.3 Relation to VDSL MIB module

VDSL modems, similar to the PME(s) comprising a 10PASS-TS port, are described in the VDSL-LINE-EXT-MCM-MIB module defined in IETF RFC 4070. Note that not all attributes of VDSL modems reflected in the VDSL-LINE-EXT-MCM-MIB module have adequate management objects (Clause 30 attributes and Clause 45 registers) in IEEE Std 802.3.

Because of these differences and for the purposes of simplicity, unification of attributes common to both 2BASE-TL and 10PASS-TS PMEs, and name consistency, it was decided not to reference VDSL-LINE-EXT-MCM-MIB objects, but define all the relevant objects in the EFM-CU-MIB module.

However, if some functionality not available in the EFM-CU-MIB module is required and supported by the PME, relevant VDSL-LINE-EXT-MCM-MIB groups may be included and applied for PMEs of 10PASS-TS subtype.

11.2.4 Relation to Ethernet-Like and MAU MIB modules

An agent implementing the objects defined in this clause shall also implement the objects required by the Ethernet-like interface MIB module defined in Clause 10 and the objects required by the MAU MIB module defined in Clause 13.

Two new values of ifMauType (OBJECT-IDENTITIES of dot3MauType) and corresponding bit definitions of ifMauTypeListBits (IANAifMauTypeListBits) have been defined in the IANA-MAU-MIB module for EFMCu MAUs:

— dot3MauType2BaseTL and b2BaseTL, for 2BASE-TL MAU
— dot3MauType10PassTS and b10PassTS, for 10PASS-TS MAU

Additionally, the IANA-MAU-MIB module defines two new values of ifMauMediaAvailable, specifically for EFMCu ports: availableReduced and ready (in textual convention IANAifMauMediaAvailable). Due to the PME aggregation, the EFMCu interpretation of some possible ifMauMediaAvailable values differs from other MAUs as follows:

— unknown: the EFMCu interface (PCS with connected PMEs) is Initializing
— ready: the interface is Down, at least one PME in the aggregation group (all PMEs connected to the PCS) is ready for handshake
— available: the interface is Up, all PMEs in the aggregation group are up

- notAvailable: the interface is Down, all PMEs in the aggregation group are Down, no handshake tones are detected by any PME
- availableReduced: the interface is Up, a link fault is detected at the receive direction by one or more PMEs in the aggregation group, but at least one PME is Up
- pmdLinkFault: a link fault is detected at the receive direction by all PMEs in the aggregation group

As an Ethernet-like interface, every EFMCu port [an ifEntry representing a consolidation of LLC, MAC, and PCS (sub)layers] shall return an ifType of ethernetCsmacd(6). While most of the MAU characteristics are not applicable to the EFMCu ports (no Auto-Negotiation, false carriers, or jabber), they shall return an appropriate ifMauType (dot3MauType2BaseTL or dot3mauType10PassTS) in order to direct the management software to look in the EFM-CU-MIB module for the desired information. For example, the information on the particular EFMCu flavor that an EFMCu port is running is available from efmCuOperSubType, defined in the EFM-CU-MIB module.

Since EFMCu PMEs are not Ethernet-like interfaces, they cannot be instantiated as MAU interface objects.

11.3 MIB structure

11.3.1 EFM copper MIB overview

The main management objects defined in the EFM-CU-MIB module are split into two groups:

- efmCuPort—containing objects for configuration, capabilities, status, and notifications, common to all EFMCu PHYs.
- efmCuPme—containing objects for configuration, capabilities, status, and notifications of EFMCu PMEs.

The efmCuPme group in turn contains efmCuPme2B and efmCuPme10P groups, which define PME profiles specific to 2BASE-TL and 10PASS-TS PMEs, respectively, as well as PME-specific status information.

11.3.2 Interface stack capability MIB overview

The IF-CAP-STACK-MIB module contains two tables:

- ifCapStackTable—containing objects that define possible relationships among the sublayers of an interface with flexible cross-connect (cross-connect capability).
- ifInvCapStackTable—an inverse of the ifCapstackTable.

11.3.3 PME profiles

Since a managed node can have a large number of EFMCu PHYs, provisioning every parameter on every EFMCu PHY may become burdensome. Moreover, most PMEs are provisioned identically with the same set of parameters. To simplify the provisioning process, the EFM-CU-MIB module makes use of configuration profiles, similar to the HDSL2-SHDSL-LINE-MIB and VDSL-LINE-EXT-MCM-MIB modules. A profile is a set of parameters, used either for configuration or representation of a PME. The same profile can be shared by multiple PME ports using the same configuration.

The PME profiles are defined in the efmCuPme2BProfileTable and efmCuPme10PProfileTable for 2BASE-TL and 10PASS-TS PMEs, respectively. There are 12 predefined standard profiles for 2BASE-TL and 22 standard profiles for 10PASS-TS, defined in IEEE Std 802.3 and dedicated for rapid provisioning of EFMCu PHYs in most scenarios. In addition, the EFM-CU-MIB defines two additional predefined profiles for "best-effort" provisioning of 2BASE-TL PMEs. An ability to define new configuration profiles is also provided to allow for EFMCu deployment tailored to specific copper environments and spectral regulations.

A specific configuration or administrative profile is assigned to a specific PME via the efmCuPmeAdminProfile object. If efmCuPmeAdminProfile is zero, then the efmCuAdminProfile object of the PCS port connected to the PME determines the configuration profile (or a list of possible profiles) for that PME. This mechanism allows specifying a common profile for all PMEs connected to the PCS port, with an ability to change individual PME profiles by setting efmCuPmeAdminProfile object, which overwrites the profile set by efmCuAdminProfile.

A current operating PME profile is pointed to by the efmCuPmeOperProfile object. Note that this profile entry can be created automatically to reflect achieved parameters in adaptive (not fixed) initialization.

11.3.4 Mapping of IEEE 802.3 managed objects

This subclause contains the mapping between relevant managed objects (attributes) defined in IEEE Std 802.3 Clause 30, and managed objects defined in this clause and in associated MIB modules, i.e., the IF-MIB defined in IETF RFC 2863. Note that the majority of the objects defined in the EFM-CU-MIB module do not have direct counterparts in Clause 30 and instead refer to Clause 45 registers.

Table 11–2—Mapping of IEEE 802.3 managed objects

IEEE 802.3 managed object		Corresponding SNMP object
oMAU - Basic Package (Mandatory)	aMAUType	ifMauType (MAU-MIB)
	aMAUTypeList	ifMauTypeListBits (MAU-MIB)
	aMediaAvailable	ifMediaAvailable (MAU-MIB)
oPAF - Basic Package (Mandatory)	aPAFID	ifIndex (IF-MIB)
	aPhyEnd	efmCuPhySide
	aPHYCurrentStatus	efmCuStatus
	aPAFSupported	efmCuPAFSupported

Table 11–2—Mapping of IEEE 802.3 managed objects *(continued)*

IEEE 802.3 managed object		Corresponding SNMP object
oPAF - PME Aggregation Package (Optional)	aPAFAdminState	efmCuPAFAdminState
	aLocalPAFCapacity	efmCuPAFCapacity
	aLocalPMEAvailable	ifCapStackTable
	aLocalPMEAggregate	ifStackTable (IF-MIB)
	aRemotePAFSupported	efmCuRemotePAFSupported
	aRemotePAFCapacity	efmCuRemotePAFCapacity
	aRemotePMEAggregate	
oPME - 10P/2B Package (Mandatory)	aPMEID	ifIndex (IF-MIB)
	aPMEAdminState aPMEStatus	ifAdminState (IF-MIB) efmCuPmeStatus
	aPMESNRMgn	efmCuPmeSnrMgn
	aTCCodingViolations	efmCuPmeTCCodingErrors
	aTCCRCErrors	efmCuPmeTCCrcErrors
	aProfileSelect	efmCuAdminProfile, efmCuPmeAdminProfile
	aOperatingProfile	efmCuPmeOperProfile
	aPMEFECCorrectedBlocks	efmCuPme10PFECCorrectedBlocks
	aPMEFECUncorrectableBlocks	efmCuPme10PFECUncorrectedBlocks

11.4 Security considerations for Ethernet in the First Mile copper interfaces MIB module

There are a number of managed objects defined in the EFM-CU-MIB module that have a MAX-ACCESS clause of read-write or read-create. Most objects are writeable only when the link is Down. Writing to these objects can have potentially disruptive effects on network operation, for example:

— Changing of efmCuPmeAdminSubType may lead to a potential locking of the link, as peer PMEs of the same subtype cannot exchange handshake messages.
— Changing of efmCuPAFAdminState to enabled may lead to a potential locking of the link, if the peer PHY does not support PAF.
— Changing of efmCuPAFDiscoveryCode, before the discovery operation, may lead to a wrongful discovery, for example, when two -O ports are connected to the same multi-PME -R port and both -O ports have the same Discovery register value.
— Changing PCS or PME configuration parameters (e.g., profile of a PCS or PME via efmCuAdminProfile or efmCuPmeAdminProfile) may lead to anything from link quality and rate degradation to a complete link initialization failure, as ability of an EFMCu port to support a particular configuration depends on the copper environment.
— Activation of a PME can cause a severe degradation of service for another EFMCu PHY, whose PME(s) may be affected by the cross-talk from the newly activated PME.
— Removal of a PME from an operationally ‘up’ EFMCu port, aggregating several PMEs, may cause port’s rate degradation.

The user of the EFM-CU-MIB module should 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 the EFM-CU-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 particular, since EFMCu can be carried over Unshielded Twisted Pair (UTP) voice-grade copper in a bundle with other pairs belonging to another operator/customer, it is theoretically possible to eavesdrop to an EFMCu transmission simply by "listening" to a cross-talk from the EFMCu pairs, especially if the parameters of the EFMCu link in question are known.

In such environments, it is important to control also GET and NOTIFY access to these objects and possibly to encrypt their values when sending them over the network via SNMP.

11.5 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL:[18]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C11mib.txt

[18]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
IEEE8023-IF-CAP-STACK-MIB DEFINITIONS ::= BEGIN

  IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, org
      FROM SNMPv2-SMI         -- [RFC2578]
    TruthValue
      FROM SNMPv2-TC          -- [RFC2579]
    MODULE-COMPLIANCE, OBJECT-GROUP
      FROM SNMPv2-CONF        -- [RFC2580]
    ifStackGroup2, ifStackHigherLayer, ifStackLowerLayer
      FROM IF-MIB             -- [RFC2863]
    ifInvStackGroup
      FROM IF-INVERTED-STACK-MIB -- [RFC2864]
    ;

  ieee8023ifCapStackMIB MODULE-IDENTITY
     LAST-UPDATED "201102020000Z" -- February 2, 2011
     ORGANIZATION
       "IEEE 802.3 working group"
     CONTACT-INFO
         "WG-URL: http://www.ieee802.org/3/index.html
         WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

         Contact: Howard Frazier
         Postal:  3151 Zanker Road
                  San Jose, CA 95134
                  USA
         Tel:     +1.408.922.8164
         E-mail:  hfrazier@broadcom.com"

    DESCRIPTION
      "The objects in this MIB module are used to describe
      cross-connect capabilities of stacked (layered) interfaces,
      complementing ifStackTable and ifInvStackTable defined in
      IF-MIB and IF-INVERTED-STACK-MIB, respectively."

    REVISION    "201102020000Z" -- February 2, 2011
    DESCRIPTION
         "Initial version, based on an earlier version published as RFC 5066."

        ::= { org ieee(111) standards-association-numbers-series-standards(2)
              lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1)
              ieee8023efmcu(11) 1}


   -- Sections of the module
   -- Structured as recommended by [RFC4181], see
   -- Appendix D: Suggested OID Layout

   ifCapStackObjects     OBJECT IDENTIFIER ::= { ieee8023ifCapStackMIB 1 }

   ifCapStackConformance OBJECT IDENTIFIER ::= { ieee8023ifCapStackMIB 2 }

   -- Groups in the module

   --
   -- ifCapStackTable group
   --
```

```
ifCapStackTable  OBJECT-TYPE
  SYNTAX      SEQUENCE OF IfCapStackEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table, modeled after ifStackTable from IF-MIB,
    contains information on the possible 'on-top-of'
    relationships between the multiple sub-layers of network
    interfaces (as opposed to actual relationships described in
    ifStackTable). In particular, it contains information on
    which sub-layers may possibly run 'on top of' which other
    sub-layers, as determined by cross-connect capability of the
    device, where each sub-layer corresponds to a conceptual row
    in the ifTable. For example, when the sub-layer with ifIndex
    value x can be connected to run on top of the sub-layer with
    ifIndex value y, then this table contains:

      ifCapStackStatus.x.y=true

    The ifCapStackStatus.x.y row does not exist if it is
    impossible to connect between the sub-layers x and y.

    Note that for most stacked interfaces (e.g., 2BASE-TL)
    there's always at least one higher-level interface (e.g., PCS
    port) for each lower-level interface (e.g., PME) and at
    least one lower-level interface for each higher-level
    interface, that is, there is at least a single row with a
    'true' status for any such existing value of x or y.

    This table is read-only as it describes device capabilities."
  REFERENCE
    "IF-MIB, ifStackTable"
  ::= { ifCapStackObjects 1 }

ifCapStackEntry  OBJECT-TYPE
  SYNTAX      IfCapStackEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Information on a particular relationship between two
    sub-layers, specifying that one sub-layer may possibly run
    on 'top' of the other sub-layer. Each sub-layer corresponds
    to a conceptual row in the ifTable (interface index for
    lower and higher layer, respectively)."
  INDEX {
    ifStackHigherLayer,
    ifStackLowerLayer
  }

  ::= { ifCapStackTable 1 }

IfCapStackEntry ::= SEQUENCE {
     ifCapStackStatus       TruthValue
   }

ifCapStackStatus  OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-only
  STATUS      current
```



```
  DESCRIPTION
    "The status of the 'cross-connect capability' relationship
    between two sub-layers. The following values can be returned:
      true(1)         - indicates that the sub-layer interface,
                        identified by the ifStackLowerLayer may
                        be connected to run 'below' the sub-layer
                        interface, identified by the
                        ifStackHigherLayer index.
      false(2)        - the sub-layer interfaces cannot be
                        connected temporarily due to
                        unavailability of the interface(s), e.g.,
                        one of the interfaces is located on an
                        absent pluggable module.

    Note that lower-layer interface availability per higher-layer,
    indicated by the value of 'true', can be constrained by
    other parameters, for example, by the aggregation capacity of
    a higher-layer interface or by the lower-layer interface in
    question being already connected to another higher-layer
    interface. In order to ensure that a particular sub-layer can
    be connected to another sub-layer, all respective objects
    (e.g., ifCapStackTable, ifStackTable, and efmCuPAFCapacity for
    EFMCu interfaces) shall be inspected.

    This object is read-only, unlike ifStackStatus, as it
    describes a cross-connect capability."
  ::= { ifCapStackEntry 1 }

ifInvCapStackTable  OBJECT-TYPE
  SYNTAX        SEQUENCE OF IfInvCapStackEntry
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
    "A table containing information on the possible relationships
    between the multiple sub-layers of network interfaces. This
    table, modeled after ifInvStackTable from
    IF-INVERTED-STACK-MIB, is an inverse of the ifCapStackTable
    defined in this MIB module.

    In particular, this table contains information on which
    sub-layers may run 'underneath' which other sub-layers, where
    each sub-layer corresponds to a conceptual row in the ifTable.
    For example, when the sub-layer with ifIndex value x may be
    connected to run underneath the sub-layer with ifIndex value
    y, then this table contains:

       ifInvCapStackStatus.x.y=true

    This table contains exactly the same number of rows as the
    ifCapStackTable, but the rows appear in a different order.

    This table is read-only as it describes a cross-connect
    capability."
  REFERENCE
     "IF-INVERTED-STACK-MIB, ifInvStackTable"
  ::= { ifCapStackObjects 2 }

ifInvCapStackEntry  OBJECT-TYPE
  SYNTAX        IfInvCapStackEntry
```

```
  MAX-ACCESS    not-accessible
  STATUS        current
  DESCRIPTION
     "Information on a particular relationship between two sub-
     layers, specifying that one sub-layer may run underneath the
     other sub-layer. Each sub-layer corresponds to a conceptual
     row in the ifTable."
  INDEX { ifStackLowerLayer, ifStackHigherLayer }
  ::= { ifInvCapStackTable 1 }

 IfInvCapStackEntry ::= SEQUENCE {
   ifInvCapStackStatus       TruthValue
 }

ifInvCapStackStatus  OBJECT-TYPE
  SYNTAX         TruthValue
  MAX-ACCESS     read-only
  STATUS         current
  DESCRIPTION
     "The status of the possible 'cross-connect capability'
     relationship between two sub-layers.

     An instance of this object exists for each instance of the
     ifCapStackStatus object, and vice versa. For example, if the
     variable ifCapStackStatus.H.L exists, then the variable
     ifInvCapStackStatus.L.H also exists, and vice versa. In
     addition, the two variables always have the same value.

     The ifInvCapStackStatus object is read-only, as it describes
     a cross-connect capability."
  REFERENCE
     "ifCapStackStatus"
  ::= { ifInvCapStackEntry 1 }

--
-- Conformance Statements
--

 ifCapStackGroups      OBJECT IDENTIFIER ::=
      { ifCapStackConformance 1 }

 ifCapStackCompliances OBJECT IDENTIFIER ::=
      { ifCapStackConformance 2 }

 -- Conformance statements

 ifCapStackGroup OBJECT-GROUP
   OBJECTS {
     ifCapStackStatus,
     ifInvCapStackStatus
   }
   STATUS  current
   DESCRIPTION
     "A collection of objects providing information on the
     cross-connect capability of multi-layer (stacked) network
     interfaces."
   ::= { ifCapStackGroups 1 }
```

```
  -- Compliance statements

   ifCapStackCompliance MODULE-COMPLIANCE
     STATUS      current
     DESCRIPTION
       "The compliance statement for SNMP entities, which provide
       information on the cross-connect capability of multi-layer
       (stacked) network interfaces, with flexible cross-connect
       between the sub-layers."


     MODULE  -- this module
       MANDATORY-GROUPS {
         ifCapStackGroup
       }

       OBJECT       ifCapStackStatus

       SYNTAX       TruthValue { true(1) }
       DESCRIPTION
         "Support for the false(2) value is optional for
         implementations supporting pluggable interfaces."

       OBJECT       ifInvCapStackStatus
       SYNTAX       TruthValue { true(1) }
       DESCRIPTION
         "Support for the false(2) value is optional for
         implementations supporting pluggable interfaces."

     MODULE  IF-MIB
       MANDATORY-GROUPS {
         ifStackGroup2
       }

     MODULE  IF-INVERTED-STACK-MIB
       MANDATORY-GROUPS {
         ifInvStackGroup
       }

     ::= { ifCapStackCompliances 1 }
END


IEEE8023-EFM-CU-MIB DEFINITIONS ::= BEGIN

  IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, Integer32,
    Unsigned32, Counter32, org
      FROM SNMPv2-SMI         -- [RFC2578]
    TEXTUAL-CONVENTION, TruthValue, RowStatus, PhysAddress
      FROM SNMPv2-TC          -- [RFC2579]
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
      FROM SNMPv2-CONF        -- [RFC2580]
    SnmpAdminString
      FROM SNMP-FRAMEWORK-MIB -- [RFC3411]
    ifIndex, ifSpeed
      FROM IF-MIB             -- [RFC2863]
    ;
```



```
ieee8023efmCuMIB MODULE-IDENTITY
   LAST-UPDATED "201102020000Z" -- February 2, 2011
   ORGANIZATION
     "IEEE 802.3 working group"
   CONTACT-INFO
       "WG-URL: http://www.ieee802.org/3/index.html
       WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

       Contact: Howard Frazier
       Postal:  3151 Zanker Road
                San Jose, CA 95134
                USA
       Tel:     +1.408.922.8164
       E-mail:  hfrazier@broadcom.com"

  DESCRIPTION
    "The objects in this MIB module are used to manage
    the Ethernet in the First Mile (EFM) Copper (EFMCu) Interfaces
    2BASE-TL and 10PASS-TS, defined in IEEE Std 802.3.

    Of particular interest are Clause 61, 'Physical Coding
    Sublayer (PCS) and common specifications, type 10PASS-TS and
    type 2BASE-TL', Clause 30, 'Management', Clause 45,
    'Management Data Input/Output (MDIO) Interface', Annex 62A,
    'PMD profiles for 10PASS-TS' and Annex 63A, 'PMD profiles for
    2BASE-TL'."

  REVISION    "201102020000Z" -- February 2, 2011
  DESCRIPTION
       "Initial version, based on an earlier version published as RFC 5066."

      ::= { org ieee(111) standards-association-numbers-series-standards(2)
           lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1)
           ieee8023efmcu(11) 2 }


 -- Sections of the module

 efmCuObjects     OBJECT IDENTIFIER ::= { ieee8023efmCuMIB 1 }

 efmCuConformance OBJECT IDENTIFIER ::= { ieee8023efmCuMIB 2 }

 -- Groups in the module

 efmCuPort        OBJECT IDENTIFIER ::= { efmCuObjects 1 }

 efmCuPme         OBJECT IDENTIFIER ::= { efmCuObjects 2 }

 -- Textual Conventions

 EfmProfileIndex ::= TEXTUAL-CONVENTION
   DISPLAY-HINT "d"
   STATUS       current
   DESCRIPTION
     "A unique value, greater than zero, for each PME configuration
      profile in the managed EFMCu port. Values should be assigned
      contiguously starting from 1. The value for each profile shall
      remain constant at least from one re-initialization of the
      entity's network management system to the next re-initialization."
```

```
    SYNTAX       Unsigned32 (1..255)

  EfmProfileIndexOrZero ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
      "This textual convention is an extension of the
      EfmProfileIndex convention. The latter defines a greater than
      zero value used to identify a PME profile in the managed EFMCu
      port. This extension permits the additional value of zero.
      The value of zero is object-specific and shall therefore be
      defined as part of the description of any object that uses
      this syntax.
      Examples of the usage of zero value might include situations
      where the current operational profile is unknown."
    SYNTAX       Unsigned32 (0..255)

  EfmProfileIndexList ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "1d:"

    STATUS       current
    DESCRIPTION
      "This textual convention represents a list of up to 6
      EfmProfileIndex values, any of which can be chosen for
      configuration of a PME in a managed EFMCu port.
      The EfmProfileIndex textual convention defines a greater than
      zero value used to identify a PME profile.
      The value of this object is a concatenation of zero or
      more (up to 6) octets, where each octet contains an 8-bit
      EfmProfileIndex value.
      A zero-length octet string is object-specific and shall
      therefore be defined as part of the description of any object
      that uses this syntax. Examples of the usage of a zero-length
      value might include situations where an object using this
      textual convention is irrelevant for a specific EFMCu port
      type."
    SYNTAX       OCTET STRING (SIZE(0..6))

  EfmTruthValueOrUnknown ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
      "This textual convention is an extension of the TruthValue
      convention. The latter defines a boolean value with possible
      values of true(1) and false(2). This extension permits the
      additional value of unknown(0), which can be returned as the
      result of a GET operation when an exact true or false value
      of the object cannot be determined."
    SYNTAX       INTEGER { unknown(0), true(1), false(2) }

 -- Port Notifications Group

  efmCuPortNotifications OBJECT IDENTIFIER ::= { efmCuPort 0 }

  efmCuLowRateCrossing NOTIFICATION-TYPE
    OBJECTS {
      ifSpeed,
      efmCuThreshLowRate
    }
    STATUS      current
```



```
  DESCRIPTION
    "This notification indicates that the EFMCu port's data rate
    has reached/dropped below or exceeded the low rate threshold,
    specified by efmCuThreshLowRate.

    This notification may be sent for the -O subtype ports
    (2BaseTL-O/10PassTS-O) while the port is Up, on the crossing
    event in both directions: from normal (rate is above the
    threshold) to low (rate equals the threshold or below it) and
    from low to normal. This notification is not applicable to
    the -R subtypes.

    A small debouncing period of 2.5 sec, between the detection
    of the condition and the notification, should be implemented to
    prevent simultaneous LinkUp/LinkDown and efmCuLowRateCrossing
    notifications to be sent.

    The adaptive nature of the EFMCu technology allows the port to
    adapt itself to the changes in the copper environment, e.g.,
    an impulse noise, alien crosstalk, or a micro-interruption may
    temporarily drop one or more PMEs in the aggregation group,
    causing a rate degradation of the aggregated EFMCu link.
    The dropped PMEs would then try to re-initialize, possibly at
    a lower rate than before, adjusting the rate to provide
    required target SNR margin.

    Generation of this notification is controlled by the
    efmCuLowRateCrossingEnable object."
  ::= { efmCuPortNotifications 1 }

-- PCS Port group

efmCuPortConfTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF EfmCuPortConfEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Table for Configuration of EFMCu 2BASE-TL/10PASS-TS (PCS)
    Ports. Entries in this table shall be maintained in a
    persistent manner."
  ::= { efmCuPort 1 }

efmCuPortConfEntry OBJECT-TYPE
  SYNTAX      EfmCuPortConfEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An entry in the EFMCu Port Configuration table.
    Each entry represents an EFMCu port indexed by the ifIndex.
    Note that an EFMCu PCS port runs on top of a single
    or multiple PME port(s), which are also indexed by ifIndex."
  INDEX  { ifIndex }
  ::= { efmCuPortConfTable 1 }

EfmCuPortConfEntry ::=
  SEQUENCE {
    efmCuPAFAdminState              INTEGER,
    efmCuPAFDiscoveryCode           PhysAddress,
    efmCuAdminProfile               EfmProfileIndexList,
```



```
      efmCuTargetDataRate              Unsigned32,
      efmCuTargetSnrMgn                Unsigned32,
      efmCuAdaptiveSpectra             TruthValue,
      efmCuThreshLowRate               Unsigned32,
      efmCuLowRateCrossingEnable       TruthValue
    }

  efmCuPAFAdminState  OBJECT-TYPE
    SYNTAX      INTEGER {
      enabled(1),
      disabled(2)
    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Administrative (desired) state of the PAF of the EFMCu port
      (PCS).
      When 'disabled', PME aggregation will not be performed by the
      PCS. No more than a single PME can be assigned to this PCS in
      this case.
      When 'enabled', PAF will be performed by the PCS when the link
      is Up, even on a single attached PME, if PAF is supported.

      PCS ports incapable of supporting PAF shall return a value of
      'disabled'. Attempts to 'enable' such ports shall be
      rejected.

      A PAF 'enabled' port with multiple PMEs assigned cannot be
      'disabled'. Attempts to 'disable' such port shall be
      rejected, until at most one PME is left assigned.

      Changing PAFAdminState is a traffic-disruptive operation and
      as such shall be done when the link is Down. Attempts to
      change this object shall be rejected if the link is Up or
      Initializing.

      This object maps to the Clause 30 attribute aPAFAdminState.

      If a Clause 45 MDIO Interface to the PCS is present, then this
      object maps to the PAF enable bit in the 10P/2B PCS control
      register.

      This object shall be maintained in a persistent manner."
    REFERENCE
      "IEEE Std 802.3, 61.2.2, 45.2.3.18.3"
    ::= { efmCuPortConfEntry 1 }
  efmCuPAFDiscoveryCode  OBJECT-TYPE
    SYNTAX      PhysAddress (SIZE(0|6))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "PAF Discovery Code of the EFMCu port (PCS).
      A unique 6-octet code used by the Discovery function,
      when PAF is supported.
      PCS ports incapable of supporting PAF shall return a
      zero-length octet string on an attempt to read this object.
      An attempt to write to this object shall be rejected for such
      ports.
      This object shall be instantiated for the -O subtype PCS before
```



```
    writing operations on the efmCuPAFRemoteDiscoveryCode
    (Set_if_Clear and Clear_if_Same) are performed by PMEs
    associated with the PCS.
    The initial value of this object for -R subtype ports after
    reset is all zeroes. For -R subtype ports, the value of this
    object cannot be changed directly. This value may be changed
    as a result of writing operation on the
    efmCuPAFRemoteDiscoveryCode object of remote PME of -O
    subtype, connected to one of the local PMEs associated with
    the PCS.

    Discovery shall be performed when the link is Down.
    Attempts to change this object shall be rejected (in case of
    SNMP with the error inconsistentValue), if the link is Up or
    Initializing.

    The PAF Discovery Code maps to the local Discovery code
    variable in PAF (note that it does not have a corresponding
    Clause 45 register)."
  REFERENCE
    "IEEE Std 802.3, 61.2.2.8.3, 61.2.2.8.4, 45.2.6.6.1, 45.2.6.8,
    61A.2"
  ::= { efmCuPortConfEntry 2 }

efmCuAdminProfile  OBJECT-TYPE
  SYNTAX      EfmProfileIndexList
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Desired configuration profile(s), common for all PMEs in the
    EFMCu port. This object is a list of pointers to entries in
    either efmCuPme2BProfileTable or
    efmCuPme10PProfileTable, depending on the current
    operating SubType of the EFMCu port as indicated by
    efmCuPortSide.
    The value of this object is a list of up to 6 indices of
    profiles. If this list consists of a single profile index,
    then all PMEs assigned to this EFMCu port shall be configured
    according to the profile referenced by that index, unless it
    is overwritten by a corresponding non-zero
    efmCuPmeAdminProfile instance, which takes precedence over
    efmCuAdminProfile.
    A list consisting of more than one index allows each PME
    in the port to be configured according to any profile
    specified in the list.
    By default, this object has a value of 0x01, referencing the
    1st entry in efmCuPme2BProfileTable or
    efmCuPme10PProfileTable.

    This object is writable and readable for the -O subtype
    (2BaseTL-O or 10PassTS-O) EFMCu ports. It is irrelevant for
    the -R  subtype (2BaseTL-R or 10PassTS-R) ports -- a
    zero-length octet string shall be returned on an attempt to
    read this object and an attempt to change this object shall be
    rejected in this case.

    Note that the current operational profile value is available
    via the efmCuPmeOperProfile object.
```

```
      Any modification of this object shall be performed when the
      link is Down. Attempts to change this object shall be
      rejected, if the link is Up or Initializing.
      Attempts to set this object to a list with a member value that
      is not the value of the index for an active entry in the
      corresponding profile table shall be rejected.

      This object maps to the Clause 30 attribute aProfileSelect.

      This object shall be maintained in a persistent manner."
    REFERENCE
      "IEEE Std 802.3, 30.11.2.1.6"
    DEFVAL { '01'H }
    ::= { efmCuPortConfEntry 3 }

  efmCuTargetDataRate  OBJECT-TYPE
    SYNTAX      Unsigned32(1..100000|999999)
    UNITS       "Kbps"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Desired EFMCu port 'net' (as seen across MII) Data Rate in
      kb/s, to be achieved during initialization, under spectral
      restrictions placed on each PME via efmCuAdminProfile or
      efmCuPmeAdminProfile, with the desired SNR margin specified by
      efmCuTargetSnrMgn.
      In case of PAF, this object represents a sum of individual PME
      data rates, modified to compensate for fragmentation and
      64/65-octet encapsulation overhead (e.g., target data rate of
      10 Mb/s shall allow lossless transmission of a full-duplex
      10 Mb/s Ethernet frame stream with minimal inter-frame gap).

      The value is limited above by 100 Mb/s as this is the max
      burst rate across MII for EFMCu ports.

      The value between 1 and 100000 indicates that the total data
      rate (ifSpeed) of the EFMCu port after initialization shall be
      equal to the target data rate or less, if the target data rate
      cannot be achieved under spectral restrictions specified by
      efmCuAdminProfile/efmCuPmeAdminProfile and with the desired
      SNR margin. In case the copper environment allows a higher
      total data rate to be achieved than that specified by the
      target, the excess capability shall be either converted to
      additional SNR margin or reclaimed by minimizing transmit
      power as controlled by efmCuAdaptiveSpectra.

      The value of 999999 means that the target data rate is not
      fixed and shall be set to the maximum attainable rate during
      initialization (Best Effort), under specified spectral
      restrictions and with the desired SNR margin.

      This object is read-write for the -O subtype EFMCu ports
      (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.

      Changing of the Target Data Rate shall be performed when the
      link is Down. Attempts to change this object shall be rejected
      (in case of SNMP with the error inconsistentValue), if the
      link is Up or Initializing.
```

```
    Note that the current Data Rate of the EFMCu port is
    represented by the ifSpeed object of IF-MIB.

    This object shall be maintained in a persistent manner."
  ::= { efmCuPortConfEntry 4 }

efmCuTargetSnrMgn  OBJECT-TYPE
  SYNTAX      Unsigned32(0..21)
  UNITS       "dB"
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Desired EFMCu port SNR margin to be achieved on all PMEs
    assigned to the port, during initialization. (The SNR margin
    is the difference between the desired SNR and the actual SNR).

    Note that 802.3ah recommends using a default target SNR margin
    of 5 dB for 2BASE-TL ports and 6 dB for 10PASS-TS ports in
    order to achieve a mean bit error ratio (BER) of 10^-7 at the
    PMA service interface.

    This object is read-write for the -O subtype EFMCu ports
    (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.

    Changing of the target SNR margin shall be performed when the
    link is Down. Attempts to change this object shall be rejected
    (in case of SNMP with the error inconsistentValue), if the
    link is Up or Initializing.

    Note that the current SNR margin of the PMEs comprising the
    EFMCu port is represented by efmCuPmeSnrMgn.

    This object shall be maintained in a persistent manner."
  REFERENCE
    "IEEE Std 802.3, 61.1.2"
  ::= { efmCuPortConfEntry 5 }

efmCuAdaptiveSpectra  OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Indicates how to utilize excess capacity when the copper
    environment allows a higher total data rate to be achieved
    than that specified by the efmCuTargetDataRate.

    A value of true(1) indicates that the excess capability shall
    be reclaimed by minimizing transmit power, e.g., using higher
    constellations and Power Back-Off, in order to reduce
    interference to other copper pairs in the binder and the
    adverse impact to link/system performance.

    A value of false(2) indicates that the excess capability shall
    be converted to additional SNR margin and spread evenly across
    all active PMEs assigned to the (PCS) port, to increase link
    robustness.

    This object is read-write for the -O subtype EFMCu ports
    (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.
```



```
      Changing of this object shall be performed when the link is
      Down. Attempts to change this object shall be rejected (in
      case of SNMP with the error inconsistentValue), if the link
      is Up or Initializing.

      This object shall be maintained in a persistent manner."
    ::= { efmCuPortConfEntry 6 }

  efmCuThreshLowRate  OBJECT-TYPE
    SYNTAX      Unsigned32(1..100000)
    UNITS       "Kbps"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "This object configures the EFMCu port low-rate crossing alarm
      threshold. When the current value of ifSpeed for this port
      reaches/drops below or exceeds this threshold, an
      efmCuLowRateCrossing notification may be generated if enabled
      by efmCuLowRateCrossingEnable.

      This object is read-write for the -O subtype EFMCu ports
      (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.

      This object shall be maintained in a persistent manner."
    ::= { efmCuPortConfEntry 7 }

  efmCuLowRateCrossingEnable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Indicates whether efmCuLowRateCrossing notifications should
      be generated for this interface.

      A value of true(1) indicates that efmCuLowRateCrossing
      notification is enabled. A value of false(2) indicates that
      the notification is disabled.

      This object is read-write for the -O subtype EFMCu ports
      (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.

      This object shall be maintained in a persistent manner."
    ::= { efmCuPortConfEntry 8 }


  efmCuPortCapabilityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EfmCuPortCapabilityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "Table for Capabilities of EFMCu 2BASE-TL/10PASS-TS (PCS)
      Ports. Entries in this table shall be maintained in a
      persistent manner"
    ::= { efmCuPort 2 }

  efmCuPortCapabilityEntry OBJECT-TYPE
    SYNTAX      EfmCuPortCapabilityEntry
    MAX-ACCESS  not-accessible
```

```
  STATUS      current
  DESCRIPTION
    "An entry in the EFMCu Port Capability table.
    Each entry represents an EFMCu port indexed by the ifIndex.
    Note that an EFMCu PCS port runs on top of a single
    or multiple PME port(s), which are also indexed by ifIndex."
  INDEX  { ifIndex }
  ::= { efmCuPortCapabilityTable 1 }

EfmCuPortCapabilityEntry ::=
  SEQUENCE {
    efmCuPAFSupported               TruthValue,
    efmCuPeerPAFSupported           EfmTruthValueOrUnknown,
    efmCuPAFCapacity                Unsigned32,
    efmCuPeerPAFCapacity            Unsigned32
  }

efmCuPAFSupported  OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "PME Aggregation Function (PAF) capability of the EFMCu port
    (PCS).
    This object has a value of true(1) when the PCS can perform
    PME aggregation on the available PMEs.
    Ports incapable of PAF shall return a value of false(2).

    This object maps to the Clause 30 attribute aPAFSupported.

    If a Clause 45 MDIO Interface to the PCS is present,
    then this object maps to the PAF available bit in the
    10P/2B capability register."
  REFERENCE
    "IEEE Std 802.3, 61.2.2, 30.11.1.1.4, 45.2.3.17.1"
  ::= { efmCuPortCapabilityEntry 1 }

efmCuPeerPAFSupported  OBJECT-TYPE
  SYNTAX      EfmTruthValueOrUnknown
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "PME Aggregation Function (PAF) capability of the EFMCu port
    (PCS) link partner.
    This object has a value of true(1) when the remote PCS can
    perform PME aggregation on its available PMEs.
    Ports whose peers are incapable of PAF shall return a value
    of false(2).
    Ports whose peers cannot be reached because of the link
    state shall return a value of unknown(0).

    This object maps to the Clause 30 attribute
    aRemotePAFSupported.

    If a Clause 45 MDIO Interface to the PCS is present, then
    this object maps to the Remote PAF supported bit in the
    10P/2B capability register."
  REFERENCE
    "IEEE Std 802.3, 61.2.2, 30.11.1.1.9, 45.2.3.17.2"
```



```
  ::= { efmCuPortCapabilityEntry 2 }

efmCuPAFCapacity  OBJECT-TYPE
  SYNTAX      Unsigned32 (1..32)
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "Number of PMEs that can be aggregated by the local PAF.
    The number of PMEs currently assigned to a particular
    EFMCu port (efmCuNumPMEs) is never greater than
    efmCuPAFCapacity.

    This object maps to the Clause 30 attribute
    aLocalPAFCapacity."
  REFERENCE
    "IEEE Std 802.3, 61.2.2, 30.11.1.1.6"
  ::= { efmCuPortCapabilityEntry 3 }

efmCuPeerPAFCapacity  OBJECT-TYPE
  SYNTAX      Unsigned32 (0|1..32)
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "Number of PMEs that can be aggregated by the PAF of the peer
    PHY (PCS port).
    A value of 0 is returned when peer PAF capacity is unknown
    (peer cannot be reached).
    This object maps to the Clause 30 attribute
    aRemotePAFCapacity."
  REFERENCE
    "IEEE Std 802.3, 61.2.2, 30.11.1.1.10"
  ::= { efmCuPortCapabilityEntry 4 }

efmCuPortStatusTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF EfmCuPortStatusEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table provides overall status information of EFMCu
    2BASE-TL/10PASS-TS ports, complementing the generic status
    information from the ifTable of IF-MIB and ifMauTable of the
    MAU-MIB module. Additional status information about connected PMEs
    is available from the efmCuPmeStatusTable.

