IEEE STANDARDS ASSOCIATION

IEEE Standard for Ethernet YANG Data Model Definitions

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

IEEE Std 802.3.2™-2019

IEEE Std 802.3.2™-2019

IEEE Standard for Ethernet YANG Data Model Definitions

Developed by the

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 21 March 2019

IEEE-SA Standards Board

Abstract: YANG models for IEEE Std 802.3 are defined in this standard. This standard also publishes these models in a machine-readable format.

Keywords: 802.3, 802.3.2, Ethernet, YANG

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

PDF: ISBN 978-1-5044-5671-5 STD23620
Print: ISBN 978-1-5044-5672-2 STDPD23620

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document 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

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted 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 these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to

photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents 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.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

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-SA Website at http://ieeexplore.ieee.org/xpl/standards.jsp or contact IEEE at the address listed previously. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. 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 standard 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 P802.3.2 (IEEE 802.3cf) 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*
Adam Healey, *IEEE 802.3 Working Group Vice-Chair*
Pete Anslow, *IEEE 802.3 Working Group Secretary*
Steven B. Carlson, *IEEE 802.3 Working Group Executive Secretary*
Valerie Maguire, *IEEE 802.3 Working Group Treasurer*

Yan Zhuang, *IEEE P802.3.2 (IEEE 802.3cf)YANG Data Model Definitions Task Force Chair*
Marek Hajduczenia, *IEEE P802.3.2 (IEEE 802.3cf) YANG Data Model Definitions Task Force Editor-in-Chief*

David Abramson
Dale Amason
Ali Angha
Eric Baden
Amrik Bains
Thananya Baldwin
Denis Beaudoin
Piergiorgio Beruto
Mark Bohm
Brad Booth
Martin Bouda
David Brandt
Ralf-Peter Braun
Theodore Brillhart
Paul Brooks
Matthew Brown
Phillip Brownlee
Chris Bullock
Jairo Bustos Heredia
Adrian Butter
Clark Carty
Craig Chabot
Mandeep Chadha
David Chalupsky
Frank Chang
Xin Chang
Chan Chen
Weiying Cheng
Golam Choudhury
Keng Hua Chuang
Keith Conroy
John D'Ambrosia
Yair Darshan
Piers Dawe
Fred Dawson
Eric DiBiaso
Chris Diminico
Mike Dudek
Frank Effenberger
Eitel Cornelia
Estes David
John Ewen
Ramin Farjad
Shahar Feldman
Alan Flatman
Brian Franchuk
Matthias Fritsche
Richard Frosch
Mike Gardner
Ali Ghiasi
Joel Goergen
Steven Gorshe
Jens Gottron
Steffen Graber
Olaf Grau
Robert M. Grow
Yong Guo
Mark Gustlin
Tomohiro Hashimoto
Akinori Hayakawa
Rajmohan Hegde
David Hess
Yasuo Hidaka
David Hoglund
Rita Horner
Bernd Horrmeyer
Xi Huang
Yasuhiro Hyakutake
Jonathan Ingham
Kazuhiko Ishibe
Hideki Isono
Tom Issenhuth
Kenneth Jackson
Andrew Jimenez
John Johnson
Chad Jones
Peter Jones
Upen Kareti
Yasuaki Kawatsu
Yong Kim
Kimber Mark
Klaus Andrew
Michael Klempa
Curtis Knittle
Elizabeth Kochuparambil
Paul Kolesar
Hans Lackner
Jeffrey Lapak
Mark Laubach
Greg Le Cheminant
Han Hyub Lee
Hanan Leizerovich
David Lewis
Jon Lewis
Mike Peng Li
Jane Lim
Alex Lin
Robert Lingle
Dekun Liu
Hai-Feng Liu
Zhenyu Liu
Miklos Lukacs
Kent Lusted
Zahy Madgar
Jeffery Maki
David Malicoat
Arthur Marris
Takeo Masuda
Kirsten Matheus
Erdem Matoglu
Mick McCarthy
Brett Mcclellan
Thomas Mcdermott
Larry McMillan
Greg McSorley
Marcel Medina
Richard Mei
Richard Mellitz
Phil Miguelez
Harald Mueller
Ron Muir

Henry Muyshondt
Edward Nakamoto
Gary Nicholl
Kevin Noll
Ronald Nordin
Mark Nowell
David Ofelt
Tom Palkert
Sujan Pandey
Earl Parsons
Harsh Patel
Arkadiy Peker
Wanquan Peng
Gerald Pepper
Michael Peters
Phong Pham
Jean Picard
David Piehler
Rick Pimpinella
William Powell
Rick Rabinovich
Adee Ran
Alon Regev
Duane Remein
Michael Ressl
Salvatore Rotolo
Sam Sambasivan
Edward Sayre
Fred Schindler
Laura Schweitz
Masood Shariff
Ramin Shirani
Mizuki Shirao
Kapil Shrikhande
Jeff Slavick
Daniel Smith
Scott Sommers
Bryan Sparrowhawk
Edward Sprague
Atul Srivastava
Peter Stassar
Heath Stewart
David Stover
Junqing Sun
Steve Swanson
Andre Szczepanek
Bharat Tailor
Tomoo Takahara
Kohichi Tamura
Ronald Tellas
Geoffrey O. Thompson
Pirooz Tooyserkani
Nathan Tracy
Matthew Traverso
David Tremblay
Stephen Trowbridge
Ed Ulrichs
Daisuke Umeda
Alexander Umnov
Sterling A. Vaden
Paul Vanderlaan
Dylan Walker
Edward Walter
Haifei Wang
Roy Wang
Tongtong Wang
Xinyuan Wang
Christoph Wechsler
Brian Welch
Matthias Wendt
Oded Wertheim
Natalie Wienckowski
Ludwig Winkel
Peter Wu
Markus Wucher
Dayin Xu
Yu Xu
Adrian Young
Lennart Yseboodt
Andrew Zambell
Conrad Zerna
Huanlin Zhang
Qiwen Zhong
Martin Zielinski
George Zimmerman
Helge Zinner
Pavel Zivny

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

Andrea Agnes
Richard Alfvin
Niranth Amogh
Peter Anslow
Butch Anton
Ralf-Peter Braun
Nancy Bravin
Theodore Brillhart
Matthew Brown
Demetrio Bucaneg
Jairo Bustos Heredia
William Byrd
Steven Carlson
Keith Chow
Rodney Cummings
Andrew Fieldsend
Avraham Freedman
Matthias Fritsche
Yukihiro Fujimoto
Eric W. Gray
Randall Groves
Robert M. Grow
Marek Hajduczenia
Adam Healey
Marco Hernandez
Werner Hoelzl
Noriyuki Ikeuchi
Sergiu Iordanescu
Atsushi Ito
Raj Jain
SangKwon Jeong
Stuart Kerry
Yongbum Kim
Patrick Kinney
Mark Laubach
David J. Law
Hyeong Ho Lee
Jon Lewis
Elvis Maculuba
Valerie Maguire
Jeffery Maki
Vinayagam Mariappan
Roger Marks
Arthur Marris
Michael Maytum
Brett Mcclellan
Thomas Mcdermott
Richard Mellitz
Nick S.A. Nikjoo
Satoshi Obara
Bansi Patel
Rick Pimpinella
William Powell
R. K. Rannow
Alon Regev
Duane Remein
Maximilian Riegel
Robert Robinson
Benjamin Rolfe
Daniel Smith
Thomas Starai
Peter Stassar
Walter Struppler
Ganesh Subramanian
Mitsutoshi Sugawara
Kohichi Tamura
David Tremblay
Stephen Trowbridge
Mark-Rene Uchida
Alexander Umnov
Dmitri Varsanofiev
George Vlantis
Khurram Waheed
Karl Weber
Hung-Yu Wei
Scott Willy
Adrian Young
James Young
Oren Yuen
Zhen Zhou
Yan Zhuang
Pavel Zivny

When the IEEE-SA Standards Board approved this standard on 21 March 2019, it had the following membership:

Gary Hoffman, *Chair*
Ted Burse, *Vice Chair*
Jean-Philippe Faure, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Stephen D. Dukes
J. Travis Griffith
Guido Hiertz
Christel Hunter
Thomas Koshy
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
David J. Law
Joseph Levy
Howard Li
Xiaohui Liu
Kevin Lu
Daleep Mohla
Andrew Myles
Annette D. Reilly
Dorothy Stanley
Sha Wei
Phil Wennblom
Philip Winston
Howard Wolfman
Feng Wu
Jingyi Zhou

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 802.3.2-2019, IEEE Standard for Ethernet YANG Data Model Definitions.

The YANG modules included in this standard provide YANG versions of attributes defined in IEEE Std 802.3™-2018, Clause 30, as well as derivative attributes defined in other management information bases (e.g., SNMP attributes included in IEEE Std 802.3.1, YANG versions of IETF Etherlike MIB attributes, etc.). The YANG modules defined in this standard accommodate IEEE Std 802.3-2018, excluding any currently published or future amendments. As IEEE Std 802.3 continues to evolve, new revisions of this standard may be published in the future to address new technologies and features.

Contents

IEEE Standard for Ethernet YANG Data Model Definitions

1. Overview

This standard defines YANG modules for various Ethernet devices specified in IEEE Std 802.3. This includes half-duplex and full-duplex data terminal equipment (DTE) using either Carrier Sense Multiple Access/Collision Detection (CSMA/CD) or Multipoint Control Protocol (MPCP), and Power Sourcing Equipment (PSE).