    This table contains live data from the equipment. As such,
    it is not persistent."
  ::= { efmCuPort 3 }

efmCuPortStatusEntry OBJECT-TYPE
  SYNTAX      EfmCuPortStatusEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An entry in the EFMCu Port Status table.
    Each entry represents an EFMCu port indexed by the ifIndex.
    Note that an EFMCu PCS port runs on top of a single
    or multiple PME port(s), which are also indexed by ifIndex."
  INDEX  { ifIndex }
  ::= { efmCuPortStatusTable 1 }
```



```
EfmCuPortStatusEntry ::=
  SEQUENCE {
    efmCuFltStatus                   BITS,
    efmCuPortSide                    INTEGER,
    efmCuNumPMEs                     Unsigned32,
    efmCuPAFInErrors                 Counter32,
    efmCuPAFInSmallFragments         Counter32,
    efmCuPAFInLargeFragments         Counter32,
    efmCuPAFInBadFragments           Counter32,
    efmCuPAFInLostFragments          Counter32,
    efmCuPAFInLostStarts             Counter32,
    efmCuPAFInLostEnds               Counter32,
    efmCuPAFInOverflows              Counter32
  }

efmCuFltStatus  OBJECT-TYPE
  SYNTAX      BITS {
    noPeer(0),
    peerPowerLoss(1),
    pmeSubTypeMismatch(2),
    lowRate(3)
  }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "EFMCu (PCS) port Fault Status. This is a bitmap of possible
    conditions. The various bit positions are:
      noPeer              - the peer PHY cannot be reached (e.g.,
                            no PMEs attached, all PMEs are Down,
                            etc.). More info is available in
                            efmCuPmeFltStatus.
      peerPowerLoss       - the peer PHY has indicated impending
                            unit failure due to loss of local
                            power ('Dying Gasp').
      pmeSubTypeMismatch  - local PMEs in the aggregation group
                            are not of the same subtype, e.g.,
                            some PMEs in the local device are -O
                            while others are -R subtype.
      lowRate             - ifSpeed of the port reached or dropped
                            below efmCuThreshLowRate.

    This object is intended to supplement the ifOperStatus object
    in IF-MIB and ifMauMediaAvailable in the MAU-MIB module.

    Additional information is available via the efmCuPmeFltStatus
    object for each PME in the aggregation group (single PME if
    PAF is disabled)."
  REFERENCE
    "IF-MIB, ifOperStatus; MAU-MIB, ifMauMediaAvailable;
     efmCuPmeFltStatus"
  ::= { efmCuPortStatusEntry 1 }

efmCuPortSide  OBJECT-TYPE
  SYNTAX      INTEGER {
    subscriber(1),
    office(2),
    unknown(3)
  }
```



```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "EFM port mode of operation (subtype).
      The value of 'subscriber' indicates that the port is
      designated as '-R' subtype (all PMEs assigned to this port are
      of subtype '-R').
      The value of the 'office' indicates that the port is
      designated as '-O' subtype (all PMEs assigned to this port are
      of subtype '-O').
      The value of 'unknown' indicates that the port has no assigned
      PMEs yet or that the assigned PMEs are not of the same side
      (subTypePMEMismatch).

      This object partially maps to the Clause 30 attribute
      aPhyEnd."
    REFERENCE
       "IEEE Std 802.3, 61.1, 30.11.1.1.2"
    ::= { efmCuPortStatusEntry 2 }

  efmCuNumPMEs  OBJECT-TYPE
    SYNTAX      Unsigned32 (0..32)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The number of PMEs that is currently aggregated by the local
      PAF (assigned to the EFMCu port using the ifStackTable).
      This number is never greater than efmCuPAFCapacity.

      This object shall be automatically incremented or decremented
      when a PME is added or deleted to/from the EFMCu port using
      the ifStackTable."
    REFERENCE
      "IEEE Std 802.3, 61.2.2, 30.11.1.1.6"
    ::= { efmCuPortStatusEntry 3 }

  efmCuPAFInErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The number of fragments that have been received across the
      gamma interface with RxErr asserted and discarded.
      This read-only counter is inactive (not incremented) when the
      PAF is unsupported or disabled. Upon disabling the PAF, the
      counter retains its previous value.

      If a Clause 45 MDIO Interface to the PCS is present, then
      this object maps to the 10P/2B PAF RX error register.

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 45.2.3.21"
    ::= { efmCuPortStatusEntry 4 }

  efmCuPAFInSmallFragments OBJECT-TYPE
```

```
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The number of fragments smaller than minFragmentSize
    (64 bytes) that have been received across the gamma interface
    and discarded.
    This read-only counter is inactive when the PAF is
    unsupported or disabled. Upon disabling the PAF, the counter
    retains its previous value.

    If a Clause 45 MDIO Interface to the PCS is present, then
    this object maps to the 10P/2B PAF small fragments register.

    Discontinuities in the value of this counter can occur at
    re-initialization of the management system, and at other times
    as indicated by the value of ifCounterDiscontinuityTime,
    defined in IF-MIB."
  REFERENCE
    "IEEE Std 802.3, 45.2.3.22"
  ::= { efmCuPortStatusEntry 5 }

efmCuPAFInLargeFragments OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The number of fragments larger than maxFragmentSize
    (512 bytes) that have been received across the gamma interface
    and discarded.
    This read-only counter is inactive when the PAF is
    unsupported or disabled. Upon disabling the PAF, the counter
    retains its previous value.

    If a Clause 45 MDIO Interface to the PCS is present, then
    this object maps to the 10P/2B PAF large fragments register.

    Discontinuities in the value of this counter can occur at
    re-initialization of the management system, and at other times
    as indicated by the value of ifCounterDiscontinuityTime,
    defined in IF-MIB."
  REFERENCE
    "IEEE Std 802.3, 45.2.3.23"
  ::= { efmCuPortStatusEntry 6 }

efmCuPAFInBadFragments OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The number of fragments that do not fit into the sequence
    expected by the frame assembly function and that have been
    received across the gamma interface and discarded (the
    frame buffer is flushed to the next valid frame start).
    This read-only counter is inactive when the PAF is
    unsupported or disabled. Upon disabling the PAF, the counter
    retains its previous value.

    If a Clause 45 MDIO Interface to the PCS is present, then
```

```
      this object maps to the 10P/2B PAF bad fragments register.

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 45.2.3.25"
    ::= { efmCuPortStatusEntry 7 }

  efmCuPAFInLostFragments OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The number of gaps in the sequence of fragments that have
      been received across the gamma interface (the frame buffer is
      flushed to the next valid frame start, when fragment/fragments
      expected by the frame assembly function is/are not received).
      This read-only counter is inactive when the PAF is
      unsupported or disabled. Upon disabling the PAF, the counter
      retains its previous value.

      If a Clause 45 MDIO Interface to the PCS is present, then
      this object maps to the 10P/2B PAF lost fragment register.

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 45.2.3.26"
    ::= { efmCuPortStatusEntry 8 }

  efmCuPAFInLostStarts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The number of missing StartOfPacket indicators expected by
      the frame assembly function.
      This read-only counter is inactive when the PAF is
      unsupported or disabled. Upon disabling the PAF, the counter
      retains its previous value.

      If a Clause 45 MDIO Interface to the PCS is present, then
      this object maps to the 10P/2B PAF lost start of fragment
      register.

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 45.2.3.27"
    ::= { efmCuPortStatusEntry 9 }