1.1 Scope

This standard defines YANG data models for IEEE Std 802.3 Ethernet.

1.2 Purpose

The purpose of the standard is to define YANG modules for IEEE Std 802.3 and publish these modules in a machine-readable format.

1.3 Machine-readable YANG modules

The machine-readable files are available for download at the following URL: https://github.com/YangModels/yang/tree/master/standard/ieee/published/802.3 as text files with a *.yang* extension, e.g., *ieee802-ethernet-interface.yang*. The use of specialized tools to view YANG modules may be useful to create tree, UML image, and HTML outputs from the YANG modules.

Like other languages, YANG (see IETF RFC 7950) has an accepted style for machine-readable files, which was followed during the development of this standard. This formatting may not be preserved when importing the machine-readable YANG modules into the PDF. In case of any formatting discrepancies, the published machine-readable files should be consulted.

1.4 Summary of YANG-based management framework

The structure of YANG-based management framework closely resembles the structure of the Internet-Standard Management Framework, described in detail in section 7 of IETF RFC 3410.

Managed objects defined using YANG modeling language are hosted on the managed device and accessed through NETCONF (see IETF RFC 7803) or RESTCONF (see IETF RFC 8040). This standard specifies YANG modules that are compliant to YANG 1.1 (see IETF RFC 7950).

1.5 Security considerations

The YANG modules defined in this standard are designed to be accessed via network management protocols, including NETCONF (see IETF RFC 7803) or RESTCONF (see IETF RFC 8040). The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) (see IETF RFC 6242) or TLS (see IETF RFC 8446). The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS (see IETF RFC 8446).

The NETCONF access control model (see IETF RFC 8341) provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in these YANG modules that are writable/creatable/deletable, i.e., have the config property set to true, which is the default setting. These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

Some of the readable data nodes in these YANG modules may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes.

Some of the RPC operations in these YANG modules may be considered sensitive or vulnerable in some network environments. Therefore, it is important to control access to these operations.

1.6 YANG module syntax validation

All YANG modules included in this standard are YANG 1.1 (see IETF RFC 7950) compliant and pass automated checks using tools available at the time of publication.

The following open source and/or free versions of YANG validation tools may be used: Pyang (see https://github.com/mbj4668/pyang), ConfD (see http://www.tail-f.com/confd-basic), as well as other YANG model validation tools listed at http://www.yangvalidator.com.

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.

IEEE Std 802®-2014, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.[1, 2]

IEEE Std 802d™-2017, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture Amendment 1: Allocation of Uniform Resource Name (URN) Values in IEEE 802 Standards.

IEEE Std 802.1Q™-2014, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks.

IEEE Std 802.3™-2018, IEEE Standard for Ethernet.

IEEE Std 802.3.1™-2013, IEEE Standard for Management Information Base (MIB) Definitions for Ethernet.

IETF RFC 2819, Remote Network Monitoring Management Information Base, S. Waldbusser, May 2000.[3]

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

IETF RFC 3621, *Power Ethernet MIB*, A. Berger, December 2003

IETF RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*, J. Flick, September 2003.

IETF RFC 6242, *Using the NETCONF Protocol over Secure Shell (SSH)*, Wasserman M, June 2011.

IETF RFC 6991, *Common YANG Data Types*, Schoenwaelder J., July 2013.

IETF RFC 7803, *Changing the Registration Policy for the NETCONF Capability URNs Registry*, B. Leiba February 2016.

IETF RFC 7950, *The YANG 1.1 Data Modeling Language*, Bjorklund M., August 2016.

IETF RFC 8040, *RESTCONF Protocol*, Bierman A., Bjorklund M., and Watsen K., January 2017.

IETF RFC 8342, *Network Management Datastore Architecture (NMDA)*, M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, and R. Wilton, March 2018.

IETF RFC 8341, *Network Configuration Access Control Model*, A. Bierman and M. Bjorklund, March 2018.

---

[1]The IEEE standards or products referred to in this clause are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

[2]IEEE publications are available from the Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

[3]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 RFC 8343, *A YANG Data Model for Interface Management*, Bjorklund, M., March 2018.

IETF RFC 8407, *Guidelines for Authors and Reviewers of YANG Data Model Documents*, Bierman A., October 2018.

IETF RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*, E. Rescorla, August 2018.

3. Definitions

For the purposes of this document, the following terms and definitions apply. Some terms used in this document are defined in IEEE Std 802.3, and where alternative definitions occur in the IEEE Standards Dictionary, the IEEE Std 802.3 definition should be used. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[4]

3.1 data model: A data model describes how data is represented and accessed.

3.2 YANG module: A YANG module defines a hierarchy of nodes that can be used for NETCONF-based (see IETF RFC 7803) and RESTCONF-based (see IETF RFC 8040) operations. With its definitions and the definitions it imports or includes from elsewhere, a module is self-contained and can be compiled.

[4] *IEEE Standards Dictionary Online* is available at: http://dictionary.ieee.org/.

4. Abbreviations

This standard contains the following abbreviations:

CO	Central Office
CPE	Customer Premise Equipment
CSMA/CD	carrier sense multiple access with collision detection
DTE	data terminal equipment
EFM	Ethernet in the First Mile
ELO	Ethernet Link OAM
EPON	Ethernet passive optical networks
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
NETCONF	Network Configuration Protocol
OAM	Operations, Administration, and Maintenance
PoE	Power over Ethernet
RESTCONF	RESTful Configuration Protocol
YANG	Yet Another Next Generation

5. Ethernet YANG Module

5.1 YANG module structure

Two modules defined in this clause are focused on the configuration and monitoring of IEEE Std 802.3 Ethernet interfaces. The *ieee802-ethernet-interface* YANG module contains definitions of current attributes used widely in the industry in current products, while the *ieee802-ethernet-interface-half-duplex* YANG module contains definitions of half-duplex attributes.

This standard does not have a normative requirement for data nodes of the base ietf-interfaces YANG module, but the following data nodes are supported: name, description, type, enabled, admin-status, oper-status, if-index, and phys-address.

5.2 Mapping of IEEE Std 802.3, Clause 30 managed objects

This subclause contains the mapping between YANG data nodes included in *ieee802-ethernet-interface* (see Table 5–1) and *ieee802-ethernet-interface-half-duplex* (see Table 5–4) YANG modules, managed objects, and attributes defined in IEEE Std 802.3, Clause 30.

Table 5–1—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-interface* YANG data nodes

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-interface* YANG data nodes		
Managed object(s)	**Attribute(s)**		**Container(s)**	**Data node(s)**	**R/W**
oAutoNegoti-aion	acAutoNegAdminControl	30.6.1.2.2	interfaces/interface/ethernet/	auto-negotiation/enable	R/W
	aAutoNegAutoConfig	30.6.1.1.4		negotiation-status	R
N/A	N/A			flow-control/pause/direction	R/W
oMACControl-FunctionEntity	aPAUSEMACCtrlFramesReceived	30.3.4.3		flow-control/pause/statistics/in-frames-pause	R
	aPAUSEMACCtrlFramesTransmitted	30.3.4.2		flow-control/pause/statistics/out-frames-pause	R
N/A	dot3HCOutPFCFrames				
N/A	N/A			flow-control/force-flow-control	R/W
N/A	N/A			speed	R/W
oMACEntity	aDuplexStatus	30.3.1.1.32		duplex	R/W
	aMaxFrameLength	30.3.1.1.37		max-frame-length	R
	aSlowProtocolFrameLimit	30.3.1.1.38		frame-limit-slow-protocol	R
oEXTENSION	aEXTENSIONMACCtrlStatus	30.3.8.3		mac-control-extension-control	R
N/A	N/A			capabilities/auto-negotiation	R

Table 5–1—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-interface* YANG data nodes *(continued)*

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-interface* YANG data nodes		
Managed object(s)	**Attribute(s)**		**Container(s)**	**Data node(s)**	**R/W**
oMACEntity	aFramesReceivedOK	30.3.1.1.5	interfaces/interface/ethernet/statistics/frame	in-frames	R
	aMulticastFramesReceivedOK	30.3.1.1.21		in-multicast-frames	R
	aBroadcastFramesReceivedOK	30.3.1.1.22		in-broadcast-frames	R
	aFrameCheckSequenceErrors + aAlignmentErrors	30.4.3.1.6, 30.4.3.1.7		in-error-fcs-frames	R
oMACEntity	aFrameTooLongErrors	30.3.1.1.25		in-error-oversize-frames	R
	aFramesLostDueToIntMACRcvError	30.3.1.1.15		in-error-mac-internal-frames	R
	aFramesTransmittedOK	30.3.1.1.2		out-frames	R
	aMulticastFramesXmittedOK	30.3.1.1.18		out-multicast-frames	R
	aBroadcastFramesXmittedOK	30.3.1.1.19		out-broadcast-frames	R
	aFramesLostDueToIntMACXmitError	30.3.1.1.12		out-error-mac-internal-frames	R
oPHYEntity	aSymbolErrorDuringCarrier	30.3.2.1.5	interfaces/interface/ethernet/statistics/phy	in-error-symbol	R
	aReceiveLPITransitions	30.3.2.1.11	interfaces/interface/ethernet/statistics/phy/lpi	in-lpi-transitions	R
	aReceiveLPIMicroseconds	30.3.2.1.9		in-lpi-time	R
	aTransmitLPITransitions	30.3.2.1.10		out-lpi-transitions	R
	aTransmitLPIMicroseconds	30.3.2.1.8		out-lpi-time	R

Table 5–1—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-interface* YANG data nodes *(continued)*

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-interface* YANG data nodes		
Managed object(s)	Attribute(s)		Container(s)	Data node(s)	R/W
oMACControlEntity	aUnsupportedOpcodesReceived	30.3.3.5	interfaces/interface/ethernet/statistics/mac-control	in-frames-mac-control-unknown	R
oEXTENSION	aEXTENSIONMACCtrlFramesReceived	30.3.8.2		in-frames-mac-control-extension	R
	aEXTENSIONMACCtrlFramesTransmitted	30.3.8.1		out-frames-mac-control-extension	R

Table 5–2—Mapping between IETF RFC 2819 managed objects and *ieee802-ethernet-interface* YANG data nodes

IETF RFC 2819 Attribute(s)	Corresponding *ieee802-ethernet-interface* YANG data nodes		
	Container(s)	Data node(s)	R/W
no direct object[a]	interfaces/interface/ethernet/statistics/frame	in-total-frames	R
etherStatsOctets		in-total-octets	R
etherStatsUndersizePkts + etherStatsFragments		in-error-undersize-frames	R

[a] Can be calculated as: aFramesReceivedOK + aFrameCheckSequenceErrors + aAlignmentErrors + aFrameTooLongErrors + aFramesLostDueToIntMACRcvError.

Table 5–3—Mapping between IETF RFC 3635 managed objects and *ieee802-ethernet-interface* YANG data nodes

ETHERLIKE MIB Attribute(s)	Corresponding *ieee802-ethernet-interface* YANG data nodes		
	Container(s)	Data node(s)	R/W
dot3HCInPFCFrames	interfaces/interface/ethernet/	flow-control/pfc{ethernet-pfc} / statistics/in-frames-pfc	R
dot3HCOutPFCFrames		flow-control/pfc{ethernet-pfc} / statistics/out-frames-pfc	R

Table 5–4—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-interface-half-duplex* YANG data nodes

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-interface-half-duplex* YANG data nodes		
Managed object(s)	Attribute(s)		Container(s)	Data node(s)	R/W
oMACEntity	aRateControlAbility	30.3.1.1.33	interfaces/interface/ethernet	dynamic-rate-control	R/W
			interfaces/interface/ethernet/capability	dynamic-rate-control-supported	R
oPHYEntity	aSQETestErrors	30.3.2.1.4	interfaces/interface/ethernet/statistics/frame/ csmacd{csma-cd}	in-errors-sqe-test	R
oMACEntity	aSingleCollisionFrames	30.3.1.1.3		out-frames-collision-single	R
	aMultipleCollisionFrames	30.3.1.1.4		out-frames-collision-multiple	R
	aFramesWithDeferredXmissions	30.3.1.1.9		out-frames-deferred	R
	aFramesAbortedDueToXSColls	30.3.1.1.11		out-frames-collisions-excessive	R
	aLateCollisions	30.3.1.1.10		out-collisions-late	R
	aCarrierSenseErrors	30.3.1.1.13		out-errors-carrier-sense	R
	aCollisionFrames	30.3.1.1.30		collision-histogram/collision-count	R
				collision-histogram/collision-count-frames	R

5.3 YANG module definition[5]

The YANG module tree hierarchy uses terms defined in IETF RFC 8407.

5.3.1 Tree hierarchy

```
module: ieee802-ethernet-interface
  augment /if:interfaces/if:interface:
    +--rw ethernet
       +--rw auto-negotiation!
       |  +--rw enable?               boolean
       |  +--ro negotiation-status?   enumeration
       +--rw duplex?                          duplex-type
       +--rw speed?                           eth-if-speed-type
       +--rw flow-control
       |  +--rw pause {ethernet-pause}?
       |  |  +--rw direction?    pause-fc-direction-type
       |  |  +--ro statistics
       |  |     +--ro in-frames-pause?    yang:counter64
       |  |     +--ro out-frames-pause?   yang:counter64
       |  +--rw pfc {ethernet-pfc}?
       |  |  +--rw enable?       boolean
       |  |  +--ro statistics
       |  |     +--ro in-frames-pfc?    yang:counter64
       |  |     +--ro out-frames-pfc?   yang:counter64
       |  +--rw force-flow-control?   boolean
       +--ro max-frame-length?                uint16
       +--ro mac-control-extension-control?   boolean
       +--ro frame-limit-slow-protocol?       uint64
       +--ro capabilities
       |  +--ro auto-negotiation?   boolean
       +--ro statistics
          +--ro frame
          |  +--ro in-total-frames?                yang:counter64
```

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

```
     |  +--ro in-total-octets?                yang:counter64
     |  +--ro in-frames?                      yang:counter64
     |  +--ro in-multicast-frames?            yang:counter64
     |  +--ro in-broadcast-frames?            yang:counter64
     |  +--ro in-error-fcs-frames?            yang:counter64
     |  +--ro in-error-undersize-frames?      yang:counter64
     |  +--ro in-error-oversize-frames?       yang:counter64
     |  +--ro in-error-mac-internal-frames?   yang:counter64
     |  +--ro out-frames?                     yang:counter64
     |  +--ro out-multicast-frames?           yang:counter64
     |  +--ro out-broadcast-frames?           yang:counter64
     |  +--ro out-error-mac-internal-frames?  yang:counter64
     +--ro phy
     |  +--ro in-error-symbol?   yang:counter64
     |  +--ro lpi
     |     +--ro in-lpi-transitions?    yang:counter64
     |     +--ro in-lpi-time?           decimal64
     |     +--ro out-lpi-transitions?   yang:counter64
     |     +--ro out-lpi-time?          decimal64
     +--ro mac-control
        +--ro in-frames-mac-control-unknown?      yang:counter64
        +--ro in-frames-mac-control-extension?    yang:counter64
        +--ro out-frames-mac-control-extension?   yang:counter64

module: ieee802-ethernet-interface-half-duplex
  augment /if:interfaces/if:interface/ieee802-eth-if:ethernet:
    +--rw dynamic-rate-control?   dynamic-rate-control-type {dynamic-rate-control}?
  augment /if:interfaces/if:interface/ieee802-eth-if:ethernet/ieee802-eth-if:capabilities:
    +--ro dynamic-rate-control-supported?   boolean {dynamic-rate-control}?
       augment    /if:interfaces/if:interface/ieee802-eth-if:ethernet/ieee802-eth-if:statistics/ieee802-eth-
if:frame:
    +--ro csma-cd {csma-cd}?
       +--ro in-errors-sqe-test?                yang:counter64
       +--ro out-frames-collision-single?       yang:counter64
       +--ro out-frames-collision-multiple?     yang:counter64
       +--ro out-frames-deferred?               yang:counter64
```

```
+--ro out-frames-collisions-excessive?   yang:counter64
+--ro out-collisions-late?               yang:counter64
+--ro out-errors-carrier-sense?          yang:counter64
+--ro collision-histogram* [collision-count]
   +--ro collision-count           yang:counter64
   +--ro collision-count-frames?   yang:counter64
```

5.3.2 YANG module

In the following YANG module definitions, should any discrepancy between the text of the description for individual YANG nodes and the corresponding definition in 5.2 through 5.3 of this clause occur, the definitions and mappings in 5.3 shall take precedence.

An ASCII text version of the Ethernet YANG module can be found at the following URL:[6]
https://github.com/YangModels/yang/tree/master/standard/ieee/published/802.3.

5.3.2.1 Ethernet interface module

```
module ieee802-ethernet-interface {
  yang-version 1.1;

  namespace
    "urn:ieee:std:802.3:yang:ieee802-ethernet-interface";

  prefix ieee802-eth-if;

  revision 2019-06-21 {
    description "Initial revision.";
  }

  import ietf-yang-types {
    prefix yang;
    reference "IETF RFC 6991";
  }

  import ietf-interfaces {
    prefix if;
    reference "IETF RFC 8343";
  }

  import iana-if-type {
    prefix ianaift;
    reference "http://www.iana.org/assignments/yang-parameters/
      iana-if-type@2018-07-03.yang";
  }

  organization
    "IEEE Std 802.3 Ethernet Working Group
     Web URL: http://www.ieee802.org/3/";

  contact
    "Web URL: http://www.ieee802.org/3/";

  description
    "This module contains YANG definitions for configuring IEEE Std
     802.3 Ethernet Interfaces.
     In this YANG module, 'Ethernet interface' can be interpreted
     as referring to 'IEEE Std 802.3 compliant Ethernet
```