  efmCuPAFInLostEnds OBJECT-TYPE
    SYNTAX      Counter32
```

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The number of missing EndOfPacket indicators expected by the
      frame assembly function.
      This read-only counter is inactive when the PAF is
      unsupported or disabled. Upon disabling the PAF, the counter
      retains its previous value.

      If a Clause 45 MDIO Interface to the PCS is present, then
      this object maps to the 10P/2B PAF lost start of fragment
      register.

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 45.2.3.28"
    ::= { efmCuPortStatusEntry 10 }
  efmCuPAFInOverflows OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The number of fragments, received across the gamma interface
      and discarded, which would have caused the frame assembly
      buffer to overflow.
      This read-only counter is inactive when the PAF is
      unsupported or disabled. Upon disabling the PAF, the counter
      retains its previous value.

      If a Clause 45 MDIO Interface to the PCS is present, then
      this object maps to the 10P/2B PAF overflow register.

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 45.2.3.24"
    ::= { efmCuPortStatusEntry 11 }

-- PME Notifications Group

  efmCuPmeNotifications OBJECT IDENTIFIER ::= { efmCuPme 0 }

  efmCuPmeLineAtnCrossing NOTIFICATION-TYPE
    OBJECTS {
      efmCuPmeLineAtn,
      efmCuPmeThreshLineAtn
    }
    STATUS      current
    DESCRIPTION
      "This notification indicates that the loop attenuation
      threshold (as per the efmCuPmeThreshLineAtn
      value) has been reached/exceeded for the 2BASE-TL/10PASS-TS
      PME. This notification may be sent on the crossing event in
      both directions: from normal to exceeded and from exceeded
```



```
        to normal.

        A small debouncing period of 2.5 sec, between the detection
        of the condition and the notification, should be implemented
        to prevent intermittent notifications from being sent.

        Generation of this notification is controlled by the
        efmCuPmeLineAtnCrossingEnable object."
      ::= { efmCuPmeNotifications 1 }

    efmCuPmeSnrMgnCrossing NOTIFICATION-TYPE
      OBJECTS {
        efmCuPmeSnrMgn,
        efmCuPmeThreshSnrMgn
      }
      STATUS      current
      DESCRIPTION
        "This notification indicates that the SNR margin threshold
        (as per the efmCuPmeThreshSnrMgn value) has been
        reached/exceeded for the 2BASE-TL/10PASS-TS PME.
        This notification may be sent on the crossing event in
        both directions: from normal to exceeded and from exceeded
        to normal.

        A small debouncing period of 2.5 sec, between the detection
        of the condition and the notification, should be implemented
        to prevent intermittent notifications from being sent.

        Generation of this notification is controlled by the
        efmCuPmeSnrMgnCrossingEnable object."
      ::= { efmCuPmeNotifications 2 }

    efmCuPmeDeviceFault NOTIFICATION-TYPE
      OBJECTS {
        efmCuPmeFltStatus
      }
      STATUS      current
      DESCRIPTION
        "This notification indicates that a fault in the PME has been
        detected by a vendor-specific diagnostic or a self-test.

        Generation of this notification is controlled by the
        efmCuPmeDeviceFaultEnable object."
      ::= { efmCuPmeNotifications 3 }

    efmCuPmeConfigInitFailure NOTIFICATION-TYPE
      OBJECTS {
        efmCuPmeFltStatus,
        efmCuAdminProfile,
        efmCuPmeAdminProfile
      }
      STATUS      current
      DESCRIPTION
        "This notification indicates that PME initialization has
        failed, due to inability of the PME link to achieve the
        requested configuration profile.

        Generation of this notification is controlled by the
        efmCuPmeConfigInitFailEnable object."
```

```
  ::= { efmCuPmeNotifications 4 }

efmCuPmeProtocolInitFailure NOTIFICATION-TYPE
  OBJECTS {
    efmCuPmeFltStatus,
    efmCuPmeOperSubType
  }
  STATUS     current
  DESCRIPTION
    "This notification indicates that the peer PME was using
    an incompatible protocol during initialization.

    Generation of this notification is controlled by the
    efmCuPmeProtocolInitFailEnable object."
  ::= { efmCuPmeNotifications 5 }

-- The PME group

efmCuPmeConfTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF EfmCuPmeConfEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Table for Configuration of common aspects for EFMCu
    2BASE-TL/10PASS-TS PME ports (modems). Configuration of
    aspects specific to 2BASE-TL or 10PASS-TS PME types is
    represented in efmCuPme2BConfTable and efmCuPme10PConfTable,
    respectively.

    Entries in this table shall be maintained in a persistent
    manner."
  ::= { efmCuPme 1 }

efmCuPmeConfEntry OBJECT-TYPE
  SYNTAX      EfmCuPmeConfEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An entry in the EFMCu PME Configuration table.
    Each entry represents common aspects of an EFMCu PME port
    indexed by the ifIndex. Note that an EFMCu PME port can be
    stacked below a single PCS port, also indexed by ifIndex,
    possibly together with other PME ports if PAF is enabled."
  INDEX  { ifIndex }
  ::= { efmCuPmeConfTable 1 }

EfmCuPmeConfEntry ::=
  SEQUENCE {
    efmCuPmeAdminSubType           INTEGER,
    efmCuPmeAdminProfile           EfmProfileIndexOrZero,
    efmCuPAFRemoteDiscoveryCode    PhysAddress,
    efmCuPmeThreshLineAtn          Integer32,
    efmCuPmeThreshSnrMgn           Integer32,
    efmCuPmeLineAtnCrossingEnable  TruthValue,
    efmCuPmeSnrMgnCrossingEnable   TruthValue,
    efmCuPmeDeviceFaultEnable      TruthValue,
    efmCuPmeConfigInitFailEnable   TruthValue,
    efmCuPmeProtocolInitFailEnable TruthValue
  }
```

```
efmCuPmeAdminSubType  OBJECT-TYPE
  SYNTAX      INTEGER {
    ieee2BaseTLO(1),
    ieee2BaseTLR(2),
    ieee10PassTSO(3),
    ieee10PassTSR(4),
    ieee2BaseTLor10PassTSR(5),
    ieee2BaseTLor10PassTSO(6),
    ieee10PassTSor2BaseTLO(7)
  }
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Administrative (desired) subtype of the PME.
    Possible values are:
      ieee2BaseTLO           - PME shall operate as 2BaseTL-O
      ieee2BaseTLR           - PME shall operate as 2BaseTL-R
      ieee10PassTSO          - PME shall operate as 10PassTS-O
      ieee10PassTSR          - PME shall operate as 10PassTS-R
      ieee2BaseTLor10PassTSR - PME shall operate as 2BaseTL-R or
                               10PassTS-R. The actual value will
                               be set by the -O link partner
                               during initialization (handshake).
      ieee2BaseTLor10PassTSO - PME shall operate as 2BaseTL-O
                               (preferred) or 10PassTS-O. The
                               actual value will be set during
                               initialization depending on the -R
                               link partner capability (i.e., if
                               -R is incapable of the preferred
                               2BaseTL mode, 10PassTS will be
                               used).
      ieee10PassTSor2BaseTLO - PME shall operate as 10PassTS-O
                               (preferred) or 2BaseTL-O. The
                               actual value will be set during
                               initialization depending on the -R
                               link partner capability (i.e., if
                               -R is incapable of the preferred
                               10PassTS mode, 2BaseTL will be
                               used).

    Changing efmCuPmeAdminSubType is a traffic-disruptive
    operation and as such shall be done when the link is Down.
    Attempts to change this object shall be rejected if the link
    is Up or Initializing.
    Attempts to change this object to an unsupported subtype
    (see efmCuPmeSubTypesSupported) shall be rejected.

    The current operational subtype is indicated by the
    efmCuPmeOperSubType variable.

    If a Clause 45 MDIO Interface to the PMA/PMD is present, then
    this object combines values of the Port subtype select bits
    and the PMA/PMD type selection bits in the 10P/2B PMA/PMD
    control register."
  REFERENCE
    "IEEE Std 802.3, 61.1, 45.2.1.11.4, 45.2.1.11.7"
  ::= { efmCuPmeConfEntry 1 }
```

```
efmCuPmeAdminProfile  OBJECT-TYPE
  SYNTAX      EfmProfileIndexOrZero
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Desired PME configuration profile. This object is a pointer
    to an entry in either the efmCuPme2BProfileTable or the
    efmCuPme10PProfileTable, depending on the current operating
    SubType of the PME. The value of this object is the index of
    the referenced profile.
    The value of zero (default) indicates that the PME is
    configured via the efmCuAdminProfile object for the PCS port
    to which this PME is assigned. That is, the profile
    referenced by efmCuPmeAdminProfile takes precedence
    over the profile(s) referenced by efmCuAdminProfile.

    This object is writable and readable for the CO subtype PMEs
    (2BaseTL-O or 10PassTS-O). It is irrelevant for the CPE
    subtype (2BaseTL-R or 10PassTS-R) -- a zero value shall be
    returned on an attempt to read this object and any attempt
    to change this object shall be rejected in this case.
    Note that the current operational profile value is available
    via efmCuPmeOperProfile object.

    Any modification of this object shall be performed when the
    link is Down. Attempts to change this object shall be
    rejected, if the link is Up or Initializing.

    Attempts to set this object to a value that is not the value
    of the index for an active entry in the corresponding profile
    table shall be rejected.

    This object maps to the Clause 30 attribute aProfileSelect.

    This object shall be maintained in a persistent manner."
  REFERENCE
    "IEEE Std 802.3, 30.11.2.1.6"
  DEFVAL { 0 }
  ::= { efmCuPmeConfEntry 2 }

efmCuPAFRemoteDiscoveryCode  OBJECT-TYPE
  SYNTAX      PhysAddress (SIZE(0|6))
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "PAF Remote Discovery Code of the PME port at the CO.
    The 6-octet Discovery Code of the peer PCS connected via
    the PME.
    Reading this object results in a Discovery Get operation.
    Setting this object to all zeroes results in a Discovery
    Clear_if_Same operation (the value of efmCuPAFDiscoveryCode
    at the peer PCS shall be the same as efmCuPAFDiscoveryCode of
    the local PCS associated with the PME for the operation to
    succeed).
    Writing a non-zero value to this object results in a
    Discovery Set_if_Clear operation.
    A zero-length octet string shall be returned on an attempt to
    read this object when PAF aggregation is not enabled.
```



```
      This object is irrelevant in CPE port (-R) subtypes: in this
      case, a zero-length octet string shall be returned on an
      attempt to read this object; writing to this object shall
      be rejected.

      Discovery shall be performed when the link is Down.
      Attempts to change this object shall be rejected (in case of
      SNMP with the error inconsistentValue), if the link is Up or
      Initializing.

      If a Clause 45 MDIO Interface to the PMA/PMD is present, then
      this object is a function of 10P/2B aggregation discovery
      control register, Discovery operation result bits in 10P/2B
      aggregation and discovery status register and
      10P/2B aggregation discovery code register."
    REFERENCE
      "IEEE Std 802.3, 61.2.2.8.4, 45.2.6.6-45.2.6.8"
    ::= { efmCuPmeConfEntry 3 }

  efmCuPmeThreshLineAtn  OBJECT-TYPE
    SYNTAX  Integer32(-127..128)
    UNITS       "dB"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Desired Line Attenuation threshold for the 2B/10P PME.
      This object configures the line attenuation alarm threshold.
      When the current value of Line Attenuation reaches or
      exceeds this threshold, an efmCuPmeLineAtnCrossing
      notification may be generated, if enabled by
      efmCuPmeLineAtnCrossingEnable.

      This object is writable for the CO subtype PMEs (-O).
      It is read-only for the CPE subtype (-R).

      Changing of the Line Attenuation threshold shall be performed
      when the link is Down. Attempts to change this object shall be
      rejected (in case of SNMP with the error inconsistentValue),
      if the link is Up or Initializing.

      If a Clause 45 MDIO Interface to the PME is present, then this
      object maps to the loop attenuation threshold bits in
      the 2B PMD line quality thresholds register."
    REFERENCE
      "IEEE Std 802.3, 45.2.1.36"
    ::= { efmCuPmeConfEntry 4 }

  efmCuPmeThreshSnrMgn  OBJECT-TYPE
    SYNTAX      Integer32(-127..128)
    UNITS       "dB"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Desired SNR margin threshold for the 2B/10P PME.
      This object configures the SNR margin alarm threshold.
      When the current value of SNR margin reaches or exceeds this
      threshold, an efmCuPmeSnrMgnCrossing notification may be
      generated, if enabled by efmCuPmeSnrMgnCrossingEnable.
      This object is writable for the CO subtype PMEs
```

```
      (2BaseTL-O/10PassTS-O). It is read-only for the CPE subtype
      (2BaseTL-R/10PassTS-R).

      Changing of the SNR margin threshold shall be performed when
      the link is Down. Attempts to change this object shall be
      rejected (in case of SNMP with the error inconsistentValue),
      if the link is Up or Initializing.

      If a Clause 45 MDIO Interface to the PME is present, then this
      object maps to the SNR margin threshold bits in the 2B PMD
      line quality thresholds register."
    REFERENCE
      "IEEE Std 802.3, 45.2.1.36"
    ::= { efmCuPmeConfEntry 5 }

  efmCuPmeLineAtnCrossingEnable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Indicates whether efmCuPmeLineAtnCrossing notifications
      should be generated for this interface.

      A value of true(1) indicates that efmCuPmeLineAtnCrossing
      notification is enabled. A value of false(2) indicates that
      the notification is disabled."
    ::= { efmCuPmeConfEntry 6 }

  efmCuPmeSnrMgnCrossingEnable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Indicates whether efmCuPmeSnrMgnCrossing notifications
      should be generated for this interface.

      A value of true(1) indicates that efmCuPmeSnrMgnCrossing
      notification is enabled. A value of false(2) indicates that
      the notification is disabled."
    ::= { efmCuPmeConfEntry 7 }

  efmCuPmeDeviceFaultEnable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
      "Indicates whether efmCuPmeDeviceFault notifications
      should be generated for this interface.

      A value of true(1) indicates that efmCuPmeDeviceFault
      notification is enabled. A value of false(2) indicates that
      the notification is disabled."
    ::= { efmCuPmeConfEntry 8 }

  efmCuPmeConfigInitFailEnable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
```

```
    "Indicates whether efmCuPmeConfigInitFailure notifications
    should be generated for this interface.

    A value of true(1) indicates that efmCuPmeConfigInitFailure
    notification is enabled. A value of false(2) indicates that
    the notification is disabled."
  ::= { efmCuPmeConfEntry 9 }

efmCuPmeProtocolInitFailEnable  OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Indicates whether efmCuPmeProtocolInitFailure notifications
    should be generated for this interface.

    A value of true(1) indicates that efmCuPmeProtocolInitFailure
    notification is enabled. A value of false(2) indicates that
    the notification is disabled."
  ::= { efmCuPmeConfEntry 10 }


efmCuPmeCapabilityTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF EfmCuPmeCapabilityEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Table for the configuration of common aspects for EFMCu
    2BASE-TL/10PASS-TS PME ports (modems). The configuration of
    aspects specific to 2BASE-TL or 10PASS-TS PME types is
    represented in the efmCuPme2BConfTable and the
    efmCuPme10PConfTable, respectively.

    Entries in this table shall be maintained in a persistent
    manner."
  ::= { efmCuPme 2 }
efmCuPmeCapabilityEntry OBJECT-TYPE
  SYNTAX      EfmCuPmeCapabilityEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An entry in the EFMCu PME Capability table.
    Each entry represents common aspects of an EFMCu PME port
    indexed by the ifIndex. Note that an EFMCu PME port can be
    stacked below a single PCS port, also indexed by ifIndex,
    possibly together with other PME ports if PAF is enabled."
  INDEX  { ifIndex }
  ::= { efmCuPmeCapabilityTable 1 }

EfmCuPmeCapabilityEntry ::=
  SEQUENCE {
    efmCuPmeSubTypesSupported     BITS
  }

efmCuPmeSubTypesSupported  OBJECT-TYPE
  SYNTAX      BITS {
    ieee2BaseTLO(0),
    ieee2BaseTLR(1),
    ieee10PassTSO(2),
```

```
    ieee10PassTSR(3)
  }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "PME supported subtypes. This is a bitmap of possible
    subtypes. The various bit positions are:
      ieee2BaseTLO    - PME is capable of operating as 2BaseTL-O
      ieee2BaseTLR    - PME is capable of operating as 2BaseTL-R
      ieee10PassTSO   - PME is capable of operating as 10PassTS-O
      ieee10PassTSR   - PME is capable of operating as 10PassTS-R

    The desired mode of operation is determined by
    efmCuPmeAdminSubType, while efmCuPmeOperSubType reflects the
    current operating mode.

    If a Clause 45 MDIO Interface to the PCS is present, then this
    object combines the 10PASS-TS capable and 2BASE-TL capable
    bits in the 10P/2B PMA/PMD speed ability register and the
    CO supported and CPE supported bits in the 10P/2B PMA/PMD
    status register."
  REFERENCE
    "IEEE Std 802.3, 61.1, 45.2.1.4.1, 45.2.1.4.2, 45.2.1.12.2,
    45.2.1.12.3"
  ::= { efmCuPmeCapabilityEntry 1 }
efmCuPmeStatusTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF EfmCuPmeStatusEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table provides common status information of EFMCu
    2BASE-TL/10PASS-TS PME ports. Status information specific
    to 10PASS-TS PME is represented in efmCuPme10PStatusTable.

    This table contains live data from the equipment. As such,
    it is not persistent."
  ::= { efmCuPme 3 }

efmCuPmeStatusEntry OBJECT-TYPE
  SYNTAX      EfmCuPmeStatusEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An entry in the EFMCu PME Status table.
    Each entry represents common aspects of an EFMCu PME port
    indexed by the ifIndex. Note that an EFMCu PME port can be
    stacked below a single PCS port, also indexed by ifIndex,
    possibly together with other PME ports if PAF is enabled."
  INDEX  { ifIndex }
  ::= { efmCuPmeStatusTable 1 }

EfmCuPmeStatusEntry ::=
  SEQUENCE {
    efmCuPmeOperStatus            INTEGER,
    efmCuPmeFltStatus             BITS,
    efmCuPmeOperSubType           INTEGER,
    efmCuPmeOperProfile           EfmProfileIndexOrZero,
    efmCuPmeSnrMgn                Integer32,
    efmCuPmePeerSnrMgn            Integer32,
```



```
    efmCuPmeLineAtn               Integer32,
    efmCuPmePeerLineAtn           Integer32,
    efmCuPmeEquivalentLength      Unsigned32,
    efmCuPmeTCCodingErrors        Counter32,
    efmCuPmeTCCrcErrors           Counter32
  }

efmCuPmeOperStatus  OBJECT-TYPE
  SYNTAX      INTEGER {
    up(1),
    downNotReady(2),
    downReady(3),
    init(4)
  }

  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "Current PME link Operational Status. Possible values are:
      up(1)           - The link is Up and ready to pass
                        64/65-octet encoded frames or fragments.
      downNotReady(2) - The link is Down and the PME does not
                        detect Handshake tones from its peer.
                        This value may indicate a possible
                        problem with the peer PME.
      downReady(3)    - The link is Down and the PME detects
                        Handshake tones from its peer.
      init(4)         - The link is Initializing, as a result of
                        ifAdminStatus being set to 'up' for a
                        particular PME or a PCS to which the PME
                        is connected.

    This object is intended to supplement the Down(2) state of
    ifOperStatus.

    This object partially maps to the Clause 30 attribute
    aPMEStatus.

    If a Clause 45 MDIO Interface to the PME is present, then this
    object partially maps to PMA/PMD link status bits in 10P/2B
    PMA/PMD status register."
  REFERENCE
    "IEEE Std 802.3, 30.11.2.1.3, 45.2.1.12.4"
  ::= { efmCuPmeStatusEntry 1 }

efmCuPmeFltStatus  OBJECT-TYPE
  SYNTAX      BITS {
    lossOfFraming(0),
    snrMgnDefect(1),
    lineAtnDefect(2),
    deviceFault(3),
    configInitFailure(4),
    protocolInitFailure(5)
  }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "Current/Last PME link Fault Status. This is a bitmap of
    possible conditions. The various bit positions are:
```



```
        lossOfFraming       - Loss of Framing for 10P or
                              Loss of Sync word for 2B PMD or
                              Loss of 64/65-octet framing.
        snrMgnDefect        - SNR margin dropped below the
                              threshold.
        lineAtnDefect       - Line Attenuation exceeds the
                              threshold.
        deviceFault         - Indicates a vendor-dependent
                              diagnostic or self-test fault
                              has been detected.
        configInitFailure   - Configuration initialization failure,
                              due to inability of the PME link to
                              support the configuration profile,
                              requested during initialization.
        protocolInitFailure - Protocol initialization failure, due
                              to an incompatible protocol used by
                              the peer PME during init (that could
                              happen if a peer PMD is a regular
                              G.SDHSL/VDSL modem instead of a
                              2BASE-TL/10PASS-TS PME).

      This object is intended to supplement ifOperStatus in IF-MIB.

      This object holds information about the last fault.
      efmCuPmeFltStatus is cleared by the device restart.
      In addition, lossOfFraming, configInitFailure, and
      protocolInitFailure are cleared by PME init;
      deviceFault is cleared by successful diagnostics/test;
      snrMgnDefect and lineAtnDefect are cleared by SNR margin
      and Line attenuation, respectively, returning to norm and by
      PME init.

      This object partially maps to the Clause 30 attribute
      aPMEStatus.

      If a Clause 45 MDIO Interface to the PME is present, then this
      object consolidates information from various PMA/PMD
      registers, namely: Fault bit in PMA/PMD status 1 register,
      10P/2B PMA/PMD link loss register,
      10P outgoing indicator bits status register,
      10P incoming indicator bits status register,
      2B state defects register."
    REFERENCE
      "IEEE Std 802.3, 30.11.2.1.3, 45.2.1.2.1, 45.2.1.38,
      45.2.1.39, 45.2.1.54"
    ::= { efmCuPmeStatusEntry 2 }

  efmCuPmeOperSubType  OBJECT-TYPE
    SYNTAX      INTEGER {
      ieee2BaseTLO(1),
      ieee2BaseTLR(2),
      ieee10PassTSO(3),
      ieee10PassTSR(4)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "Current operational subtype of the PME.
```

```
      Possible values are:
        ieee2BaseTLO          - PME operates as 2BaseTL-O
        ieee2BaseTLR          - PME operates as 2BaseTL-R
        ieee10PassTSO         - PME operates as 10PassTS-O
        ieee10PassTSR         - PME operates as 10PassTS-R

      The desired operational subtype of the PME can be configured
      via the efmCuPmeAdminSubType variable.

      If a Clause 45 MDIO Interface to the PMA/PMD is present, then
      this object combines values of the Port subtype select
      bits, the PMA/PMD type selection bits in the 10P/2B
      PMA/PMD control register, and the PMA/PMD link status bits in
      the 10P/2B PMA/PMD status register."
    REFERENCE
      "IEEE Std 802.3, 61.1, 45.2.1.11.4, 45.2.1.11.7, 45.2.1.12.4"
    ::= { efmCuPmeStatusEntry 3 }

  efmCuPmeOperProfile  OBJECT-TYPE
    SYNTAX      EfmProfileIndexOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "PME current operating profile. This object is a pointer to
      an entry in either the efmCuPme2BProfileTable or the
      efmCuPme10PProfileTable, depending on the current operating
      SubType of the PME as indicated by efmCuPmeOperSubType.
      Note that a profile entry to which efmCuPmeOperProfile is
      pointing can be created automatically to reflect achieved
      parameters in adaptive (not fixed) initialization,
      i.e., values of efmCuPmeOperProfile and efmCuAdminProfile or
      efmCuPmeAdminProfile may differ.
      The value of zero indicates that the PME is Down or
      Initializing.

      This object partially maps to the aOperatingProfile attribute
      in Clause 30."
    REFERENCE
      "IEEE Std 802.3, 30.11.2.1.7"
    ::= { efmCuPmeStatusEntry 4 }
  efmCuPmeSnrMgn OBJECT-TYPE
    SYNTAX      Integer32(-127..128|65535)
    UNITS       "dB"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The current Signal to Noise Ratio (SNR) margin with respect
      to the received signal as perceived by the local PME.
      The value of 65535 is returned when the PME is Down or
      Initializing.

      This object maps to the aPMESNRMgn attribute in Clause 30.

      If a Clause 45 MDIO Interface is present, then this
      object maps to the 10P/2B RX SNR margin register."
    REFERENCE
      "IEEE Std 802.3, 30.11.2.1.4, 45.2.1.16"
    ::= { efmCuPmeStatusEntry 5 }
```

```
efmCuPmePeerSnrMgn OBJECT-TYPE
  SYNTAX      Integer32(-127..128|65535)
  UNITS       "dB"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The current SNR margin in dB with respect to the received
    signal, as perceived by the remote (link partner) PME.
    The value of 65535 is returned when the PME is Down or
    Initializing.

    This object is irrelevant for the -R PME subtypes. The value
    of 65535 shall be returned in this case.

    If a Clause 45 MDIO Interface is present, then this
    object maps to the 10P/2B link partner RX SNR margin
    register."
  REFERENCE
    "IEEE Std 802.3, 45.2.1.17"
  ::= { efmCuPmeStatusEntry 6}

efmCuPmeLineAtn OBJECT-TYPE
  SYNTAX      Integer32(-127..128|65535)
  UNITS       "dB"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The current Line Attenuation in dB as perceived by the local
    PME.
    The value of 65535 is returned when the PME is Down or
    Initializing.

    If a Clause 45 MDIO Interface is present, then this
    object maps to the Line Attenuation register."
  REFERENCE
    "IEEE Std 802.3, 45.2.1.18"
  ::= { efmCuPmeStatusEntry 7 }

efmCuPmePeerLineAtn OBJECT-TYPE
  SYNTAX      Integer32(-127..128|65535)
  UNITS       "dB"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The current Line Attenuation in dB as perceived by the remote
    (link partner) PME.
    The value of 65535 is returned when the PME is Down or
    Initializing.

    This object is irrelevant for the -R PME subtypes. The value
    of 65535 shall be returned in this case.

    If a Clause 45 MDIO Interface is present, then this
    object maps to the 20P/2B link partner Line Attenuation
    register."
  REFERENCE
    "IEEE Std 802.3, 45.2.1.19"
  ::= { efmCuPmeStatusEntry 8 }
```

```
efmCuPmeEquivalentLength  OBJECT-TYPE
  SYNTAX      Unsigned32(0..8192|65535)
  UNITS       "m"
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "An estimate of the equivalent loop's physical length in
    meters, as perceived by the PME after the link is established.
    An equivalent loop is a hypothetical 26AWG (0.4mm) loop with a
    perfect square root attenuation characteristic, without any
    bridged taps.
    The value of 65535 is returned if the link is Down or
    Initializing or the PME is unable to estimate the equivalent
    length.

    For a 10BASE-TL PME, if a Clause 45 MDIO Interface to the PME
    is present, then this object maps to the 10P Electrical Length
    register."
  REFERENCE
    "IEEE Std 802.3, 45.2.1.21"
  ::= { efmCuPmeStatusEntry 9 }

efmCuPmeTCCodingErrors OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The number of 64/65-octet encapsulation errors. This counter
    is incremented for each 64/65-octet encapsulation error
    detected by the 64/65-octet receive function.

    This object maps to aTCCodingViolations attribute in
    Clause 30.

    If a Clause 45 MDIO Interface to the PME TC is present, then
    this object maps to the TC coding violations register
    (see 45.2.6.12).

    Discontinuities in the value of this counter can occur at
    re-initialization of the management system, and at other times
    as indicated by the value of ifCounterDiscontinuityTime,
    defined in IF-MIB."
  REFERENCE
    "IEEE Std 802.3, 61.3.3.1, 30.11.2.1.5, 45.2.6.12"
  ::= { efmCuPmeStatusEntry 10 }

efmCuPmeTCCrcErrors OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The number of TC-CRC errors. This counter is incremented for
    each TC-CRC error detected by the 64/65-octet receive function
    (see 61.3.3.3 and Figure 61-19).

    This object maps to aTCCRCErrors attribute in
    Clause 30.

    If a Clause 45 MDIO Interface to the PME TC is present, then
```

```
      this object maps to the TC CRC error register
      (see 45.2.6.11).

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 61.3.3.3, 30.11.2.1.10, 45.2.6.11"
    ::= { efmCuPmeStatusEntry 11 }

-- 2BASE-TL specific PME group

 efmCuPme2B      OBJECT IDENTIFIER ::= { efmCuPme 5 }

 efmCuPme2BProfileTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EfmCuPme2BProfileEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "This table supports definitions of administrative and
      operating profiles for 2BASE-TL PMEs.
      The first 14 entries in this table shall always be defined as
      follows (see 802.3ah Annex 63A):
      -------+-------+-------+-----+------+-------------+-----------
      Profile MinRate MaxRate Power Region Constellation Comment
       index  (kb/s)  (kb/s)  (dBm)
      -------+-------+-------+-----+------+-------------+-----------
         1     5696    5696    13.5    1   32-TCPAM      default
         2     3072    3072    13.5    1   32-TCPAM
         3     2048    2048    13.5    1   16-TCPAM
         4     1024    1024    13.5    1   16-TCPAM
         5      704     704    13.5    1   16-TCPAM
         6      512     512    13.5    1   16-TCPAM
         7     5696    5696    14.5    2   32-TCPAM
         8     3072    3072    14.5    2   32-TCPAM
         9     2048    2048    14.5    2   16-TCPAM
        10     1024    1024    13.5    2   16-TCPAM
        11      704     704    13.5    2   16-TCPAM
        12      512     512    13.5    2   16-TCPAM
        13      192    5696       0    1   0             best effort
        14      192    5696       0    2   0             best effort
      -------+-------+-------+-----+------+-------------+-----------

      These default entries shall be created during agent
      initialization and shall not be deleted.

      Entries following the first 14 can be dynamically created and
      deleted to provide custom administrative (configuration)
      profiles and automatic operating profiles.

      This table shall be maintained in a persistent manner."
    REFERENCE
      "IEEE Std 802.3, Annex 63A, 30.11.2.1.6"
    ::= { efmCuPme2B  2 }
 efmCuPme2BProfileEntry OBJECT-TYPE
    SYNTAX      EfmCuPme2BProfileEntry
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
      "Each entry corresponds to a single 2BASE-TL PME profile.
      Each profile contains a set of parameters, used either for
      configuration or representation of a 2BASE-TL PME.
      In case a particular profile is referenced via the
      efmCuPmeAdminProfile object (or efmCuAdminProfile if
      efmCuPmeAdminProfile is zero), it represents the desired
      parameters for the 2BaseTL-O PME initialization.
      If a profile is referenced via an efmCuPmeOperProfile object,
      it represents the current operating parameters of an
      operational PME.

      Profiles may be created/deleted using the row creation/
      deletion mechanism via efmCuPme2BProfileRowStatus. If an
      active entry is referenced, the entry shall remain 'active'
      until all references are removed.
      Default entries shall not be removed."
    INDEX { efmCuPme2BProfileIndex }
    ::= { efmCuPme2BProfileTable 1 }

  EfmCuPme2BProfileEntry ::=
    SEQUENCE {
      efmCuPme2BProfileIndex           EfmProfileIndex,
      efmCuPme2BProfileDescr           SnmpAdminString,
      efmCuPme2BRegion                 INTEGER,
      efmCuPme2BsMode                  EfmProfileIndexOrZero,
      efmCuPme2BMinDataRate            Unsigned32,
      efmCuPme2BMaxDataRate            Unsigned32,
      efmCuPme2BPower                  Unsigned32,
      efmCuPme2BConstellation          INTEGER,
      efmCuPme2BProfileRowStatus       RowStatus
    }

  efmCuPme2BProfileIndex OBJECT-TYPE
    SYNTAX      EfmProfileIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "2BASE-TL PME profile index.
      This object is the unique index associated with this profile.
      Entries in this table are referenced via efmCuAdminProfile or
      efmCuPmeAdminProfile objects."
    ::= { efmCuPme2BProfileEntry 1 }
  efmCuPme2BProfileDescr OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "A textual string containing information about a 2BASE-TL PME
      profile. The string may include information about the data
      rate and spectral limitations of this particular profile."
    ::= { efmCuPme2BProfileEntry 2 }

  efmCuPme2BRegion  OBJECT-TYPE
    SYNTAX      INTEGER {
      region1(1),
      region2(2)
    }
    MAX-ACCESS  read-create
```

```
  STATUS      current
  DESCRIPTION
    "Regional settings for a 2BASE-TL PME, as specified in the
    relevant Regional Annex of ITU-T Recommendation G.991.2.
    Regional settings specify the Power Spectral Density (PSD)
    mask and the Power Back-Off (PBO) values, and place
    limitations on the max allowed data rate, power, and
    constellation.

    Possible values for this object are:
      region1      - Annexes A and F (e.g., North America)
      region2      - Annexes B and G (e.g., Europe)

    Annex A/B specify regional settings for data rates from
    192 kb/s to 2304 kb/s using 16-TCPAM encoding.
    Annex F/G specify regional settings for rates from
    2320 kb/s to 3840 kb/s using 16-TCPAM encoding and from
    768 kb/s to 5696 kb/s using 32-TCPAM encoding.

    If a Clause 45 MDIO Interface to the PME is present, then this
    object partially maps to the Region bits in the 2B general
    parameter register."
  REFERENCE
    "IEEE Std 802.3, 45.2.1.42; ITU-T Recommendation G.991.2,
     Annexes A, B, F and G"
  ::= { efmCuPme2BProfileEntry 3 }

efmCuPme2BsMode  OBJECT-TYPE
  SYNTAX      EfmProfileIndexOrZero
  MAX-ACCESS  read-create
  STATUS      current
  DESCRIPTION
    "Desired custom Spectral Mode for a 2BASE-TL PME. This object
    is a pointer to an entry in efmCuPme2BsModeTable and a block
    of entries in efmCuPme2BRateReachTable, which together define
    (country-specific) reach-dependent rate limitations in
    addition to those defined by efmCuPme2BRegion.

    The value of this object is the index of the referenced
    spectral mode.
    The value of zero (default) indicates that no specific
    spectral mode is applicable.

    Attempts to set this object to a value that is not the value
    of the index for an active entry in the corresponding spectral
    mode table shall be rejected."
  REFERENCE
    "efmCuPme2BsModeTable, efmCuPme2BRateReachTable"
  DEFVAL { 0 }
  ::= { efmCuPme2BProfileEntry 4 }

efmCuPme2BMinDataRate  OBJECT-TYPE
  SYNTAX  Unsigned32(192..5696)
  UNITS       "Kbps"
  MAX-ACCESS  read-create
  STATUS      current
  DESCRIPTION
    "Minimum Data Rate for the 2BASE-TL PME.
    This object can take values of (n x 64)kb/s,
```



```
      where n=3..60 for 16-TCPAM and n=12..89 for 32-TCPAM encoding.

      The data rate of the 2BASE-TL PME is considered 'fixed' when
      the value of this object equals that of efmCuPme2BMaxDataRate.
      If efmCuPme2BMinDataRate is less than efmCuPme2BMaxDataRate in
      the administrative profile, the data rate is considered
      'adaptive', and shall be set to the maximum attainable rate
      not exceeding efmCuPme2BMaxDataRate, under the spectral
      limitations placed by the efmCuPme2BRegion and
      efmCuPme2BsMode.

      Note that the current operational data rate of the PME is
      represented by the ifSpeed object of IF-MIB.

      If a Clause 45 MDIO Interface to the PME is present, then this
      object maps to the Min Data Rate1 bits in the 2B PMD
      parameters register.

      This object shall be maintained in a persistent manner."
    REFERENCE
      "IEEE Std 802.3, 45.2.1.43"
    ::= { efmCuPme2BProfileEntry 5 }
  efmCuPme2BMaxDataRate  OBJECT-TYPE
    SYNTAX  Unsigned32(192..5696)
    UNITS       "Kbps"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "Maximum Data Rate for the 2BASE-TL PME.
      This object can take values of (n x 64)kb/s,
      where n=3..60 for 16-TCPAM and n=12..89 for 32-TCPAM encoding.

      The data rate of the 2BASE-TL PME is considered 'fixed' when
      the value of this object equals that of efmCuPme2BMinDataRate.
      If efmCuPme2BMinDataRate is less than efmCuPme2BMaxDataRate in
      the administrative profile, the data rate is considered
      'adaptive', and shall be set to the maximum attainable rate
      not exceeding efmCuPme2BMaxDataRate, under the spectral
      limitations placed by the efmCuPme2BRegion and
      efmCuPme2BsMode.

      Note that the current operational data rate of the PME is
      represented by the ifSpeed object of IF-MIB.

      If a Clause 45 MDIO Interface to the PME is present, then this
      object maps to the Max Data Rate1 bits in the 2B PMD
      parameters register.

      This object shall be maintained in a persistent manner."
    REFERENCE
      "IEEE Std 802.3, 45.2.1.43"
    ::= { efmCuPme2BProfileEntry 6 }

  efmCuPme2BPower  OBJECT-TYPE
    SYNTAX      Unsigned32(0|10..42)
    UNITS       "0.5 dBm"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
```



```
    "Signal Transmit Power. Multiple of 0.5 dBm.
    The value of 0 in the administrative profile means that the
    signal transmit power is not fixed and shall be set to
    maximize the attainable rate, under the spectral limitations
    placed by the efmCuPme2BRegion and efmCuPme2BsMode.

    If a Clause 45 MDIO Interface to the PME is present, then this
    object maps to the Power1 bits in the 2B PMD parameters
    register."
  REFERENCE
    "IEEE Std 802.3, 45.2.1.43"
  ::= { efmCuPme2BProfileEntry 7 }

efmCuPme2BConstellation  OBJECT-TYPE
  SYNTAX      INTEGER {
    adaptive(0),
    tcpam16(1),
    tcpam32(2)
  }
  MAX-ACCESS  read-create
  STATUS      current
  DESCRIPTION
    "TCPAM Constellation of the 2BASE-TL PME.
    The possible values are:
      adaptive(0)    - either 16- or 32-TCPAM
      tcpam16(1)     - 16-TCPAM
      tcpam32(2)     - 32-TCPAM

    The value of adaptive(0) in the administrative profile means
    that the constellation is not fixed and shall be set to
    maximize the attainable rate, under the spectral limitations
    placed by the efmCuPme2BRegion and efmCuPme2BsMode.

    If a Clause 45 MDIO Interface to the PME is present, then this
    object maps to the Constellation1 bits in the 2B general
    parameter register."
  REFERENCE
     "IEEE Std 802.3, 45.2.1.43"
  ::= { efmCuPme2BProfileEntry 8 }

efmCuPme2BProfileRowStatus OBJECT-TYPE
  SYNTAX      RowStatus
  MAX-ACCESS  read-create
  STATUS      current
  DESCRIPTION
    "This object controls the creation, modification, or deletion
    of the associated entry in the efmCuPme2BProfileTable per the
    semantics of RowStatus.

    If an ‘active’ entry is referenced via efmCuAdminProfile or
    efmCuPmeAdminProfile instance(s), the entry shall remain
    ‘active’.

    An ‘active’ entry shall not be modified. In order to modify
    an existing entry, it shall be taken out of service (by setting
    this object to ‘notInService’), modified, and set ‘active’
    again."
  ::= { efmCuPme2BProfileEntry 9 }
efmCuPme2BsModeTable OBJECT-TYPE
```

```
  SYNTAX      SEQUENCE OF EfmCuPme2BsModeEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table, together with efmCu2BReachRateTable, supports
    definition of administrative custom spectral modes for
    2BASE-TL PMEs, describing spectral limitations in addition to
    those specified by efmCuPme2BRegion.

    In some countries, spectral regulations (e.g., UK ANFP) limit
    the length of the loops for certain data rates. This table
    allows these country-specific limitations to be specified.

    Entries in this table referenced by the efmCuPme2BsMode
    shall not be deleted until all the active references are
    removed.

    This table shall be maintained in a persistent manner."
  REFERENCE
    "efmCu2BReachRateTable"
  ::= { efmCuPme2B  3 }

efmCuPme2BsModeEntry OBJECT-TYPE
  SYNTAX      EfmCuPme2BsModeEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Each entry specifies a spectral mode description and its
    index, which is used to reference corresponding entries in the
    efmCu2BReachRateTable.

    Entries may be created/deleted using the row creation/
    deletion mechanism via efmCuPme2BsModeRowStatus."
  INDEX { efmCuPme2BsModeIndex }
  ::= { efmCuPme2BsModeTable 1 }

EfmCuPme2BsModeEntry ::=
  SEQUENCE {
    efmCuPme2BsModeIndex             EfmProfileIndex,
    efmCuPme2BsModeDescr             SnmpAdminString,
    efmCuPme2BsModeRowStatus         RowStatus
  }

efmCuPme2BsModeIndex OBJECT-TYPE
  SYNTAX      EfmProfileIndex
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "2BASE-TL PME Spectral Mode index.
    This object is the unique index associated with this spectral
    mode.
    Entries in this table are referenced via the efmCuPme2BsMode
    object."
  ::= { efmCuPme2BsModeEntry 1 }

efmCuPme2BsModeDescr OBJECT-TYPE
  SYNTAX      SnmpAdminString
  MAX-ACCESS  read-create
  STATUS      current
```

```
    DESCRIPTION
      "A textual string containing information about a 2BASE-TL PME
      spectral mode. The string may include information about
      corresponding (country-specific) spectral regulations
      and rate/reach limitations of this particular spectral mode."
    ::= { efmCuPme2BsModeEntry 2 }

  efmCuPme2BsModeRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "This object controls creation, modification, or deletion of
      the associated entry in efmCuPme2BsModeTable per the semantics
      of RowStatus.

      If an 'active' entry is referenced via efmCuPme2BsMode
      instance(s), the entry shall remain 'active'.

      An 'active' entry shall not be modified. In order to modify
      an existing entry, it shall be taken out of service (by setting
      this object to 'notInService'), modified, and set 'active'
      again."
    ::= { efmCuPme2BsModeEntry 3 }


  efmCuPme2BReachRateTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EfmCuPme2BReachRateEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "This table supports the definition of administrative custom
      spectral modes for 2BASE-TL PMEs, providing spectral
      limitations in addition to those specified by
      efmCuPme2BRegion.
      The spectral regulations in some countries (e.g., UK ANFP)
      limit the length of the loops for certain data rates.
      This table allows these country-specific limitations to be
      specified.

      Below is an example of this table for NICC Document ND1602:2005/08:
      ----------+-------+-------
      Equivalent MaxRate MaxRate
        Length    PAM16   PAM32
          (m)     (kb/s)  (kb/s)
      ----------+-------+-------
           975      2304    5696
          1125      2304    5504
          1275      2304    5120
          1350      2304    4864
          1425      2304    4544
          1500      2304    4288
          1575      2304    3968
          1650      2304    3776
          1725      2304    3520
          1800      2304    3264
          1875      2304    3072
          1950      2048    2688
          2100      1792    2368
```

```
        2250      1536       0
        2400      1408       0
        2550      1280       0
        2775      1152       0
        2925      1152       0
        3150      1088       0
        3375      1024       0
     ----------+-------+-------

     Entries in this table referenced by an efmCuPme2BsMode
     instance shall not be deleted.

     This table shall be maintained in a persistent manner."
   REFERENCE
     "NICC Document ND1602:2005/08"
   ::= { efmCuPme2B  4 }

 efmCuPme2BReachRateEntry OBJECT-TYPE
   SYNTAX      EfmCuPme2BReachRateEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
     "Each entry specifies maximum 2BASE-TL PME data rates
     allowed for a certain equivalent loop length, when using
     16-TCPAM or 32-TCPAM encoding.

     When a 2BASE-TL PME is initialized, its data rate shall not
     exceed the following limitations:
     - the value of efmCuPme2BMaxDataRate
     - maximum data rate allowed by efmCuPme2BRegion and
       efmCuPme2BPower
     - maximum data rate for a given encoding specified in the
       efmCuPme2BsModeEntry, corresponding to the equivalent loop
       length, estimated by the PME

     efmCuPme2BEquivalentLength values should be assigned
     in increasing order, starting from the minimum value.

     Entries may be created/deleted using the row creation/
     deletion mechanism via efmCuPme2ReachRateRowStatus."
   INDEX { efmCuPme2BsModeIndex, efmCuPme2BReachRateIndex }
   ::= { efmCuPme2BReachRateTable 1 }

 EfmCuPme2BReachRateEntry ::=
   SEQUENCE {
     efmCuPme2BReachRateIndex         EfmProfileIndex,
     efmCuPme2BEquivalentLength       Unsigned32,
     efmCuPme2BMaxDataRatePam16       Unsigned32,
     efmCuPme2BMaxDataRatePam32       Unsigned32,
     efmCuPme2BReachRateRowStatus     RowStatus
   }

 efmCuPme2BReachRateIndex OBJECT-TYPE
   SYNTAX      EfmProfileIndex
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
     "2BASE-TL custom spectral mode Reach-Rate table index.
     This object is the unique index associated with each entry."
```



```
    ::= { efmCuPme2BReachRateEntry 1 }

  efmCuPme2BEquivalentLength  OBJECT-TYPE
    SYNTAX      Unsigned32(0..8192)
    UNITS       "m"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "Maximum allowed equivalent loop's physical length in meters
      for the specified data rates.
      An equivalent loop is a hypothetical 26AWG (0.4mm) loop with a
      perfect square root attenuation characteristic, without any
      bridged taps."
    ::= { efmCuPme2BReachRateEntry 2 }

  efmCuPme2BMaxDataRatePam16  OBJECT-TYPE
    SYNTAX      Unsigned32(0|192..5696)
    UNITS       "Kbps"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "Maximum data rate for a 2BASE-TL PME at the specified
      equivalent loop's length using TC-PAM16 encoding.
      The value of zero means that TC-PAM16 encoding should not be
      used at this distance."
    ::= { efmCuPme2BReachRateEntry 3 }

  efmCuPme2BMaxDataRatePam32  OBJECT-TYPE
    SYNTAX      Unsigned32(0|192..5696)
    UNITS       "Kbps"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "Maximum data rate for a 2BASE-TL PME at the specified
      equivalent loop's length using TC-PAM32 encoding.
      The value of zero means that TC-PAM32 encoding should not be
      used at this distance."
    ::= { efmCuPme2BReachRateEntry 4 }

  efmCuPme2BReachRateRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "This object controls the creation, modification, or deletion
      of the associated entry in the efmCuPme2BReachRateTable per
      the semantics of RowStatus.

      If an 'active' entry is referenced via efmCuPme2BsMode
      instance(s), the entry shall remain 'active'.

      An 'active' entry shall not be modified. In order to modify
      an existing entry, it shall be taken out of service (by setting
      this object to 'notInService'), modified, and set 'active'
      again."
    ::= { efmCuPme2BReachRateEntry 5 }

-- 10PASS-TS specific PME group
```



```
efmCuPme10P      OBJECT IDENTIFIER ::= { efmCuPme 6 }

efmCuPme10PProfileTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF EfmCuPme10PProfileEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "This table supports definitions of configuration profiles for
    10PASS-TS PMEs.
    The first 22 entries in this table shall always be defined as
    follows (see 802.3ah Annex 62B.3, table 62B-1):
    -------+--------+----+---------+-----+-----+---------------
    Profile Bandplan UPBO BandNotch DRate URate Comment
     Index  PSDMask#  p#    p#        p#    p#
    -------+--------+----+---------+-----+-----+---------------
       1       1      3    2,6,10,11    20    20 default profile
       2      13      5    0            20    20
       3       1      1    0            20    20
       4      16      0    0           100   100
       5      16      0    0            70    50
       6       6      0    0            50    10
       7      17      0    0            30    30
       8       8      0    0            30     5
       9       4      0    0            25    25
      10       4      0    0            15    15
      11      23      0    0            10    10
      12      23      0    0             5     5
      13      16      0    2,5,9,11    100   100
      14      16      0    2,5,9,11     70    50
      15       6      0    2,6,10,11    50    10
      16      17      0    2,5,9,11     30    30
      17       8      0    2,6,10,11    30     5
      18       4      0    2,6,10,11    25    25
      19       4      0    2,6,10,11    15    15
      20      23      0    2,5,9,11     10    10
      21      23      0    2,5,9,11      5     5
      22      30      0    0           200    50
    -------+--------+----+---------+-----+-----+---------------

    These default entries shall be created during agent
    initialization and shall not be deleted.

    Entries following the first 22 can be dynamically created and
    deleted to provide custom administrative (configuration)
    profiles and automatic operating profiles.

    This table shall be maintained in a persistent manner."
  REFERENCE
    "IEEE Std 802.3, Annex 62B.3, 30.11.2.1.6"
  ::= { efmCuPme10P  1 }

efmCuPme10PProfileEntry OBJECT-TYPE
  SYNTAX      EfmCuPme10PProfileEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Each entry corresponds to a single 10PASS-TS PME profile.

    Each profile contains a set of parameters, used either for
```



```
    configuration or representation of a 10PASS-TS PME.
    In case a particular profile is referenced via the
    efmCuPmeAdminProfile object (or efmCuAdminProfile if
    efmCuPmeAdminProfile is zero), it represents the desired
    parameters for the 10PassTS-O PME initialization.
    If a profile is referenced via an efmCuPmeOperProfile object,
    it represents the current operating parameters of the PME.

    Profiles may be created/deleted using the row creation/
    deletion mechanism via efmCuPme10PProfileRowStatus. If an
    'active' entry is referenced, the entry shall remain 'active'
    until all references are removed.
    Default entries shall not be removed."
  INDEX { efmCuPme10PProfileIndex }
  ::= { efmCuPme10PProfileTable 1 }

EfmCuPme10PProfileEntry ::=
  SEQUENCE {
    efmCuPme10PProfileIndex           EfmProfileIndex,
    efmCuPme10PProfileDescr           SnmpAdminString,
    efmCuPme10PBandplanPSDMskProfile  INTEGER,
    efmCuPme10PUPBOReferenceProfile   INTEGER,
    efmCuPme10PBandNotchProfiles      BITS,
    efmCuPme10PPayloadDRateProfile    INTEGER,
    efmCuPme10PPayloadURateProfile    INTEGER,
    efmCuPme10PProfileRowStatus       RowStatus
  }

efmCuPme10PProfileIndex OBJECT-TYPE
  SYNTAX      EfmProfileIndex
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "10PASS-TS PME profile index.
    This object is the unique index associated with this profile.
    Entries in this table are referenced via efmCuAdminProfile or
    efmCuPmeAdminProfile."
  ::= { efmCuPme10PProfileEntry 1 }

efmCuPme10PProfileDescr OBJECT-TYPE
  SYNTAX      SnmpAdminString
  MAX-ACCESS  read-create
  STATUS      current
  DESCRIPTION
    "A textual string containing information about a 10PASS-TS PME
    profile. The string may include information about data rate
    and spectral limitations of this particular profile."
  ::= { efmCuPme10PProfileEntry 2 }

efmCuPme10PBandplanPSDMskProfile  OBJECT-TYPE
  SYNTAX  INTEGER {
    profile1(1),
    profile2(2),
    profile3(3),
    profile4(4),
    profile5(5),
    profile6(6),
    profile7(7),
    profile8(8),
```

```
      profile9(9),
      profile10(10),
      profile11(11),
      profile12(12),
      profile13(13),
      profile14(14),
      profile15(15),
      profile16(16),
      profile17(17),
      profile18(18),
      profile19(19),
      profile20(20),
      profile21(21),
      profile22(22),
      profile23(23),
      profile24(24),
      profile25(25),
      profile26(26),
      profile27(27),
      profile28(28),
      profile29(29),
      profile30(30)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "The 10PASS-TS PME Bandplan and PSD Mask Profile, as specified
      in IEEE Std 802.3 Annex 62A, table 62A-1. Possible values are:
      -------------+-----------------------+-----------+--------
      Profile Name    PSD Mask                    Bands      ITU-T G.993.1
                                                0/1/2/3/4/5  Bandplan
      -------------+-----------------------+-----------+--------
      profile1(1)    ANSI T1.424 FTTCab.M1         x/D/U/D/U    A
      profile2(2)    ANSI T1.424 FTTEx.M1          x/D/U/D/U    A
      profile3(3)    ANSI T1.424 FTTCab.M2         x/D/U/D/U    A
      profile4(4)    ANSI T1.424 FTTEx.M2          x/D/U/D/U    A
      profile5(5)    ANSI T1.424 FTTCab.M1         D/D/U/D/U    A
      profile6(6)    ANSI T1.424 FTTEx.M1          D/D/U/D/U    A
      profile7(7)    ANSI T1.424 FTTCab.M2         D/D/U/D/U    A
      profile8(8)    ANSI T1.424 FTTEx.M2          D/D/U/D/U    A
      profile9(9)    ANSI T1.424 FTTCab.M1         U/D/U/D/x    A
      profile10(10)  ANSI T1.424 FTTEx.M1          U/D/U/D/x    A
      profile11(11)  ANSI T1.424 FTTCab.M2         U/D/U/D/x    A
      profile12(12)  ANSI T1.424 FTTEx.M2          U/D/U/D/x    A
      profile13(13)  ETSI TS 101 270-1 Pcab.M1.A   x/D/U/D/U    B
      profile14(14)  ETSI TS 101 270-1 Pcab.M1.B   x/D/U/D/U    B
      profile15(15)  ETSI TS 101 270-1 Pex.P1.M1   x/D/U/D/U    B
      profile16(16)  ETSI TS 101 270-1 Pex.P2.M1   x/D/U/D/U    B
      profile17(17)  ETSI TS 101 270-1 Pcab.M2     x/D/U/D/U    B
      profile18(18)  ETSI TS 101 270-1 Pex.P1.M2   x/D/U/D/U    B
      profile19(19)  ETSI TS 101 270-1 Pex.P2.M2   x/D/U/D/U    B
      profile20(20)  ETSI TS 101 270-1 Pcab.M1.A   U/D/U/D/x    B
      profile21(21)  ETSI TS 101 270-1 Pcab.M1.B   U/D/U/D/x    B
      profile22(22)  ETSI TS 101 270-1 Pex.P1.M1   U/D/U/D/x    B
      profile23(23)  ETSI TS 101 270-1 Pex.P2.M1   U/D/U/D/x    B
      profile24(24)  ETSI TS 101 270-1 Pcab.M2     U/D/U/D/x    B
      profile25(25)  ETSI TS 101 270-1 Pex.P1.M2   U/D/U/D/x    B
      profile26(26)  ETSI TS 101 270-1 Pex.P2.M2   U/D/U/D/x    B
      profile27(27)  ITU-T G.993.1 F.1.2.1         x/D/U/D/U    Annex F
```



```
      profile28(28)  ITU-T G.993.1 F.1.2.2         x/D/U/D/U    Annex F
      profile29(29)  ITU-T G.993.1 F.1.2.3         x/D/U/D/U    Annex F
      profile30(30)  ANSI T1.424 FTTCab.M1 (ext.)  x/D/U/D/U/D  Annex A
      --------------+------------------------+------------+--------
      "
    REFERENCE
      "IEEE Std 802.3, Annex 62A"
    ::= { efmCuPme10PProfileEntry 3 }

  efmCuPme10PUPBOReferenceProfile  OBJECT-TYPE
    SYNTAX  INTEGER {
      profile0(0),
      profile1(1),
      profile2(2),
      profile3(3),
      profile4(4),
      profile5(5),
      profile6(6),
      profile7(7),
      profile8(8),
      profile9(9)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "The 10PASS-TS PME Upstream Power Back-Off (UPBO) Reference
      PSD Profile, as specified in 802.3 Annex 62A, table 62A-3.
      Possible values are:
      ------------+----------------------------
      Profile Name   Reference        PSD
      ------------+----------------------------
      profile0(0)  no profile
      profile1(1)  ANSI T1.424        Noise A    M1
      profile2(2)  ANSI T1.424        Noise A    M2
      profile3(3)  ANSI T1.424        Noise F    M1
      profile4(4)  ANSI T1.424        Noise F    M2
      profile5(5)  ETSI TS 101 270-1  Noise A&B
      profile6(6)  ETSI TS 101 270-1  Noise C
      profile7(7)  ETSI TS 101 270-1  Noise D
      profile8(8)  ETSI TS 101 270-1  Noise E
      profile9(9)  ETSI TS 101 270-1  Noise F
      ------------+----------------------------
      "
    REFERENCE
      "IEEE Std 802.3, Annex 62A.3.5"
    ::= { efmCuPme10PProfileEntry 4 }

  efmCuPme10PBandNotchProfiles  OBJECT-TYPE
    SYNTAX  BITS {
      profile0(0),
      profile1(1),
      profile2(2),
      profile3(3),
      profile4(4),
      profile5(5),
      profile6(6),
      profile7(7),
      profile8(8),
      profile9(9),
```

```
      profile10(10),
      profile11(11)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "The 10PASS-TS PME Egress Control Band Notch Profile bitmap,
      as specified in IEEE Std 802.3 Annex 62A, table 62A-4. Possible
      values are:
      --------------+--------+------+-----------+------+------
      Profile Name   G.991.3  T1.424 TS 101 270-1 StartF EndF
                     table    table  table        (MHz)  (MHz)
      --------------+--------+------+-----------+------+------
      profile0(0)    no profile
      profile1(1)    F-5 #01  -      -            1.810  1.825
      profile2(2)    6-2      15-1   17           1.810  2.000
      profile3(3)    F-5 #02  -      -            1.907  1.912
      profile4(4)    F-5 #03  -      -            3.500  3.575
      profile5(5)    6-2      -      17           3.500  3.800
      profile6(6)    -        15-1   -            3.500  4.000
      profile7(7)    F-5 #04  -      -            3.747  3.754
      profile8(8)    F-5 #05  -      -            3.791  3.805
      profile9(9)    6-2      -      17           7.000  7.100
      profile10(10)  F-5 #06  15-1   -            7.000  7.300
      profile11(11)  6-2      15-1   1            10.100 10.150
      --------------+--------+------+-----------+------+------

      Any combination of profiles can be specified by ORing
      individual profiles, for example, a value of 0x2230 selects
      profiles 2, 6, 10, and 11."
    REFERENCE
      "IEEE Std 802.3, Annex 62A.3.5"
    ::= { efmCuPme10PProfileEntry 5 }

  efmCuPme10PPayloadDRateProfile  OBJECT-TYPE
    SYNTAX      INTEGER {
      profile5(5),
      profile10(10),
      profile15(15),
      profile20(20),
      profile25(25),
      profile30(30),
      profile50(50),
      profile70(70),
      profile100(100),
      profile140(140),
      profile200(200)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "The 10PASS-TS PME Downstream Payload Rate Profile, as
      specified in IEEE Std 802.3 Annex 62A. Possible values are:
        profile5(5)      - 2.5 Mb/s
        profile10(10)    - 5 Mb/s
        profile15(15)    - 7.5 Mb/s
        profile20(20)    - 10 Mb/s
        profile25(25)    - 12.5 Mb/s
        profile30(30)    - 15 Mb/s
```

```
        profile50(50)    - 25 Mb/s
        profile70(70)    - 35 Mb/s
        profile100(100)  - 50 Mb/s
        profile140(140)  - 70 Mb/s
        profile200(200)  - 100 Mb/s

      Each value represents a target for the PME’s Downstream
      Payload Bitrate as seen at the MII. If the payload rate of
      the selected profile cannot be achieved based on the loop
      environment, bandplan, and PSD mask, the PME initialization
      shall fail."
    REFERENCE
      "IEEE Std 802.3, Annex 62A.3.6"
    ::= { efmCuPme10PProfileEntry 6 }

  efmCuPme10PPayloadURateProfile  OBJECT-TYPE
    SYNTAX      INTEGER {
      profile5(5),
      profile10(10),
      profile15(15),
      profile20(20),
      profile25(25),
      profile30(30),
      profile50(50),
      profile70(70),
      profile100(100)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "The 10PASS-TS PME Upstream Payload Rate Profile, as specified
       in 802.3 Annex 62A. Possible values are:
        profile5(5)       - 2.5 Mb/s
        profile10(10)     - 5 Mb/s
        profile15(15)     - 7.5 Mb/s
        profile20(20)     - 10 Mb/s
        profile25(25)     - 12.5 Mb/s
        profile30(30)     - 15 Mb/s
        profile50(50)     - 25 Mb/s
        profile70(70)     - 35 Mb/s
        profile100(100)   - 50 Mb/s
      Each value represents a target for the PME’s Upstream Payload
      Bitrate as seen at the MII. If the payload rate of the
      selected profile cannot be achieved based on the loop
      environment, bandplan, and PSD mask, the PME initialization
      shall fail."
    REFERENCE
      "IEEE Std 802.3, Annex 62A.3.6"
    ::= { efmCuPme10PProfileEntry 7 }

  efmCuPme10PProfileRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
      "This object controls creation, modification, or deletion of
      the associated entry in efmCuPme10PProfileTable per the
      semantics of RowStatus.
```



```
      If an active entry is referenced via efmCuAdminProfile or
      efmCuPmeAdminProfile, the entry shall remain 'active' until
      all references are removed.

      An 'active' entry shall not be modified. In order to modify
      an existing entry, it shall be taken out of service (by setting
      this object to 'notInService'), modified, and set 'active'
      again."
    ::= { efmCuPme10PProfileEntry 8 }


  efmCuPme10PStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EfmCuPme10PStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "This table provides status information of EFMCu 10PASS-TS
      PMEs (modems).

      This table contains live data from the equipment. As such,
      it is not persistent."
    ::= { efmCuPme10P 2 }

  efmCuPme10PStatusEntry OBJECT-TYPE
    SYNTAX      EfmCuPme10PStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "An entry in the EFMCu 10PASS-TS PME Status table."
    INDEX  { ifIndex }
    ::= { efmCuPme10PStatusTable 1 }

  EfmCuPme10PStatusEntry ::=
    SEQUENCE {
      efmCuPme10PFECCorrectedBlocks     Counter32,
      efmCuPme10PFECUncorrectedBlocks   Counter32
    }

  efmCuPme10PFECCorrectedBlocks  OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The number of received and corrected Forward Error Correction
      (FEC) codewords in this 10PASS-TS PME.

      This object maps to the aPMEFECCorrectedBlocks attribute in
      Clause 30.

      If a Clause 45 MDIO Interface to the PMA/PMD is present,
      then this object maps to the 10P FEC correctable errors
      register.

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 45.2.1.22, 30.11.2.1.8"
```

```
    ::= { efmCuPme10PStatusEntry 1 }

  efmCuPme10PFECUncorrectedBlocks  OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "The number of received uncorrectable FEC codewords in this
      10PASS-TS PME.

      This object maps to the aPMEFECUncorrectableBlocks attribute
      in Clause 30.

      If a Clause 45 MDIO Interface to the PMA/PMD is present,
      then this object maps to the 10P FEC uncorrectable errors
      register.

      Discontinuities in the value of this counter can occur at
      re-initialization of the management system, and at other times
      as indicated by the value of ifCounterDiscontinuityTime,
      defined in IF-MIB."
    REFERENCE
      "IEEE Std 802.3, 45.2.1.23, 30.11.2.1.9"
    ::= { efmCuPme10PStatusEntry 2 }

--
-- Conformance statements
--

 efmCuGroups      OBJECT IDENTIFIER ::= { efmCuConformance 1 }

 efmCuCompliances OBJECT IDENTIFIER ::= { efmCuConformance 2 }

 -- Object Groups

 efmCuBasicGroup OBJECT-GROUP
   OBJECTS {
     efmCuPAFSupported,
     efmCuAdminProfile,
     efmCuTargetDataRate,
     efmCuTargetSnrMgn,
     efmCuAdaptiveSpectra,
     efmCuPortSide,
     efmCuFltStatus
   }
   STATUS      current
   DESCRIPTION
     "A collection of objects representing management information
     common for all types of EFMCu ports."
   ::= { efmCuGroups 1 }

 efmCuPAFGroup OBJECT-GROUP
   OBJECTS {
     efmCuPeerPAFSupported,
     efmCuPAFCapacity,
     efmCuPeerPAFCapacity,
     efmCuPAFAdminState,
     efmCuPAFDiscoveryCode,
     efmCuPAFRemoteDiscoveryCode,
```

```
      efmCuNumPMEs
    }
    STATUS      current
    DESCRIPTION
      "A collection of objects supporting optional PME
      Aggregation Function (PAF) and PAF discovery in EFMCu ports."
    ::= { efmCuGroups 2 }
  efmCuPAFErrorsGroup OBJECT-GROUP
    OBJECTS {
      efmCuPAFInErrors,
      efmCuPAFInSmallFragments,
      efmCuPAFInLargeFragments,
      efmCuPAFInBadFragments,
      efmCuPAFInLostFragments,
      efmCuPAFInLostStarts,
      efmCuPAFInLostEnds,
      efmCuPAFInOverflows
    }
    STATUS      current
    DESCRIPTION
      "A collection of objects supporting optional error counters
      of PAF on EFMCu ports."
    ::= { efmCuGroups 3 }

  efmCuPmeGroup OBJECT-GROUP
    OBJECTS {
      efmCuPmeAdminProfile,
      efmCuPmeOperStatus,
      efmCuPmeFltStatus,
      efmCuPmeSubTypesSupported,
      efmCuPmeAdminSubType,
      efmCuPmeOperSubType,
      efmCuPAFRemoteDiscoveryCode,
      efmCuPmeOperProfile,
      efmCuPmeSnrMgn,
      efmCuPmePeerSnrMgn,
      efmCuPmeLineAtn,
      efmCuPmePeerLineAtn,
      efmCuPmeEquivalentLength,
      efmCuPmeTCCodingErrors,
      efmCuPmeTCCrcErrors,
      efmCuPmeThreshLineAtn,
      efmCuPmeThreshSnrMgn
    }
    STATUS      current
    DESCRIPTION
      "A collection of objects providing information about
      a 2BASE-TL/10PASS-TS PME."
    ::= { efmCuGroups 4 }

  efmCuAlarmConfGroup OBJECT-GROUP
    OBJECTS {
      efmCuThreshLowRate,
      efmCuLowRateCrossingEnable,
      efmCuPmeThreshLineAtn,
      efmCuPmeLineAtnCrossingEnable,
      efmCuPmeThreshSnrMgn,
      efmCuPmeSnrMgnCrossingEnable,
      efmCuPmeDeviceFaultEnable,
```

```
      efmCuPmeConfigInitFailEnable,
      efmCuPmeProtocolInitFailEnable
    }
    STATUS      current
    DESCRIPTION
      "A collection of objects supporting configuration of alarm
      thresholds and notifications in EFMCu ports."
    ::= { efmCuGroups 5 }

  efmCuNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
      efmCuLowRateCrossing,
      efmCuPmeLineAtnCrossing,
      efmCuPmeSnrMgnCrossing,
      efmCuPmeDeviceFault,
      efmCuPmeConfigInitFailure,
      efmCuPmeProtocolInitFailure
    }
    STATUS      current
    DESCRIPTION
      "This group supports notifications of significant conditions
      associated with EFMCu ports."
    ::= { efmCuGroups 6 }

  efmCuPme2BProfileGroup OBJECT-GROUP
    OBJECTS {
      efmCuPme2BProfileDescr,
      efmCuPme2BRegion,
      efmCuPme2BsMode,
      efmCuPme2BMinDataRate,
      efmCuPme2BMaxDataRate,
      efmCuPme2BPower,
      efmCuPme2BConstellation,
      efmCuPme2BProfileRowStatus,
      efmCuPme2BsModeDescr,
      efmCuPme2BsModeRowStatus,
      efmCuPme2BEquivalentLength,
      efmCuPme2BMaxDataRatePam16,
      efmCuPme2BMaxDataRatePam32,
      efmCuPme2BReachRateRowStatus
    }
    STATUS      current
    DESCRIPTION
      "A collection of objects that constitute a configuration
      profile for configuration of 2BASE-TL ports."
    ::= { efmCuGroups 7}

  efmCuPme10PProfileGroup OBJECT-GROUP
    OBJECTS {
      efmCuPme10PProfileDescr,
      efmCuPme10PBandplanPSDMskProfile,
      efmCuPme10PUPBOReferenceProfile,
      efmCuPme10PBandNotchProfiles,
      efmCuPme10PPayloadDRateProfile,
      efmCuPme10PPayloadURateProfile,
      efmCuPme10PProfileRowStatus
    }
    STATUS  current
    DESCRIPTION
```

```
      "A collection of objects that constitute a configuration
      profile for configuration of 10PASS-TS ports."
    ::= { efmCuGroups 8 }

  efmCuPme10PStatusGroup OBJECT-GROUP
    OBJECTS {
      efmCuPme10PFECCorrectedBlocks,
      efmCuPme10PFECUncorrectedBlocks
    }
    STATUS  current
    DESCRIPTION
      "A collection of objects providing status information
      specific to 10PASS-TS PMEs."
    ::= { efmCuGroups 9 }

 -- Compliance statements

  efmCuCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
      "The compliance statement for 2BASE-TL/10PASS-TS interfaces.
      Compliance with the following external compliance statements
      is required:

      MIB module              Compliance Statement
      ----------              --------------------
      IF-MIB                  ifCompliance3
      IEEE8023-EtherLike-MIB dot3Compliance2
      MAU-MIB                 mauModIfCompl3

      Compliance with the following external compliance statements
      is optional for implementations supporting PME Aggregation
      Function (PAF) with flexible cross-connect between the PCS
      and PME ports:

      MIB module              Compliance Statement
      ----------              --------------------
      IF-INVERTED-STACK-MIB  ifInvCompliance
      IF-CAP-STACK-MIB       ifCapStackCompliance"

    MODULE  -- this module
      MANDATORY-GROUPS {
        efmCuBasicGroup,
        efmCuPmeGroup,
        efmCuAlarmConfGroup,
        efmCuNotificationGroup
      }

      GROUP       efmCuPme2BProfileGroup
      DESCRIPTION
        "Support for this group is only required for implementations
        supporting 2BASE-TL PHY."

      GROUP       efmCuPme10PProfileGroup
      DESCRIPTION
        "Support for this group is only required for implementations
        supporting 10PASS-TS PHY."

      GROUP       efmCuPAFGroup
```

```
        DESCRIPTION
          "Support for this group is only required for
          implementations supporting PME Aggregation Function (PAF)."

        GROUP       efmCuPAFErrorsGroup
        DESCRIPTION
          "Support for this group is optional for implementations
          supporting PME Aggregation Function (PAF)."

        GROUP       efmCuPme10PStatusGroup
        DESCRIPTION
          "Support for this group is optional for implementations
          supporting 10PASS-TS PHY."

        OBJECT      efmCuPmeSubTypesSupported
        SYNTAX      BITS {
          ieee2BaseTLO(0),
          ieee2BaseTLR(1),
          ieee10PassTSO(2),
          ieee10PassTSR(3)
        }
        DESCRIPTION
          "Support for all subtypes is not required. However, at
          least one value shall be supported."

        OBJECT      efmCuPmeAdminSubType
        MIN-ACCESS  read-only
        DESCRIPTION
          "Write access is not required (needed only for PMEs
          supporting more than a single subtype, e.g.,
          ieee2BaseTLO and ieee2BaseTLR or ieee10PassTSO and
          ieee10PassTSR)."

        OBJECT      efmCuTargetSnrMgn
        MIN-ACCESS  read-only
        DESCRIPTION
          "Write access is optional. For PHYs without write access,
          the target SNR margin shall be fixed at 5dB for 2BASE-TL
          and 6dB for 10PASS-TS."

        OBJECT      efmCuAdaptiveSpectra
        MIN-ACCESS  read-only
        DESCRIPTION
          "Write access is optional. For PHYs without write access,
          the default value should be false."

      ::= { efmCuCompliances 1 }
END
```


12. Ethernet wide area network (WAN) interface sublayer (WIS) MIB module

This clause defines a portion of the Management Information Base (MIB) for use with SNMP. In particular, it defines objects for managing IEEE Std 802.3 WAN interface sublayers.