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

```
   interfaces'.";

reference "IEEE Std 802.3-2018, unless dated explicitly";

typedef eth-if-speed-type {
  type decimal64 {
    fraction-digits 3;
  }
  units "Gb/s";
  description
    "Used to represent the configured, negotiated, or actual speed
     of an Ethernet interface in Gigabits per second (Gb/s),
     accurate to 3 decimal places (i.e., accurate to 1 Mb/s).";
}

typedef duplex-type {
  type enumeration {
    enum full {
      description
        "Full duplex.";
    }
    enum half {
      description
        "Half duplex.";
    }
    enum unknown {
      description
        "Link is currently disconnected or initializing.";
    }
  }
  default full;
  description
    "Used to represent the configured, negotiated, or actual
     duplex mode of an Ethernet interface.";
  reference "IEEE Std 802.3, 30.3.1.1.32, aDuplexStatus";
}

typedef pause-fc-direction-type {
  type enumeration {
    enum "disabled" {
      description
        "Flow-control disabled in both ingress and egress
         directions, i.e., PAUSE frames are not transmitted and
         PAUSE frames received in the ingress direction are
         discarded without processing.";
    }
    enum "ingress-only" {
      description
        "PAUSE frame based flow control is enabled in the ingress
         direction only, i.e., PAUSE frames may be transmitted to
         reduce the ingress traffic flow, but PAUSE frames received
         in the ingress direction are discarded without reducing
         the egress traffic rate.";
    }
```

```
      enum "egress-only" {
        description
          "PAUSE frame based flow control is enabled in the egress
           direction only, i.e., PAUSE frames are not transmitted,
           but PAUSE frames received in the ingress direction are
           processed to reduce the egress traffic rate.";
      }
      enum "bi-directional" {
        description
          "PAUSE frame based flow control is enabled in both ingress
           and egress directions, i.e., PAUSE frames may be
           transmitted to reduce the ingress traffic flow, and
           PAUSE frames received on ingress are processed to reduce
           the egress traffic rate.";
      }
      enum "undefined" {
        description
          "Link is currently disconnected or initializing.";
      }
    }
    description
      "Used to represent the configured, negotiated, or actual
       PAUSE frame-based flow control setting.";

    reference
      "IEEE Std 802.3.1, dot3PauseAdminMode and dot3PauseOperMode";
  }

  feature ethernet-pfc {
    description
      "This device supports Ethernet priority flow-control.";
  }

  feature ethernet-pause {
    description
      "This device supports Ethernet PAUSE.";
  }

  augment "/if:interfaces/if:interface" {
    when "derived-from-or-self(if:type, 'ianaift:ethernetCsmacd')" {
      description
        "Applies to all P2P Ethernet interfaces.";
    }
    description
      "Augment interface model with Ethernet interface
       specific configuration nodes.";

    container ethernet {
      description
        "Contains all Ethernet interface related configuration.";

      container auto-negotiation {
        presence
          "The presence of this container indicates that
```

```
    auto-negotiation is supported on this Ethernet
    interface.";
  description
    "Contains auto-negotiation transmission parameters

    This container contains a data node that allows the
    advertised duplex value in the negotiation to be
    restricted.

    If not specified then the default behavior for the duplex
    data node is to negotiate all available values for the
    particular type of Ethernet PHY associated with the
    interface.

    If auto-negotiation is enabled, and PAUSE frame based flow
    control has not been explicitly configured, then the
    default PAUSE frame based flow control capabilities that
    are negotiated allow for bi-directional or egress-only
    PAUSE frame based flow control.

    If auto-negotiation is enabled, and PAUSE frame based flow
    control has been explicitly configured, then the
    configuration settings restrict the values that may be
    negotiated. However, it should be noted that the protocol
    does not allow only egress PAUSE frame based flow control
    to be negotiated without also allowing bi-directional
    PAUSE frame based flow control.";
  reference
    "IEEE Std 802.3, Clause 28 and Annexes 28A-D";

  leaf enable {
    type boolean;
    default true;

    description
      "Controls whether auto-negotiation is enabled or
       disabled.
       For interface types that support auto-negotiation then
       it defaults to being enabled.

       For interface types that do not support auto-negotiation,
       the related configuration data is ignored.";
  }
  leaf negotiation-status {
    when "../enable = 'true'";
    type enumeration {
      enum in-progress {
        description
          "The auto-negotiation protocol is running and
           negotiation is currently in-progress.";
      }
      enum complete {
        description
          "The auto-negotiation protocol has completed
```

```
            successfully.";
        }
        enum failed {
          description
            "The auto-negotiation protocol has failed.";
        }
        enum unknown {
          description
            "The auto-negotiation status is not currently known,
             this could be because it is still negotiating or the
             protocol cannot run (e.g., if no medium is present).";
        }
        enum no-negotiation {
          description
            "No auto-negotiation is executed.
             The auto-negotation function is either not supported
             on this interface or has not been enabled.";
        }
      }
      config false;
      description
        "The status of the auto-negotiation protocol.";
      reference
        "IEEE 802.3, 30.6.1.1.4, aAutoNegAutoConfig";
    }

  }

  leaf duplex {
    type duplex-type;
    description
      "Operational duplex mode of the Ethernet interface.";
    reference
      "IEEE Std 802.3, 30.3.1.1.32 aDuplexStatus";
  }

  leaf speed {
    type eth-if-speed-type;
    units "Gb/s";
    description
      "Operational speed (data rate) of the Ethernet interface.
       The default value is implementation-dependent.";
  }

  container flow-control {
    description
      "Holds the different types of Ethernet PAUSE frame based
       flow control that can be enabled.";
    container pause {
      if-feature "ethernet-pause";
      description
        "IEEE Std 802.3 PAUSE frame based PAUSE frame based flow
         control.";
      reference
```

```
      "IEEE Std 802.3, Annex 31B";
    leaf direction {
      type pause-fc-direction-type;
      description
        "Indicates which direction PAUSE frame based flow
         control is enabled in, or whether it is disabled.
         The default flow-control settings are vendor specific.
         If auto-negotiation is enabled, then PAUSE based
         flow-control is negotiated by default.
         The default value is implementation-dependent.";
    }

    container statistics {
      config false;
      description
        "Contains the number of PAUSE frames received or
         transmitted.";
      leaf in-frames-pause {
        type yang:counter64;
        units frames;
        description
          "A count of PAUSE MAC Control frames transmitted on
           this Ethernet interface.

           Discontinuities in the values of counters in
           this container can occur at re-initialization of the
           management system, and at other times as indicated
           by the value of the 'discontinuity-time' leaf
           defined in the ietf-interfaces YANG module
           (IETF RFC 8343).";
        reference
          "IEEE Std 802.3, 30.3.4.3 aPAUSEMACCtrlFramesReceived";
      }
      leaf out-frames-pause {
        type yang:counter64;
        units frames;
        description
          "A count of PAUSE MAC Control frames transmitted on
           this Ethernet interface.

           Discontinuities in the values of counters in
           this container can occur at re-initialization of the
           management system, and at other times as indicated
           by the value of the 'discontinuity-time' leaf
           defined in the ietf-interfaces YANG module
           (IETF RFC 8343).";
        reference
          "IEEE Std 802.3, 30.3.4.2
           aPAUSEMACCtrlFramesTransmitted";
      }
    }
  }

  container pfc {
```

```
if-feature "ethernet-pfc";
description
  "IEEE Std 802.3 Priority-based flow control.";
reference
  "IEEE Std 802.3, Annex 31D";

leaf enable {
  type boolean;

  description
    "True indicates that IEEE Std 802.3 priority-based
     flow control is enabled, false indicates that
     IEEE Std 802.3 priority-based flow control is disabled.
     For interfaces that have auto-negotiation,
     the priority-based flow control is enabled by default.";
}

container statistics {
  config false;
  description
    "This container collects all statistics for
     Ethernet interfaces.";

  leaf in-frames-pfc {
    type yang:counter64;
    units frames;
    description
      "A count of PFC MAC Control frames received on this
       Ethernet interface.

       Discontinuities in the values of counters in
       this container can occur at re-initialization of the
       management system, and at other times as indicated
       by the value of the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module
       (IETF RFC 8343).";
    reference
      "IEEE Std 802.3.1, dot3HCInPFCFrames";
  }

  leaf out-frames-pfc {
    type yang:counter64;
    units frames;
    description
      "A count of PFC MAC Control frames transmitted on
       this interface.

       Discontinuities in the values of counters in
       this container can occur at re-initialization of the
       management system, and at other times as indicated
       by the value of the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module
       (IETF RFC 8343).";
    reference
```

```
            "IEEE Std 802.3.1, dot3HCInPFCFrames";
        }
      }
    }

    leaf force-flow-control {
      type boolean;
      default false;
      description
        "Explicitly forces the local PAUSE frame based flow control
         settings regardless of what has been negotiated.

         Since the auto-negotiation of flow-control settings
         does not allow all sane combinations to be negotiated
         (e.g., consider a device that is only capable of sending
         PAUSE frames connected to a peer device that is only
         capable of receiving and acting on PAUSE frames) and
         failing to agree on the flow-control settings does not
         cause the auto-negotiation to fail completely, then it is
         sometimes useful to be able to explicitly enable
         particular PAUSE frame based flow control settings on
         the local device regardless of what is being advertised
         or negotiated.";
      reference
        "IEEE Std 802.3, Table 28B-3";
    }
  }

  leaf max-frame-length {
    type uint16;
    units octets;
    config false;
    description
      "This indicates the MAC frame length (including FCS bytes)
       at which frames are dropped for being too long.";
    reference
      "IEEE Std 802.3, 30.3.1.1.37 aMaxFrameLength";
  }

  leaf mac-control-extension-control {
    type boolean;
    config false;
    description
      "A value that identifies the current EXTENSION MAC Control
       function, as specified in IEEE Std 802.3, Annex 31C.";
    reference
      "IEEE Std 802.3, 30.3.8.3 aEXTENSIONMACCtrlStatus
       IEEE Std 802.3.1, dot3ExtensionMacCtrlStatus ";
  }

  leaf frame-limit-slow-protocol {
    type uint64;
    units f/s;
    default 10;
```

```
    config false;
    description
      "The maximum number of Slow Protocol frames of a given
       subtype that can be transmitted in a one second interval.
       The default value is 10.";
    reference
      "IEEE Std 802.3, 30.3.1.1.38 aSlowProtocolFrameLimit";
  }

  container capabilities {
    config false;
    description
      "Container all Ethernet interface specific capabilities.";

    leaf auto-negotiation {
      type boolean;
      description
        "Indicates whether auto-negotiation may be configured on
         this interface.";
    }

  }

  container statistics {
    config false;
    description
      "Contains statistics specific to Ethernet interfaces.

       Discontinuities in the values of counters in the
       container can occur at re-initialization of the management
       system, and at other times as indicated by the value of
       the 'discontinuity-time' leaf defined in the
       ietf-interfaces YANG module (IETF RFC 8343).";

    container frame {
      description
        "Contains frame statistics specific to Ethernet
         interfaces.

         All octet frame lengths include the 4 byte FCS.

         Error counters are only reported once ... 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 Std 802.3 Layer
         Management, counted exclusively according to the error
         status presented to the LLC.

         A frame that is counted by an instance of this object is
         also counted by the corresponding instance of 'in-errors'
         leaf defined in the ietf-interfaces YANG module
         (IETF RFC 8343).
```

```
    Discontinuities in the values of counters in the
    container can occur at re-initialization of the
    management system, and at other times as indicated by
    the value of the 'discontinuity-time' leaf defined in
    the ietf-interfaces YANG module (IETF RFC 8343).";

  leaf in-total-frames {
    type yang:counter64;