12.1 Overview

The objects defined in this clause are used in conjunction with objects defined in the Interfaces Group MIB in IETF RFC 2863, the SONET/SDH Interface MIB in IETF RFC 3592, and the IEEE 802.3 MAU MIB defined in Clause 13 of this document to manage the Ethernet WAN interface sublayer (WIS) defined in IEEE Std 802.3. 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 IEEE Std 802.3 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 clause 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 IEEE 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 of IEEE Std 802.3. 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 IEEE Std 802.3.

12.1.1 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 in IETF RFC 3592. Thus, the MIB module defined in this clause is a sparse augmentation of the SONET-MIB—in other words, every table defined here is an extension of some table in the SONET-MI—and its compliance statement REQUIRES that an agent implementing the objects defined in this clause 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 12.1.6. 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 clause requires no hardware support beyond that mandated by 50.3.11 of IEEE Std 802.3.

12.1.2 Relationship to the Ethernet-like interface MIB

An interface that includes the Ethernet WIS is, by definition, an Ethernet-like interface, and an agent implementing the objects defined in this clause shall also implement the objects required by the Ethernet-like interface MIB module defined in Clause 10.

12.1.3 Relationship to the IEEE 802.3 MAU MIB

Support for the mauModIfCompl3 compliance statement of the MAU-MIB module defined in Clause 13 is required for all Ethernet-like interfaces. The MAU-MIB module is needed in order to allow applications to control and/or determine the media type in use; this is important for devices that 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 module 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.

12.1.4 Use of the ifTable

This subclause specifies how the ifTable, as defined in IETF RFC 2863, is used for the Ethernet WIS application.

12.1.4.1 Layering model

Ethernet interfaces that employ the WIS are layered as defined in IEEE Std 802.3. The corresponding use of the ifTable defined in IETF RFC 2863 is shown in Figure 12–1.

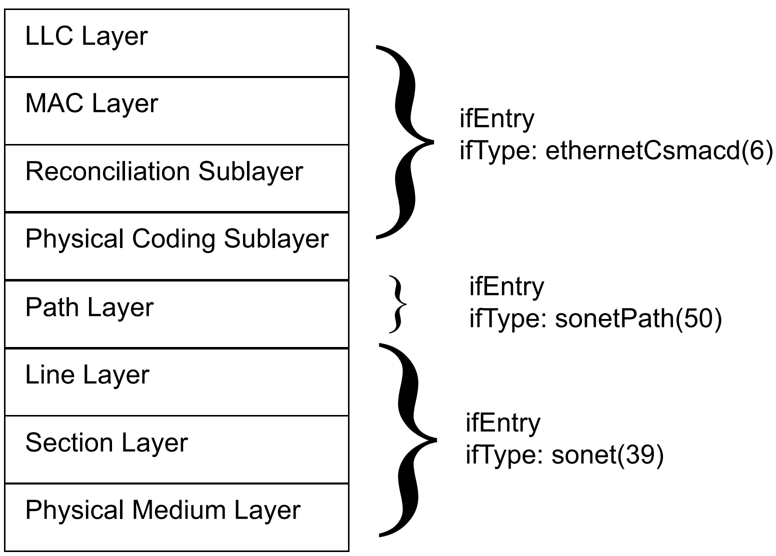


Figure 12–1—Use of ifTable for an Ethernet WIS port

The exact configuration and multiplexing of the layers is maintained in the ifStackTable in IETF RFC 2863 and in the ifInvStackTable in IETF RFC 2864.

12.1.4.2 Use of ifTable for LLC layer/MAC sublayer/reconciliation sublayer/physical coding sublayer

The ifTable shall be used as specified in Clause 10 and Clause 13 for the LLC Layer/MAC sublayer/reconciliation sublayer/physical coding sublayer.

12.1.4.3 Use of ifTable for SONET/SDH path layer

The ifTable shall be used as specified in IETF RFC 3592 for the SONET/SDH path layer. The value of ifHighSpeed is set to 9585. ifSpeed reports a value of 4294967295.

12.1.4.4 Use of ifTable for SONET/SDH medium/section/line layer

The ifTable shall be used as specified in IETF RFC 3592 for the SONET/SDH Medium/Section/Line Layer. The value of ifHighSpeed is set to 9953. ifSpeed reports a value of 4294967295.

12.1.5 SONET/SDH terminology

The SONET/SDH terminology used in IEEE Std 802.3 is mostly the same as in IETF RFC 3592, but there are a few differences. In those cases, the definitions in Clause 3 take precedence.

12.1.6 Mapping of IEEE 802.3 managed objects

Table 12–1 contains the mapping between oWIS managed objects in the pWIS Basic package defined in IEEE Std 802.3 and managed objects defined in this clause and in associated MIB modules, i.e., the IF-MIB in IETF RFC 2863, the SONET-MIB in IETF RFC 3592, and the IEEE 802.3 MAU-MIB module defined in Clause 13 of this document.

Table 12–1—Mapping of IEEE 802.3 managed objects (pWIS Basic package)

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

The Unequipped defect is not defined by IEEE Std 802.3.

Table 12–2 contains the same mapping information for the pWIS optional package.

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 presented in Table 12–3. 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 50.3.11 of IEEE Std 802.3 suffices for both. See Annex 12A for details.

Table 12–2—Mapping of IEEE 802.3 managed objects (pWIS optional package)

IEEE 802.3 managed object		Corresponding SNMP object
oWIS – pWISOptional package	aSectionSESThreshold	SONET-MIB – sonetSESthresholdSe
	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

Table 12–2—Mapping of IEEE 802.3 managed objects (pWIS optional package) *(continued)*

IEEE 802.3 managed object		Corresponding SNMP object
oWIS - pWISOptional package (continued)	aJ1ValueRX	etherWisPathCurrentJ1Received
	aFarEndPathSESs	SONET-MIB – sonetFarEndPathCurrentSESs + sonetFarEndPathIntervalSESs
	aFarEndPathESs	SONET-MIB – sonetFarEndPathCurrentESs + sonetFarEndPathIntervalESs
	aFarEndPathCVs	SONET-MIB – sonetFarEndPathCurrentCVs + sonetFarEndPathIntervalCVs

Table 12–3—IEEE 802.3 managed object and SNMP object differences

IEEE 802.3 managed object	How corresponding SNMP object differs
aSectionSESThreshold	This object is defined in IEEE Std 802.3 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 IEEE Std 802.3 as a generalized nonresettable counter. The objects sonetSectionCurrentSESs and sonetSectionIntervalSESs are 15-minute interval counters.
aSectionESs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3. The objects sonetSectionCurrentESs and sonetSectionIntervalESs are 15-minute interval counters.
aSectionSEFSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3. The objects sonetSectionCurrentSEFSs and sonetSectionIntervalSEFSs are 15-minute interval counters.
aSectionCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, 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 IEEE Std 802.3 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 nonresettable counter in IEEE Std 802.3, 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 nonresettable counter in IEEE Std 802.3, 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.

Table 12–3—IEEE 802.3 managed object and SNMP object differences *(continued)*

IEEE 802.3 managed object	How corresponding SNMP object differs
aLineCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, 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 nonresettable counter in IEEE Std 802.3, 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 nonresettable counter in IEEE Std 802.3, 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 nonresettable counter in IEEE Std 802.3, 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 IEEE Std 802.3 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 nonresettable counter in IEEE Std 802.3, 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, IEEE Std 802.3 includes PLM-P and LCD-P defects in the criteria for declaring path layer severely errored seconds, while IETF RFC 3592 does not.
aPathESs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, 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, IEEE Std 802.3 includes PLM-P and LCD-P defects in the criteria for declaring path layer errored seconds, while IETF RFC 3592 does not
aPathCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, 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

Table 12–3—IEEE 802.3 managed object and SNMP object differences *(continued)*

IEEE 802.3 managed object	How corresponding SNMP object differs
aFarEndPathSESs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, 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, IEEE Std 802.3 includes far-end PLM-P and LCD-P defects in the criteria for declaring far-end path layer severely errored seconds, while IETF RFC 3592 does not.
aFarEndPathESs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, 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, IEEE Std 802.3 includes far-end PLM-P and LCD-P defects in the criteria for declaring far-end path layer errored seconds, while IETF RFC 3592 does not.
aFarEndPathCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, 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.

12.1.7 Mapping of SNMP objects to WIS station management registers

Some of the objects defined in this clause or incorporated by reference from the SONET-MIB IETF RFC 3592, or the MAU-MIB module defined in Clause 13 require WIS-specific hardware support. 50.3.11 of IEEE Std 802.3 specifies WIS management interface requirements, including a required subset of the WIS Management Data Input/Output (MDIO) registers defined in 45.2.2 of IEEE Std 802.3. Table 12–4 provides a cross- reference between those managed objects and the WIS MDIO registers from the subset in 50.3.11 of IEEE Std 802.3 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 shall be provided by other means.

Table 12–4—Cross reference between SNMP objects and WIS MDIO registers

SMNP 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

Table 12–4—Cross reference between SNMP objects and WIS MDIO registers *(continued)*

SMNP object	WIS MDIO register(s)
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	10G WIS status 3 + 10G WIS section BIP error count
SONET-MIB - sonetSectionCurrentSESs	
SONET-MIB - sonetSectionCurrentSEFSs	
SONET-MIB - sonetSectionCurrentCVs	
SONET-MIB - sonetSectionIntervalESs	
SONET-MIB - sonetSectionIntervalSESs	
SONET-MIB - sonetSectionIntervalSEFSs	
SONET-MIB - sonetSectionIntervalCVs	
SONET-MIB - sonetSectionIntervalValidData	None required
SONET-MIB - sonetLineCurrentStatus	10G WIS status 3
SONET-MIB - sonetLineCurrentESs	10G WIS status 3 + 10G WIS line BIP errors
SONET-MIB - sonetLineCurrentSESs	
SONET-MIB - sonetLineCurrentCVs	
SONET-MIB - sonetLineCurrentUASs	
SONET-MIB - sonetLineIntervalESs	
SONET-MIB - sonetLineIntervalSESs	
SONET-MIB - sonetLineIntervalCVs	
SONET-MIB - sonetLineIntervalUASs	
SONET-MIB - sonetLineIntervalValidData	None required

Table 12–4—Cross reference between SNMP objects and WIS MDIO registers *(continued)*

SMNP object	WIS MDIO register(s)
SONET-MIB - sonetFarEndLineCurrentESs	10G WIS status 3 + 10G WIS far end line BIP errors
SONET-MIB - sonetFarEndLineCurrentSESs	
SONET-MIB - sonetFarEndLineCurrentCVs	
SONET-MIB - sonetFarEndLineCurrentUASs	
SONET-MIB - sonetFarEndLineIntervalESs	
SONET-MIB - sonetFarEndLineIntervalSESs	
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 - sonetPathIntervalCVs	
SONET-MIB - sonetPathIntervalUASs	
SONET-MIB - sonetPathIntervalValidData	None required
ETHER-WIS - etherWisFarEndPathCurrentStatus	10G WIS status 3

Table 12–4—Cross reference between SNMP objects and WIS MDIO registers *(continued)*

SMNP object	WIS MDIO register(s)
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	
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 + 10G WIS status 3
MAU-MIB - ifMauMediaAvailableStateExits	
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

12.1.8 Structure of the MIB module

Four tables are defined in this MIB module.

12.1.8.1 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 50.3.8 of IEEE Std 802.3.

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

12.1.8.2 etherWisSectionCurrentTable

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

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

12.1.8.3 etherWisPathCurrentTable

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

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

12.1.8.4 etherWisFarEndPathCurrentTable

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

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

12.2 Security considerations for Ethernet wide area network (WAN) interface sublayer (WIS) MIB module

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:

— 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.

The user of this MIB module should 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 GET and NOTIFY access to these objects and possibly to encrypt their values when sending them over the network via SNMP.

12.3 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL:[19]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C12mib.txt

[19]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
IEEE8023-ETHER-WIS-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE,
       Gauge32, org
           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;

   ieee8023etherWisMIB MODULE-IDENTITY
        LAST-UPDATED "201102020000Z" -- February 2, 2011
        ORGANIZATION
          "IEEE 802.3 working group"
        CONTACT-INFO
            "WG-URL: http://www.ieee802.org/3/index.html
            WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

            Contact: Howard Frazier
            Postal:  3151 Zanker Road
                     San Jose, CA 95134
                     USA
            Tel:     +1.408.922.8164
            E-mail:  hfrazier@broadcom.com"
DESCRIPTION
         "The objects in this MIB module are used in conjunction
         with objects in the SONET-MIB module and the MAU-MIB module to manage
         the Ethernet WAN Interface Sublayer (WIS) defined in
         IEEE Std 802.3.

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

       REVISION "201102020000Z" -- February 2, 2011
       DESCRIPTION
           "Initial version, based on an earlier version published as RFC 3637."

           ::= { org ieee(111) standards-association-numbers-series-standards(2)
                 lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 12 }


   -- The main sections of the module

   etherWisObjects     OBJECT IDENTIFIER ::= { ieee8023etherWisMIB 1 }

   etherWisObjectsPath OBJECT IDENTIFIER ::= { ieee8023etherWisMIB 2 }
```



```
etherWisConformance OBJECT IDENTIFIER ::= { ieee8023etherWisMIB 3 }

-- groups in the Ethernet WIS MIB module

etherWisDevice      OBJECT IDENTIFIER ::= { etherWisObjects 1 }

etherWisSection     OBJECT IDENTIFIER ::= { etherWisObjects 2 }

etherWisPath        OBJECT IDENTIFIER ::= { etherWisObjectsPath 1 }

etherWisFarEndPath  OBJECT IDENTIFIER ::= { etherWisObjectsPath 2 }

-- The Device group

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

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

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

EtherWisDeviceEntry ::=
    SEQUENCE {
        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 Std 802.3, 50.3.8.1.
       The value prbs31(3) puts the WIS transmit path into the
```



```
        PRBS31 test pattern mode described in IEEE Std 802.3
        50.3.8.2. The value mixedFrequency(4) puts the
        WIS transmit path into the mixed frequency test pattern
        mode described in IEEE Std 802.3, 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) shall 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)
        shall be rejected with the error inconsistentValue."
     REFERENCE
        "IEEE Std 802.3, 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 Std 802.3, 50.3.8.2. The value
        mixedFrequency(4) puts the WIS receive path into the mixed
        frequency test pattern mode described in IEEE Std 802.3,
        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) shall 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) shall be rejected with the error inconsistentValue."
     REFERENCE
        "IEEE Std 802.3, 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 Std 802.3, 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 Std 802.3, 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 shall 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 Std 802.3, 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 Std 802.3, 30.8.1.1.9, aJ0ValueRX."
      ::= { etherWisSectionCurrentEntry 2 }

-- The Path group

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

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

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

EtherWisPathCurrentEntry ::=
     SEQUENCE {
         etherWisPathCurrentStatus           BITS,
         etherWisPathCurrentJ1Transmitted    OCTET STRING,
         etherWisPathCurrentJ1Received       OCTET STRING
         }
etherWisPathCurrentStatus OBJECT-TYPE
     SYNTAX  BITS {
                 etherWisPathLOP(0),
                 etherWisPathAIS(1),
                 etherWisPathPLM(2),
                 etherWisPathLCD(3)
             }
     MAX-ACCESS  read-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. When this
           bit is set, sonetPathSTSLOP shall 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. When
           this bit is set, sonetPathSTSAIS shall 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. When
           this bit is set, sonetPathSignalLabelMismatch
           shall 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 Std 802.3, 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 Std 802.3, 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 Std 802.3, 30.8.1.1.24, aJ1ValueRX."
     ::= { etherWisPathCurrentEntry 3 }
-- The Far End Path group

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

etherWisFarEndPathCurrentTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF EtherWisFarEndPathCurrentEntry
```

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

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

EtherWisFarEndPathCurrentEntry ::=
    SEQUENCE {
        etherWisFarEndPathCurrentStatus     BITS
        }

etherWisFarEndPathCurrentStatus OBJECT-TYPE
    SYNTAX  BITS {
                etherWisFarEndPayloadDefect(0),
                etherWisFarEndServerDefect(1)
            }
    MAX-ACCESS  read-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. When this bit is set,
          sonetPathSTSRDI shall be set in the corresponding
          instance of sonetPathCurrentStatus."
    REFERENCE
       "IEEE Std 802.3, 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
           IEEE8023-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
```


Annex 12A

(informative)

Collection of performance data using WIS MDIO registers

The purpose of this annex is to illustrate how the WIS MDIO registers specified in IEEE Std 802.345.2.2 (and more specifically the subset required by IEEE Std 802.3 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 IEEE Std 802.3 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 IETF RFC 3592. 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 ANSI T1.231-1997 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 ANSI T1.231-1997 Section 9.1.2.2 for details).

An agent implementing the IEEE Std 802.3 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.

13. Ethernet medium attachment units (MAUs) MIB module

13.1 Introduction

This clause defines a portion of the Management Information Base (MIB) for use with SNMP. In particular, it defines objects for managing IEEE 802.3 medium attachment units (MAUs).

A previous version of this clause, IETF RFC 3636, defined a single MIB module. IETF RFC 4836 split the original MIB module into two, putting frequently updated object identities and textual conventions into a separate, IANA-maintained MIB module, in order to decrease the need of updating the basic MAU-MIB module. The MIB module defined in this clause incorporates the IANA-MAU-MIB module by reference.

13.2 Overview

Instances of these object types represent attributes of an IEEE 802.3 MAU. Several types of MAUs are defined in IEEE Std 802.3. These MAUs may be connected to IEEE 802.3 repeaters or to IEEE 802.3 (Ethernet-like) interfaces. For convenience, this clause refers to these devices as "repeater MAUs" and "interface MAUs."

The definitions presented here are based on 30.5 and 30.6 of IEEE Std 802.3. This specification is intended to provide for management of all types of Ethernet/IEEE 802.3 MAUs.

13.2.1 Relationship to IETF RFC 3636 and IETF RFC 4836

The management definitions provided in this clause are intended to be a superset of those defined by IETF RFC 3636 and IETF RFC 4836.

In order to decrease the need of updating the basic MAU MIB module due to the new MAU type, Media Available state, Auto Negotiation capability and/or Jack type introduction, all relevant object identities and textual conventions have been moved to a separate, IANA-maintained MIB module, IANA-MAU-MIB. Thus when a new MAU type, Media Available state, Auto Negotiation capability, and/or Jack type is defined by the IEEE 802.3 working group, only the IANA-maintained module needs to be revised, leaving the basic MAU-MIB module defined in this clause unchanged.

The changes made in this revision are not entirely backward-compatible with MIB modules that currently import MAU type object identity descriptors from the MAU-MIB; such modules need to be revised to import those DESCRIPTORS from the IANA-MAU-MIB. Similarly, any management applications that process the object identity definitions (e.g., to present the DESCRIPTION text to a user) need to get those definitions from the IANA-MAU-MIB instead of the MAU-MIB. While it is true that changes that require such adjustments are not strictly compliant with the SMIv2 rules governing MIB module revisions (see IETF STD 58, RFC 2578 Section 10), in this case continued high maintenance costs that would result from not making these changes make the deviation from the rules justified.

13.2.2 Relationship to other MIBs

It is assumed that an agent implementing the MAU-MIB module will also implement (at least) the 'system' group defined in the SNMPv2 MIB of IETF RFC 3418. The following subclauses identify other MIBs that such an agent should implement.

13.2.2.1 Relationship to the Interfaces Group MIB

The subclauses of this clause that define interface MAU-related objects specify an extension to the Interfaces Group MIB of IETF RFC 2863. An agent implementing these interface-MAU related objects shall also implement the relevant groups of the ifCompliance3 MODULE-COMPLIANCE statement of the Interfaces Group MIB. The value of the object ifMauIfIndex is the same as the value of ‘ifIndex’ used to instantiate the interface to which the given MAU is connected.

An agent implementing the interface-MAU related objects in the MAU-MIB module shall also fully comply with the dot3Compliance2 MODULE-COMPLIANCE statement of the Ethernet-like interface MIB defined in Clause 10. Furthermore, when the interface-MAU related objects are used to manage a 10GBASE-W PHY—i.e., when ifMauType is equal to dot3MauType10GigBaseW or any other 10GBASE-W variant—then the agent shall also support the Ethernet WAN Interface Sublayer (WIS) MIB module defined in Clause 12, and shall follow the interface layering model specified therein. In that case the value of the object ifMauIfIndex is the same as the value of ‘ifIndex’ for the layer at the top of the stack, i.e., for the ifTable entry that has ‘ifType’ equal to ethernetCsmacd(6). If the interface-MAU related objects are used to manage a PHY that allows the MAU type to be changed dynamically, then the agent shall create ifTable, ifStackTable, and ifInvStackTable entries that pertain to the WIS when ifMauDefaultType is changed to a 10GBASEW variant (i.e., one of dot3MauType10GigBaseW, dot3MauType10GigBaseEW, dot3MauType10GigBaseLW, or dot3MauType10GigBaseSW) from any other type, and shall destroy the WIS-related entries when ifMauDefaultType is changed to a non-10GBASE-W type. The agent shall also change the values of ‘ifConnectorPresent’ and ‘ifHighSpeed’ in the ifTable entry indexed by ifMauIfIndex as specified in Clause 10 and Clause 12 when ifMauDefaultType is manipulated in this way, but shall NOT otherwise alter that entry.

NOTE—Repeater ports are not represented as interfaces in the Interfaces Group MIB.

13.2.2.2 Relationship to the IEEE 802.3 repeater MIB module

The subclause of this clause that defines repeater MAU-related objects specifies an extension to the IEEE 802.3 repeater MIB module defined in Clause 7. An agent implementing these repeater-MAU related objects shall also comply with the snmpRptrModCompl compliance statement of the IEEE 802.3 repeater MIB module.

The values of ‘rpMauGroupIndex’ and ‘rpMauPortIndex’ used to instantiate a repeater MAU variable shall be the same as the values of ‘rptrPortGroupIndex’ and ‘rptrPortIndex’ used to instantiate the port to which the given MAU is connected.

13.2.3 Management of internal MAUs

In some situations, a MAU can be “internal”, i.e., its functionality is implemented entirely within a device. For example, a managed repeater may contain an internal repeater-MAU and/or an internal interface-MAU through which management communications originating on one of the repeater’s external ports pass, in order to reach the management agent associated with the repeater. Such internal MAUs may or may not be managed. If they are managed, objects describing their attributes should appear in the appropriate MIB subtree: dot3RpMauBasicGroup for internal repeater-MAUs and dot3IfMauBasicGroup for internal interface-MAUs.

13.2.4 Mapping of IEEE 802.3 managed objects

Table 13–1 depicts the mapping between relevant managed objects (attributes) defined in IEEE Std 802.3 Clause 30, and managed objects defined in this clause.|

Table 13–1—Mapping of IEEE 802.3 managed objects

IEEE 802.3 managed object		Corresponding SNMP object
oMAU	.aMAUID	rpMauIndex or ifMauIndex or broadMauIndex
	.aMAUType	rpMauType or ifMauType
	.aMAUTypeList	ifMauTypeListBits
	.aMediaAvailable	rpMauMediaAvailable or ifMauMediaAvailable
	.aLoseMediaCounter	rpMauMediaAvailableStateExits or ifMauMediaAvailableStateExits
	.aJabber	rpMauJabberState and rpMauJabberingStateEnters or ifMauJabberState and ifMauJabberingStateEnters
	.aMAUAdminState	rpMauStatus or ifMauStatus
	.aFalseCarriers	rpMauFalseCarriers or ifMauFalseCarriers
	.acResetMAU	rpMauStatus or ifMauStatus
	.acMAUAdminControl	rpMauStatus or ifMauStatus
	.nJabber	rpMauJabberTrap or ifMauJabberTrap
oAutoNegotiation	.aAutoNegID	ifMauIndex
	.aAutoNegAdminState	ifMauAutoNegAdminStatus
	.aAutoNegRemoteSignalling	ifMauAutoNegRemoteSignalling
	.aAutoNegAutoConfig	ifMauAutoNegConfig
	.aAutoNegLocalTechnologyAbility	ifMauAutoNegCapabilityBits
	.aAutoNegAdvertisedTechnologyAbility	ifMauAutoNegAdvertisedBits and ifMauAutoNegRemoteFaultAdvertised
	.aAutoNegReceivedTechnologyAbility	ifMauAutoNegReceivedBits and ifMauAutoNegRemoteFaultReceived
	.acAutoNegRestartAutoConfig	ifMauAutoNegRestart
	.acAutoNegAdminControl	ifMauAutoNegAdminStatus

Table 13–2 depicts the IEEE 802.3 managed objects that have not been included in the MAU-MIB module, and the reason for the exclusion.

Table 13–2—Unmapped IEEE 802.3 managed objects

IEEE 802.3 managed object		Reason for exclusion
oMAU	.aIdleErrorCount	Only useful for 100BaseT2, which is not widely implemented
oAutoNegotiation	.aAutoNegLocalSeletorAbility	Only needed for support of isoethernet (IEEE 802.9a), which is not supported by MAU-MIB
	.aAutoNegAdvertisedSelectorAbility	
	.aAutoNegReceivedSelectorAbility	

13.2.5 Addition of new MAU types

13.2.5.1 dot3MauType OBJECT-IDENTITIES

The dot3MauType OBJECT IDENTIFIER and its OBJECT-IDENTITY definitions has been moved from the MAU-MIB module into the IANA-maintained IANA-MAU-MIB module.

When a new IEEE 802.3 MAU is defined, IANA can re-issue a version of IANA-MAU-MIB module with the new dot3MauType OBJECT-IDENTITY and its matching IANAifMauTypeListBits textual convention value and, possibly, with new IANAifMauMediaAvailable, IANAifMauAutoNegCapBits, and/or IANAif-JackType values.

An Expert Review, as defined in IETF RFC 2434, is required for the addition of the new MAU, Media Available states, Auto Negotiation capabilities, and/or Jack types.

In some cases, new MAU types may require additional managed objects or may have side effects on the behavior of existing managed objects. In such cases a standards-track specification (which may be a new document or a revision of this document) is also required. Any such document is required to note any special properties of the MAU types that it defines—for example, side effects on the ifStackTable as noted in this document for 10GBASE-W MAUs.

13.2.5.2 IANAifMauTypeListBits TEXTUAL-CONVENTION

The syntax of ifMauTypeListBits is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauTypeListBits, which can be re-specified (with additional values, when defined by IEEE 802.3) in the IANA-maintained MIB module without issuing a new version of this document.

13.2.5.3 IANAifMauMediaAvailable TEXTUAL-CONVENTION

The syntax of ifMauMediaAvailable and rpMauMediaAvailable is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauMediaAvailable, which can be re-specified (with additional values, when defined by IEEE 802.3) in the IANA-maintained MIB module without issuing a new version of this document.

13.2.5.4 IANAifMauAutoNegCapBits TEXTUAL-CONVENTION

The syntax of ifMauAutoNegCapabilityBits, ifMauAutoNegCapAdvertisedBits, and ifMauAutoNegCapReceivedBits objects is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauAutoNegCapBits, which can be re-specified (with additional values, when defined by IEEE 802.3) in the IANA-maintained MIB module without issuing a new version of this document.

13.2.5.5 JackType TEXTUAL-CONVENTION

The JackType Textual Convention has been deprecated in favor of the IANAifJackType defined in the IANA-maintained MIB module, so the new Jack types can be added (when defined by IEEE 802.3) without issuing a new version of this document.

13.3 Security considerations for Ethernet medium attachment units (MAUs) MIB module

The IANA-MAU-MIB module does not define any management objects. Instead, it defines a set of textual conventions that are used by the MAU-MIB module and may be used by other MIB modules to define management objects. Meaningful security considerations can only be written for MIB modules that define management objects.

There are a number of management objects defined in the MAU-MIB module that have a MAX-ACCESS clause of read-write. Setting these objects can have a serious effect on the operation of the network, including:

- Enabling or disabling a MAU
- Changing a MAU's default type
- Enabling, disabling, or restarting autonegotiation
- Modifying the capabilities that a MAU advertises during autonegotiation.

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.

Some of the readable objects in the MAU-MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. In some environments, it may be undesirable to allow unauthorized parties to access statistics or status information about individual links in a network. It is thus important to control GET and/or NOTIFY access to these objects and possibly to encrypt the values of these objects when sending them over the network via SNMP.

13.4 IANA considerations

It is intended that each new MAU type, Media Available state, Auto Negotiation capability, and/or Jack type defined by the IEEE 802.3 working group and approved for publication in a revision of IEEE Std 802.3 will be added to the IANA-maintained MIB module, provided that it is suitable for being managed by the base objects in the MAU-MIB module.

For each new MAU type added, a short description of the MAU technology and, wherever possible, a reference to a publicly available specification should be specified. An Expert Review, as defined in IETF RFC 2434, is required, for each modification.

13.5 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL:[20]

http://www.ieee802.org/3/be/public/mib_modules/20110202/802dot3dot1C13mib.txt

The IANA-MAU-MIB module can be found at the following URL:

http://www.iana.org/assignments/ianamau-mib

[20]Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.


```
IEEE8023-MAU-MIB DEFINITIONS ::= BEGIN

     IMPORTS
       Counter32, Integer32, Counter64,
       OBJECT-TYPE, MODULE-IDENTITY, NOTIFICATION-TYPE, org
         FROM SNMPv2-SMI         -- RFC 2578
       TruthValue, AutonomousType
         FROM SNMPv2-TC          -- RFC 2579
       OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
         FROM SNMPv2-CONF        -- RFC 2580
       InterfaceIndex
         FROM IF-MIB             -- RFC 2863
       IANAifMauTypeListBits, IANAifMauMediaAvailable,
       IANAifMauAutoNegCapBits, IANAifJackType
         FROM IANA-MAU-MIB
                          -- http://www.iana.org/assignments/ianamau-mib
       ;

     ieee8023mauMIB MODULE-IDENTITY
        LAST-UPDATED "201102020000Z" -- February 2, 2011
        ORGANIZATION
          "IEEE 802.3 working group"
        CONTACT-INFO
            "WG-URL: http://www.ieee802.org/3/index.html
            WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG

            Contact: Howard Frazier
            Postal:  3151 Zanker Road
                     San Jose, CA 95134
                     USA
            Tel:     +1.408.922.8164
            E-mail:  hfrazier@broadcom.com"
DESCRIPTION
         "Management information for 802.3 MAUs."

       REVISION    "201102020000Z" -- February 2, 2011
       DESCRIPTION
            "Initial version, based on an earlier version published as RFC 4836."

           ::= { org ieee(111) standards-association-numbers-series-standards(2)
                 lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 13 }


      ieee8023snmpDot3MauMgt OBJECT IDENTIFIER ::= { ieee8023mauMIB 1 }

      dot3RpMauBasicGroup
          OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 1 }
      dot3IfMauBasicGroup
          OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 2 }
      -- The following object is a placeholder
      -- to preserve the arc assignments that follow it.
      dot3PlaceholderGroup
          OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 3 }

      -- OIDs under the following branch are reserved for
      -- the IANA-MAU-MIB to assign as MAU type values:
      --                        { ieee8023snmpDot3MauMgt 4 }

      dot3IfMauAutoNegGroup
```

```
    OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 5 }

--
-- The Basic Repeater MAU Table
--

rpMauTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF RpMauEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION "Table of descriptive and status information
              about the MAU(s) attached to the ports of a
              repeater."
  ::= { dot3RpMauBasicGroup 1 }

rpMauEntry OBJECT-TYPE
  SYNTAX      RpMauEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION "An entry in the table, containing information
              about a single MAU."
  INDEX       { rpMauGroupIndex,
                rpMauPortIndex,
                rpMauIndex
              }
  ::= { rpMauTable 1 }

RpMauEntry ::=
  SEQUENCE {
      rpMauGroupIndex                     Integer32,
      rpMauPortIndex                      Integer32,
      rpMauIndex                          Integer32,
      rpMauType                           AutonomousType,
      rpMauStatus                         INTEGER,
      rpMauMediaAvailable                 IANAifMauMediaAvailable,
      rpMauMediaAvailableStateExits       Counter32,
      rpMauJabberState                    INTEGER,
      rpMauJabberingStateEnters           Counter32,
      rpMauFalseCarriers                  Counter32
}

rpMauGroupIndex OBJECT-TYPE
  SYNTAX      Integer32 (1..2147483647)
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION "This variable uniquely identifies the group
              containing the port to which the MAU described
              by this entry is connected.

              Note:  In practice, a group will generally be
              a field-replaceable unit (i.e., module, card,
              or board) that can fit in the physical system
              enclosure, and the group number will correspond
              to a number marked on the physical enclosure.

              The group denoted by a particular value of this
              object is the same as the group denoted by the
              same value of rptrGroupIndex."
  REFERENCE   "RFC 2108, rptrGroupIndex."
```



```
  ::= { rpMauEntry 1 }

rpMauPortIndex OBJECT-TYPE
  SYNTAX      Integer32 (1..2147483647)
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION "This variable uniquely identifies the repeater
              port within group rpMauGroupIndex to which the
              MAU described by this entry is connected."
  REFERENCE   "RFC 2108, rptrPortIndex."
  ::= { rpMauEntry 2 }

rpMauIndex OBJECT-TYPE
  SYNTAX      Integer32 (1..2147483647)
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION "This variable uniquely identifies the MAU
              described by this entry from among other
              MAUs connected to the same port
              (rpMauPortIndex)."
  REFERENCE   "IEEE Std 802.3, 30.5.1.1.1, aMAUID."
  ::= { rpMauEntry 3 }

rpMauType OBJECT-TYPE
  SYNTAX      AutonomousType
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION "This object identifies the MAU type. Values for
              standard IEEE 802.3 MAU types are defined in the
              IANA maintained IANA-MAU-MIB module, as
              OBJECT-IDENTITIES of dot3MauType.
              If the MAU type is unknown, the object identifier
              zeroDotZero is returned."
  REFERENCE   "IEEE Std 802.3, 30.5.1.1.2, aMAUType."
  ::= { rpMauEntry 4 }

rpMauStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),
                    unknown(2),
                    operational(3),
                    standby(4),
                    shutdown(5),
                    reset(6)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The current state of the MAU. This object may
                be implemented as a read-only object by those
                agents and MAUs that do not implement software
                control of the MAU state. Some agents may not
                support setting the value of this object to some
                of the enumerated values.

                The value other(1) is returned if the MAU is in
                a state other than one of the states 2 through
                6.
                The value unknown(2) is returned when the MAU's
                true state is unknown; for example, when it is
```

```
                being initialized.

                A MAU in the operational(3) state is fully
                functional; it operates, and passes signals to its
                attached DTE or repeater port in accordance to
                its specification.

                A MAU in standby(4) state forces DI and CI to
                idle, and the media transmitter to idle or fault,
                if supported. Standby(4) mode only applies to
                link type MAUs. The state of
                rpMauMediaAvailable is unaffected.

                A MAU in shutdown(5) state assumes the same
                condition on DI, CI, and the media transmitter,
                as though it were powered down or not connected.
                The MAU may return other(1) value for the
                rpMauJabberState and rpMauMediaAvailable objects
                when it is in this state. For an AUI, this
                state will remove power from the AUI.

                Setting this variable to the value reset(6)
                resets the MAU in the same manner as a
                power-off, power-on cycle of at least one-half
                second would. The agent is not required to
                return the value reset(6).

                Setting this variable to the value
                operational(3), standby(4), or shutdown(5)
                causes the MAU to assume the respective state,
                except that setting a mixing-type MAU or an AUI
                to standby(4) will cause the MAU to enter the
                shutdown state."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.7, aMAUAdminState,
                30.5.1.2.2, acMAUAdminControl, and 30.5.1.2.1,
                acResetMAU."
    ::= { rpMauEntry 5 }

rpMauMediaAvailable OBJECT-TYPE
    SYNTAX      IANAifMauMediaAvailable
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "This object identifies Media Available state of
                the MAU, complementary to the rpMauStatus. Values
                for the standard IEEE 802.3 Media Available states
                are defined in the IANA maintained IANA-MAU-MIB
                module, as IANAifMauMediaAvailable TC."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.4, aMediaAvailable."
    ::= { rpMauEntry 6 }

rpMauMediaAvailableStateExits OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of the number of times that
                rpMauMediaAvailable for this MAU instance leaves
                the state available(3).

                Discontinuities in the value of this counter can
```

```
                occur at re-initialization of the management
                system and at other times, as indicated by the
                value of rptrMonitorPortLastChange."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.5, aLoseMediaCounter.
                RFC 2108, rptrMonitorPortLastChange"
    ::= { rpMauEntry 7 }

rpMauJabberState OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),
                    unknown(2),
                    noJabber(3),
                    jabbering(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The value other(1) is returned if the jabber
                state is not 2, 3, or 4. The agent shall always
                return other(1) for MAU type dot3MauTypeAUI.

                The value unknown(2) is returned when the MAU's
                true state is unknown; for example, when it is
                being initialized.

                If the MAU is not jabbering the agent returns
                noJabber(3). This is the 'normal' state.

                If the MAU is in jabber state the agent returns
                the jabbering(4) value."
    REFERENCE "IEEE Std 802.3, 30.5.1.1.6, aJabber.jabberFlag."
    ::= { rpMauEntry 8 }

rpMauJabberingStateEnters OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of the number of times that
                mauJabberState for this MAU instance enters the
                state jabbering(4). For MAUs of type
                dot3MauTypeAUI, dot3MauType100BaseT4,
                dot3MauType100BaseTX, dot3MauType100BaseFX, and
                all 1000 Mb/s types, this counter will always
                indicate zero.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system and at other times, as indicated by the
                value of rptrMonitorPortLastChange."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.6, aJabber.jabberCounter.
                RFC 2108, rptrMonitorPortLastChange"
    ::= { rpMauEntry 9 }

rpMauFalseCarriers OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of the number of false carrier events
                during IDLE in 100BASE-X links. This counter
                does not increment at the symbol rate. It can
```

```
                 increment after a valid carrier completion at a
                 maximum rate of once per 100 ms until the next
                 carrier event.

                 This counter increments only for MAUs of type
                 dot3MauType100BaseT4, dot3MauType100BaseTX,
                 dot3MauType100BaseFX, and all 1000 Mb/s types.

                 For all other MAU types, this counter will
                 always indicate zero.

                 The approximate minimum time for rollover of
                 this counter is 7.4 hours.

                 Discontinuities in the value of this counter can
                 occur at re-initialization of the management
                 system and at other times, as indicated by the
                 value of rptrMonitorPortLastChange."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.10, aFalseCarriers.
                 RFC 2108, rptrMonitorPortLastChange"
    ::= { rpMauEntry 10 }

-- The rpJackTable applies to MAUs attached to repeaters
-- which have one or more external jacks (connectors).
rpJackTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF RpJackEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Information about the external jacks attached
                 to MAUs attached to the ports of a repeater."
    ::= { dot3RpMauBasicGroup 2 }

rpJackEntry OBJECT-TYPE
    SYNTAX      RpJackEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the table, containing information
                 about a particular jack."
    INDEX       { rpMauGroupIndex,
                  rpMauPortIndex,
                  rpMauIndex,
                  rpJackIndex
                }
    ::= { rpJackTable 1 }

RpJackEntry ::=
    SEQUENCE {
        rpJackIndex                         Integer32,
        rpJackType                          IANAifJackType
    }

rpJackIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "This variable uniquely identifies the jack
                 described by this entry from among other jacks
                 attached to the same MAU (rpMauIndex)."
    ::= { rpJackEntry 1 }
```

```
rpJackType OBJECT-TYPE
    SYNTAX      IANAifJackType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The jack connector type, as it appears on the
                outside of the system."
    ::= { rpJackEntry 2 }

--
-- The Basic Interface MAU Table
--
ifMauTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IfMauEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Table of descriptive and status information
                about MAU(s) attached to an interface."
    ::= { dot3IfMauBasicGroup 1 }

ifMauEntry OBJECT-TYPE
    SYNTAX      IfMauEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the table, containing information
                about a single MAU."
    INDEX       { ifMauIfIndex,
                  ifMauIndex
                }
    ::= { ifMauTable 1 }

IfMauEntry ::=
    SEQUENCE {
        ifMauIfIndex                      InterfaceIndex,
        ifMauIndex                        Integer32,
        ifMauType                         AutonomousType,
        ifMauStatus                       INTEGER,
        ifMauMediaAvailable               IANAifMauMediaAvailable,
        ifMauMediaAvailableStateExits     Counter32,
        ifMauJabberState                  INTEGER,
        ifMauJabberingStateEnters         Counter32,
        ifMauFalseCarriers                Counter32,
        ifMauDefaultType                  AutonomousType,
        ifMauAutoNegSupported             TruthValue,
        ifMauTypeListBits                 IANAifMauTypeListBits,
        ifMauHCFalseCarriers              Counter64,
        ifMauPCSCodingViolations          Counter64,
        ifMauFECAbility                   INTEGER,
        ifMauFECMode                      INTEGER,
        ifMauFECCorrectedBlocks           Counter64,
        ifMauFECUnCorrectableBlocks       Counter64,
        ifMauSNROpMarginChnlA             Integer32,
        ifMauSNROpMarginChnlB             Integer32,
        ifMauSNROpMarginChnlC             Integer32,
        ifMauSNROpMarginChnlD             Integer32
    }

ifMauIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "This variable uniquely identifies the interface
                to which the MAU described by this entry is
                connected."
    REFERENCE   "RFC 2863, ifIndex"
    ::= { ifMauEntry 1 }

ifMauIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "This variable uniquely identifies the MAU
                described by this entry from among other MAUs
                connected to the same interface (ifMauIfIndex)."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.1, aMAUID."
    ::= { ifMauEntry 2 }

ifMauType OBJECT-TYPE
  SYNTAX      AutonomousType
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION "This object identifies the MAU type. Values for
              standard IEEE 802.3 MAU types are defined in the
              IANA maintained IANA-MAU-MIB module, as
              OBJECT-IDENTITIES of dot3MauType.
              If the MAU type is unknown, the object identifier
              zeroDotZero is returned.

              This object represents the operational type of
              the MAU, as determined by either 1) the result
              of the Auto-Negotiation function or 2) if
              Auto-Negotiation is not enabled or is not
              implemented for this MAU, by the value of the
              object ifMauDefaultType. In case 2), a set to
              the object ifMauDefaultType will force the MAU
              into the new operating mode."
  REFERENCE   "IEEE Std 802.3, 30.5.1.1.2, aMAUType."
  ::= { ifMauEntry 3 }

ifMauStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),
                    unknown(2),
                    operational(3),
                    standby(4),
                    shutdown(5),
                    reset(6)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The current state of the MAU. This object may
                be implemented as a read-only object by those
                agents and MAUs that do not implement software
                control of the MAU state. Some agents may not
                support setting the value of this object to some
                of the enumerated values.

                The value other(1) is returned if the MAU is in
```

```
                a state other than one of the states 2 through
                6.

                The value unknown(2) is returned when the MAU’s
                true state is unknown; for example, when it is
                being initialized.

                A MAU in the operational(3) state is fully
                functional; it operates, and passes signals to its
                attached DTE or repeater port in accordance to
                its specification.

                A MAU in standby(4) state forces DI and CI to
                idle and the media transmitter to idle or fault,
                if supported. Standby(4) mode only applies to
                link type MAUs. The state of
                ifMauMediaAvailable is unaffected.

                A MAU in shutdown(5) state assumes the same
                condition on DI, CI, and the media transmitter,
                as though it were powered down or not connected.
                The MAU may return other(1) value for the
                ifMauJabberState and ifMauMediaAvailable objects
                when it is in this state. For an AUI, this
                state will remove power from the AUI.

                Setting this variable to the value reset(6)
                resets the MAU in the same manner as a
                power-off, power-on cycle of at least one-half
                second would. The agent is not required to
                return the value reset(6).

                Setting this variable to the value
                operational(3), standby(4), or shutdown(5)
                causes the MAU to assume the respective state,
                except that setting a mixing-type MAU or an AUI
                to standby(4) will cause the MAU to enter the
                shutdown state."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.7, aMAUAdminState,
                30.5.1.2.2, acMAUAdminControl, and 30.5.1.2.1,
                acResetMAU."
    ::= { ifMauEntry 4 }

ifMauMediaAvailable OBJECT-TYPE
    SYNTAX      IANAifMauMediaAvailable
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "This object identifies Media Available state of
                the MAU, complementary to the ifMauStatus. Values
                for the standard IEEE 802.3 Media Available states
                are defined in the IANA maintained IANA-MAU-MIB
                module, as IANAifMauMediaAvailable TC."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.4, aMediaAvailable."
    ::= { ifMauEntry 5 }

ifMauMediaAvailableStateExits OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
```

```
    DESCRIPTION "A count of the number of times that
                ifMauMediaAvailable for this MAU instance leaves
                the state available(3).

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system and at other times, as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.5, aLoseMediaCounter.
                RFC 2863, ifCounterDiscontinuityTime."
    ::= { ifMauEntry 6 }

ifMauJabberState OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),
                    unknown(2),
                    noJabber(3),
                    jabbering(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The value other(1) is returned if the jabber
                state is not 2, 3, or 4. The agent shall always
                return other(1) for MAU type dot3MauTypeAUI.

                The value unknown(2) is returned when the MAU's
                true state is unknown; for example, when it is
                being initialized.

                If the MAU is not jabbering the agent returns
                noJabber(3). This is the 'normal' state.

                If the MAU is in jabber state the agent returns
                the jabbering(4) value."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.6, aJabber.jabberFlag."
    ::= { ifMauEntry 7 }

ifMauJabberingStateEnters OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of the number of times that
                mauJabberState for this MAU instance enters the
                state jabbering(4). This counter will always
                indicate zero for MAUs of type dot3MauTypeAUI
                and those of speeds above 10 Mb/s.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system and at other times, as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.6, aJabber.jabberCounter.
                RFC 2863, ifCounterDiscontinuityTime."
    ::= { ifMauEntry 8 }

ifMauFalseCarriers OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
```



```
    DESCRIPTION "A count of the number of false carrier events
                during IDLE in 100BASE-X and 1000BASE-X links.

                For all other MAU types, this counter will
                always indicate zero. This counter does not
                increment at the symbol rate.

                It can increment after a valid carrier
                completion at a maximum rate of once per 100 ms
                for 100BASE-X and once per 10us for 1000BASE-X
                until the next CarrierEvent.

                This counter can roll over very quickly. A
                management station is advised to poll the
                ifMauHCFalseCarriers instead of this counter in
                order to avoid loss of information.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system and at other times, as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.10, aFalseCarriers.
                RFC 2863, ifCounterDiscontinuityTime."
    ::= { ifMauEntry 9 }

ifMauDefaultType OBJECT-TYPE
    SYNTAX      AutonomousType
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "This object identifies the default
                administrative baseband MAU type to be used in
                conjunction with the operational MAU type
                denoted by ifMauType.

                The set of possible values for this object is
                the same as the set defined for the ifMauType
                object.

                This object represents the
                administratively-configured type of the MAU. If
                Auto-Negotiation is not enabled or is not
                implemented for this MAU, the value of this
                object determines the operational type of the
                MAU. In this case, a set to this object will
                force the MAU into the specified operating mode.

                If Auto-Negotiation is implemented and enabled
                for this MAU, the operational type of the MAU
                is determined by Auto-Negotiation, and the value
                of this object denotes the type to which the MAU
                will automatically revert if/when
                Auto-Negotiation is later disabled.

                It may be necessary to provide for underlying hardware
                implementations which do not follow the exact behavior
                specified above.
                In particular, when ifMauAutoNegAdminStatus transitions
                from enabled to disabled, the agent implementation shall
                ensure that the operational type of the MAU
```

```
                (as reported by ifMauType) correctly transitions to
                the value specified by this object, rather than
                continuing to operate at the value earlier
                determined by the Auto-Negotiation function."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.1, aMAUID, and 22.2.4.1.4."
    ::= { ifMauEntry 10 }

ifMauAutoNegSupported OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "This object indicates whether or not
                Auto-Negotiation is supported on this MAU."
    ::= { ifMauEntry 11 }

ifMauTypeListBits OBJECT-TYPE
    SYNTAX      IANAifMauTypeListBits
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A value that uniquely identifies the set of
                possible IEEE 802.3 types that the MAU could be.
                If Auto-Negotiation is present on this MAU, this
                object will map to ifMauAutoNegCapabilityBits.

                Note that this MAU may be capable of operating
                as a MAU type that is beyond the scope of this
                MIB. This is indicated by returning the
                bit value bOther in addition to any bit values
                for standard capabilities that are listed in the
                IANAifMauTypeListBits TC."
    ::= { ifMauEntry 12 }

ifMauHCFalseCarriers OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A count of the number of false carrier events
                during IDLE in 100BASE-X and 1000BASE-X links.

                For all other MAU types, this counter will
                always indicate zero. This counter does not
                increment at the symbol rate.

                This counter is a 64-bit version of
                ifMauFalseCarriers. Since the 32-bit version of
                this counter can roll over very quickly,
                management stations are advised to poll the
                64-bit version instead, in order to avoid loss
                of information.

                Discontinuities in the value of this counter can
                occur at re-initialization of the management
                system and at other times, as indicated by the
                value of ifCounterDiscontinuityTime."
    REFERENCE   "IEEE Std 802.3, 30.5.1.1.10, aFalseCarriers.
                RFC 2863, ifCounterDiscontinuityTime."
    ::= { ifMauEntry 13 }

ifMauPCSCodingViolations OBJECT-TYPE
```



```
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Generalized nonresettable counter. This counter
                 has a maximum increment rate of 25 000 000
                 counts per second for 100 Mb/s implementations and
                 125 000 000 counts per second for 1000 Mb/s
                 implementations.

                 For 100 Mb/s operation it is a count of the number
                 of events that cause the PHY to indicate 'Data
                 reception with errors' on the MII (see IEEE Std 802.3
                 Table 22-2).

                 For 1000 Mb/s operation it is a count of the
                 number of events that cause the PHY to indicate 'Data
                 reception error' or 'Carrier Extend Error' on the GMII
                 (see IEEE Std 802.3 Table 35-2). The contents of this
                 attribute is undefined when FEC is operating."
      REFERENCE "IEEE Std 802.3, 30.5.1.1.12 aPCSCodingViolations."
      ::= {ifMauEntry 14}

ifMauFECAbility OBJECT-TYPE
    SYNTAX     INTEGER {
                     unknown(1),
                     supported(2),
                     notsupported(3)
                 }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A read-only value that indicates if the
                PHY supports an optional FEC sublayer for
                forward error correction (see IEEE Std 802.3 65.2
                and Clause 74).

                If an IEEE Std 802.3 Clause 45 MDIO Interface to the
                PCS is present, then this attribute will map to the
                FEC capability register (see 45.2.8.2)."
    REFERENCE   "IEEE Std 802.3. 30.5.1.1.13 aFECAbility."
    ::= {ifMauEntry 15}

ifMauFECMode OBJECT-TYPE
    SYNTAX     INTEGER {
                     unknown(1),
                     disabled(2),
                     enabled(3)
                 }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "A read-write value that indicates the mode of
                 operation of the optional FEC sublayer for forward
                 error correction (see 65.2 and Clause 74).

                 A GET operation returns the current mode of operation
                 of the PHY. A SET operation changes the mode of
                 operation of the PHY to the indicated value. When
                 IEEE Std 802.3 Clause 73 Auto-Negotiation is enabled
                 a SET operation is not allowed and a GET operation maps
                 to the variable FEC enabled in Clause 74.
```

```
                 If an IEEE Std 802.3 Clause 45 MDIO Interface to the
                 PCS is present, then this object will map to the
                 FEC controlregister (see 45.2.7.3) for 1000BASE-PX or
                 FEC enable bit in 10GBASE-R FEC control register
                 (see 45.2.1.85)."
    REFERENCE   "IEEE Std 802.3. 30.5.1.1.14 aFECMode."
    ::= {ifMauEntry 16}

ifMauFECCorrectedBlocks OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Generalized nonresettable counter. This counter
                 has a maximum increment rate of 1 200 000
                 counts per second for 1000 Mb/s implementations,
                 and 5 000 000 counts per second for 10 Gb/s
                 implementations.

                 For 1000BASE-PX PHYs or 10GBASE-R PHYs, a count
                 of corrected FEC blocks. This counter will not
                 increment for other PHY types.
                 Increment the counter by one for each received block
                 that is corrected by the FEC function in the PHY.
                 If a Clause 45 MDIO Interface to the PCS is present,
                 then this object will map to the FEC corrected blocks
                 counter (see IEEE Std 802.3 45.2.7.5 and 45.2.1.86)"
    REFERENCE   "IEEE Std 802.3. 30.5.1.1.15 aFECCorrectedBlocks."
    ::= {ifMauEntry 17}

ifMauFECUnCorrectableBlocks OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Generalized nonresettable counter. This counter
                 has a maximum increment rate of 1 200 000
                 counts per second for 1000 Mb/s implementations,
                 and 5 000 000 counts per second for 10 Gb/s
                 implementations.

                 For 1000BASE-PX PHYs or 10GBASE-R PHYs, a count
                 of uncorrectable FEC blocks. This counter will not
                 increment for other PHY types.
                 Increment the counter by one for each received block
                 that is determined to be uncorrectable by the FEC
                 function in the PHY.

                 If a Clause 45 MDIO Interface to the PCS is present,
                 then this object will map to the FEC uncorrectable
                 blocks counter (see IEEE Std 802.3 45.2.7.6 and
                 45.2.1.87)"
    REFERENCE   "IEEE Std 802.3. 30.5.1.1.16 aFECUnCorrectableBlocks."
    ::= {ifMauEntry 18}

ifMauSNROpMarginChnlA OBJECT-TYPE
    SYNTAX      Integer32 (-127..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current SNR operating margin measured at the
```



```
                 slicer input for channel A for the 10GBASE-T PMA.
                 It is reported in units of 0.1 dB to an accuracy of
                 0.5 dB within the range of -12.7 dB to 12.7 dB.
                 If an IEEE Std 802.3 Clause 45 MDIO Interface to the
                 PMA/PMD is present, then this attribute maps to the SNR
                 operating margin channel A register (see 45.2.1.63)."
    REFERENCE   "IEEE Std 802.3. 30.5.1.1.17 aSNROpMarginChnlA."
    ::= {ifMauEntry 19}

ifMauSNROpMarginChnlB OBJECT-TYPE
    SYNTAX      Integer32 (-127..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current SNR operating margin measured at the
                 slicer input for channel B for the 10GBASE-T PMA.
                 It is reported in units of 0.1 dB to an accuracy of
                 0.5 dB within the range of -12.7 dB to 12.7 dB.
                 If an IEEE Std 802.3 Clause 45 MDIO Interface to the
                 PMA/PMD is present, then this attribute maps to the SNR
                 operating margin channel B register (see 45.2.1.64)."
    REFERENCE   "IEEE Std 802.3. 30.5.1.1.18 aSNROpMarginChnlB."
    ::= {ifMauEntry 20}

ifMauSNROpMarginChnlC OBJECT-TYPE
    SYNTAX      Integer32 (-127..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current SNR operating margin measured at the
                 slicer input for channel C for the 10GBASE-T PMA.
                 It is reported in units of 0.1 dB to an accuracy of
                 0.5 dB within the range of -12.7 dB to 12.7 dB.
                 If an IEEE Std 802.3 Clause 45 MDIO Interface to the
                 PMA/PMD is present, then this attribute maps to the SNR
                 operating margin channel C register (see 45.2.1.65)."
    REFERENCE   "IEEE Std 802.3. 30.5.1.1.19 aSNROpMarginChnlC."
    ::= {ifMauEntry 21}

ifMauSNROpMarginChnlD OBJECT-TYPE
    SYNTAX      Integer32 (-127..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current SNR operating margin measured at the
                 slicer input for channel D for the 10GBASE-T PMA.
                 It is reported in units of 0.1 dB to an accuracy of
                 0.5 dB within the range of -12.7 dB to 12.7 dB.
                 If an IEEE Std 802.3 Clause 45 MDIO Interface to the
                 PMA/PMD is present, then this attribute maps to the SNR
                 operating margin channel D register (see 45.2.1.66)."
    REFERENCE   "IEEE Std 802.3. 30.5.1.1.20 aSNROpMarginChnlD."
    ::= {ifMauEntry 22}

-- The ifJackTable applies to MAUs attached to interfaces
-- which have one or more external jacks (connectors).

ifJackTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IfJackEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Information about the external jacks attached
```

```
                to MAUs attached to an interface."
    ::= { dot3IfMauBasicGroup 2 }

ifJackEntry OBJECT-TYPE
    SYNTAX      IfJackEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the table, containing information
                about a particular jack."
    INDEX       { ifMauIfIndex,
                  ifMauIndex,
                  ifJackIndex
                }
    ::= { ifJackTable 1 }

IfJackEntry ::=
    SEQUENCE {
        ifJackIndex                         Integer32,
        ifJackType                          IANAifJackType
    }

ifJackIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "This variable uniquely identifies the jack
                described by this entry from among other jacks
                attached to the same MAU."
    ::= { ifJackEntry 1 }

ifJackType OBJECT-TYPE
    SYNTAX      IANAifJackType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The jack connector type, as it appears on the
                outside of the system."
    ::= { ifJackEntry 2 }
--
-- The MAU Auto-Negotiation Table
--

ifMauAutoNegTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IfMauAutoNegEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Configuration and status objects for the
                Auto-Negotiation function of MAUs attached to
                interfaces.

                The ifMauAutoNegTable applies to systems in
                which Auto-Negotiation is supported on one or
                more MAUs attached to interfaces. Note that if
                Auto-Negotiation is present and enabled, the
                ifMauType object reflects the result of the
                Auto-Negotiation function."
    ::= { dot3IfMauAutoNegGroup 1 }

ifMauAutoNegEntry OBJECT-TYPE
    SYNTAX      IfMauAutoNegEntry
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the table, containing configuration
                and status information for the Auto-Negotiation
                function of a particular MAU."
    INDEX       { ifMauIfIndex,
                  ifMauIndex
                }
    ::= { ifMauAutoNegTable 1 }

IfMauAutoNegEntry ::=
    SEQUENCE {
        ifMauAutoNegAdminStatus           INTEGER,
        ifMauAutoNegRemoteSignaling       INTEGER,
        ifMauAutoNegConfig                INTEGER,
        ifMauAutoNegRestart               INTEGER,
        ifMauAutoNegCapabilityBits        IANAifMauAutoNegCapBits,
        ifMauAutoNegCapAdvertisedBits     IANAifMauAutoNegCapBits,
        ifMauAutoNegCapReceivedBits       IANAifMauAutoNegCapBits,
        ifMauAutoNegRemoteFaultAdvertised INTEGER,
        ifMauAutoNegRemoteFaultReceived   INTEGER
    }
ifMauAutoNegAdminStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    enabled(1),
                    disabled(2)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Setting this object to enabled(1) will cause
                the interface that has the Auto-Negotiation
                signaling ability to be enabled.

                If the value of this object is disabled(2) then
                the interface will act as it would if it had no
                Auto-Negotiation signaling. Under these
                conditions, an IEEE 802.3 MAU will immediately
                be forced to the state indicated by the value of
                the object ifMauDefaultType.

                When ifMauAutoNegAdminStatus transitions from enabled
                to disabled, the agent implementation shall
                ensure that the operational type of the MAU (as
                reported by ifMauType) correctly transitions to
                the value specified by the ifMauDefaultType
                object, rather than continuing to operate at the
                value earlier determined by the Auto-Negotiation
                function."
    REFERENCE   "IEEE Std 802.3, 30.6.1.1.2, aAutoNegAdminState,
                and 30.6.1.2.2, acAutoNegAdminControl."
    ::= { ifMauAutoNegEntry 1 }

ifMauAutoNegRemoteSignaling OBJECT-TYPE
    SYNTAX      INTEGER {
                    detected(1),
                    notdetected(2)
                }
    MAX-ACCESS  read-only
    STATUS      current
```



```
    DESCRIPTION "A value indicating whether the remote end of
                the link is using Auto-Negotiation signaling. It
                takes the value detected(1) if and only if,
                during the previous link negotiation, FLP Bursts
                were received."
    REFERENCE   "IEEE Std 802.3, 30.6.1.1.3,
                aAutoNegRemoteSignaling."
    ::= { ifMauAutoNegEntry 2 }

ifMauAutoNegConfig OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),
                    configuring(2),
                    complete(3),
                    disabled(4),
                    parallelDetectFail(5)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A value indicating the current status of the
                Auto-Negotiation process. The enumeration
                parallelDetectFail(5) maps to a failure in
                parallel detection as defined in 28.2.3.1 of
                IEEE Std 802.3."
    REFERENCE   "IEEE Std 802.3, 30.6.1.1.4, aAutoNegAutoConfig."
    ::= { ifMauAutoNegEntry 4 }

ifMauAutoNegRestart OBJECT-TYPE
    SYNTAX      INTEGER {
                    restart(1),
                    norestart(2)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "If the value of this object is set to
                restart(1) then this will force Auto-Negotiation
                to begin link renegotiation. If Auto-Negotiation
                signaling is disabled, a write to this object
                has no effect.
                Setting the value of this object to norestart(2)
                has no effect."
    REFERENCE   "IEEE Std 802.3, 30.6.1.2.1,
                acAutoNegRestartAutoConfig."
    ::= { ifMauAutoNegEntry 5 }

ifMauAutoNegCapabilityBits OBJECT-TYPE
    SYNTAX      IANAifMauAutoNegCapBits
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A value that uniquely identifies the set of
                capabilities of the local Auto-Negotiation
                entity. Note that interfaces that support this
                MIB may have capabilities that extend beyond the
                scope of this MIB.

                Note that the local Auto-Negotiation entity may
                support some capabilities beyond the scope of
                this MIB. This is indicated by returning the
                bit value bOther in addition to any bit values
```

```
                for standard capabilities that are listed in the
                IANAifMauAutoNegCapBits TC."
    REFERENCE   "IEEE Std 802.3, 30.6.1.1.5,
                aAutoNegLocalTechnologyAbility."
    ::= { ifMauAutoNegEntry 6 }

ifMauAutoNegCapAdvertisedBits OBJECT-TYPE
    SYNTAX      IANAifMauAutoNegCapBits
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "A value that uniquely identifies the set of
                capabilities advertised by the local
                Auto-Negotiation entity.

                Capabilities in this object that are not
                available in ifMauAutoNegCapabilityBits cannot
                be enabled.

                Note that the local Auto-Negotiation entity may
                advertise some capabilities beyond the scope of
                this MIB. This is indicated by returning the
                bit value bOther in addition to any bit values
                for standard capabilities that are listed in the
                IANAifMauAutoNegCapBits TC."
    REFERENCE   "IEEE Std 802.3, 30.6.1.1.6,
                aAutoNegAdvertisedTechnologyAbility."
    ::= { ifMauAutoNegEntry 7 }

ifMauAutoNegCapReceivedBits OBJECT-TYPE
    SYNTAX      IANAifMauAutoNegCapBits
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A value that uniquely identifies the set of
                capabilities received from the remote
                Auto-Negotiation entity.
                Note that interfaces that support this MIB may
                be attached to remote Auto-Negotiation entities
                that have capabilities beyond the scope of this
                MIB. This is indicated by returning the bit
                value bOther in addition to any bit values for
                standard capabilities that are listed in the
                IANAifMauAutoNegCapBits TC."
    REFERENCE   "IEEE Std 802.3, 30.6.1.1.7,
                aAutoNegReceivedTechnologyAbility."
    ::= { ifMauAutoNegEntry 8 }

ifMauAutoNegRemoteFaultAdvertised OBJECT-TYPE
    SYNTAX      INTEGER {
                    noError(1),
                    offline(2),
                    linkFailure(3),
                    autoNegError(4)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "A value that identifies any local fault
                indications that this MAU has detected and will
                advertise at the next Auto-Negotiation
                interaction for 1000 Mb/s MAUs."
```

```
    REFERENCE   "IEEE Std 802.3, 30.6.1.1.6,
                aAutoNegAdvertisedTechnologyAbility."
    ::= { ifMauAutoNegEntry 9 }

ifMauAutoNegRemoteFaultReceived OBJECT-TYPE
    SYNTAX      INTEGER {
                    noError(1),
                    offline(2),
                    linkFailure(3),
                    autoNegError(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A value that identifies any fault indications
                received from the far end of a link by the
                local Auto-Negotiation entity for 1000 Mb/s
                MAUs."
    REFERENCE   "IEEE Std 802.3, 30.6.1.1.7,
                aAutoNegReceivedTechnologyAbility."
    ::= { ifMauAutoNegEntry 10 }

-- Placeholder to preserve module structure and assignments
dot3Placeholder OBJECT-TYPE
   SYNTAX       INTEGER {
                    placeholder(1)
                }
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION  "A placeholder object to preserve the assignments
                 that follow in the module. The assignment was given
                 to the object broadMauBasicTable in earlier
                 versions of this module. Preserving the assignments that
                 follow is considered important because they are used for
                 the IANA-MAU-MIB to assign as MAU type values."
   REFERENCE    "none"
   ::= { dot3PlaceholderGroup 1 }

-- Notifications for use by 802.3 MAUs

snmpDot3MauTraps OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 0 }

rpMauJabberTrap NOTIFICATION-TYPE
    OBJECTS     { rpMauJabberState }
    STATUS      current
    DESCRIPTION "This trap is sent whenever a managed repeater
                MAU enters the jabber state.

                The agent shall limit the generation of
                consecutive rpMauJabberTraps so that there is at
                least a five-second gap between them."
    REFERENCE   "IEEE Std 802.3, 30.5.1.3.1, nJabber notification."
    ::= { snmpDot3MauTraps 1 }

ifMauJabberTrap NOTIFICATION-TYPE
    OBJECTS     { ifMauJabberState }
    STATUS      current
    DESCRIPTION "This trap is sent whenever a managed interface
                MAU enters the jabber state.
```

```
                  The agent shall limit the generation of
                  consecutive ifMauJabberTraps so that there is at
                  least a five-second gap between them."

    REFERENCE     "IEEE Std 802.3, 30.5.1.3.1, nJabber notification."
    ::= { snmpDot3MauTraps 2 }

-- Conformance statements

mauModConf
        OBJECT IDENTIFIER ::= { ieee8023mauMIB 2 }
  mauModCompls
        OBJECT IDENTIFIER ::= { mauModConf 1 }
  mauModObjGrps
        OBJECT IDENTIFIER ::= { mauModConf 2 }
  mauModNotGrps
        OBJECT IDENTIFIER ::= { mauModConf 3 }

-- Object groups
mauRpGrpBasic OBJECT-GROUP
    OBJECTS       { rpMauType,
                    rpMauStatus,
                    rpMauMediaAvailable,
                    rpMauMediaAvailableStateExits,
                    rpMauJabberState,
                    rpMauJabberingStateEnters
                  }
    STATUS        current
    DESCRIPTION "Basic conformance group for MAUs attached to
                  repeater ports. This group is also the
                  conformance specification for RFC 1515
                  implementations."
    ::= { mauModObjGrps 1 }

mauRpGrp100Mbs OBJECT-GROUP
    OBJECTS       { rpMauFalseCarriers }
    STATUS        current
    DESCRIPTION "Conformance group for MAUs attached to
                  repeater ports with 100 Mb/s or greater
                  capability."
    ::= { mauModObjGrps 2 }

mauRpGrpJack OBJECT-GROUP
    OBJECTS       { rpJackType }
    STATUS        current
    DESCRIPTION "Conformance group for MAUs attached to
                  repeater ports with managed jacks."
    ::= { mauModObjGrps 3 }

mauIfGrpBasic OBJECT-GROUP
    OBJECTS       { ifMauType,
                    ifMauStatus,
                    ifMauMediaAvailable,
                    ifMauMediaAvailableStateExits,
                    ifMauJabberState,
                    ifMauJabberingStateEnters,
                    dot3Placeholder
                  }
    STATUS        current
```

```
        DESCRIPTION "Basic conformance group for MAUs attached to
                    interfaces. This group also provides a
                    conformance specification for RFC 1515
                    implementations."
        ::= { mauModObjGrps 4 }

    mauIfGrpJack OBJECT-GROUP
        OBJECTS     { ifJackType }
        STATUS      current
        DESCRIPTION "Conformance group for MAUs attached to
                    interfaces with managed jacks."
        ::= { mauModObjGrps 5 }

    mauIfGrpHighCapacity OBJECT-GROUP
        OBJECTS     { ifMauFalseCarriers,
                      ifMauTypeListBits,
                      ifMauDefaultType,
                      ifMauAutoNegSupported
                    }
        STATUS      current
        DESCRIPTION "Conformance group for MAUs attached to
                    interfaces with 100 Mb/s or greater capability."
        ::= { mauModObjGrps 6 }

    mauIfGrpAutoNeg2 OBJECT-GROUP
        OBJECTS     { ifMauAutoNegAdminStatus,
                      ifMauAutoNegRemoteSignaling,
                      ifMauAutoNegConfig,
                      ifMauAutoNegCapabilityBits,
                      ifMauAutoNegCapAdvertisedBits,
                      ifMauAutoNegCapReceivedBits,
                      ifMauAutoNegRestart
                    }
        STATUS      current
        DESCRIPTION "Conformance group for MAUs attached to
                    interfaces with managed Auto-Negotiation."
        ::= { mauModObjGrps 7 }

    mauIfGrpAutoNeg1000Mbps OBJECT-GROUP
        OBJECTS     { ifMauAutoNegRemoteFaultAdvertised,
                      ifMauAutoNegRemoteFaultReceived
                    }
        STATUS      current
        DESCRIPTION "Conformance group for 1000 Mb/s MAUs attached to
                    interfaces with managed Auto-Negotiation."
        ::= { mauModObjGrps 8 }

    mauIfGrpHCStats OBJECT-GROUP
        OBJECTS     { ifMauHCFalseCarriers,
                      ifMauPCSCodingViolations
                    }
        STATUS      current
        DESCRIPTION "Conformance for high capacity statistics for
                    MAUs attached to interfaces."
        ::= { mauModObjGrps 9 }

   mauIfGrpFEC OBJECT-GROUP
        OBJECTS     { ifMauFECAbility,
                      ifMauFECMode,
```

```
                    ifMauFECCorrectedBlocks,
                    ifMauFECUnCorrectableBlocks
                  }
        STATUS      current
        DESCRIPTION "Conformance for FEC capable
                    MAUs attached to interfaces."
        ::= { mauModObjGrps 10 }

    mauIfGrpSNR OBJECT-GROUP
        OBJECTS     { ifMauSNROpMarginChnlA,
                      ifMauSNROpMarginChnlB,
                      ifMauSNROpMarginChnlC,
                      ifMauSNROpMarginChnlD
                    }
        STATUS      current
        DESCRIPTION "Conformance for SNR operating margin reporting
                    MAUs attached to interfaces."
        ::= { mauModObjGrps 11 }

    -- Notification groups

    rpMauNotifications NOTIFICATION-GROUP
        NOTIFICATIONS { rpMauJabberTrap }
        STATUS      current
        DESCRIPTION "Notifications for repeater MAUs."
        ::= { mauModNotGrps 1 }

    ifMauNotifications NOTIFICATION-GROUP
        NOTIFICATIONS { ifMauJabberTrap }
        STATUS      current
        DESCRIPTION "Notifications for interface MAUs."
        ::= { mauModNotGrps 2 }

    -- Compliance statements

    mauModRpCompl2 MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION "Compliance for MAUs attached to repeater
                    ports.

                    Note that compliance with this compliance
                    statement requires compliance with the
                    snmpRptrModCompl MODULE-COMPLIANCE statement of
                    the IEEE8023-SNMP-REPEATER-MIB defined in Clause 7."

        MODULE -- this module
            MANDATORY-GROUPS { mauRpGrpBasic }

            GROUP       mauRpGrp100Mbs
            DESCRIPTION "Implementation of this optional group is
                        recommended for MAUs that have 100 Mb/s or
                        greater capability."

            GROUP       mauRpGrpJack
            DESCRIPTION "Implementation of this optional group is
                        recommended for MAUs that have one or more
                        external jacks."

            GROUP       rpMauNotifications
```

```
        DESCRIPTION "Implementation of this group is recommended
                    for MAUs attached to repeater ports."

        OBJECT      rpMauStatus
        MIN-ACCESS  read-only
        DESCRIPTION "Write access is not required."
    ::= { mauModCompls 1 }

mauModIfCompl3 MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION "Compliance for MAUs attached to interfaces.

                Note that compliance with this compliance
                statement requires compliance with the
                ifCompliance3 MODULE-COMPLIANCE statement of the
                IF-MIB (RFC 2863) and the dot3Compliance2
                MODULE-COMPLIANCE statement of the
                IEEE8023-EtherLike-MIB defined in Clause 10."

    MODULE -- this module
        MANDATORY-GROUPS { mauIfGrpBasic }

        GROUP       mauIfGrpHighCapacity
        DESCRIPTION "Implementation of this optional group is
                    recommended for MAUs that have 100 Mb/s
                    or greater capability."

        GROUP       mauIfGrpHCStats
        DESCRIPTION "Implementation of this group is mandatory
                    for MAUs that have 1000 Mb/s capacity, and
                    is recommended for MAUs that have 100 Mb/s
                    capacity."

        GROUP       mauIfGrpJack
        DESCRIPTION "Implementation of this optional group is
                    recommended for MAUs that have one or more
                    external jacks."
        GROUP       mauIfGrpAutoNeg2
        DESCRIPTION "Implementation of this group is mandatory
                    for MAUs that support managed
                    Auto-Negotiation."
        GROUP       mauIfGrpAutoNeg1000Mbps
        DESCRIPTION "Implementation of this group is mandatory
                    for MAUs that have 1000 Mb/s or greater
                    capability and support managed
                    Auto-Negotiation."

        GROUP       ifMauNotifications
        DESCRIPTION "Implementation of this group is recommended
                    for MAUs attached to interfaces."

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

        GROUP       mauIfGrpFEC
        DESCRIPTION "Implementation of this optional group is
                     recommended for MAUs that incorporate FEC."
```

```
            GROUP       mauIfGrpSNR
            DESCRIPTION "Implementation of this optional group is
                         recommended for MAUs that report SNR operating
                         margin."

      ::= { mauModCompls 2 }

   END
```

Annex A

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] Case, J., and C. Partridge, "Case Diagrams: A First Step to Diagrammed Management Information Bases," Computer Communications Review, 19(1):13–16, January 1989.

[B2] Faulk, Jr., R. L., C. A. Black, and A. S. Fraley. 1991. Detection on a network by a mapping application of a relative location of a first device to a second device. US Patent 5,293,635, filed Apr. 30, 1991 and issued Mar. 8, 1994.

[B3] Faulk, Jr. R. L., and C. A. Black. 1993. Detection of a relative location of a network device using a multicast packet processed only by hubs. US Patent 5,421,024, filed Aug. 30, 1993 and issued May 30, 1995.

[B4] IEEE Std 802.1Q™, IEEE Standards for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks.

[B5] IEEE Std 802.9™, IEEE Standards for Local and Metropolitan Area Networks: Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers.

[B6] *IEEE Standards Dictionary: Glossary of Terms & Definitions,* January 2008.[21]

[B7] IETF RFC 1157, *Simple Network Management Protocol*, Case, J., Fedor, M., Schoffstall, M., and J. Davin, May 1990.

[B8] IETF RFC 1573, *Evolution of the Interfaces Group of MIB-II*, McCloghrie, K., and F. Kastenholz, January 1994.

[B9] IETF RFC 1212, *Concise MIB Definitions*, Rose, M. and K. McCloghrie, March 1991.

[B10] IETF RFC 1284, *Definitions of Managed Objects for Ethernet-Like Interface Types*, Cook, J., December 1991.

[B11] IETF RFC 1369, *Implementation Notes and Experience for The Internet Ethernet MIB*, Kastenholz, F., October 1992.

[B12] IETF RFC 1398, *Definitions of Managed Objects for the Ethernet-like Interface Types*, Kastenholz, F., January 1993.

[B13] IETF RFC 1515, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)*, McMaster, D., McCloghrie, K., and S. Roberts, September 1993.

[21] The IEEE Standards Dictionary: Glossary of Terms & Definitions is available at http://shop.ieee.org/.

[B14] IETF RFC 1525, *Definitions of Managed Objects for Source Routing Bridges*, Decker, E., McCloghrie, K., Langille, P., and A. Rijsinghani, September 1993.

[B15] IETF RFC 1643, *Definitions of Managed Objects for the Ethernet-like Interface Types*, Kastenholz, F., July 1994.

[B16] IETF RFC 1650, *Definitions of Managed Objects for the Ethernet-like Interface Types using SMIv2*, Kastenholz, F., August 1994.

[B17] IETF RFC 1905, *Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)*, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, January 1996.

[B18] IETF RFC 2026, *The Internet Standards Process - Revision 3*, Bradner, S., October 1996.

[B19] IETF RFC 2108, *Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIv2*, de Graaf, K, Romascanu, D., McMaster, D., and K. McCloghrie, February 1997.

[B20] IETF RFC 2239, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs) using SMIv2*, de Graaf, K., Romascanu, D., McMaster, D., McCloghrie, K., and S. Roberts, November 1997.

[B21] IETF RFC 2358, *Definitions of Managed Objects for the Ethernet-like Interface Types*, Flick, J. and J. Johnson, June 1998.

[B22] IETF RFC 2665, *Definitions of Managed Objects for the Ethernet-like Interface Types*, Flick, J. and J. Johnson, August 1999.

[B23] IETF RFC 2666, *Definitions of Object Identifiers for Identifying Ethernet Chip Sets*, Flick, J., August 1999.

[B24] IETF RFC 2668, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)*, Smith, A., Flick, J., de Graaf, K., Romascanu, D., McMaster, D., McCloghrie, K., and S. Roberts, August 1999.

[B25] IETF RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*, Presuhn, R., December 2002.

[B26] IETF RFC 3636, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)*, Flick, J., September 2003.

[B27] IETF RFC 3637, *Definitions of Managed Objects for the Ethernet WAN Interface Sublayer*, Heard, C., September 2003.

[B28] IETF RFC 4070, *Definitions of Managed Object Extensions for Very High Speed Digital Subscriber Lines (VDSL) Using Multiple Carrier Modulation (MCM) Line Coding*, Dodge, M. and B. Ray, May 2005.

[B29] IETF RFC 4181, *Guidelines for Authors and Reviewers of MIB Documents*, Heard, C., September 2005.

[B30] IETF RFC 4188, *Definitions of Managed Objects for Bridges*, Norseth, K. and E. Bell, September 2005.