    units frames;
    description
      "The total number of frames (including bad frames)
       received on the Ethernet interface.

       This counter is calculated by summing the following
       IEEE Std 802.3, Clause 30 counters:
       aFramesReceivedOK +
       aFrameCheckSequenceErrors +
       aAlignmentErrors +
       aFrameTooLongErrors +
       aFramesLostDueToIntMACRcvError

       Also see the 'description' statement associated with
       the parent 'statistics' container for additional
       common semantics related to this counter.";

    reference
      "IEEE Std 802.3, Clause 30 counters, as specified
       in the description above.";
  }

  leaf in-total-octets {
    type yang:counter64;
    units octets;
    description
      "The total number of octets of data (including those in
       bad frames) received on the Ethernet interface.

       Includes the 4-octet FCS.

       Also see the 'description' statement associated with
       the parent 'statistics' container for additional
       common semantics related to this counter.";

    reference
      "IETF RFC 2819, etherStatsOctets";
  }

  leaf in-frames {
    type yang:counter64;
    units frames;
    description
      "A count of frames (including unicast, multicast and
       broadcast) that have been successfully received on the
```

```
        Ethernet interface.

        This count does not include frames received with
        frame-too-long, FCS, length or alignment errors, or
        frames lost due to internal MAC sublayer error.

        Also see the 'description' statement associated with
        the parent 'statistics' container for additional
        common semantics related to this counter.";

    reference
      "IEEE Std 802.3, 30.3.1.1.5 aFramesReceivedOK";
  }

  leaf in-multicast-frames {
    type yang:counter64;
    units frames;
    description
      "A count of multicast frames that have been
       successfully received on the Ethernet interface.

       This counter represents a subset of the frames counted
       by in-frames.

       This count does not include frames received with
       frame-too-long, FCS, length or alignment errors, or
       frames lost due to internal MAC sublayer error.

       Also see the 'description' statement associated with
       the parent 'statistics' container for additional
       common semantics related to this counter.";

    reference
      "IEEE Std 802.3, 30.3.1.1.21 aMulticastFramesReceivedOK";
  }

  leaf in-broadcast-frames {
    type yang:counter64;
    units frames;
    description
      "A count of broadcast frames that have been
       successfully received on the Ethernet interface.

       This counter represents a subset of the frames counted
       by in-frames.

       This count does not include frames received with
       frame-too-long, FCS, length or alignment errors, or
       frames lost due to internal MAC sublayer error.

       Also see the 'description' statement associated with
       the parent 'statistics' container for additional
       common semantics related to this counter.";
```

```
    reference
      "IEEE Std 802.3, 30.3.1.1.22 aBroadcastFramesReceivedOK";
  }

  leaf in-error-fcs-frames {
    type yang:counter64;
    units frames;
    description
      "A count of receive frames that are of valid length,
       but do not pass the FCS check, regardless of whether
       or not the frames are an integral number of octets in
       length.

       This count effectively comprises
       aFrameCheckSequenceErrors and aAlignmentErrors added
       together.

       Also see the 'description' statement associated with
       the parent 'statistics' container for additional
       common semantics related to this counter.";

    reference
      "IEEE Std 802.3, 30.3.1.1.6 aFrameCheckSequenceErrors;
       IEEE Std 802.3, 30.3.1.1.7 aAlignmentErrors";
  }

  leaf in-error-undersize-frames {
    type yang:counter64;
    units frames;
    description
      "A count of frames received on a particular Ethernet
       interface that are less than 64 bytes in length, and
       are discarded.

       This counter is incremented regardless of whether the
       frame passes the FCS check.

       Also see the 'description' statement associated with
       the parent 'statistics' container for additional
       common semantics related to this counter.";

    reference
      "IETF RFC 2819, etherStatsUndersizePkts and
       etherStatsFragments";
  }

  leaf in-error-oversize-frames {
    type yang:counter64;
    units frames;
    description
      "A count of frames received on a particular Ethernet
       interface that exceed the maximum permitted frame
       size, that is specified in max-frame-length, and are
       discarded.
```

```
        This counter is incremented regardless of whether the
        frame passes the FCS check.

        Also see the 'description' statement associated with
        the parent 'statistics' container for additional
        common semantics related to this counter.";

    reference "IEEE Std 802.3, 30.3.1.1.25 aFrameTooLongErrors";
  }

  leaf in-error-mac-internal-frames {
    type yang:counter64;
    units frames;
    description
      "A count of frames for which reception on a particular
       Ethernet 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 in-error-fcs-frames, in-error-undersize-frames,
       or in-error-oversize-frames.  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
       Ethernet interface that are not otherwise counted.

       Also see the 'description' statement associated with
       the parent 'statistics' container for additional
       common semantics related to this counter.";

    reference
      "IEEE Std 802.3, 30.3.1.1.15
       aFramesLostDueToIntMACRcvError";
  }

  leaf out-frames {
    type yang:counter64;
    units frames;
    description
      "A count of frames (including unicast, multicast and
       broadcast) that have been successfully transmitted on
       the Ethernet interface.

       Also see the 'description' statement associated with
       the parent 'statistics' container for additional
       common semantics related to this counter.";

    reference
      "IEEE Std 802.3, 30.3.1.1.2 aFramesTransmittedOK";
  }
```

```
leaf out-multicast-frames {
  type yang:counter64;
  units frames;
  description
    "A count of multicast frames that have been
     successfully transmitted on the Ethernet interface.

     This counter represents a subset of the frames counted
     by out-frames.

     Also see the 'description' statement associated with
     the parent 'statistics' container for additional
     common semantics related to this counter.";

  reference
    "IEEE Std 802.3, 30.3.1.1.18 aMulticastFramesXmittedOK";
}

leaf out-broadcast-frames {
  type yang:counter64;
  units frames;
  description
    "A count of broadcast frames that have been
     successfully transmitted on the Ethernet interface.

     This counter represents a subset of the frames counted
     by out-frames.

     Also see the 'description' statement associated with
     the parent 'statistics' container for additional
     common semantics related to this counter.";

  reference
    "IEEE Std 802.3, 30.3.1.1.19 aBroadcastFramesXmittedOK";
}

leaf out-error-mac-internal-frames {
  type yang:counter64;
  units frames;
  description
    "A count of frames for which transmission on a
     particular Ethernet interface fails due to an internal
     MAC sublayer transmit error.

     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 Ethernet
     interface that are not otherwise counted.

     Also see the 'description' statement associated with
     the parent 'statistics' container for additional
     common semantics related to this counter.";
```

```
    reference
      "IEEE Std 802.3, 30.3.1.1.12
       aFramesLostDueToIntMACXmitError";
  }
}

container phy {
  description
    "Ethernet statistics related to the PHY layer.

     Discontinuities in the values of counters in the
     container can occur at re-initialization of the
     management system, and at other times as indicated by
     the value of the 'discontinuity-time' leaf defined in
     the ietf-interfaces YANG module (IETF RFC 8343).";

  leaf in-error-symbol {
    type yang:counter64;
    units errors;
    description
      "A count of the number of symbol errors that have
       occurred.

       For the precise definition of when the symbol error
       counter is incremented, please see the 'description'
       text associated with aSymbolErrorDuringCarrier,
       specified in IEEE Std 802.3, 30.3.2.1.5.

       Also see the 'description' statement associated with
       the parent 'phy-statistics' container for additional
       common semantics related to this counter.";
    reference
      "IEEE Std 802.3, 30.3.2.1.5 aSymbolErrorDuringCarrier";
  }

  container lpi {
    description
      "Physical Ethernet statistics for the energy efficiency
       related low power idle indications.";

    leaf in-lpi-transitions {
      type yang:counter64;
      units transitions;
      description
        "A count of occurrences of the transition from
         DEASSERT to ASSERT of the LPI_INDICATE
         parameter. The indication reflects the state of the
         PHY according to the requirements of the RS (see
         IEEE Std 802.3, 22.7, 35.4, and 46.4).

         Also see the 'description' statement associated with
         the parent 'phy-statistics' container for additional
         common semantics related to this counter.";
```

```
    reference
      "IEEE Std 802.3, 30.3.2.1.11 aReceiveLPITransitions";
  }

  leaf in-lpi-time {
    type decimal64 {
      fraction-digits 6;
    }
    units seconds;
    description
      "A count reflecting the total amount of time (in
       seconds) that the LPI_REQUEST parameter has the
       value ASSERT.  The request is indicated to the PHY
       according to the requirements of the RS (see IEEE Std
       802.3, 22.7, 35.4, and 46.4).

       Also see the 'description' statement associated with
       the parent 'phy-statistics' container for additional
       common semantics related to this counter.";

    reference
      "IEEE Std 802.3, 30.3.2.1.9 aReceiveLPIMicroseconds";
  }

  leaf out-lpi-transitions {
    type yang:counter64;
    units transitions;
    description
      "A count of occurrences of the transition from state
       LPI_DEASSERTED to state LPI_ASSERTED in the LPI
       transmit state diagram of the RS. The state
       transition corresponds to the assertion of the
       LPI_REQUEST parameter. The request is indicated to
       the PHY according to the requirements of the RS (see
       IEEE Std 802.3, 22.7, 35.4, 46.4.)

       Also see the 'description' statement associated with
       the parent 'phy-statistics' container for additional
       common semantics related to this counter.";

    reference
      "IEEE Std 802.3, 30.3.2.1.10 aTransmitLPITransitions";
  }

  leaf out-lpi-time {
    type decimal64 {
      fraction-digits 6;
    }
    units seconds;
    description
      "A count reflecting the total amount of time (in
       seconds) that the LPI_INDICATION parameter has the
       value ASSERT.  The request is indicated to the PHY
```

```
          according to the requirements of the RS (see IEEE
          802.3, 22.7, 35.4, and 46.4).

          Also see the 'description' statement associated with
          the parent 'phy-statistics' container for additional
          common semantics related to this counter.";

        reference
          "IEEE Std 802.3, 30.3.2.1.8 aTransmitLPIMicroseconds";
      }
    }
  }

  container mac-control {
    description
      "A group of statistics specific to MAC Control operation
       of selected Ethernet interfaces.

       Discontinuities in the values of counters in the
       container can occur at re-initialization of the
       management system, and at other times as indicated by
       the value of the 'discontinuity-time' leaf defined in
       the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3.1, dot3ExtensionTable";

    leaf in-frames-mac-control-unknown {
      type yang:counter64;
      units frames;
      description
        "A count of MAC Control frames with an unsupported
         opcode received on this Ethernet interface.

         Frames counted against this counter are also counted
         against in-discards defined in the ietf-interfaces
         YANG module (IETF RFC 8343).

         Also see the 'description' statement associated with
         the parent 'mac-control-statistics' container for
         additional semantics.";
      reference
        "IEEE Std 802.3, 30.3.3.5 aUnsupportedOpcodesReceived";
    }

    leaf in-frames-mac-control-extension {
      type yang:counter64;
      units frames;
      description
        "The count of Extension MAC Control frames received on
         this Ethernet interface.

         Also see the 'description' statement associated with
         the parent 'mac-control-statistics' container for
```

```
              additional semantics.";
            reference
              "IEEE Std 802.3, 30.3.8.2
               aEXTENSIONMACCtrlFramesReceived";
          }

          leaf out-frames-mac-control-extension {
            type yang:counter64;
            units frames;
            description
              "The count of Extension MAC Control frames transmitted
               on this Ethernet interface.

               Also see the 'description' statement associated with
               the parent 'mac-control-statistics' container for
               additional semantics.";
            reference
              "IEEE Std 802.3, 30.3.8.1
               aEXTENSIONMACCtrlFramesTransmitted";
          }
        }
      }
    }
  }
}
```

5.3.2.2 Ethernet interface module (half-duplex)

```
module ieee802-ethernet-interface-half-duplex {

  yang-version 1.1;

  namespace
    "urn:ieee:std:802.3:yang:ieee802-ethernet-interface-half-duplex";

  prefix ieee802-eth-half-duplex;

revision 2019-06-21 {
    description "Initial revision.";
  }

  import ietf-yang-types {
    prefix yang;
    reference "IETF RFC 6991";
  }

  import ietf-interfaces {
    prefix if;
    reference "IETF RFC 8343";
  }

  import iana-if-type {
    prefix ianaift;
    reference "http://www.iana.org/assignments/yang-parameters/
      iana-if-type@2018-07-03.yang";
  }
```

```
import ieee802-ethernet-interface {
  prefix ieee802-eth-if;
}

organization
  "IEEE Std 802.3 Ethernet Working Group
   Web URL: http://www.ieee802.org/3/";

contact
  "Web URL: http://www.ieee802.org/3/";

description
  "This module contains YANG definitions for configuring Ethernet
   interfaces that are deprecated, and are no longer
   widely used in the industry. The definitions are maintained for
   backwards compatibility purposes, but the general expectation is
   that this module is not anticipated to be widely implemented.";
reference
  "IEEE Std 802.3-2018, unless dated explicitly";

feature dynamic-rate-control {
  description
    "This feature indicates that the device supports Ethernet
     interfaces lowering the average data rate of the MAC sublayer,
     with frame granularity, by using Rate Control to dynamically
     increase the inter-packet gap for some types of Ethernet
     interface.
     Only valid for Ethernet interfaces operating at speeds (data rates)
     above 1000 Mb/s.";
  reference "IEEE Std 802.3, 30.3.1.1.33 aRateControlAbility";
}

feature csma-cd {
  description
    "This feature indicates that the device supports Ethernet
     interfaces running at half-duplex using CSMA/CD.";
}

typedef dynamic-rate-control-type {
  type enumeration {
    enum disabled {
      description
        "Dynamic rate control is disabled";
    }

    enum "sonet-oc192" {
      value 2;
      description
        "Dynamic rate control is enabled for a 10 Gb/s Ethernet
         interface to SONET/SDH OC192/STM64.";
    }
  }
  default disabled;
  description
    "Allowed values for dynamic-rate-control.";
  reference
    "IEEE Std 802.3, 4.4.2 ipgStretchRatio and 30.3.1.1.34
     aRateControlStatus";
```

```
}

 augment "/if:interfaces/if:interface/ieee802-eth-if:ethernet" {
 when "derived-from-or-self(../if:type, 'ianaift:ethernetCsmacd')
    and ieee802-eth-if:duplex = 'half'" {
    description
      "Applies to half-duplex Ethernet interfaces.";
  }

  description
    "Augment with Ethernet interface configuration parameters
     for half-duplex operation.";

  leaf dynamic-rate-control {
    if-feature "dynamic-rate-control";
    type dynamic-rate-control-type;
    description
      "Enables dynamic rate control and specifies what speed (data rate)
       the dynamic rate control is operating at. The value of this attribute
       is constrained by the MAC data rate and hardware support.
       The default value is implementation-dependent.";
    reference
      "IEEE Std 802.3, 30.3.1.1.34 aRateControlStatus";
  }
}

augment "/if:interfaces/if:interface/ieee802-eth-if:ethernet/" +
  "ieee802-eth-if:capabilities"{
   when  "derived-from-or-self(../../if:type,
     'ianaift:ethernetCsmacd') and ../ieee802-eth-if:duplex = 'half'" {
     description "Applies to half-duplex Ethernet interfaces";
  }

  description
    "Augment with configuration capabilities for half-duplex
     Ethernet interface.";

  leaf dynamic-rate-control-supported {
    if-feature "dynamic-rate-control";
    type boolean;
    default false;
    description
      "Indicates whether the Ethernet interface supports lowering
       the average data rate of the MAC sublayer, with frame
       granularity, by using Rate Control to dynamically increase
       the inter-packet gap.
       Only valid for Ethernet interfaces operating at speeds (data rates)
       above 1000 Mb/s.";
    reference
      "IEEE Std 802.3, 30.3.1.1.33 aRateControlAbility";
  }
}

augment "/if:interfaces/if:interface/ieee802-eth-if:ethernet/" +
  "ieee802-eth-if:statistics/ieee802-eth-if:frame" {
  when "derived-from-or-self(../../../if:type,
    'ianaift:ethernetCsmacd') and ../../ieee802-eth-if:duplex = 'half'" {
    description
      "Applies to half-duplex Ethernet interfaces.";
```

```
  }
  description
    "Augment with statistics for half-duplex Ethernet interface.";

  container "csma-cd" {
    if-feature "csma-cd";
    description
      "Holds counters that are specific to CDMA/CD half-duplex
       operation of Ethernet interfaces.
       Discontinuities in the values of the counters in this
       container can occur at re-initialization of the management
       system, and at other times as indicated by the value of the
       'discontinuity-time' leaf defined in the ietf-interfaces
       YANG module (IETF RFC 8343).";

    leaf in-errors-sqe-test {
      type yang:counter64;
      units errors;
      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 Ethernet interfaces
         operating at speeds (data rates) greater than 10 Mb/s, or on
         Ethernet 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 the
         'discontinuity-time' leaf defined in the ietf-interfaces
         YANG module (IETF RFC 8343).";
      reference
        "IEEE Std 802.3, 7.2.4.6, and 30.3.2.1.4 aSQETestErrors";
    }

    leaf out-frames-collision-single {
      type yang:counter64;
      units frames;
      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
         'out-unicast-frames', 'out-broadcast-frames', or
         'out-multicast-frames', and is not counted by the
         corresponding instance of the
         'out-frames-collision-multiple'.

         This counter does not increment when the Ethernet
         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 the
         'discontinuity-time' leaf defined in the ietf-interfaces
         YANG module (IETF RFC 8343).";
      reference
        "IEEE Std 802.3, 30.3.1.1.3 aSingleCollisionFrames";
    }
```

```
leaf out-frames-collision-multiple {
  type yang:counter64;
  units frames;
  description
    "A count of frames that are involved in multiple
     collisions, 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 'out-unicast-frames', 'out-broadcast-frames', or
     'out-multicast-frames', and is not counted by the
     corresponding instance of the 'out-frames-collision-single'.
     This counter does not increment when the Ethernet
     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 the
     'discontinuity-time' leaf defined in the ietf-interfaces
     YANG module (IETF RFC 8343).";
  reference
    "IEEE Std 802.3, 30.3.1.1.4 aMultipleCollisionFrames";
}

leaf out-frames-deferred {
  type yang:counter64;
  units frames;
  description
    "A count of frames for which the first transmission attempt
     on a particular Ethernet interface is delayed because the
     medium is busy.
     A deferred frame that is not subject to any number of
     collisions is not counted by an instance of
     'out-frames-collision-single' or
     'out-frames-collision-multiple' objects.
     This counter does not increment when the Ethernet
     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 the
     'discontinuity-time' leaf defined in the ietf-interfaces
     YANG module (IETF RFC 8343).";
  reference
    "IEEE Std 802.3, 30.3.1.1.9 aFramesWithDeferredXmissions";
}

leaf out-frames-collisions-excessive {
  type yang:counter64;
  units frames;
  description
    "A count of frames for which transmission on a particular
     Ethernet interface fails due to excessive collisions.

     This counter does not increment when the Ethernet
     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 the
     'discontinuity-time' leaf defined in the ietf-interfaces
     YANG module (IETF RFC 8343).";
```

```
    reference
      "IEEE Std 802.3, 30.3.1.1.11 aFramesAbortedDueToXSColls";
  }

  leaf out-collisions-late {
    type yang:counter64;
    units collisions;
    description
      "The number of times that a collision is detected on a
       particular Ethernet 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 Ethernet
       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 the
       'discontinuity-time' leaf defined in the ietf-interfaces
       YANG module (IETF RFC 8343).";
    reference
      "IEEE Std 802.3, 30.3.1.1.10 aLateCollisions";
  }

  leaf out-errors-carrier-sense {
    type yang:counter64;
    units errors;
    description
      "The number of times that the carrier sense condition was
       lost or never asserted when attempting to transmit a frame
       on a particular Ethernet 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 Ethernet
       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 the
       'discontinuity-time' leaf defined in the ietf-interfaces
       YANG module (IETF RFC 8343).";
    reference
      "IEEE Std 802.3, 30.3.1.1.13 aCarrierSenseErrors";
  }

  list collision-histogram {
    key collision-count;
    description
      "A collection of collision histograms for a particular
       interface.";
    reference
      "IEEE Std 802.3, 30.3.1.1.30 aCollisionFrames";
    leaf collision-count {
      type yang:counter64;
      units collisions;
      description
```

```
            "The number of per-frame media collisions for which a
             particular collision histogram cell represents the
             frequency on a particular interface.";
          }
          leaf collision-count-frames {
            type yang:counter64;
            units frames;
            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 an
               interface after experiencing exactly 4 collisions would
               be indicated by incrementing only collision-count-frames
               object associated with the collision-count value of
               4. No other instance of collision-count-frames 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 the
               'discontinuity-time' leaf defined in the ietf-interfaces
               YANG module (IETF RFC 8343).";
          }
        }
      }
    }
  }
}
```