[B31] IETF RFC 4319, *Definitions of Managed Objects for High Bit-Rate DSL—2nd generation (HDSL2) and Single-Pair High-Speed Digital Subscriber Line (SHDSL) Lines*, Sikes, C., Ray, B., and R. Abbi, December 2005.

[B32] IETF RFC 4837, *Managed Objects of Ethernet Passive Optical Networks (EPON)*, Khermosh, L., July 2007.

[B33] IETF RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*, Squire, M., June 2007.

[B34] IETF, "Ethernet Interfaces and Hub MIB (hubmib) Charter," <http://www.ietf.org/html.charters/OLD/hubmib-charter.html>.

[B35] Internet Assigned Numbers Authority (IANA), "IANAifType Textual Convention definition," <http://www.iana.org/assignments/ianaiftype-mib>.

[B36] ISO/IEC 8824:1990, Information technology -- Abstract Syntax Notation One (ASN.1).

[B37] ISO/IEC 10165-4:1992, Information technology—Open Systems Interconnection—Structure of management information—Part 4: Guidelines for the definition of managed objects.

[B38] ISO/IEC 10165-5:1994, Information technology—Open Systems Interconnection—Structure of management information: Generic management information.

[B39] Kastenholz, F., "IEEE 802.3 Layer Management Draft compatible MIB for TCP/IP Networks," electronic mail message to mib-wg@nnsc.nsf.net, 9 June 1989.

[B40] Network Interoperability Consultative Committee (NICC), "Specification of the Access Network Frequency Plan (ANFP) applicable to transmission systems used on the BT Access Network," NICC Document ND1602:2005/08, August 2005.

Annex B

(normative)

GDMO specification for IEEE 802.3 managed object classes

This annex formally defines the protocol encodings for CMIP and ISO/IEC 15802-2:1995 [IEEE Std 802.1B and IEEE Std 802.1k, 1995 Edition] for the IEEE 802.3 managed objects using the templates specified in ISO/IEC 10165-4:1992. The application of a GDMO template compiler against B.1 to B.18 will produce the proper protocol encodings.

NOTE 1—The arcs (that is, object identifier values) defined in this annex deprecate the arcs previously defined in Annexes H.1 (Layer Management), H.2 (Repeater Management), and H.3 (MAU Management). See IEEE Std 802.1F-1993, Annex C.4.

NOTE 2—During the update for IEEE Std 802.3-2005, differences between objects in the root of all registration arcs were harmonized and aligned with the recommendation of IEEE Std 802b-2004. All instances were changed to *{iso(1) std(0) iso8802(8802) csma(3)...* For maximum compatibility with previous implementations it is recommended that all implementations respond equally to requests for communication based on registration arc *{iso(1) member-body(2) us(840) ieee802dot3(10006)...*

NOTE 3—The arcs defined in this annex have not been updated to conform to the currently preferred root of “org ieee(111) standards-association-numbers-series-standards(2) lan-man-stds(802) ieee802dot3(3)” because the GDMO object definitions are of historical value only, and are not recommended for current or future implementations.

Each attribute definition in this clause references directly by means of the WITH ATTRIBUTE SYNTAX construct or indirectly by means of the DERIVED FROM construct an ASN.1 type or subtype that defines the attribute’s type and range. Those ASN.1 types and subtypes defined exclusively for CSMA/CD Management appear in a single ASN.1 module in C.2.

Counters for these protocol encodings are specified as either 32 or 64 bits wide. Thirty-two bit counters are used for the protocol encoding of counter attributes, providing the minimum rollover time is 58 min or more. Sixty-four bit counters are used for the protocol encoding of counter attributes that could roll over in less than 58 min with a 32-bit counter. Approximate counter rollover times are provided as notes below each counter BEHAVIOUR definition. 10 Gb/s counters are 10 times faster than 1000 Mb/s counters, 1000 Mb/s counters are 10 times faster than 100 Mb/s counters, and 100 Mb/s counters are 10 times faster than 10 Mb/s counters. For a formal definition of the counter, refer to the BEHAVIOUR bCMCounter in C.2.

B.1 DTE MAC entity managed object class

B.1.1 DTE MAC entity formal definition

```
oMACEntity                                  MANAGED OBJECT CLASS

    DERIVED FROM            'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992':top;
    CHARACTERIZED BY
            pBasic          PACKAGE
                    ATTRIBUTES   aMACID                          GET;
                    ACTIONS      acInitializeMAC;
            ;
    ;
    CONDITIONAL PACKAGES
```

```
        pMandatory      PACKAGE
                ATTRIBUTES      aFramesTransmittedOK            GET,
                                aSingleCollisionFrames          GET,
                                aMultipleCollisionFrames        GET,
                                aFramesReceivedOK               GET,
                                aFrameCheckSequenceErrors       GET,
                                aAlignmentErrors                GET,
                                aMACCapabilities                GET,
                                aDuplexStatus                   GET-REPLACE,
                                aRateControlAbility             GET,
                                aRateControlStatus              GET-REPLACE,
                                aDeferControlAbility            GET,
                                aDeferControlStatus             GET-REPLACE;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) macMandato-
                ryPkg(1)};
PRESENT IF      Conformance to DTE Management is desired. Attributes aMACCapabilities and
                aDuplexStatus are mandatory in systems that can operate in full duplex mode and
                are recommended in systems that can only operate in half duplex mode.,
        pRecommended    PACKAGE
                ATTRIBUTES      aOctetsTransmittedOK            GET,
                                aFramesWithDeferredXmissions    GET,
                                aLateCollisions                 GET,
                                aFramesAbortedDueToXSColls      GET,
                                aFramesLostDueToIntMACXmitError GET,
                                aCarrierSenseErrors             GET,
                                aOctetsReceivedOK               GET,
                                aFramesLostDueToIntMACRcvError  GET,
                                aPromiscuousStatus              GET-REPLACE,
                                aReadMulticastAddressList       GET;
                ACTIONS         acAddGroupAddress,
                                acDeleteGroupAddress;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) macRecom-
                mendedPkg(2)};
PRESENT IF      The Recommended Package is implemented.,
        pOptional               PACKAGE
                ATTRIBUTES      aMulticastFramesXmittedOK       GET,
                                aBroadcastFramesXmittedOK       GET,
                                aMulticastFramesReceivedOK      GET,
                                aBroadcastFramesReceivedOK      GET,
                                aInRangeLengthErrors            GET,
                                aOutOfRangeLengthField          GET,
                                aFrameTooLongErrors             GET,
                                aMaxFrameLength                 GET,
                                aMACEnableStatus                GET-REPLACE,
                                aTransmitEnableStatus           GET-REPLACE,
                                aMulticastReceiveStatus         GET-REPLACE,
                                aReadWriteMACAddress            GET-REPLACE;
                ACTIONS         acExecuteSelfTest;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) option-
                alPkg(3)};
PRESENT IF      The Optional Package and the Recommended Package are implemented.,
        pArray                  PACKAGE
                ATTRIBUTES      aCollisionFrames                GET;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) arrayPkg(4)};
```


PRESENT IF The Array Package and the Recommended Package are implemented.,
pExcessiveDeferral PACKAGE
ATTRIBUTES aFramesWithExcessiveDeferral GET;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) excessiveDeferralPkg(5)};
PRESENT IF The ExcessiveDeferral Package and the Recommended Package are implemented.;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) managedObjectClass(3) macObjectClass(1)};

nbMACName **NAME BINDING**

SUBORDINATE OBJECT CLASS oMACEntity;
NAMED BY SUPERIOR OBJECT CLASS
“ISO/IEC 10165-2”:system;
WITH ATTRIBUTE aMACID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) macName(1)};

nbMACMonitor **NAME BINDING**

SUBORDINATE OBJECT CLASS “IEEE802.1F”:oEWMAMetricMonitor;
NAMED BY SUPERIOR OBJECT CLASS
“ISO/IEC 10165-2”:system;
WITH ATTRIBUTE “IEEE802.1F”:aScannerId;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) macMonitor(2)};

nbMAC-MACControl **NAME BINDING**

SUBORDINATE OBJECT CLASS oMACEntity;
NAMED BY SUPERIOR OBJECT CLASS oMACControlEntity;
WITH ATTRIBUTE aMACID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) nbMAC-MACControl(16)};

nbMAC-Aggregator **NAME BINDING**

SUBORDINATE OBJECT CLASS oMACEntity;
NAMED BY SUPERIOR OBJECT CLASS oAggregator;
WITH ATTRIBUTE aMACID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) nbMAC-Aggregator(17)};

aDeferControlAbility **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bDeferControlAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) deferControlAbility(311)};

bDeferControlAbility **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.35;

aDeferControlStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.DeferControl;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bDeferControlStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) deferControlStatus(312)};

bDeferControlStatus **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.36;

B.1.2 DTE MAC entity attributes

aMACID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bMACID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) macID(1)};

bMACID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.1;

aFramesTransmittedOK **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesTransmittedOK;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesTransmittedOK(2)};

bFramesTransmittedOK **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.2

NOTE 1—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.

NOTE 2—This maps to framesSent (of the mandatory macPackage) in ISO/IEC 10742:1994.;

aSingleCollisionFrames **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bSingleCollisionFrames;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) singleCollisionFrames(3)};

bSingleCollisionFrames **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.3

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 103 h.;

aMultipleCollisionFrames **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bMultipleCollisionFrames;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) multipleCollisionFrames(4)};

bMultipleCollisionFrames **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.4

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 125 h.;

aFramesReceivedOK **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesReceivedOK;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesReceivedOK(5)};

bFramesReceivedOK **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.5

NOTE 1—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.

NOTE 2—This maps to framesReceived (of the mandatory macPackage) in ISO/IEC 10742:1994.;

aFrameCheckSequenceErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFrameCheckSequenceErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) frameCheckSequenceErrors(6)};

bFrameCheckSequenceErrors **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.6

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aAlignmentErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;

BEHAVIOUR bAlignmentErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) alignmentErrors(7)};

bAlignmentErrors **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.7

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aOctetsTransmittedOK **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bOctetsTransmittedOK;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) octetsTransmittedOK(8)};

bOctetsTransmittedOK **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.8

NOTE 1—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.

NOTE 2—This maps to octetsSent (of the mandatory macPackage) in ISO/IEC 10742:1994.;

aFramesWithDeferredXmissions **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesWithDeferredXmissions;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesWithDeferredXmissions(9)};

bFramesWithDeferredXmissions **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.9

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 103 h.;

aLateCollisions **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bLateCollisions;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) lateCollisions(10)};

bLateCollisions **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.10

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aFramesAbortedDueToXSColls **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesAbortedDueToXSColls;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesAbortedDueToXSColls(11)};

bFramesAbortedDueToXSColls **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.11

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 53 days.;

aFramesLostDueToIntMACXmitError **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesLostDueToIntMACXmitError;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesLostDueToIntMACXmitError(12)};

bFramesLostDueToIntMACXmitError **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.12

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 16 h.;

aCarrierSenseErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bCarrierSenseErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) carrierSenseErrors(13)};

bCarrierSenseErrors **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.13

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aOctetsReceivedOK **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bOctetsReceivedOK;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) octetsReceivedOK(14)};

bOctetsReceivedOK **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.14

NOTE 1—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.

NOTE 2—This maps to octetsReceived (of the mandatory macPackage) in ISO/IEC 10742:1994.;

aFramesLostDueToIntMACRcvError ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesLostDueToIntMACRcvError;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesLostDueToIntMACRcvError(15)};

bFramesLostDueToIntMACRcvError BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.15

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aPromiscuousStatus ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
BEHAVIOUR bPromiscuousStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) promiscuousStatus(16)};

bPromiscuousStatus BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.16;

aReadMulticastAddressList ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule. MulticastAddressList;
BEHAVIOUR bReadMulticastAddressList;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) readMulticastAddressList(17)};

bReadMulticastAddressList BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.17;

aMulticastFramesXmittedOK ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bMulticastFramesXmittedOK;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) multicastFramesXmittedOK(18)};

bMulticastFramesXmittedOK BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.18

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aBroadcastFramesXmittedOK **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bBroadcastFramesXmittedOK;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) broadcastFramesXmittedOK(19)};

bBroadcastFramesXmittedOK **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.19

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aFramesWithExcessiveDeferral **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesWithExcessiveDeferral;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesWithExcessiveDeferral(20)};

bFramesWithExcessiveDeferral **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.20

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 days.;

aMulticastFramesReceivedOK **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bMulticastFramesReceivedOK;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) multicastFramesReceivedOK(21)};

bMulticastFramesReceivedOK **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.21

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aBroadcastFramesReceivedOK **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bBroadcastFramesReceivedOK;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) broadcastFramesReceivedOK(22)};

bBroadcastFramesReceivedOK **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.22

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aInRangeLengthErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bInRangeLengthErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) inRangeLengthErrors(23)};

bInRangeLengthErrors **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.23

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aOutOfRangeLengthField **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bOutOfRangeLengthField;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) outOfRangeLengthField(24)};

bOutOfRangeLengthField **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.24

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aFrameTooLongErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFrameTooLongErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) frameTooLongErrors(25)};

bFrameTooLongErrors **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.25

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 61 days.;

aMACEnableStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
BEHAVIOUR bMACEnableStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mACEnableStatus(26)};

bMACEnableStatus **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.1.1.26;

aTransmitEnableStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
BEHAVIOUR bTransmitEnableStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) transmitEnableStatus(27)};

bTransmitEnableStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.27;

aMulticastReceiveStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
BEHAVIOUR bMulticastReceiveStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) multicastReceiveStatus(28)};

bMulticastReceiveStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.28;

aReadWriteMACAddress **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802CommonDefinitions.MACAddress;
BEHAVIOUR bReadWriteMACAddress;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) modifyMACAddress(29)};

bReadWriteMACAddress **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.29

NOTE—This maps to localMACAddress (of the mandatory macPackage) in ISO/IEC 10742:1994.;

aCollisionFrames **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AttemptArray;
BEHAVIOUR bCollisionFrames;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) collisionFrames(30)};

bCollisionFrames **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.30

NOTE—The approximate minimum time for any single counter rollover for 10 Mb/s operation is 103 h.;

aMACCapabilities **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MACCapabilitiesList;

MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMACCapabilities;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) aMACCapabilities(89)};

bMACCapabilities **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.1.1.31;

aMaxFrameLength **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MaxFrameLengthList;

MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMaxFrameLength;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) aMaxFrameLength(357)};

bMaxFrameLength **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.37;

aDuplexStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.DuplexValues;
MATCHES FOR EQUALITY;
BEHAVIOUR bDuplexStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) aDuplexStatus(90)};

bDuplexStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.1.32;

aRateControlAbility **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
BEHAVIOUR bRateControlAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) rateControlAbility(179)};

bRateControlAbility **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.1.1.33;

aRateControlStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.RateValues;
MATCHES FOR EQUALITY;
BEHAVIOUR bRateControlStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) rateControlStatus(180)};

bRateControlStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.1.1.34;

B.1.3 DTE MAC entity actions

acInitializeMAC **ACTION**

BEHAVIOUR bInitializeMAC;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) initializeMAC(1)};

bInitializeMAC **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.2.1;

acAddGroupAddress **ACTION**

BEHAVIOUR bAddGroupAddress;
MODE CONFIRMED;
WITH INFORMATION SYNTAX IEEE802CommonDefinitions.MACAddress;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) addGroupAddress(2)};

bAddGroupAddress **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.2.2;

acDeleteGroupAddress **ACTION**

BEHAVIOUR bDeleteGroupAddress;
MODE CONFIRMED;
WITH INFORMATION SYNTAX IEEE802CommonDefinitions.MACAddress;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) deleteGroupAddress(3)};

bDeleteGroupAddress **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.1.2.3;

acExecuteSelfTest **ACTION**

BEHAVIOUR bExecuteSelfTestMAC;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) executeSelfTestMAC(4)};

bExecuteSelfTestMAC **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.1.2.4;

B.2 DTE physical entity managed object class

B.2.1 DTE physical entity formal definition

```
oPHYEntity                              MANAGED OBJECT CLASS

    DERIVED FROM                        'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                        :1992':top;
    CHARACTERIZED BY
        pBasic                          PACKAGE
            ATTRIBUTES                  aPHYID                  GET,
                                        aPHYType                GET,
                                        aPHYTypeList            GET,
                                        aMIIDetect              GET,
                                        aPHYAdminState          GET;
        ;
    ;
    CONDITIONAL PACKAGES
        pRecommended                    PACKAGE
            ATTRIBUTES                  aSQETestErrors          GET;
            REGISTERED AS               {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4)
                                        phyRecommendedPkg(6)};
            PRESENT IF                  The Recommended Package is implemented.,
        pMultiplePhy                    PACKAGE
            ACTIONS                     acPHYAdminControl;
            REGISTERED AS               {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4)
                                        phyMultiplePhyPkg(7)};
            PRESENT IF                  There is more than one PHY per MAC.,
        p100MbpsMonitor                 PACKAGE
            ATTRIBUTES                  aSymbolErrorDuringCarrier   GET;
            REGISTERED AS               {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4)
                                        phy100MbpsMonitor(8)};
            PRESENT IF                  The 100/1000 Mb/s Monitor capability is
                                        implemented.;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                            managedObjectClass(3) phyObjectClass(2)};
```

nbPHYName **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            oPHYEntity;
    NAMED BY SUPERIOR OBJECT CLASS
                                        oMACEntity;
    WITH ATTRIBUTE                      aPHYID;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            phyName(3)};
```

nbPHYMonitor **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            "IEEE802.1F":oEWMAMetricMonitor;
```

NAMED BY SUPERIOR OBJECT CLASS
"ISO/IEC 10165-2":system;
WITH ATTRIBUTE "IEEE802.1F":aScannerId;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) phyMonitor(4)};

B.2.2 DTE physical entity attributes

aPHYID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bPHYID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) phyID(31)};

bPHYID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.2.1.1;

aPHYType **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PhyTypeValue;
MATCHES FOR EQUALITY;
BEHAVIOUR bPHYType;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) phyType(32)};

bPHYType **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.2.1.2;

aPHYTypeList **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PhyTypeList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPHYTypeList;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) phyTypeList(33)};

bPHYTypeList **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.2.1.3;

aSQETestErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bSQETestErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)

sqeTestErrors(34)};

bSQETestErrors **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.2.1.4

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aSymbolErrorDuringCarrier **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bSymbolErrorDuringCarrier;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) symbolErrorDuringCarrier(35)};

bSymbolErrorDuringCarrier **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.2.1.5

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aMIIDetect **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MIIDetect;
MATCHES FOR EQUALITY;
BEHAVIOUR bMIIDetect;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mIIDetect(36)};

bMIIDetect **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.2.1.6;

aPHYAdminState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
PortAdminState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPHYAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) phyAdminState(37)};

bPHYAdminState **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.2.1.7;

B.2.3 DTE physical entity actions

acPHYAdminControl **ACTION**

BEHAVIOUR bPHYAdminControl;

```
        MODE                                    CONFIRMED;
        WITH INFORMATION SYNTAX                 IEEE802Dot3-MgmtAttributeModule.
                                                PortAdminState;
        REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
                                phyAdminControl(5)};
```

bPHYAdminControl **BEHAVIOUR**

```
        DEFINED AS              See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.2.2.1;
```

B.3 DTE MAC control entity managed object class

B.3.1 DTE MAC control entity formal definition

oMACControlEntity **MANAGED OBJECT CLASS**

```
    DERIVED FROM                    'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :1992':top;
    CHARACTERIZED BY
        pMandatory                  PACKAGE
            ATTRIBUTES              aMACControlFunctionsSupported       GET-REPLACE;
        ;
    ;
    CONDITIONAL PACKAGES
        pRecommended                PACKAGE
            ATTRIBUTES              aMACControlFramesTransmitted        GET,
                                    aMACControlFramesReceived           GET,
                                    aUnsupportedOpcodesReceived         GET;
            REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3)
                                    csmacdmgt(30)
                                    package(4) macControlRecommendedPkg(17)};
            PRESENT IF              The Recommended Package is implemented.;
    REGISTERED AS{iso(1) std(0) iso8802(8802) csma(3)
        csmacdmgt(30)
        managedObjectClass(3) macControlObjectClass(8)};
```

nbMACControl-System **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            oMACControlEntity;
    NAMED BY SUPERIOR OBJECT CLASS      "ISO/IEC 10165-2":system;
    WITH ATTRIBUTE                      aMACControlID;
    REGISTERED AS    {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) nbMAC-
                     Control-System(18)};
```

nbMACControl-Aggregator **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            oMACControlEntity;
    NAMED BY SUPERIOR OBJECT CLASS      oAggregator;
    WITH ATTRIBUTE                      aMACControlID;
    REGISTERED AS    {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) nbMAC-
                     Control-Aggregator(19)};
```

nbMACControlMonitor **NAME BINDING**

SUBORDINATE OBJECT CLASS“IEEE802.1F”:oEWMAMetricMonitor;
NAMED BY SUPERIOR OBJECT CLASS
“ISO/IEC 10165-2”:system;
WITH ATTRIBUTE “IEEE802.1F”:aScannerId;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS{iso(1) std(0) iso8802(8802) csma(3)
csmacdmgt(30) nameBinding(6) macControlMonitor(15)};

B.3.2 DTE MAC Control entity attributes

aMACControlID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bMACControlID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3)
csmacdmgt(30) attribute(7) aMACControlID(92)};

bMACControlID **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.3.1;

aMACControlFunctionsSupported **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
MACControlFunctionsList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMACControlFunctionsSupported;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3)
csmacdmgt(30) attribute(7) aMACControlFunctionsSupported(93)};

bMACControlFunctionsSupported **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.3.2;

aMACControlFramesTransmitted **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bMACControlFramesTransmitted;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3)
csmacdmgt(30) attribute(7) aMACControlFramesTransmitted(94)};

bMACControlFramesTransmitted **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.3.3.3;

aMACControlFramesReceived **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bMACControlFramesReceived;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3)
csmacdmgt(30) attribute(7) aMACControlFramesReceived(95)};


```
bMACControlFramesReceived               BEHAVIOUR

    DEFINED AS          See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.3.4;

aUnsupportedOpcodesReceived             ATTRIBUTE

    DERIVED FROM                        aCMCounter;
    BEHAVIOUR                           bUnsupportedOpcodesReceived;
    REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3)
                        csmacdmgt(30) attribute(7) aUnsupportedOpcodesReceived(96)};

bUnsupportedOpcodesReceived             BEHAVIOUR

    DEFINED AS          See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.3.5;
```

B.4 DTE MAC Control function entity managed object class

B.4.1 DTE MAC Control function entity formal definition

```
oMACControlFunctionEntity          MANAGED OBJECT CLASS

  DERIVED FROM                     'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :1992':top;
  CHARACTERIZED BY
    pMandatory                     PACKAGE
      ATTRIBUTES                   aPAUSELinkDelayAllowance         GET-REPLACE;
    ;
  ;
  CONDITIONAL PACKAGES
    pRecommended                   PACKAGE
      ATTRIBUTES                   aPAUSEMACCtrlFramesTransmitted   GET,
                                   aPAUSEMACCtrlFramesReceived      GET;
      REGISTERED AS                {iso(1) std(0) iso8802(8802) csma(3)
                                   csmacdmgt(30)
                                   package(4) macControlFunctionRecomendedPkg(25)};
      PRESENT IF                   The Recommended Package is implemented.;
    REGISTERED AS          {iso(1) std(0) iso8802(8802) csma(3)
                           csmacdmgt(30)
                           managedObjectClass(3)macControlFunctionObjectClass(9)};

oMPCP                              MANAGED OBJECT CLASS

  DERIVED FROM                     'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992':top;
  CHARACTERIZED BY
          pMPCPBasic               PACKAGE
                ATTRIBUTES         aMPCPID                          GET,
                                   aMPCPAdminState                  GET,
                                   aMPCPMode                        GET,
                                   aMPCPLinkID                      GET,
                                   aMPCPRemoteMACAddress            GET,
                                   aMPCPRegistrationState           GET,
                                   aMPCPMACCtrlFramesTransmitted    GET,
                                   aMPCPMACCtrlFramesReceived       GET,
```



```
                              aMPCPTxGate                     GET,
                              aMPCPTxRegAck                   GET,
                              aMPCPTxRegister                 GET,
                              aMPCPTxRegRequest               GET,
                              aMPCPTxReport                   GET,
                              aMPCPRxGate                     GET,
                              aMPCPRxRegAck                   GET,
                              aMPCPRxRegister                 GET,
                              aMPCPRxRegRequest               GET,
                              aMPCPRxReport                   GET,
                              aMPCPTransmitElapsed            GET,
                              aMPCPReceiveElapsed             GET,
                              aMPCPRoundTripTime              GET,
                              aMPCPDiscoveryWindowsSent       GET,
                              aMPCPDiscoveryTimeout           GET,
                              aMPCPMaximumPendingGrants       GET;
                  ACTIONS     acMPCPAdminControl;
            ;
    ;
    REGISTERED AS{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) managedObjectClass(3)
                  mpcpObjectClass(21)};
```

nbMPCP-macControl **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS              oMPCP;
    NAMED BY SUPERIOR OBJECT CLASS        oMACControlEntity AND SUBCLASSES;
    WITH ATTRIBUTE                        aMPCPID;
    REGISTERED AS    {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) mpcp-
                     macControl(34)};
```

aMPCPID **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX         IEEE802Dot3-MgmtAttributeModule.OneOfName;
    MATCHES FOR                   EQUALITY, ORDERING;
    BEHAVIOUR                     bMPCPID;
    REGISTERED AS    {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                     mpcpID(351)};
```

bMPCPID **BEHAVIOUR**

```
    DEFINED AS       See "BEHAVIOUR DEFINED AS" in 30.3.5.1.1;
```

aMPCPAdminState **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX         IEEE802Dot3-MgmtAttributeModule.PortAdminState;
    MATCHES FOR                   EQUALITY, ORDERING;
    BEHAVIOUR                     bMPCPAdminState;
    REGISTERED AS    {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpAdmin-
                     State(278)};
```

bMPCPAdminState **BEHAVIOUR**

```
    DEFINED AS       See "BEHAVIOUR DEFINED AS" in 30.3.5.1.2;
```

aMPCPMode **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MPCPMode;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPMode;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpMode(279)};

bMPCPMode **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.5.1.3;

aMPCPLinkID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPLinkID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpLinkID(282)};

bMPCPLinkID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.5.1.4;

aMPCPRemoteMACAddress **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MACAddress;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPRemoteMACAddress;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpRemoteMACAddress(283)};

bMPCPRemoteMACAddress **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.5.1.5;

aMPCPRegistrationState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MPCPRegState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPRegistrationState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpRegistrationState(284)};

bMPCPRegistrationState **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.5.1.6;

aMPCPMACCtrlFramesTransmitted **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPMACCtrlFramesTransmitted;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpMACCtrlFramesTransmitted(280)};

bMPCPMACCtrlFramesTransmittedBEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.7;

aMPCPMACCtrlFramesReceived ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPMACCtrlFramesReceived;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpMACCtrlFramesReceived(281)};

bMPCPMACCtrlFramesReceivedBEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.8;

aMPCPTxGate ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPTxGate;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) txGate(315)};

bMPCPTxGate BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.9;

aMPCPTxRegAck ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPTxRegAck;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) txRegAck(316)};

bMPCPTxRegAck BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.10;

aMPCPTxRegister ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPTxRegister;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) txRegister(317)};

bMPCPTxRegister BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.11;

aMPCPTxRegRequest ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPTxRegRequest;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) txRegRegister(318)};

bMPCPTxRegRequest BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.12;

aMPCPTxReport ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPTxReport;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) txReport(319)};

bMPCPTxReport BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.13;

aMPCPRxGate ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPRxGate;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) rxGate(320)};

bMPCPRxGate BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.14;

aMPCPRxRegAck ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPRxRegAck;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) rxRegAck(321)};

bMPCPRxRegAck BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.15;

aMPCPRxRegister ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPRxRegister;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) rxRegister(322)};

bMPCPRxRegister **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.16;

aMPCPRxRegRequest **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPRxRegRequest;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) rxRegRequest(323)};

bMPCPRxRegRequest **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.17;

aMPCPRxReport **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPRxReport;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) rxReport(324)};

bMPCPRxReport **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.18;

aMPCPTransmitElapsed **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer32;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPTransmitElapsed;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpTransmitElapsed(285)};

bMPCPTransmitElapsed **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.19;

aMPCPReceiveElapsed **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer32;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPReceiveElapsed;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpReceiveElapsed(286)};

bMPCPReceiveElapsed **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.20;

aMPCPRoundTripTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPRoundTripTime;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpRoundTripTime(287)};

bMPCPRoundTripTime BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.21;

aMPCPDiscoveryWindowsSent ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPDiscoveryWindowsSent;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpDiscoveryWindowsSent(288)};

bMPCPDiscoveryWindowsSent BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.22;

aMPCPDiscoveryTimeout ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPDiscoveryTimeout;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpDiscoveryTimeout(290)};

bMPCPDiscoveryTimeout BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.23;

aMPCPMaximumPendingGrants ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer8;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMPCPMaximumPendingGrants;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mpcpMaximumPendingGrants(291)};

bMPCPMaximumPendingGrants BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.5.1.24;

acMPCPAdminControl ACTION

BEHAVIOUR bMPCPAdminControl;
MODE CONFIRMED;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.AdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)

mpcpAdminCtrl(16)};

bMPCPAdminControl **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.5.2.1;

B.4.2 DTE MAC Control function entity attributes

aPAUSELinkDelayAllowance **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.LinkDelayAllowance;
BEHAVIOUR bPAUSELinkDelayAllowance;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) aPAUSELinkDelayAllowance(97)};

bPAUSELinkDelayAllowance **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.4.1;

aPAUSEMACCtrlFramesTransmitted **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bPAUSEMACCtrlFramesTransmitted;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) aPAUSEMACCtrlFramesTransmitted(98)};

bPAUSEMACCtrlFramesTransmitted **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.4.2;

aPAUSEMACCtrlFramesReceived **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bPAUSEMACCtrlFramesReceived;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) aPAUSEMACCtrlFramesReceived(99)};

bPAUSEMACCtrlFramesReceived **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.3.4.3;

B.5 Repeater managed object class

B.5.1 Repeater, formal definition

oRepeater MANAGED OBJECT CLASS

DERIVED FROM 'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 1992':top;

CHARACTERIZED BY

```
        pRepeaterBasicControl           PACKAGE
            ATTRIBUTES                  aRepeaterID                 GET,
                                        aRepeaterType               GET,
                                        aRepeaterGroupCapacity      GET,
                                        aGroupMap                   GET,
                                        aRepeaterHealthState        GET,
                                        aRepeaterHealthText         GET,
                                        aRepeaterHealthData         GET;
            ACTIONS                     acResetRepeater,
                                        acExecuteNonDisruptiveSelfTest;
            NOTIFICATIONS               nRepeaterHealth,
                                        nRepeaterReset,
                                        nGroupMapChange;
        ;
    ;
    CONDITIONAL PACKAGES
        pRepeaterPerfMonitor            PACKAGE
            ATTRIBUTES                  aTransmitCollisions         GET;
            REGISTERED AS               {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4)
                                        repeaterPerfMonitorPkg(9)};
            PRESENT IF                  The Performance Monitor Capability is implemented.;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                            managedObjectClass(3) repeaterObjectClass(3)};
```

nbRepeaterName **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            oRepeater;
    NAMED BY SUPERIOR OBJECT CLASS
                                        "ISO/IEC 10165-2":system AND SUBCLASSES;
    WITH ATTRIBUTE                      aRepeaterID;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            repeaterName(5)};
```

nbRepeaterMonitor **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            "IEEE802.1F":oEWMAMetricMonitor;
    NAMED BY SUPERIOR OBJECT CLASS
                                        "ISO/IEC 10165-2":system  AND SUBCLASSES;
    WITH ATTRIBUTE                      "IEEE802.1F":aScannerId;
        CREATE                          WITH-AUTOMATIC-INSTANCE-NAMING;
        DELETE                          ONLY-IF-NO-CONTAINED-OBJECTS;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            repeaterMonitor(6)};
```

B.5.2 Repeater attributes

aRepeaterID **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.OneOfName;
    MATCHES FOR                         EQUALITY;
    BEHAVIOUR                           bRepeaterID;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
```


repeaterID(38)};

bRepeaterID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.1.1.1.;

aRepeaterType **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.RepeaterType;
MATCHES FOR EQUALITY;
BEHAVIOUR bRepeaterType;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) repeaterType (39)};

bRepeaterType **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.1.1.2.;

a**RepeaterGroupCapacity** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bRepeaterGroupCapacity;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) repeaterGroupCapacity(40)};

bRepeaterGroupCapacity **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.1.1.3.;

aGroupMap **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.BitString;
MATCHES FOR EQUALITY;
BEHAVIOUR bGroupMap;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) groupMap(41)};

bGroupMap **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.1.1.4.;

aRepeaterHealthState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.RepeaterHealthState;
MATCHES FOR EQUALITY;
BEHAVIOUR bRepeaterHealthState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) repeaterHealthState(42)};

bRepeaterHealthState **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.1.1.5.;

aRepeaterHealthText **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
RepeaterHealthText;
MATCHES FOR EQUALITY;
BEHAVIOUR bRepeaterHealthText;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
repeaterHealthText(43)};

bRepeaterHealthText **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.1.1.6.;

aRepeaterHealthData **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
RepeaterHealthData;
MATCHES FOR EQUALITY;
BEHAVIOUR bRepeaterHealthData;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
repeaterHealthData(44)};

bRepeaterHealthData **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.1.1.7.;

aTransmitCollisions **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bTransmitCollisions;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
transmitCollisions (45)};

bTransmitCollisions **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.1.1.8

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

B.5.3 Repeater actions

acResetRepeater **ACTION**

BEHAVIOUR bResetRepeater;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
resetRepeater(6)};

bResetRepeater **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.1.2.1.;

acExecuteNonDisruptiveSelfTest **ACTION**

BEHAVIOUR bExecuteNonDisruptiveSelfTest;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) executeNonDisruptiveSelfTestAction(7)};

bExecuteNonDisruptiveSelfTest **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.1.2.2.;

B.5.4 Repeater notifications

nRepeaterHealth **NOTIFICATION**

BEHAVIOUR bRepeaterHealth;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.
RepeaterHealthInfo
AND ATTRIBUTE IDS repeaterHealthState aRepeaterHealthState,
repeaterHealthText aRepeaterHealthText,
repeaterHealthData aRepeaterHealthData
;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) notification(10) repeaterHealth(1)};

bRepeaterHealth **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.1.3.1.;

nRepeaterReset **NOTIFICATION**

BEHAVIOUR bRepeaterReset;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.
RepeaterHealthInfo
AND ATTRIBUTE IDS repeaterHealthState aRepeaterHealthState,
repeaterHealthText aRepeaterHealthText,
repeaterHealthData aRepeaterHealthData
;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) notification(10) repeaterReset(2)};

bRepeaterReset **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.1.3.2.;

nGroupMapChange **NOTIFICATION**

BEHAVIOUR bGroupMapChange;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.BitString;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) notification(10) groupMapChange(3)};

bGroupMapChange **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.1.3.3.;

B.6 Group managed object class

B.6.1 Group, formal definition

```
oGroup                                  MANAGED OBJECT CLASS

    DERIVED FROM                        'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                        1992':top;
    CHARACTERIZED BY
        pGroupBasicControl              PACKAGE
            ATTRIBUTES                  aGroupID                    GET,
                                        aGroupPortCapacity          GET,
                                        aPortMap                    GET;
            NOTIFICATIONS               nPortMapChange;
        ;
    ;

    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                            managedObjectClass(3) groupObjectClass(4)};

nbGroupName                             NAME BINDING

    SUBORDINATE OBJECT CLASS            oGroup;
    NAMED BY SUPERIOR OBJECT CLASS
                                        oRepeater  AND SUBCLASSES;
    WITH ATTRIBUTE                      aGroupID;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            groupName(7)};
```

B.6.2 Group attributes

aGroupID **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.OneOfName;
    MATCHES FOR                         EQUALITY;
    BEHAVIOUR                           bGroupID;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                            groupID(46)};
```

bGroupID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.2.1.1;

aGroupPortCapacity **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.OneOfName;
    MATCHES FOR                         EQUALITY, ORDERING;
```



```
        BEHAVIOUR                              bGroupPortCapacity;
        REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                                groupPortCapacity(47)};

bGroupPortCapacity                             BEHAVIOUR

        DEFINED AS              See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.2.1.2;

aPortMap                                       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX                  IEEE802Dot3-MgmtAttributeModule.BitString;
        MATCHES FOR                            EQUALITY;
        BEHAVIOUR                              bPortMap;
        REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                                portMap(48)};

bPortMap                                       BEHAVIOUR

        DEFINED AS              See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.2.1.3;
```

B.6.3 Group notifications

```
nPortMapChange                                 NOTIFICATION

        BEHAVIOUR                              bPortMapChange;
        WITH INFORMATION SYNTAX                IEEE802Dot3-MgmtAttributeModule.BitString;
        REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) notification(10)
                                portMapChange(4)};

bPortMapChange                                 BEHAVIOUR

        DEFINED AS              See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.2.2.1;
```

B.7 Repeater port managed object class

B.7.1 Port, formal definition

```
oRepeaterPort                                  MANAGED OBJECT CLASS

        DERIVED FROM                           'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                               1992':top;
        CHARACTERIZED BY
                pPortBasicControl              PACKAGE
                        ATTRIBUTES             aPortID                  GET,
                                               aPortAdminState          GET,
                                               aAutoPartitionState      GET;
                        ACTIONS                acPortAdminControl;
                ;
        ;
        CONDITIONAL PACKAGES
                pPortPerfMonitor               PACKAGE
```

```
                ATTRIBUTES          aReadableFrames               GET,
                                    aReadableOctets               GET,
                                    aFrameCheckSequenceErrors     GET,
                                    aAlignmentErrors              GET,
                                    aFramesTooLong                GET,
                                    aShortEvents                  GET,
                                    aRunts                        GET,
                                    aCollisions                   GET,
                                    aLateEvents                   GET,
                                    aVeryLongEvents               GET,
                                    aDataRateMismatches           GET,
                                    aAutoPartitions               GET;
                REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                    package(4)
                                    portPerfMonitorPkg(10)};
                PRESENT IF          The Performance Monitor Capability is implemented.,
            pPortAddrTracking       PACKAGE
                ATTRIBUTES          aLastSourceAddress            GET,
                                    aSourceAddressChanges         GET;
                REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3)
                                    csmacdmgt(30) package(4) portAddrTrackPkg(11)};
                PRESENT IF          The Address Tracking and Performance Monitor
                                    capabilities are implemented.,
            p100MbpsMonitor         PACKAGE
                ATTRIBUTES          aIsolates                     GET,
                                    aSymbolErrorDuringPacket      GET;
                REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3)
                                    csmacdmgt(30) package(4)
                                    port100MbpsMonitor(12)};
                PRESENT IF          The 100/1000 Mb/s Monitor capability is
                                    implemented,
            pBurst                  PACKAGE
                ATTRIBUTES          aBursts                       GET;
                REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3)
                                    csmacdmgt(30) package(4)
                                    portBurst(18)};
                PRESENT IF          The 1000 Mb/s Burst Monitor capability is
                                    implemented;
        REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                            managedObjectClass(3) repeaterPortObjectClass(5)};

nbPortName                          NAME BINDING

        SUBORDINATE OBJECT CLASS    oRepeaterPort;
        NAMED BY SUPERIOR OBJECT CLASS
                                    oGroup AND SUBCLASSES;
        WITH ATTRIBUTE              aPortID;
        REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            portName(8)};
```