6. YANG module for Ethernet data terminal equipment (DTE) power via medium dependent interface (MDI) and Power over Data Lines (PoDL)

6.1 Introduction

This clause defines a YANG module to manage power via MDI Power Sourcing Equipment (PSE) and Power over Data Line (PoDL) PSE.

IEEE Std 802.3 defines the hardware registers that allow management interfaces to be built for a DTE Power via MDI and Power over Data Line device. The YANG module defined in this clause extends the Ethernet-interface YANG data modules defined in Clause 5 with the management objects required for the management of PoE and PoDL devices and ports.

6.2 YANG module structure

The *ieee802-ethernet-pse* YANG module of this clause is focused on the configuration and monitoring of the Power over Ethernet (PoE) function defined in IEEE Std 802.3, including power via MDI, as well as Power over Data Line which can also be considered as the single pair PoE. The module augments the *ieee802-ethernet-interface* YANG module with attributes for the PoE function. The module is partitioned into two major containers.

The PoE PSE container describes a multi-pair PSE, while the PoDL PSE describes a single-pair PSE.

6.3 Security considerations for Ethernet data terminal equipment (DTE) power via medium dependent interface (MDI) and Power over Data Line Module

There are a number of data nodes defined in this YANG module that are configurable as read-write. Such data nodes may be considered sensitive or vulnerable in some network environments. The support for configuration operations in a non-secure environment without proper protection can have a negative effect on network operations.

Setting the following data nodes to incorrect values can result in improper operation of the PSE, including the possibility that the Powered Device (PD) does not receive power from the PSE port:

— pse-enable
— powering-pairs

Some of the readable operational states in this module may be considered sensitive or vulnerable in some network environments. These are as follows:

— pairs-control-ability
— classifications
— pd-power-class
— pse-type
— detected-pd-type

It is thus important to control GET access to these data nodes and to possibly encrypt their values when sending them over the network.

6.4 Mapping of IEEE Std 802.3, Clause 30 managed objects

This subclause contains the mapping between YANG data nodes included in *ieee802-ethernet-pse* (see Table 6–1) YANG module, managed objects, and attributes defined in IEEE Std 802.3, Clause 30.

Table 6–1—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-pse* YANG data nodes

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-pse* YANG data nodes		
Managed object(s)	**Attribute(s)**		**Container(s)**	**Data node(s)**	**R/W**
N/A	N/A		interfaces/interface/ethernet/pse	supported-pse-type	R
oPSE	aPSEAdminState	30.9.1	interfaces/interface/ethernet/pse/multi-pair	pse-enable	R
	aPSEPowerPairs			powering-pairs	R/W
	aPSEPowerPairsControlAbility			pairs-control-ability	R
	aPSEPowerDetectionStatus			detection-status	R
	aPSEPowerClassification			classifications	R
	aPSEActualPower			actual-power	R
	aPSEPowerAccuracy			power-accuracy	R
	aPSEInvalidSignatureCounter		interfaces/interface/ethernet/pse/multi-pair/statistics	invalid-signature	R
	aPSEPowerDeniedCounter			power-denied	R
	aPSEOverLoadCounter			overload	R
	aPSEShortCounter			short	R
	aPSEMPSAbsentCounter			mps-absent	R
	aPSECumulativeEnergy			cumulative-energy	R

Table 6–1—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-pse* YANG data nodes *(continued)*

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-pse* YANG data nodes		
Managed object(s)	**Attribute(s)**		**Container(s)**	**Data node(s)**	**R/W**
oPoDLPSE	aPoDLPSEAdminState	30.15	interfaces/interface/ethernet/pse/single-pair	pse-enable	R
	aPoDLPSEPowerDetectionStatus			detection-status	R
	aPoDLPSEType			podl-type	R
	aPoDLPSEDetectedPDType			detected-pd-type	R
	aPoDLPSEDetectedPDPowerClass			pd-power-class	R
	aPoDLPSEActualPower			actual-power	R
	aPoDLPSEPowerAccuracy			power-accuracy	R
	aPoDLPSEInvalidSignatureCounter		interfaces/interface/ethernet/pse/single-pair/statistics	invalid-signature	R
	aPoDLPSEInvalidClassCounter			invalid-class	R
	aPoDLPSEPowerDeniedCounter			power-denied	R
	aPoDLPSEOverLoadCounter			overload	R
	aPoDLPSEMaintainFullVoltageSignatureAbsentCounter			fvs-absent	R
	aPoDLPSECumulativeEnergy			cumulative-energy	R

6.5 YANG module definition[7]

The YANG module tree hierarchy uses terms defined in IETF RFC 8407.