B.7.2 Port attributes

aPortID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
BEHAVIOUR bPortID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) portID(49)};

bPortID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.1.1;

aPortAdminState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PortAdminState;
MATCHES FOR EQUALITY;
BEHAVIOUR bPortAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) portAdminState(50)};

bPortAdminState **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.1.2;

aAutoPartitionState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AutoPartitionState;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoPartition;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoPartitionState(51)};

bAutoPartition **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.1.3;

aReadableFrames **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bReadableFrames;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) readableFrames(52)};

bReadableFrames **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.1.4

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aReadableOctets **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bReadableOctets;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) readableOctets(53)};

bReadableOctets **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.5

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.;

aFrameCheckSequenceErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFCSErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) frameCheckSequenceErrors(54)};

bFCSErrors **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.6

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aAlignmentErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bAlignmentErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) alignmentErrors(55)};

bAlignmentErrors **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.7

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aFramesTooLong **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesTooLong;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesTooLong(56)};

bFramesTooLong **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.8

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 61 days.;

aShortEvents **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bShortEvents;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) shortEvents(57)};

bShortEvents **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.9

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 16 hours;

aRunts **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bRunts;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) runts(58)};

bRunts **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.10

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

aCollisions **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bCollisions;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) collisions(59)};

bCollisions **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.11

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

aLateEvents **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bLateEvents;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) lateEvents(60)};

bLateEvents **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.12

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 81 h.;

aVeryLongEvents **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bVeryLongEvents;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) veryLongEvents(61)};

bVeryLongEvents **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.13

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 198 days.;

aDataRateMismatches **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bDataRateMismatches;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) dataRateMismatches(62)};

bDataRateMismatches **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.14;

aAutoPartitions **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bAutoPartitions;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoPartitions(63)};

bAutoPartitions **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.15;

aIsolates **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bIsolates;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) isolates(64)};

bIsolates **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.4.3.1.16;

aSymbolErrorDuringPacket **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bSymbolErrorDuringPacket;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) symbolErrorDuringPacket(65)};

bSymbolErrorDuringPacket **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.1.17;

aLastSourceAddress **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802CommonDefinitions.MACAddress;
MATCHES FOR EQUALITY;
BEHAVIOUR bLastSourceAddress;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) lastSourceAddress(66)};

bLastSourceAddress **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.1.18;

aSourceAddressChanges **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bSourceAddressChanges;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sourceAddressChanges(67)};

bSourceAddressChanges **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.1.19

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 81 h.;

aBursts **ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bBursts;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) bursts(100)};

bBursts **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.1.20;

B.7.3 Port actions

acPortAdminControl **ACTION**

BEHAVIOUR bPortAdminControl;

```
    WITH INFORMATION SYNTAX         IEEE802Dot3-MgmtAttributeModule.
                                    PortAdminState;
    REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
                          portAdminControl(8)};
```

bPortAdminControl **BEHAVIOUR**

```
    DEFINED AS            See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.4.3.2.1;
```

B.8 MAU managed object class

B.8.1 MAU, formal definition

```
oMAU                                MANAGED OBJECT CLASS

  DERIVED FROM                'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :1992':top;
  CHARACTERIZED BY
          pMAUBasic           PACKAGE
          ATTRIBUTES          aMAUID                         GET,
                              aMAUType                       GET-REPLACE,
                              aMAUTypeList                   GET,
                              aMediaAvailable                GET,
                              aJabber                        GET,
                              aMAUAdminState                 GET;
          NOTIFICATIONS       nJabber;
          ;
  ;
  CONDITIONAL PACKAGES
          pMAUControl         PACKAGE
          ACTIONS             acResetMAU,
                              acMAUAdminControl;
  REGISTERED AS     {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4)
                    mauControlPkg(13)};
  PRESENT IF        The pMAUControl package is implemented.,
          pMediaLossTracking  PACKAGE
  ATTRIBUTESaLoseMediaCounterGET;
  REGISTERED AS     {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) mediaLoss-
                    TrackingPkg(14)};
  PRESENT IF        MAU TypeValue = AUI or if the pMediaLossTracking package is implemented.,
          pBroadbandDTEMAU    PACKAGE
          ATTRIBUTES          aBbMAUXmitRcvSplitType         GET,
                              aBroadbandFrequencies          GET;
  REGISTERED AS     {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) broadband-
                    MAUPkg(15)};
  PRESENT IF        The MAU is of type 10BROAD36.,
          p100MbpsMonitor               PACKAGE
          ATTRIBUTES                    aFalseCarriers       GET,
                                        aIdleErrorCount      GET;
  REGISTERED AS     {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4)
                    mau100MbpsMonitor(16)};
  PRESENT IF        The MAU is capable of 100 Mb/s operation.,
          pPHY10GBTMargin               PACKAGE
```



```
            ATTRIBUTES                    aSNROpMarginChnlA          GET,
                                          aSNROpMarginChnlB          GET,
                                          aSNROpMarginChnlC          GET,
                                          aSNROpMarginChnlD          GET;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4)
                    phy10GBTMargin(39)};
    PRESENT IF      The MAU is capable of 10GBASE-T operation.,
            pPCSMonitor                   PACKAGE
            ATTRIBUTES                    aPCSCodingViolation        GET;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) pcsMoni-
                    tor(35)};
    PRESENT IF      The MAU PCS code error monitoring capability is implemented.,
            pFECMonitor                   PACKAGE
            ATTRIBUTES                    aFECAbility                GET,
                                          aFECmode                   GET-REPLACE,
                                          aFECCorrectedBlocks        GET,
                                          aFECUncorrectableBlocks    GET;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) fecMoni-
                    tor(30)};
    PRESENT IF      The MAU is capable of FEC operation.;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) managedObjectClass(3)
                    mauObjectClass(6)};
```

nbMAU-repeaterName **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS          oMAU;
    NAMED BY SUPERIOR OBJECT CLASS                --(of oRepeaterPort)
                                      oRepeaterPort AND SUBCLASSES;
                                                  --{1.0.8802.3.30.3.5}
    WITH ATTRIBUTE                    aMAUID;
    REGISTERED AS        {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                         mau-repeaterName(9)};
```

nbMAU-dteName **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS          oMAU;
    NAMED BY SUPERIOR OBJECT CLASS                --(of oPHYEntity)
                                      oPHYEntity AND SUBCLASSES;
                                                  --{1.0.8802.3.30.3.2}
    WITH ATTRIBUTE                    aMAUID;
    REGISTERED AS        {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                         mau-dteName(10)};
```

B.8.2 MAU attributes

aMAUID **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.OneOfName;
    MATCHES FOR                       EQUALITY;
    BEHAVIOUR                         bMAUID;
    REGISTERED AS        {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                         mauID(68)};
```

bMAUID		**BEHAVIOUR**
DEFINED AS	See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.5.1.1.1;	
aMAUType		**ATTRIBUTE**
WITH ATTRIBUTE SYNTAX		IEEE802Dot3-MgmtAttributeModule.TypeValue;
MATCHES FOR		EQUALITY, ORDERING;
BEHAVIOUR		bMAUType;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauType(69)};	
bMAUType		**BEHAVIOUR**
DEFINED AS	See "BEHAVIOUR DEFINED AS" in 30.5.1.1.2;	
aMAUTypeList		**ATTRIBUTE**
WITH ATTRIBUTE SYNTAX		IEEE802Dot3-MgmtAttributeModule.TypeList;
MATCHES FOR		EQUALITY, ORDERING;
BEHAVIOUR		bMAUTypeList;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauTypeList(70)};	
bMAUTypeList		**BEHAVIOUR**
DEFINED AS	See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.5.1.1.3;	
aMediaAvailable		**ATTRIBUTE**
WITH ATTRIBUTE SYNTAX		IEEE802Dot3-MgmtAttributeModule. MediaAvailState;
MATCHES FOR		EQUALITY, ORDERING;
BEHAVIOUR		bMediaAvailable;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauMediaAvailable(71)};	
bMediaAvailable		**BEHAVIOUR**
DEFINED AS	See "BEHAVIOUR DEFINED AS" in 30.5.1.1.4;	
aLoseMediaCounter		**ATTRIBUTE**
DERIVED FROM		aCMCounter;
MATCHES FOR		EQUALITY, ORDERING;
BEHAVIOUR		bLoseMediaCounter;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauLoseMediaCounter(72)};	
bLoseMediaCounter		**BEHAVIOUR**
DEFINED AS	See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.5.1.1.5;	

aJabber **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Jabber;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bJabberAttribute;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) jabber(73)};

bJabberAttribute **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.6;

aMAUAdminState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AdminState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMAUAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauAdminState(74)};

bMAUAdminState **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.5.1.1.7;

aBbMAUXmitRcvSplitType **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule. BbandXmitRcvSplitType;
MATCHES FOR EQUALITY;
BEHAVIOUR bBbMAUXmitRcvSplitType;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) bBandSplitType(75)};

bBbMAUXmitRcvSplitType **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.5.1.1.8;

aBroadbandFrequencies **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule. BbandFrequency;
MATCHES FOR EQUALITY;
BEHAVIOUR bBroadbandFrequencies;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) bBandFrequencies(76)};

bBroadbandFrequencies **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.5.1.1.9;

aFalseCarriers **ATTRIBUTE**

DERIVED FROM aCMCounter;

MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFalseCarriers;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) falseCarriers(77)};

bFalseCarriers **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.5.1.1.10;

aIdleErrorCount **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.RegisterEight;
BEHAVIOUR bIdleErrorCount;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) idleErrorCount(91)};

bIdleErrorCount **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.5.1.1.11;

aSNROpMarginChnlA **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSNROpMarginChnlA;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sNROpMarginChnlA(353)};

bSNROpMarginChnlA **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.17;

aSNROpMarginChnlB **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSNROpMarginChnlB;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sNROpMarginChnlB(354)};

bSNROpMarginChnlB **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.18;

aSNROpMarginChnlC **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSNROpMarginChnlC;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sNROpMarginChnlC(355)};

bSNROpMarginChnlC **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.19;

aSNROpMarginChnlD **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSNROpMarginChnlD;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sNROpMarginChnlD(356)};

bSNROpMarginChnlD **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.20;

aPCSCodingViolation **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPCSCodingViolation;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pcsCodingViolation(292)};

bPCSCodingViolation **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.12;

aFECAbility **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.FECMode;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFECAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) fecAbility(313)};

bFECAbility **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.13;

aFECmode **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.FECMode;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFECmode;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) fecMode(314)};

bFECmode **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.14;

aFECCorrectedBlocks **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFECCorrectedBlocks;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) fecCorrected-Blocks(293)};

bFECCorrectedBlocks **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.15;

aFECUncorrectableBlocks **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFECUncorrectableBlocks;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) fecUncorrect-ableBlocks(294)};

bFECUncorrectableBlocks **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.16;

B.8.3 MAU actions

acResetMAU **ACTION**

BEHAVIOUR bResetMAU;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) resetMAU(9)};

bResetMAU **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.5.1.2.1;

acMAUAdminControl **ACTION**

BEHAVIOUR bMAUAdminControl;
MODE CONFIRMED;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.AdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) mauAdminCtrl(10)};

bMAUAdminControl **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.5.1.2.2;

B.8.4 MAU notifications

```
nJabber                                   NOTIFICATION

    BEHAVIOUR                             bJabberNotification;
    WITH INFORMATION SYNTAX               IEEE802Dot3-MgmtAttributeModule.Jabber;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) notification(10)
                            jabber(5)};

bJabberNotification                       BEHAVIOUR

    DEFINED AS              See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.5.1.3.1;
```

B.9 AutoNegotiation managed object class

B.9.1 AutoNegotiation, formal definition

```
oAutoNegotiation                      MANAGED OBJECT CLASS

    DERIVED FROM                      'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :1992':top;
    CHARACTERIZED BY
        pAutoNeg                      PACKAGE
            ATTRIBUTES                aAutoNegID                          GET,
                                      aAutoNegAdminState                  GET,
                                      aAutoNegRemoteSignaling             GET,
                                      aAutoNegAutoConfig                  GET-REPLACE,
                                      aAutoNegLocalTechnologyAbility      GET,
                                      aAutoNegAdvertisedTechnologyAbilityGET-REPLACE,
                                      aAutoNegReceivedTechnologyAbilityGET,
                                      aAutoNegLocalSelectorAbility        GET,
                                      aAutoNegAdvertisedSelectorAbility   GET-REPLACE,
                                      aAutoNegReceivedSelectorAbility     GET;
            ACTIONS
                                     acAutoNegRestartAutoConfig,
                                     acAutoNegAdminControl;
        ;
    ;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                            managedObjectClass(3) autoNegObjectClass(7)};

nbAutoNeg-mauName                         NAME BINDING

    SUBORDINATE OBJECT CLASS              oAutoNegotiation;
    NAMED BY SUPERIOR OBJECT CLASS                        --(of oMAU)
                                          oMAU   AND SUBCLASSES;
                                                          --{1.0.8802.3.30.3.6}
    WITH ATTRIBUTE                        aMAUID;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            autoNeg-mauName(11)};
```

B.9.2 Auto-Negotiation attributes

aAutoNegID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoNegID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegID(78)};

bAutoNegID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.6.1.1.1;

aAutoNegAdminState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegAdminState;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoNegAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegAdminState(79)};

bAutoNegAdminState **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.6.1.1.2;

aAutoNegRemoteSignaling **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegRemoteSignalingDetect;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoNegRemoteSignaling;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegRemoteSignaling(80)};

bAutoNegRemoteSignaling **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.6.1.1.3;

aAutoNegAutoConfig **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegAutoConfig;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoNegAutoConfig;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegAutoConfig(81)};

bAutoNegAutoConfig **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.6.1.1.4;

aAutoNegLocalTechnologyAbility **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegTechnologyList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegLocalTechnologyAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegLocalTechnologyAbility(82)};

bAutoNegLocalTechnologyAbility **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.6.1.1.5;

aAutoNegAdvertisedTechnologyAbility **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegTechnologyList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegAdvertisedTechnologyAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegAdvertisedTechnologyAbility(83)};

bAutoNegAdvertisedTechnologyAbility **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.6.1.1.6;

aAutoNegReceivedTechnologyAbility **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegTechnologyList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegReceivedTechnologyAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegReceivedTechnologyAbility(84)};

bAutoNegReceivedTechnologyAbility **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.6.1.1.7;

aAutoNegLocalSelectorAbility **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegSelectorList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegLocalSelectorAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegLocalSelectorAbility(85)};

bAutoNegLocalSelectorAbility **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in IEEE Std 802.3 30.6.1.1.8;

aAutoNegAdvertisedSelectorAbility ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegSelectorList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegAdvertisedSelectorAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegAdvertisedSelectorAbility(86)};

bAutoNegAdvertisedSelectorAbility BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.6.1.1.9;

aAutoNegReceivedSelectorAbility ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegSelectorList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegReceivedSelectorAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegReceivedSelectorAbility(87)};

bAutoNegReceivedSelectorAbility BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.6.1.1.10;

B.9.3 AutoNegotiation actions

acAutoNegRestartAutoConfig ACTION

BEHAVIOUR bAutoNegRestartAutoConfig;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
autoNegRestartAutoConfig(11)};

bAutoNegRestartAutoConfig BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.6.1.2.1;

acAutoNegAdminControl ACTION

BEHAVIOUR bAutoNegAdminControl;
MODE CONFIRMED;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
autoNegAdminCtrl(12)};

bAutoNegAdminControl BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in IEEE Std 802.3 30.6.1.2.2;

B.10 ResourceTypeID managed object class

B.10.1 ResourceTypeID, formal definition

```
-- Implementation of this managed object in accordance with the definition contained in IEEE Std
-- 802.1F-1993 is a conformance requirement of this standard.
-- NOTE—A single instance of the Resource Type ID managed object exists within the oMACEntity
-- managed object class, a single instance of the Resource Type ID managed object exists within
-- the oRepeater managed object class, and a single instance of the Resource Type ID managed
-- object exists within the oMAU managed object class conditional on the presence of an MII.
-- The managed object itself is contained in IEEE Std 802.1F-1993, therefore only name bindings
-- appear in this standard;

nbResourceTypeID-mac                          NAME BINDING

        SUBORDINATE OBJECT CLASS              "IEEE802.1F":oResourceTypeID;
        NAMED BY SUPERIOR OBJECT CLASS
                                              oMACEntity;
        WITH ATTRIBUTE                        "IEEE802.1F":aResourceTypeIDName;
        REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                              resourceTypeID-mac(12)};

nbResourceTypeID-repeater                     NAME BINDING

        SUBORDINATE OBJECT CLASS              "IEEE802.1F":oResourceTypeID;
        NAMED BY SUPERIOR OBJECT CLASS
                                              oRepeater AND SUBCLASSES;
        WITH ATTRIBUTE                        "IEEE802.1F":aResourceTypeIDName;
        REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                              resourceTypeID-repeater(13)};

nbResourceTypeID-mau                          NAME BINDING

        SUBORDINATE OBJECT CLASS              "IEEE802.1F":oResourceTypeID;
        NAMED BY SUPERIOR OBJECT CLASS
                                              oMAU AND SUBCLASSES;
        WITH ATTRIBUTE                        "IEEE802.1F":aResourceTypeIDName;
        REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                              resourceTypeID-mau(14)};

nbResourceTypeID-midSpan                      NAME BINDING

        SUBORDINATE OBJECT CLASS              "IEEE802.1F":oResourceTypeID;
        NAMED BY SUPERIOR OBJECT CLASS oMidSpan AND SUBCLASSES;
        WITH ATTRIBUTE                        aMidSpanID;
        REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                              resourceTypeID-midSpan(25)};
```

B.11 WIS managed object class

B.11.1 WIS, formal definition

```
oWIS                          MANAGED OBJECT CLASS

   DERIVED FROM               'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :1992':top;
   CHARACTERIZED BY
          pWISBasic           PACKAGE
               ATTRIBUTES     aWISID                       GET,
                              aSectionStatus               GET,
                              aLineStatus                  GET,
                              aPathStatus                  GET,
                              aFarEndPathStatus            GET;
          ;
   ;
   CONDITIONAL PACKAGES
          pWISOptional        PACKAGE
               ATTRIBUTES     aSectionSESThreshold         GET-REPLACE,
                              aSectionSESs                 GET,
                              aSectionESs                  GET,
                              aSectionSEFSs                GET,
                              aSectionCVs                  GET,
                              aJ0ValueTX                   GET-REPLACE,
                              aJ0ValueRX                   GET,
                              aLineSESThreshold            GET-REPLACE,
                              aLineSESs                    GET,
                              aLineESs                     GET,
                              aLineCVs                     GET,
                              aFarEndLineSESs              GET,
                              aFarEndLineESs               GET,
                              aFarEndLineCVs               GET,
                              aPathSESThreshold            GET-REPLACE,
                              aPathSESs                    GET,
                              aPathESs                     GET,
                              aPathCVs                     GET,
                              aJ1ValueTX                   GET-REPLACE,
                              aJ1ValueRX                   GET,
                              aFarEndPathSESs              GET,
                              aFarEndPathESs               GET,
                              aFarEndPathCVs               GET;
               REGISTERED AS
                              {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) pack-
               age(4) wisOptionalPkg(26)};
               PRESENT IF     The optional package is implemented;
   REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) managedObjectClass(3)
                       wisObjectClass(14)};

nbWIS-mauName                 NAME BINDING

   SUBORDINATE OBJECT CLASS          oWIS;
   NAMED BY SUPERIOR OBJECT CLASS    oMAU AND SUBCLASSES;
   WITH ATTRIBUTE                    aWISID;
```

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) wis-mau-Name(24)};

B.11.2 WIS attributes

aWISID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bWISID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) wisID(181)};

bWISID **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.1;

aSectionStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.SectionStatus;
MATCHES FOR EQUALITY;
BEHAVIOUR bSectionStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sectionStatus(182)};

bSectionStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.2;

aSectionSESThreshold **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.SESThreshold;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSectionSESThreshold;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sectionSESThreshold(183)};

bSectionSESThreshold **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.3

NOTE—This counter has a maximum increment rate of 1 count per second.;

aSectionSESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSectionSESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sectionSESs(184)};

bSectionSESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.4

NOTE—This counter has a maximum increment rate of 1 count per second.;

aSectionESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSectionESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sectionESs(185)};

bSectionESs **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.5;

aSectionSEFSs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSectionSEFSs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sectionSEFSs(186)};

bSectionSEFSs **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.6

NOTE—This counter has a maximum increment rate of 1 count per second.;

aSectionCVs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSectionCVs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sectionCVs(187)};

bSectionCVs **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.7

NOTE—This counter has a maximum increment rate of 64 000 counts per second.;

aJ0ValueTX **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.JValue;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bJ0ValueTX;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) j0ValueTX(188)};

bJ0ValueTX **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.8;

aJ0ValueRX **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.JValue;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bJ0ValueRX;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) j0ValueRX(189)};

bJ0ValueRX **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.9;

aLineStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.LineStatus;
MATCHES FOR EQUALITY;
BEHAVIOUR bLineStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) lineStatus(190)};

bLineStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.10;

aLineSESThreshold **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.SESThreshold;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bLineSESThreshold;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) lineSESThreshold(191)};

bLineSESThreshold **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.11;

aLineSESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bLineSESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) lineSESs(192)};

bLineSESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.12

NOTE—This counter has a maximum increment rate of 1 count per second.;

aLineESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;

BEHAVIOUR bLineESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) lineESs(193)};

bLineESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.13

NOTE—This counter has a maximum increment rate of 1 count per second.;

aLineCVs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bLineCVs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) lineCVs(194)};

bLineCVs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.14

NOTE—This counter has a maximum increment rate of 12 288 000 counts per second.;

aFarEndLineSESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFarEndLineSESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) farEndLineSESs(195)};

bFarEndLineSESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.15

NOTE—This counter has a maximum increment rate of 1 count per second.;

aFarEndLineESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFarEndLineESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) farEndLineESs(196)};

bFarEndLineESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.16

NOTE—This counter has a maximum increment rate of 1 count per second.;

aFarEndLineCVs **ATTRIBUTE**

DERIVED FROM aCMCounter;

MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFarEndLineCVs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) farEndLineCVs(197)};

bFarEndLineCVs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.17

NOTE—This counter has a maximum increment rate of 2 040 000 counts per second.;

aPathStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PathStatus;
MATCHES FOR EQUALITY;
BEHAVIOUR bPathStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pathStatus(198)};

bPathStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.18;

aPathSESThreshold **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.SESThreshold;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPathSESThreshold;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pathSESThreshold(199)};

bPathSESThreshold **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.19;

aPathSESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPathSESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pathSESs(200)};

bPathSESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.20

NOTE—This counter has a maximum increment rate of 1 count per second.;

aPathESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPathESs;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pathESs(201)};

bPathESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.21

NOTE—This counter has a maximum increment rate of 1 count per second.;

aPathCVs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPathCVs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pathCVs(202)};

bPathCVs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.22

NOTE—This counter has a maximum increment rate of 8000 counts per second.;

aJ1ValueTX **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.JValue;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bJ1ValueTX;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) j1ValueTX(203)};

bJ1ValueTX **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.23;

aJ1ValueRX **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.JValue;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bJ1ValueRX;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) j1ValueRX(204)};

bJ1ValueRX **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.24;

aFarEndPathStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.FarEndPathStatus;
MATCHES FOR EQUALITY;
BEHAVIOUR bFarEndPathStatus;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) farEndPathStatus(205)};

bFarEndPathStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.25;

aFarEndPathSESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFarEndPathSESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) farEndPathSESs(206)};

bFarEndPathSESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.26

NOTE—This counter has a maximum increment rate of 1 count per second.;

aFarEndPathESs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFarEndPathESs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) farEndPathESs(207)};

bFarEndPathESs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.27

NOTE—This counter has a maximum increment rate of 1 count per second.;

aFarEndPathCVs **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFarEndPathCVs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) farEndPathCVs(208)};

bFarEndPathCVs **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.8.1.1.28

NOTE—This counter has a maximum increment rate of counts 8000 per second.;

B.12 PSE managed object class

B.12.1 PSE, formal definition

```
oPSE                                        MANAGED OBJECT CLASS

     DERIVED FROM                           'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                            :1992':top;
     CHARACTERIZED BY
          pPSEBasic                         PACKAGE
               ATTRIBUTES                   aPSEID                        GET,
                                            aPSEAdminState                GET,
                                            aPSEPowerPairsControlAbility  GET,
                                            aPSEPowerPairs                GET-REPLACE,
                                            aPSEPowerDetectionStatus      GET;
               ACTIONS                      acPSEAdminControl;
          ;
     ;
     CONDITIONAL PACKAGES
          pPSERecommended                   PACKAGE
               ATTRIBUTES                   aPSEPowerClassification       GET,
                                            aPSEInvalidSignatureCounter   GET,
                                            aPSEPowerDeniedCounter        GET,
                                            aPSEOverLoadCounter           GET,
                                            aPSEShortCounter              GET,
                                            aPSEMPSAbsentCounter          GET;
               REGISTERED AS                {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                            package(4) pseRecommendedPkg(27)};
               PRESENT IF                   The recommended package is implemented;
     REGISTERED AS              {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                managedObjectClass(3) pseObjectClass(15)};

nbPSE-repeaterPortName                      NAME BINDING

     SUBORDINATE OBJECT CLASS               oPSE;
     NAMED BY SUPERIOR OBJECT CLASS         oRepeaterPort AND SUBCLASSES;
     WITH ATTRIBUTE                         aPSEID;
     REGISTERED AS              {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                                pse-repeaterPortName(26)};

nbPSE-dteName                               NAME BINDING

     SUBORDINATE OBJECT CLASS               oPSE;
     NAMED BY SUPERIOR OBJECT CLASS         oPHYEntity AND SUBCLASSES;
     WITH ATTRIBUTE                         aPSEID;
     REGISTERED AS              {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                                pse-dteName(27)};

nbPSE-pseGroupName                          NAME BINDING

     SUBORDINATE OBJECT CLASS               oPSE;
     NAMED BY SUPERIOR OBJECT CLASS         oPSEGroup AND SUBCLASSES;
     WITH ATTRIBUTE                         aPSEID;
     REGISTERED AS              {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                                pse-pseGroupName(28)};
```

B.12.2 PSE attributes

aPSEID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pseID(209)};

bPSEID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.9.1.1.1;

aPSEAdminState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PortAdminState;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pseAdminState(210)};

bPSEAdminState **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.2;

aPSEPowerPairsControlAbility **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PairCtrlAbility;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEPowerPairsControlAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) psePowerPairsControlAbility(211)};

bPSEPowerPairsControlAbility **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.3;

aPSEPowerPairs **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PSEPowerPairs;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEPowerPairs;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) psePowerPairs(212)};

bPSEPowerPairs **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.4;

aPSEPowerDetectionStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.DetectStatus;

MATCHES FOR EQUALITY;
BEHAVIOUR bPSEPowerDetectionStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) psePowerDetectionStatus(214)};

bPSEPowerDetectionStatus BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.11;

aPSEPowerClassification ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PowerClass;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEPowerClassification;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) psePowerClassification(215)};

bPSEPowerClassification BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.9.1.1.6;

aPSEInvalidSignatureCounter ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEInvalidSignatureCounter;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)pseInvalidSignatureCounter(227)};

bPSEInvalidSignatureCounter BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.8.1.1.5;

aPSEPowerDeniedCounter ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEPowerDeniedCounter;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)psePowerDeniedCounter(228)};

bPSEPowerDeniedCounter BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.9.1.1.8;

aPSEOverLoadCounter ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEOverLoadCounter;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)pseOverLoadCounter(229)};

bPSEOverLoadCounter BEHAVIOUR

```
        DEFINED AS          See "BEHAVIOUR DEFINED AS" in 30.9.1.1.9;

aPSEShortCounter                        ATTRIBUTE

        DERIVED FROM                    aCMCounter;
        MATCHES FOR                     EQUALITY;
        BEHAVIOUR                       bPSEShortCounter;
        REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                            attribute(7)pseShortCounter(230)};

bPSEShortCounter                        BEHAVIOUR

        DEFINED AS          See "BEHAVIOUR DEFINED AS" in 30.9.1.1.10;

aPSEMPSAbsentCounter                    ATTRIBUTE

        DERIVED FROM                    aCMCounter;
        MATCHES FOR                     EQUALITY;
        BEHAVIOUR                       bPSEMPSAbsentCounter;
        REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                            pseMPSAbsentCounter(217)};

bPSEMPSAbsentCounter                    BEHAVIOUR

        DEFINED AS          See "BEHAVIOUR DEFINED AS" in 30.9.1.1.11;
```

B.12.3 PSE actions

```
acPSEAdminControl                       ACTION

        BEHAVIOUR                       bPSEAdminControl;
        MODE                            CONFIRMED;
        WITH INFORMATION SYNTAX         IEEE802Dot3-
                                        MgmtAttributeModule.PortAdminState;
        REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
                            pseAdminControl(13)};

bPSEAdminControl                        BEHAVIOUR

        DEFINED AS          See "BEHAVIOUR DEFINED AS" in 30.9.1.2.1;
```

B.13 Midspan managed object class

```
oMidSpan                                MANAGED OBJECT CLASS

        DERIVED FROM                    'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                        :1992':top;
        CHARACTERIZED BY
                pMidSpanBasic           PACKAGE
                        ATTRIBUTES      aMidSpanID                  GET,
                                        aMidSpanPSEGroupCapacity    GET,
                                        aMidSpanPSEGroupMap         GET;
```

```
                    NOTIFICATIONS          nMidSpanPSEGroupMapChange;
            ;
    ;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                            managedObjectClass(3) midSpanObjectClass(17)};
```

nbMidSpanName **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            oMidSpan;
    NAMED BY SUPERIOR OBJECT CLASS      "ISO/IEC 10165-2":system AND SUBCLASSES;
    WITH ATTRIBUTE                      aMidSpanID;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            midSpanName(31)};
```

nbMidSpanMonitor **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            "IEEE802.1F ":oEWMAMetricMonitor;
    NAMED BY SUPERIOR OBJECT CLASS      "ISO/IEC 10165-2":system AND SUBCLASSES;
    WITH ATTRIBUTE                      "IEEE802.1F":aScannerId;
    CREATE                              WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE                              ONLY-IF-NO-CONTAINED-OBJECTS;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            midSpanMonitor(32)};
```

B.13.1 Midspan attributes

aMidSpanID **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.OneOfName;
    MATCHES FOR                         EQUALITY;
    BEHAVIOUR                           bMidSpanID;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                            midSpanID(221)};
```

bMidSpanID **BEHAVIOUR**

```
    DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.10.1.1.1;
```

aMidSpanPSEGroupCapacity **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.OneOfName;
    MATCHES FOR                         EQUALITY,ORDERING;
    BEHAVIOUR                           bMidSpanPSEGroupCapacity;
    REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                            midSpanPSEGroupCapacity(222)};
```

bMidSpanPSEGroupCapacity **BEHAVIOUR**

```
    DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.10.1.1.2;
```

aMidSpanPSEGroupMap **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.BitString;
```



```
    MATCHES FOR                         EQUALITY;
    BEHAVIOUR                           bMidSpanPSEGroupMap;
    REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                          midSpanPSEGroupMap(223)};
```

bMidSpanPSEGroupMap **BEHAVIOUR**

```
    DEFINED AS            See "BEHAVIOUR DEFINED AS" in 30.10.1.1.3;
```

B.13.2 Midspan notifications

nMidSpanPSEGroupMapChange **NOTIFICATION**

```
    BEHAVIOUR                           bMidSpanPSEGroupMapChange;
    WITH INFORMATION SYNTAX             IEEE802Dot3-MgmtAttributeModule.BitString;
    REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                          notification(10)midSpanPSEGroupMapChange(8)};
```

bMidSpanPSEGroupMapChange **BEHAVIOUR**

```
    DEFINED AS            See "BEHAVIOUR DEFINED AS" in 30.10.1.2.1;
```

B.14 PSE Group managed object class

oPSEGroup **MANAGED OBJECT CLASS**

```
    DERIVED FROM                        'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                        :1992':top;
    CHARACTERIZED BY
            pPSEGroupBasic              PACKAGE
                    ATTRIBUTES          aPSEGroupID             GET,
                                        aPSECapacity            GET,
                                        aPSEMap                 GET;
                    NOTIFICATIONS       nPSEMapChange;
            ;
    ;
    REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                          managedObjectClass(3) midSpanGroupObjectClass(18)};
```

nbPSEGroupName **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            oPSEGroup;
    NAMED BY SUPERIOR OBJECT CLASS      oMidSpan AND SUBCLASSES;
    WITH ATTRIBUTE                      aPSEGroupID;
    REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                          pseGroupName(33)};
```

B.14.1 PSE Group attributes

aPSEGroupID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEGroupID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pseGroupID(224)};

bPSEGroupID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.10.2.1.1;

aPSECapacity **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY,ORDERING;
BEHAVIOUR bPSECapacity;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pseCapacity(225)};

bPSECapacity **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.10.2.1.2;

aPSEMap **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.BitString;
MATCHES FOR EQUALITY;
BEHAVIOUR bPSEMap;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pseMap(226)};

bPSEMap **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.10.2.1.3;

B.14.2 PSE Group notifications

nPSEMapChange **NOTIFICATION**

BEHAVIOUR bPSEMapChange;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.BitString;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) notification(10)pseMapChange(9)};

bPSEMapChange **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.10.2.2.1;

B.15 OMP entity formal definition

oOMP **MANAGED OBJECT CLASS**

DERIVED FROM 'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992':top;

```
CHARACTERIZED BY
        pOMPBasic               PACKAGE
                ATTRIBUTES      aOMPEmulationID                 GET,
                                aOMPEmulationType               GET,
                                aSLDErrors                      GET,
                                aCRC8Errors                     GET;
        ;
;
CONDITIONAL PACKAGES
        pOMPOLTMonitor          PACKAGE
                ATTRIBUTES      aGoodLLID                       GET,
                                aONUPONcastLLID                 GET,
                                aOLTPONcastLLID                 GET,
                                aBadLLID                        GET;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) package(4) oMPEr-
                ror(37)};
PRESENT IF      This package is mandatory for a OLT and optional for a ONU.;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) managedObjectClass(3)
                ompObjectClass(19)};
```

nbOMP-MACEntity **NAME BINDING**

```
SUBORDINATE OBJECT CLASS                oOMP;
NAMED BY SUPERIOR OBJECT CLASS          oMACEntity;
WITH ATTRIBUTE                          aOMPEmulationID;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) omp-
                macEntity(36)};
```

B.15.1 OMP attributes

aOMPEmulationID **ATTRIBUTE**

```
WITH ATTRIBUTE SYNTAX           IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR                     EQUALITY, ORDERING;
BEHAVIOUR                       bOMPEmulationID;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) ompEmula-
                tionID(231)};
```

bOMPEmulationID **BEHAVIOUR**

```
DEFINED AS                      See "BEHAVIOUR DEFINED AS" in 30.3.7.1.1;
```

aOMPEmulationType **ATTRIBUTE**

```
WITH ATTRIBUTE SYNTAX           IEEE802Dot3-MgmtAttributeModule.OMPEmulationType;
MATCHES FOR                     EQUALITY, ORDERING;
BEHAVIOUR                       bOMPEmulationType;
REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) ompEmula-
                tionType(232)};
```

bOMPEmulationType **BEHAVIOUR**

```
DEFINED AS                      See "BEHAVIOUR DEFINED AS" in 30.3.7.1.2;
```


aSLDErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bSLDErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) sldErrors(233)};

bSLDErrors **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.7.1.3;

aCRC8Errors **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bCRC8Errors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) crc8Errors(234)};

bCRC8Errors **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.7.1.4;

aGoodLLID **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bGoodLLID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) goodLLID(341)};

bGoodLLID **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.7.1.5;

aONUPONcastLLID **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bONUPONcastLLID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) onuPONcastLLID(342)};

bONUPONcastLLID **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.7.1.6;

aOLTPONcastLLID **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOLTPONcastLLID;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oltPONcastLLID(343)};

bOLTPONcastLLID **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.7.1.7;

aBadLLID **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bBadLLID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) badLLID(235)};

bBadLLID **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.7.1.8;

B.16 OAM entity managed object class

B.16.1 OAM entity formal definition

oOAM **MANAGED OBJECT CLASS**

DERIVED FROM 'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992':top;
CHARACTERIZED BY
pOAMBasic PACKAGE
ATTRIBUTES aOAMID GET,
aOAMAdminState GET,
aOAMMode GET-REPLACE,
aOAMDiscoveryState GET,
aOAMRemoteMACAddress GET,
aOAMLocalConfiguration GET,
aOAMRemoteConfiguration GET,
aOAMLocalPDUConfiguration GET,
aOAMRemotePDUConfiguration GET,
aOAMLocalFlagsField GET,
aOAMRemoteFlagsField GET,
aOAMLocalRevision GET,
aOAMRemoteRevision GET,
aOAMLocalState GET,
aOAMRemoteState GET,
aOAMRemoteVendorOUI GET,
aOAMRemoteVendorSpecificInfo GET,
aOAMUnsupportedCodesTx GET,
aOAMUnsupportedCodesRx GET,
aOAMInformationTx GET,
aOAMInformationRx GET,
aOAMUniqueEventNotificationTx GET,
aOAMDuplicateEventNotificationTx GET,
aOAMUniqueEventNotificationRx GET,


```
                                aOAMDuplicateEventNotificationRx      GET,
                                aOAMLoopbackControlTx                 GET,
                                aOAMLoopbackControlRx                 GET,
                                aOAMVariableRequestTx                 GET,
                                aOAMVariableRequestRx                 GET,
                                aOAMVariableResponseTx                GET,
                                aOAMVariableResponseRx                GET,
                                aOAMOrganizationSpecificTx            GET,
                                aOAMOrganizationSpecificRx            GET,
                                aOAMLocalErrSymPeriodConfig           GET,
                                aOAMLocalErrSymPeriodEvent            GET,
                                aOAMLocalErrFrameConfig               GET,
                                aOAMLocalErrFrameEvent                GET,
                                aOAMLocalErrFramePeriodConfig         GET,
                                aOAMLocalErrFramePeriodEvent          GET,
                                aOAMLocalErrFrameSecsSummaryConfig    GET,
                                aOAMLocalErrFrameSecsSummaryEvent     GET,
                                aOAMRemoteErrSymPeriodEvent           GET,
                                aOAMRemoteErrFrameEvent               GET,
                                aOAMRemoteErrFramePeriodEvent         GET,
                                aOAMRemoteErrFrameSecsSummaryEvent    GET,
                                aFramesLostDueToOAMError              GET;
                    ACTIONS     acOAMAdminControl;
          ;
    ;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) managedObjectClass(3)
                    oamObjectClass(20)};
```

nbOAM-aggregator **NAME BINDING**

```
    SUBORDINATE OBJECT CLASS            oOAM;
    NAMED BY SUPERIOR OBJECT CLASS      oAggregator;
    WITH ATTRIBUTE                      aOAMID;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) oam-
                    aggregator(35)};
```

B.16.2 OAM entity attributes

aOAMID **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX       IEEE802Dot3-MgmtAttributeModule.OneOfName;
    MATCHES FOR                 EQUALITY, ORDERING;
    BEHAVIOUR                   bOAMID;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamID(236)};
```

bOAMID **BEHAVIOUR**

```
    DEFINED AS                  See "BEHAVIOUR DEFINED AS" in 30.3.6.1.1;
```

aOAMAdminState **ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX       IEEE802Dot3-MgmtAttributeModule.PortAdminState;
    MATCHES FOR                 EQUALITY, ORDERING;
```

BEHAVIOUR bOAMAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamAdminState(237)};

bOAMAdminState **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.2;

aOAMMode **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMMode;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMMode;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamMode(238)};

bOAMMode **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.3;

aOAMDiscoveryState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMDiscoveryState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMDiscoveryState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamDiscoveryState(333)};

bOAMDiscoveryState **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.4;

aOAMRemoteMACAddress **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MACAddress;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteMACAddress;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteMACAddress(239)};

bOAMRemoteMACAddress **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.5;

aOAMLocalConfiguration **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMConfig;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalConfiguration;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalConfiguration(334)};

bOAMLocalConfiguration **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.6;

aOAMRemoteConfiguration **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMConfig;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteConfiguration;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemote-Configuration(240)};

bOAMRemoteConfiguration **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.7;

aOAMLocalPDUConfiguration **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalPDUConfiguration;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalP-DUConfiguration(335)};

bOAMLocalPDUConfiguration **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.8;

aOAMRemotePDUConfiguration **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemotePDUConfiguration;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteP-DUConfiguration(241)};

bOAMRemotePDUConfiguration **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.9;

aOAMLocalFlagsField **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMFlagsField;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalFlagsField;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocal-FlagsField(242)};

bOAMLocalFlagsField **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.10;

aOAMRemoteFlagsField **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMFlagsField;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteFlagsField;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteFlagsField(243)};

bOAMRemoteFlagsField **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.11;

aOAMLocalRevision **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalRevision;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalRevision(336)};

bOAMLocalRevision **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.12;

aOAMRemoteRevision **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer16;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteRevision;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteRevision(244)};

bOAMRemoteRevision **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.13;

aOAMLocalState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalState(337)};

bOAMLocalState **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.14;

aOAMRemoteState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteState;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteState(245)};

bOAMRemoteState BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.15;

aOAMRemoteVendorOUI ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OUI;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteVendorOUI;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteVendorOUI(246)};

bOAMRemoteVendorOUI BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.16;

aOAMRemoteVendorSpecificInfo ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer32;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOamRemoteVendorSpecificInfo;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteVendorSpecificInfo(247)};

bOamRemoteVendorSpecificInfo BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.17;

aOAMUnsupportedCodesTx ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMUnsupportedCodesTx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamUnsupportedCodesTx(338)};

bOAMUnsupportedCodesTx BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.18;

aOAMUnsupportedCodesRx ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMUnsupportedCodesRx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamUnsupportedCodesRx(250)};

bOAMUnsupportedCodesRx BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.19;

aOAMInformationTx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMInformationTx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamInformationTx(251)};

bOAMInformationTx **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.20;

aOAMInformationRx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMInformationRx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamInformationRx(252)};

bOAMInformationRx **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.21;

aOAMUniqueEventNotificationTx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMUniqueEventNotificationTx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamUniqueEventNotificationTx(339)};

bOAMUniqueEventNotificationTx **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.22;

aOAMDuplicateEventNotificationTx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMDuplicateEventNotificationTx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamDuplicateEventNotificationTx(340)};

bOAMDuplicateEventNotificationTx **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.23;

aOAMUniqueEventNotificationRx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMUniqueEventNotificationRx;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamUniqueEventNotificationRx(254)};

bOAMUniqueEventNotificationRx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.24;

aOAMDuplicateEventNotificationRx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMDuplicateEventNotificationRx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamDuplicateEventNotificationRx(255)};

bOAMDuplicateEventNotificationRx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.25;

aOAMLoopbackControlTx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLoopbackControlTx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLoopbackControlTx(256)};

bOAMLoopbackControlTx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.26;

aOAMLoopbackControlRx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLoopbackControlRx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLoopbackControlRx(257)};

bOAMLoopbackControlRx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.27;

aOAMVariableRequestTx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMVariableRequestTx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamVariableRequestTx(258)};

bOAMVariableRequestTx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.28;

aOAMVariableRequestRx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMVariableRequestRx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamVariableRequestRx(259)};

bOAMVariableRequestRx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.29;

aOAMVariableResponseTx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMVariableResponseTx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamVariableResponseTx(260)};

bOAMVariableResponseTx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.30;

aOAMVariableResponseRx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMVariableResponseRx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamVariableResponseRx(261)};

bOAMVariableResponseRx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.31;

aOAMOrganizationSpecificTx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMOrganizationSpecificTx;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamOrganizationSpecificTx(262)};

bOAMOrganizationSpecificTx **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.32;

aOAMOrganizationSpecificRx **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMOrganizationSpecificRx;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamOrganizationSpecificRx(263)};

bOAMOrganizationSpecificRx BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.33;

aOAMLocalErrSymPeriodConfig ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorConf;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalErrSymPeriodConfig;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalErrSymPeriodConfig(264)};

bOAMLocalErrSymPeriodConfig BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.34;

aOAMLocalErrSymPeriodEvent ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorEvent;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalErrSymPeriodEvent;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalErrSymPeriodEvent(265)};

bOAMLocalErrSymPeriodEvent BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.35;

aOAMLocalErrFrameConfig ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorConf;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalErrFrameConfig;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalErrFrameConfig(266)};

bOAMLocalErrFrameConfig BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.36;

aOAMLocalErrFrameEvent ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorEvent;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalErrFrameEvent;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalErrFrameEvent(267)};

bOAMLocalErrFrameEvent BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.37;

aOAMLocalErrFramePeriodConfig ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorConf;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalErrFramePeriodConfig;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalErrFramePeriodConfig(268)};

bOAMLocalErrFramePeriodConfig BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.38;

aOAMLocalErrFramePeriodEvent ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorEvent;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalErrFramePeriodEvent;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalErrFramePeriodEvent(269)};

bOAMLocalErrFramePeriodEvent BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.39;

aOAMLocalErrFrameSecsSummaryConfig ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorConf;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalErrFrameSecsSummaryConfig;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalErrFrameSecsSummaryConfig(270)};

bOAMLocalErrFrameSecsSummaryConfig BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.40;

aOAMLocalErrFrameSecsSummaryEvent ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorEvent;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMLocalErrFrameSecsSummaryEvent;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamLocalErrFrameSecsSummaryEvent(271)};

bOAMLocalErrFrameSecsSummaryEvent BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.6.1.41;

aOAMRemoteErrSymPeriodEvent ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorEvent;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteErrSymPeriodEvent;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteErrSymPeriodEvent(272)};

bOAMRemoteErrSymPeriodEvent BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.42;

aOAMRemoteErrFrameEvent ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorEvent;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteErrFrameEvent;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteErrFrameEvent(273)};

bOAMRemoteErrFrameEvent BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.43;

aOAMRemoteErrFramePeriodEvent ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorEvent;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteErrFramePeriodEvent;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteErrFramePeriodEvent(274)};

bOAMRemoteErrFramePeriodEvent BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.44;

aOAMRemoteErrFrameSecsSummaryEvent ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OAMErrorEvent;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOAMRemoteErrFrameSecsSummaryEvent;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) oamRemoteErrFrameSecsSummaryEvent(275)};

bOAMRemoteErrFrameSecsSummaryEvent BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.45;

aFramesLostDueToOAMError ATTRIBUTE

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFramesLostDueToOAMError;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) framesLostDueToOAMError(276)};

bFramesLostDueToOAMError BEHAVIOUR

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.6.1.46;

B.16.3 OAM actions

```
acOAMAdminControl                       ACTION

    BEHAVIOUR                           bOAMAdminControl;
    MODE                                CONFIRMED;
    WITH INFORMATION SYNTAX             IEEE802Dot3-MgmtAttributeModule.AdminState;
    REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
                        oamAdminControl(15)};

bOAMAdminControl                        BEHAVIOUR

    DEFINED AS                          See "BEHAVIOUR DEFINED AS" in 30.3.6.2.1;
```

B.17 PAF managed object class

B.17.1 PAF formal definition

```
oPAF                                MANAGED OBJECT CLASS

  DERIVED FROM                      'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992':top;
  CHARACTERIZED BY
          pPAFBasic                 PACKAGE
                  ATTRIBUTES        aPAFID                          GET,
                                    aPhyEnd                         GET,
                                    aPHYCurrentStatus               GET,
                                    aPAFSupported                   GET;
          ;
  ;
  CONDITIONAL PACKAGES
          pPAFAggregation PACKAGE
                  ATTRIBUTES        aPAFAdminState                  GET-REPLACE,
                                    aLocalPAFCapacity               GET,
                                    aLocalPMEAvailable              GET,
                                    aLocalPMEAggregate              GET,
                                    aRemotePAFSupported             GET,
                                    aRemotePAFCapacity              GET,
                                    aRemotePMEAggregate             GET;
                  REGISTERED AS{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                  package(4) pafAggregation(38)};
                  PRESENT IF        The PAF aggregation capability is implemented.;
  REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) managedObjectClass(3)
                  pafObjectClass(24)};

nbPAF-PHYEntity                     NAME BINDING

  SUBORDINATE OBJECT CLASS              oPAF;
  NAMED BY SUPERIOR OBJECT CLASS        oPHYEntity;
  WITH ATTRIBUTE                        aPAFID;
  REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) paf-phyE-
                  ntity(39)};
```

aPAFID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPAFID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pafID(344)};

bPAFID **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.11.1.1.1;

aPhyEnd **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PHYEnd;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPhyEnd;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) phyEnd(326)};

bPhyEnd **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.11.1.1.2;

aPHYCurrentStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PHYCurrentState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPHYCurrentStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) phyCurrentStatus(296)};

bPHYCurrentStatus **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.11.1.1.3;

aPAFSupported **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPAFSupported;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pAFSupported(304)};

bPAFSupported **BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.11.1.1.4;

aPAFAdminState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PortAdminState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPAFAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pAFAdminState(305)};

bPAFAdminState **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.1.1.5;

aLocalPAFCapacity **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer8;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bLocalPAFCapacity;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) localPAFCapacity(327)};

bLocalPAFCapacity **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.1.1.6;

aLocalPMEAvailable **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer32;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bLocalPMEAvailable;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) localPMEAvailable(306)};

bLocalPMEAvailable **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.1.1.7;

aLocalPMEAggregate **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer32;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bLocalPMEAggregate;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) localPMEAggregate(307)};

bLocalPMEAggregate **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.1.1.8;

aRemotePAFSupported **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bRemotePAFSupported;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) remotePAFSupported(328)};

bRemotePAFSupported **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.1.1.9;

```
aRemotePAFCapacity              ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       IEEE802Dot3-MgmtAttributeModule.Integer8;
    MATCHES FOR                 EQUALITY, ORDERING;
    BEHAVIOUR                   bRemotePAFCapacity;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) remotePAFCa-
                    pacity(329)};

bRemotePAFCapacity              BEHAVIOUR

    DEFINED AS                  See "BEHAVIOUR DEFINED AS" in 30.11.1.1.10;

aRemotePMEAggregate             ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       IEEE802Dot3-MgmtAttributeModule.Integer32;
    MATCHES FOR                 EQUALITY, ORDERING;
    BEHAVIOUR                   bRemotePMEAggregate;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) remotePME-
                    Aggregate(310)};

bRemotePMEAggregate             BEHAVIOUR

    DEFINED AS                  See "BEHAVIOUR DEFINED AS" in 30.11.1.1.11;
```

B.18 PME managed object class

B.18.1 PME formal definition

```
oPME                            MANAGED OBJECT CLASS

    DERIVED FROM                'CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992':top;
    CHARACTERIZED BY
            pPMEBasic           PACKAGE
                    ATTRIBUTES  aPMEID                          GET,
                                aPMEAdminState                  GET-REPLACE,
                                aPMEStatus                      GET,
                                aPMESNRMgn                      GET,
                                aTCCodingViolations             GET,
                                aProfileSelect                  GET-REPLACE,
                                aOperatingProfile               GET,
                                aPMEFECCorrectedBlocks          GET,
                                aPMEFECUncorrectableBlocks      GET,
                                aTCCRCErrors                    GET;
            ;
    ;
    REGISTERED AS   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) managedObjectClass(3)
                    pmeObjectClass(25)};

nbPME-PAF                       NAME BINDING

    SUBORDINATE OBJECT CLASS            oPME;
    NAMED BY SUPERIOR OBJECT CLASS      oPAF;
```


WITH ATTRIBUTE aPAFID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) pme-paf(40)};

aPMEID **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPMEID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pmeID(330)};

bPMEID **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.1;

aPMEAdminState **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PortAdminState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPMEAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pmeAdmin-State(345)};

bPMEAdminState **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.2;

aPMEStatus **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PMEStatus;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPMEStatus;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pmeSta-tus(346)};

bPMEStatus **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.3;

aPMESNRMgn **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer8;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPMESNRMgn;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pmeSN-RMgn(331)};

bPMESNRMgn **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.4;

aTCCodingViolations **ATTRIBUTE**

DERIVED FROM aCMCounter;

MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bTCCodingViolations;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) tcCodingViolations(332)};

bTCCodingViolations **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.5;

aProfileSelect **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PMEProfile;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bProfileSelect;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) profileSelect(347)};

bProfileSelect **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.6;

aOperatingProfile **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PMEOperatingProfile;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bOperatingProfile;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) operatingProfile(348)};

bOperatingProfile **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.7;

aPMEFECCorrectedBlocks **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPMEFECCorrectedBlocks;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pmeFECCorrectedBlocks(349)};

bPMEFECCorrectedBlocks **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.8;

aPMEFECUncorrectableBlocks **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPMEFECUncorrectableBlocks;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) pmeFECUncorrectableBlocks(350)};

bPMEFECUncorrectableBlocks **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.9;

aTCCRCErrors **ATTRIBUTE**

DERIVED FROM aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bTCCRCErrors;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) tcCRCErrors(352)};

bTCCRCErrors **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.11.2.1.10;

Annex C

(normative)

GDMO and ASN.1 definitions for management

C.1 Common attributes template

This template defines generic facilities that are used in the standard.

aCMCounter **ATTRIBUTE**

DERIVED FROM "ISO/IEC 10165-5":genericWrappingCounter;
BEHAVIOUR bCMCounter;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) cmCounter(88)};

bCMCounter **BEHAVIOUR**

DEFINED AS Wraps at one of two sizes. Size is conditional.
Wraps at 32 bits, that is this counter reaches its maximum value at $2^{32}-1$ (i.e., approximately 4.294×10^9) and then rolls over to zero on the next increment, if maximum increment rate from zero causes a rollover in 58 min or more.
Wraps at 64 bits, that is this counter reaches its maximum value at $2^{64}-1$ (i.e., approximately 1.844×10^{19}) and then rolls over to zero on the next increment, if maximum increment rate from zero would cause a 32 bit counter to roll over in less than 58 min.
The counter that this is derived from initializes to zero. Initialization to zero is not a requirement of this standard;

C.2 ASN.1 module for CSMA/CD managed objects

This ASN.1 module defines the ASN.1 types and subtypes that are referred to immediately after the WITH ATTRIBUTE SYNTAX construct in this clause uses of the attribute template defined in ISO/IEC 10165-4:1992 [B37].

IEEE802Dot3-MgmtAttributeModule {iso(1) std(0) iso8802(8802) csma(3) global(1) asn1Module(2) commonDefinitions(0) version(2)} DEFINITIONS IMPLICIT TAGS::= BEGIN

EXPORTS--*everything*

IMPORTS--*implicitly imports ISO 8824:1990*

```
MACAddress
    FROM IEEE802CommonDefinitions
    {iso(1) member-body(2) us(840) ieee802dot1partF(10011)
    asn1Module(2) commonDefinitions(0) version1(0)};
AdminState::= ENUMERATED {
```

```
        other                   (1),     --undefined
        unknown                 (2),     --initializing, true state not yet known
        operational             (3),     --powered and connected
        standby                 (4),     --inactive but on
        shutdown                (5)      --similar to power down
        }
AggDataRate ::= INTEGER (0..2^32-1)--The data rate of an Aggregation

AggID ::= INTEGER (0..2^32-1)

AggOrInd ::= BOOLEAN

AggPortID ::= INTEGER (0..2^32-1)

AggPortList ::= SEQUENCE OF AggPortID

AggPortState ::= BIT STRING (SIZE (1..8))

AggState ::= ENUMERATED {
        up                      (0),     --operational
        down                    (1)      --disabled
        }

AttemptArray::= SEQUENCE OF aCMCounter--array [1..attempt limit - 1]

AutoNegAdminState::= ENUMERATED {
        disabled                (1),
        enabled                 (2)
        }

AutoNegAutoConfig::= ENUMERATED {
        other                   (1),
        configuring             (2),
        complete                (3),
        disabled                (4),
        parallel detect fail    (5)
        }

AutoNegRemoteSignalingDetect::= ENUMERATED {
        detected                (1),
        notdetected             (2)
        }

AutoNegSelector::= ENUMERATED {
```



```
        other                   (1),    --undefined
        ethernet                (2),    --802.3
        isoethernet             (3)     --802.9
        }

AutoNegSelectorList::= SEQUENCE OF AutoNegSelector

AutoNegTechnology::= ENUMERATED {
        global                  (0),    --reserved for future use
        other                   (1),    --undefined
        unknown                 (2),    --initializing, true ability not yet know.
        10BASE-T                (14),   --10BASE-T as defined in Clause 14
        10BASE-TFD              (142),  --Full duplex 10BASE-T as defined in Clauses 14 and 31
        100BASE-T4              (23),   --100BASE-T4 as defined in Clause 23
        100BASE-TX              (25),   --100BASE-TX as defined in Clause 25
        100BASE-TXFD            (252),  --Full duplex 100BASE-TX as defined in Clauses 25 and 31
        10GBASE-KX4             (483),  --10GBASE-KX4 PHY as defined in Clause 71
        10GBASE-KR              (495),  --10GBASE-KR PHY as defined in Clause 72
        10GBASE-T               (55),   --10GBASE-T PHY as defined in Clause 55
        FDX PAUSE               (312),  --PAUSE operation for full duplex links as defined in
                                        Annex 31B
        FDX APAUSE              (313),  --Asymmetric PAUSE operation for full duplex links as defined
                                         in Clause 37 and Annex 31B
        FDX SPAUSE              (314),  --Symmetric PAUSE operation for full duplex links as defined
                                         in Clause 37 and Annex 31B
        FDX BPAUSE              (315),  --Asymmetric and Symmetric PAUSE operation for full duplex
                                        links as defined in Clause 37 and Annex 31B
        100BASE-T2              (32),   --100BASE-T2 as defined in Clause 32
        100BASE-T2FD            (322),  --Full duplex 100BASE-T2 as defined in Clauses 31 and 32
        1000BASE-X              (36),   --1000BASE-X as defined in Clause 36
        1000BASE-XFD            (362),  --Full duplex 1000BASE-X as defined in Clause 36
        1000BASE-KX             (393),  --1000BASE-KX PHY as defined in Clause 70
        1000BASE-T              (40),   --1000BASE-T PHY as defined in Clause 40
        1000BASE-TFD            (402),  --Full duplex 1000BASE-T PHY as defined in Clause 40
        Rem Fault1              (37),   --Remote fault bit 1 (RF1) as specified in Clause 37
        Rem Fault2              (372),  --Remote fault bit 2 (RF1) as specified in Clause 37
        isoethernet             (8029)  --802.9 ISLAN-16T
        }
AutoNegTechnologyList::= SEQUENCE OF AutoNegTechnology
AutoPartitionState::= ENUMERATED {
        autoPartitioned         (1),
        notAutoPartitioned      (2)
        }

BbandFrequency::= SEQUENCE {
        xmitCarrierFrequency    [1]     INTEGER --Frequency in MHz times 4 (250 kHz resolution)
        translationFrequency    [2]     INTEGER   --Frequency in MHz times 4 (250 kHz resolution)
        }
```

```
BbandXmitRcvSplitType::= ENUMERATED {
        other                   (1),    --undefined
        single                  (2),    --single-cable system
        dual                    (3)     --dual-cable system, offset normally zero
        }


BitString::= BIT STRING (SIZE (1..1024))


ChurnState ::= ENUMERATED {
        noChurn                 (0),    --NO_ACTOR/PARTNER_CHURN
                                        --or ACTOR/PARTNER_CHURN_MONITOR
        churn                   (1)     --ACTOR/PARTNER_CHURN
                                }


CollectorMaxDelay ::= INTEGER           --16 bit value, tens of microseconds
                                        --(max = 0.65535 seconds)


CurrentStatus ::= ENUMERATED {
        MPSAbsent               (0),    -- MPS absent
        ok                      (1)     -- MPS present and over current not detected
        }


DeferControl::= ENUMERATED {
        unknown                 (1),    --defer control mode unknown
        defer control off       (2),    --defer control mode disabled
        defer control on        (3)     --defer control mode enabled
        }


DescriptionString ::= PrintableString (SIZE 0..255))


DetectStatus ::= ENUMERATED {
        disabled                (0),    -- PSE disabled
        searching               (1),    -- PSE searching
        deliveringPower         (2),    -- PSE delivering power
        test                    (3),    -- PSE test mode
        fault                   (4),    -- PSE fault detected
        otherFault              (5)     -- PSE implementation specific fault detected
        }


DuplexValues::= ENUMERATED {
        half duplex             (1),    --capable of operating in half duplex mode
        full duplex             (2),    --capable of operating in full duplex mode
        unknown                 (3)     --unknown duplex capability
        }


FarEndPathStatus ::= BIT STRING (SIZE (2))
```

```
FECMode::= ENUMERATED {
        unknown                 (1),    --initializing, true state not yet known
        supported               (2),    --FEC supported
        not supported           (3)     --FEC not supported
        }


Integer8::= INTEGER (0..2^8-1)          --8 bit value


Integer16::= INTEGER (0..2^16-1)        --16 bit value


Integer32 ::= INTEGER (0..2^32-1)       --32 bit value


Integer48::= INTEGER (0..2^48-1)        --48 bit value


Integer64::= INTEGER (0..2^64-1)        --64 bit value


Jabber::= SEQUENCE {
        jabberFlag              [1]     JabberFlag,
        jabberCounter           [2]     JabberCounter
        }


JabberFlag::= ENUMERATED {
        other                   (1),    --undefined
        unknown                 (2),    --initializing, true state not yet known
        normal                  (3),    --state is true or normal
        fault                   (4)     --state is false, fault or abnormal
        }


JabberCounter::= INTEGER (0..2^32- 1)


JValue::= OCTET STRING (SIZE (16))


KeyValue ::= INTEGER (0..2^16-1)        --16 bit value; range 0-65535


LACPActivity ::= ENUMERATED {
        active                  (0),    --Port is Active LACP
        passive                 (1)     --Port is Passive LACP
        }


LACPTimeout ::= ENUMERATED {
        short                   (0),    --Timeouts are Short
        long                    (1)     --Timeouts are Long
```



```
        }

LineStatus ::= BIT STRING (SIZE (2))

LinkDelayAllowance::= INTEGER (0..2^32 - 1)

MACControlFunctions::= ENUMERATED {
        PAUSE                   (312)   --PAUSE command implemented
        }

MACControlFunctionsList::= SEQUENCE OF MACControlFunctions

MACCapabilitiesList::= SEQUENCE OF DuplexValues

MauTypeList::= SEQUENCE OF TypeValue

MaxFrameLengthList::= ENUMERATED {
        unknown                 (1),    --Frame length capability unknown
        basicFrame              (2),    --Capable of supporting maxBasicFrameSize
                                        (1518 octet frames)
        qTaggedFrame            (3),    --Capable of supporting maxBasicFrameSize + qTagPrefixSize
                                        (1522 octet frames)
        envelopeFrame           (4)     --Capable of supporting maxEnvelopeFrameSize
                                        (2000 octet frames)
        }

MediaAvailState::= ENUMERATED {
        other                   (1),    --undefined
        unknown                 (2),    --initializing, true state not yet known
        available               (3),    --link or light normal, loopback normal
        not available           (4),    --link loss or low light, no loopback
        remote fault            (5),    --remote fault with no detail
        invalid signal          (6),    --invalid signal, applies only to 10BASE-FB
        remote jabber           (7),    --remote fault, reason known to be jabber
        remote link loss        (8),    --remote fault, reason known to be far-end link loss
        remote test             (9),    --remote fault, reason known to be test
        offline                 (10),   --offline, applies only to Clause 37 Auto-Negotiation
        auto neg error          (11)    --Auto-Negotiation error, applies only to Clause 37 Auto-
                                        Negotiation
        PMD link fault          (12)    --PMD/PMD receive link fault
        WIS frame loss          (13)    --WIS loss of frame, applies only to 10GBASE-W
        WIS signal loss         (14)    --WIS loss of signal, applies only to 10GBASE-W
        PCS link fault          (15)    --PCS receive link fault
        excessive BER           (16)    --PCS Bit Error Ratio monitor reporting excessive error ratio
        DXS link fault          (17)    --DTE XS receive link fault, applies only to XAUI
        PXS link fault          (18)    --PHY XS transmit link fault, applies only to XAUI
        }
```

```
MIIDetect::= ENUMERATED {
        unknown                 (1),
        presentNothingConnected(2),
        presentConnected        (3),
        absent                  (4)
        }


MPCPMode::= ENUMERATED {
        OLT                     (1),        --sublayer operating in OLT mode
        ONU                     (2)         --sublayer operating in ONU mode
        }


MPCPRegState::= ENUMERATED {
        unregistered            (1),        --unregistered
        registering             (2),        --registering
        registered              (3)         --registered
        }


MulticastAddressList::= SEQUENCE OF MACAddress


MuxState ::= ENUMERATED {
        detached                (0),        --DETACHED
        waiting                 (1),        --WAITING
        attached                (2),        --ATTACHED
        collecting              (3),        --COLLECTING
        distributing            (4),        --DISTRIBUTING
        collecting_distributing (5)         --COLLECTING_DISTRIBUTING
        }


NotificationEnable ::= ENUMERATED {
        enabled                 (0),        --Notifications enabled
        disabled                (1)         --Notifications disabled


OAMConfig::= BIT STRING (SIZE (5))


OAMDiscoveryState::= ENUMERATED {
        link fault              (1),
        active send local       (2),
        passive wait            (3),
        send local remote       (4),
        send any                (5),
        send local remote ok    (6)
        }


OAMErrorConf::= SEQUENCE {
```



```
        OAMEventWindow      [1]     Integer64,
        OAMEventThreshold   [2]     Integer64
        }


OAMErrorEvent::= SEQUENCE {
        OAMEventTimeStamp   [1]     Integer16,
        OAMEventWindow      [2]     Integer64,
        OAMEventThreshold   [3]     Integer64,
        OAMErrors           [4]     Integer64,
        OAMErrorRunTotal    [5]     Integer64,
        OAMEventRunTotal    [6]     Integer32
        }


OAMFlagsField::= BIT STRING (SIZE (7))


OAMMode::= ENUMERATED {
        passive             (1),    --passive
        active              (2)     --active
        }


OAMState::= BIT STRING (SIZE (3))


OMPEmulationType::= ENUMERATED {
        unknown             (1),    --Initializing, true state or type not yet known
        OLT                 (2),    --Sublayer operating in OLT mode
        ONU                 (3)     --Sublayer operating in ONU mode
        }


OneOfName::= INTEGER (1..1024)


OUI::= OCTETSTRING                  -- Length 3 octets


PairCtrlAbility ::=BOOLEAN


PathStatus ::= BIT STRING (SIZE (4))


PHYCurrentState::= ENUMERATED {
        noPMEAssigned       (1),    --no PME assigned in case of PME aggregation
        lossOfFraming       (2),    --one or more PME in aggregation indicates Loss of Framing
        lossOfSignal        (3),    --one or more PME in aggregation indicates Loss of Signal
        lossOfPower         (4),    --one or more PME in aggregation indicates Loss of Power,
                                      10PASS-TS only
        configInitFailure   (5),    --configuration initialization failure
        noPeerPMEPresent    (6),    --one or more PME in aggregation indicates no peer PME
                                      present
```



```
        snrMarginViolation      (7),    --one or more PME in aggregation indicates SNR Margin
                                          Violation
        lineAttenViolation      (8)     --one or more PME in aggregation indicates Loop Attenuation
                                          Violation
        }


PHYEnd::= ENUMERATED {
        subscriber              (1),    --Subscriber mode of operation
        office                  (2)     --Office mode of operation
        }


PhyTypeList::= SEQUENCE OF PhyTypeValue


PhyTypeValue::= ENUMERATED {
        other                   (1),    --undefined
        unknown                 (2),    --initializing, true state or type not yet known
        none                    (3),    --MII present and nothing connected
        10 Mb/s                 (7),    --Clause 7 10 Mb/s Manchester
        100BASE-T4              (23),   --Clause 23 100 Mb/s 8B/6T
        100BASE-X               (24),   --Clause 24 100 Mb/s 4B/5B
        100BASE-T2              (32),   --Clause 32 100 Mb/s PAM5x5
        1000BASE-X              (36),   --Clause 36 1000 Mb/s 8B/10B
        1000BASE-T              (40)    --Clause 40 1000 Mb/s 4D-PAM5
        10GBASE-X               (48)    --Clause 48 10 Gb/s 4
        10GBASE-R               (49)    --Clause 49 10 Gb/s 64B/66B
        10GBASE-W               (50),   --Clause 49 10 Gb/s 64B/66B and Clause 50 WIS
        10GBASE-T               (55),   --Clause 55 10 Gb/s DSQ128
        10PASS-TS               (62),   --Clause 61 2.5 Mb/s to 100 Mb/s 64/65-octet
        2BASE-TL                (63)    --Clause 61 0.5 Mb/s to 5.5 Mb/s 64/65-octet
        }


PMEOperatingProfile::= SEQUENCE {
        PMEProfileState         [1]     PMEProfileState,
        PMEProfileNumber        [2]     Integer8
        }


PMEProfile::= SEQUENCE {
        profile1                [1]     Integer8,
        profile2                [2]     Integer8,
        profile3                [3]     Integer8,
        profile4                [4]     Integer8,
        profile4                [5]     Integer8,
        profile4                [6]     Integer8
        }


PMEProfileState::= ENUMERATED {
        no link                 (1),    --link is not up
        match                   (2),    --link up using a profile
```



```
        no match               (3),     --link up not using a profile
        activate failure       (4)      --link activate failure
        }

PMEStatus::= ENUMERATED {
        down not ready         (1),     --link is down, not ready
        down ready             (2),     --link is down, ready
        initializing           (3),     --link is initializing
        10PASS-TS              (4),     --link is up as 10PASS-TS
        2BASE-TL               (5)      --link is up as 2BASE-TL
        }

PortAdminState::= ENUMERATED {
        disabled               (1),
        enabled                (2)
        }

PortNumber ::= INTEGER (0..2^16-1)

PowerClass ::= ENUMERATED {
        class0                 (0),     -- Class 0 PD
        class1                 (1),     -- Class 1 PD
        class2                 (2),     -- Class 2 PD
        class3                 (3),     -- Class 3 PD
        class4                 (4)      -- Class 4 PD
        }

PriorityValue ::= INTEGER (0..2^16-1) --16 bits

PSEPowerPairs ::= ENUMERATED {
        signal                 (0),     -- PSE Pinout Alternative A
        spare                  (1)      -- PSE Pinout Alternative B
        }

RateValues ::= ENUMERATED {
        rate control off       (1),     -- rate control enabled
        rate control on        (2),     -- rate control disabled
        unknown                (3)      -- rate control mode unknown
        }

RegisterEight::= INTEGER (0..255)

RepeaterHealthData::= OCTET STRING (SIZE (0..255))

RepeaterHealthInfo::= SEQUENCE {
```



```
    repeaterHealthState     [1]     RepeaterHealthState,
    repeaterHealthText      [2]     RepeaterHealthText  OPTIONAL,
    repeaterHealthData      [3]     RepeaterHealthData  OPTIONAL
    }

RepeaterHealthState::= ENUMERATED {
    other                   (1),    --undefined or unknown
    ok                      (2),    --no known failures
    repeaterFailure         (3),    --known to have a repeater-related failure
    groupFailure            (4),    --known to have a group-related failure
    portFailure             (5),    --known to have a port-related failure
    generalFailure          (6)     --has a failure condition, unspecified type
    }

RepeaterType::= ENUMERATED {
    other                   (1),    --See 30.2.5:
    unknown                 (2),    --initializing, true state or type not yet known
    10 Mb/s                 (9),    --Clause 9 10 Mb/s Baseband repeater
    100 Mb/sClassI          (271),  --Clause 27 class I 100 Mb/s Baseband repeater
    100 Mb/sClassII         (272),  --Clause 27 class II 100 Mb/s Baseband repeater
    1000 Mb/s               (41),   --Clause 41 1000 Mb/s Baseband repeater
    802.9a                  (99)    --Integrated services repeater
    }

RepeaterHealthText::= PrintableString (SIZE (0..255))

RxState ::= ENUMERATED {
    current                 (0),
    expired                 (1),
    defaulted               (2),
    initialize              (3),
    lacpDisabled            (4),
    portDisabled            (5)
    }

SectionStatus ::= BIT STRING (SIZE (2))

SESThreshold ::= INTEGER (0..2^32-1)

TrueFalse::= BOOLEAN

TypeList::= SEQUENCE OF TypeValue

TypeValue::= ENUMERATED {
    global                  (0),    --undefined
    other                   (1),    --undefined
```

```
unknown          (2),    --initializing, true state not yet known
AUI              (7),    --no internal MAU, view from AUI
2BASE-TL         (63),   --see IEEE Std 802.3, Clause 61 and 63
10BASE5          (8),    --see IEEE Std 802.3, Clause 8
FOIRL            (9),    --see IEEE Std 802.3, 9.9
10BASE2          (10),   --see IEEE Std 802.3, Clause 10
10BROAD36        (11),   --see IEEE Std 802.3, Clause 11
10BASE-T         (14),   --see IEEE Std 802.3, Clause 14, duplex mode unknown
10BASE-THD       (141),  --see IEEE Std 802.3, Clause 14, half duplex mode
10BASE-TFD       (142),  --see IEEE Std 802.3, Clause 14, full duplex mode
10PASS-TS        (62),   --see IEEE Std 802.3, Clause 61 and 62
10BASE-FP        (16),   --see IEEE Std 802.3, Clause 16
10BASE-FB        (17),   --see IEEE Std 802.3, Clause 17
10BASE-FL        (18),   --see IEEE Std 802.3, Clause 18, duplex mode unknown
10BASE-FLHD      (181),  --see IEEE Std 802.3, Clause 18, half duplex mode
10BASE-FLFD      (182),  --see IEEE Std 802.3, Clause 18, full duplex mode
100BASE-T4       (23),   --see IEEE Std 802.3, Clause 23
100BASE-TX       (25),   --see IEEE Std 802.3, Clause 25, duplex mode unknown
100BASE-TXHD     (251),  --see IEEE Std 802.3, Clause 25, half duplex mode
100BASE-TXFD     (252),  --see IEEE Std 802.3, Clause 25, full duplex mode
100BASE-BX10D    (581),  --see IEEE Std 802.3, Clause 58, OLT
100BASE-BX10U    (582),  --see IEEE Std 802.3, Clause 58, ONU
100BASE-FX       (26),   --see IEEE Std 802.3, Clause 26, duplex mode unknown
100BASE-FXHD     (261),  --see IEEE Std 802.3, Clause 26, half duplex mode
100BASE-FXFD     (262),  --see IEEE Std 802.3, Clause 26, full duplex mode
100BASE-LX10     (58),   --see IEEE Std 802.3, Clause 58
100BASE-T2       (32),   --see IEEE Std 802.3, Clause 32, duplex mode unknown
100BASE-T2HD     (321),  --see IEEE Std 802.3, Clause 32, half duplex mode
100BASE-T2FD     (322),  --see IEEE Std 802.3, Clause 32, full duplex mode
1000BASE-X       (36),   --see IEEE Std 802.3, Clause 36, duplex mode unknown
1000BASE-BX10D   (591),  --see IEEE Std 802.3, Clause 59, OLT
1000BASE-BX10U   (592),  --see IEEE Std 802.3, Clause 59, ONU
1000BASE-XHD     (361),  --see IEEE Std 802.3, Clause 36, half duplex mode
1000BASE-XFD     (362),  --see IEEE Std 802.3, Clause 36, full duplex mode
1000BASE-LX      (381),  --see IEEE Std 802.3, Clause 38, duplex mode unknown
1000BASE-LXHD    (382),  --see IEEE Std 802.3, Clause 38, half duplex mode
1000BASE-LXFD    (383),  --see IEEE Std 802.3, Clause 38, full duplex mode
1000BASE-LX10    (59),   --see IEEE Std 802.3, Clause 59
1000BASE-PX10D   (601),  --see IEEE Std 802.3, Clause 60, OLT
1000BASE-PX10U   (602),  --see IEEE Std 802.3, Clause 60, ONU
1000BASE-PX20D   (603),  --see IEEE Std 802.3, Clause 60, OLT
1000BASE-PX20U   (604),  --see IEEE Std 802.3, Clause 60, ONU
1000BASE-SX      (384),  --see IEEE Std 802.3, Clause 38, duplex mode unknown
1000BASE-SXHD    (385),  --see IEEE Std 802.3, Clause 38, half duplex mode
1000BASE-SXFD    (386),  --see IEEE Std 802.3, Clause 38, full duplex mode
1000BASE-CX      (39),   --see IEEE Std 802.3, Clause 39, duplex mode unknown
1000BASE-CXHD    (391),  --see IEEE Std 802.3, Clause 39, half duplex mode
1000BASE-CXFD    (392),  --see IEEE Std 802.3, Clause 39, full duplex mode
1000BASE-KX      (393),  --see IEEE Std 802.3, Clause 70
1000BASE-T       (40),   --see IEEE Std 802.3, Clause 40, duplex mode unknown
1000BASE-THD     (401),  --see IEEE Std 802.3, Clause 40, half duplex mode
1000BASE-TFD     (402),  --see IEEE Std 802.3, Clause 40, full duplex mode
10GBASE-X        (48),   --see IEEE Std 802.3, Clause 48
```

```
        10GBASE-LX4         (481),   --see IEEE Std 802.3, Clause 53
        10GBASE-CX4         (482),   --see IEEE Std 802.3, Clause 54
        10GBASE-KX4         (483),   --see IEEE Std 802.3, Clause 71
        10GBASE-R           (49),    --see IEEE Std 802.3, Clause 49
        10GBASE-ER          (491),   --see IEEE Std 802.3, Clause 52
        10GBASE-LR          (492),   --see IEEE Std 802.3, Clause 52
        10GBASE-SR          (493),   --see IEEE Std 802.3, Clause 52
        10GBASE-LRM         (494),   --see IEEE Std 802.3, Clause 68
        10GBASE-KR          (495),   --see IEEE Std 802.3, Clause 72
        10GBASE-W           (50),    --see IEEE Std 802.3, Clauses 49 and 50
        10GBASE-EW          (501),   --see IEEE Std 802.3, Clause 52
        10GBASE-LW          (502),   --see IEEE Std 802.3, Clause 52
        10GBASE-SW          (503),   --see IEEE Std 802.3, Clause 52
        10GBASE-T           (55),    --see IEEE Std 802.3, Clause 55
        802.9a              (99)     --see IEEE Std 802.3, IEEE Std 802.9
        }

    END
```