6.5.1 Tree hierarchy

```
module: ieee802-ethernet-pse
  augment /if:interfaces/if:interface/ieee802-eth-if:ethernet:
    +--rw pse
       +--ro supported-pse-type?   identityref
       +--rw multi-pair!
       |  +--rw pse-enable?              boolean
       |  +--rw powering-pairs?          identityref
       |  +--ro pairs-control-ability?   boolean
       |  +--ro detection-status?        multi-pair-detection-state
       |  +--ro classifications?         power-class
       |  +--ro statistics
       |  |  +--ro power-denied?        yang:counter64
       |  |  +--ro invalid-signature?   yang:counter64
       |  |  +--ro mps-absent?          yang:counter64
       |  |  +--ro overload?            yang:counter64
       |  |  +--ro short?               yang:counter64
       |  |  +--ro cumulative-energy?   yang:counter64
       |  +--ro actual-power?            decimal64
       |  +--ro power-accuracy?          int64
       +--rw single-pair!
          +--rw pse-enable?         boolean
          +--ro detection-status?   single-pair-detection-state
          +--ro podl-type?          enumeration
          +--ro detected-pd-type?   enumeration
          +--ro pd-power-class?     power-class
          +--ro statistics
          |  +--ro power-denied?        yang:counter64
          |  +--ro invalid-signature?   yang:counter64
          |  +--ro invalid-class?       yang:counter64
          |  +--ro overload?            yang:counter64
          |  +--ro fvs-absence?         yang:counter64
          |  +--ro cumulative-energy?   yang:counter64
          +--ro actual-power?       decimal64
          +--ro power-accuracy?     int64
```

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

6.5.2 YANG module

In the following YANG module definition, should any discrepancy between the text of the description for individual YANG nodes and the corresponding definition in 6.2 through 6.5 of this clause occur, the definitions and mappings in 6.5 shall take precedence.

An ASCII text version of the YANG module can be found at the following URL:[8] https://github.com/YangModels/yang/tree/master/standard/ieee/published/802.3.

```
module ieee802-ethernet-pse{
  yang-version 1.1;
  namespace "urn:ieee:std:802.3:yang:ieee802-ethernet-pse";
  prefix ieee802-pse;

  revision 2019-06-21 {
    description "Initial revision.";
  }

  import ietf-interfaces {
    prefix "if";
    reference "IETF RFC 8343";
  }

  import ietf-yang-types {
    prefix yang;
    reference "IETF RFC 6991";
  }

  import ieee802-ethernet-interface {
    prefix ieee802-eth-if;
  }

  organization
    "IEEE 802.3 Ethernet Working Group
     Web URL: http://www.ieee802.org/3/";

  contact
    "Web URL: http://www.ieee802.org/3/";

  description
    "This module contains YANG definitions for configuring and
     managing ports with Power Over Ethernet feature defined by
     IEEE 802.3. It provides functionality roughly equivalent to
     that of the POWER-ETHERNET-MIB defined in IETF RFC 3621.";

  reference
    "IEEE Std 802.3-2018, unless dated explicitly";
  typedef multi-pair-detection-state {
    type enumeration {
      enum disabled {
        value 1;
```

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

```
        description "PSE disabled.";
      }
      enum searching {
        value 2;
        description "PSE is searching.";
      }
      enum deliveringPower {
        value 3;
        description "PSE is delivering power.";
      }
      enum fault {
        value 4;
        description "PSE fault detected.";
      }
      enum test {
        value 5;
        description "PSE test mode.";
      }
      enum otherFault {
        value 6;
        description "PSE implementation specific fault detected.";
      }
    }
    description
      "Detection state of a multi-pair PSE.";
    reference
      "IEEE Std 802.3, 30.9.1.1.5";
  }

  typedef single-pair-detection-state{
    type enumeration {
      enum unknown {
        value 1;
        description "True detection state unknown.";
      }
      enum disabled {
        value 2;
        description "PoDL PSE is disabled.";
      }
      enum searching {
        value 3;
        description "PoDL PSE is searching.";
      }
      enum deliveringPower {
        value 4;
        description "PoDL PSE is delivering power.";
      }
      enum sleep {
        value 5;
        description "PoDL PSE is in sleep state.";
      }
      enum idle {
        value 6;
        description "PoDL PSE is idle.";
```

```
      }
      enum error {
        value 7;
        description "PoDL PSE error.";
      }
    }

    description
      "Detection state of a PoDL PSE.";
    reference
      "IEEE Std 802.3, 30.15.1.1.3";
  }

  typedef power-class {
    type enumeration {
      enum class0 {
        value 1;
        description "Class 0";
      }
      enum class1 {
        value 2;
        description "Class 1";
      }
      enum class2 {
        value 3;
        description "Class 2";
      }
      enum class3 {
        value 4;
        description "Class 3";
      }
      enum class4 {
        value 5;
        description "Class 4";
      }
      enum class5 {
        value 6;
        description "Class 5 (for PoDL-only)";
      }
      enum class6 {
        value 7;
        description "Class 6 (for PoDL-only)";
      }
      enum class7 {
        value 8;
        description "Class 7 (for PoDL-only)";
      }
      enum class8 {
        value 9;
        description "Class 8 (for PoDL-only)";
      }
      enum class9 {
        value 10;
        description "Class 9 (for PoDL-only)";
```

```
      }
      enum unknown {
        value 11;
        description
          "Initializing, true Power Class not yet known
           (only for PoDL PSE).";
      }
    }
    description
      "Power class.";
    reference
      "IEEE Std 802.3, 30.9.1.1.6 aPSEPowerClassification and
       IEEE Std 802.3, 30.15.1.1.6 aPoDLPSEDetectedPDPowerClass.";
  }

  identity pse-type {
    description "Base type for PSE.";
  }

  identity multi-pair {
    base pse-type;
    description "PSE supports IEEE Std 802.3, Clause 33.";
  }

  identity single-pair {
    base pse-type;
    description "PSE support IEEE Std 802.3, Clause 104.";
  }

  identity powering-pairs {
    description "Base type for powering pairs.";
  }
  identity signal {
    base powering-pairs;
    description "The signal pair is in use.";
  }
  identity spare {
    base powering-pairs;
    description "The spare pair is in use.";
  }

  augment "/if:interfaces/if:interface/ieee802-eth-if:ethernet" {
    description
      "Augments ethernet interface configuration model with
       nodes specific to DTE Power via MDI devices and ports";

    container pse {
      description
        "DTE Power via MDI port configuration";
      reference
        "IEEE Std 802.3, 30.9.1 PoE PSE & IEEE Std 802.3, 30.15.1 PoDL
         PSE";

      leaf supported-pse-type {
```

```
    type identityref {
      base ieee802-pse:pse-type ;
    }
    config false;
    description
      "PSE may support IEEE Std 802.3, Clause 33 or
       IEEE Std 802.3, Clause 104.";
  }

  container multi-pair {
    presence "PSE port supports IEEE Std 802.3, Clause 33.";

    description
      "PSE port configuration in IEEE Std 802.3, 30.9.1.";

    leaf pse-enable {
      type boolean;
      default false;
      description
        "When true enables the PSE function on the interface,
         when false disables the PSE function on the interface.";
      reference
        "IEEE Std 802.3, 30.9.1.1.2 aPSEAdminState";
    }

    leaf powering-pairs {
      type identityref{
        base powering-pairs;
    }
      description
        "Describes or controls the PSE pairs in use. If the value of
         pairs-control-ability is true, this object is
         writeable.";

      reference
        "IEEE Std 802.3, 30.9.1.1.4 aPSEPowerPairs";
    }

    leaf pairs-control-ability {
      type boolean;
      default true;
      config false;
      description
        "Describes the ability to control switching the
         power sourcing pins of the PSE.";
      reference
        "IEEE Std 802.3, 30.9.1.1.3 aPSEPowerPairsControlAbility";
    }

    leaf detection-status {
      type multi-pair-detection-state;
      config false;
      description
        "Describes the operational status of the port
```

```
      PD detection.";
    reference
      "IEEE Std 802.3, 30.9.1.1.5 aPSEPowerDetectionStatus";
  }

  leaf classifications {
    when "../detection-status = 'deliveringPower'" {
      description
        "This node only applies when the detection status is
         delivering power.";
    }
    type power-class;
    config false;
    description
      "The power class of the PSE port.";
    reference
      "IEEE Std 802.3, 30.9.1.1.6 aPSEPowerClassfication";
  }

  container statistics {
    config false;
    description
      "statistics information of the multi-pair port.";

    leaf power-denied {
      type yang:counter64;
      description
        "This counter is incremented when the PSE state diagram
         enters the POWER_DENIED state, per IEEE Std 802.3,
         Figure 33-9.";
      reference
        "IEEE Std 802.3, 30.9.1.1.8 aPSEPowerDeniedCounter";
    }

    leaf invalid-signature {
      type yang:counter64;
      description
        "This counter is incremented when the PSE state diagram
         enters the SIGNATURE_INVALID state per IEEE Std 802.3,
         Figure 33-9.";
      reference
        "IEEE Std 802.3, 30.9.1.1.7 aPSEInvalidSignatureCounter";
    }

    leaf mps-absent {
      type yang:counter64;
      description
        "This counter is incremented when the PSE
         transitions directly from the POWER_ON state to the
         IDLE state due to tmpdo_timer_done being asserted,
         per IEEE Std 802.3, Figure 33-9.";
      reference
        "IEEE Std 802.3, 30.9.1.1.11 aPSEMPSAbsentCounter";
    }
```

```
    leaf overload {
      type yang:counter64;
      description
        "This counter is incremented when the PSE state diagram
         enters the ERROR_DELAY state due to the ovld_detected
         variable being TRUE, per IEEE Std 802.3, Figure 33-9.";
      reference
        "IEEE Std 802.3, 30.9.1.1.9 aPSEOverLoadCounter";
    }

    leaf short {
      type yang:counter64;
      description
        "This counter is incremented when the PSE state diagram
         enters the ERROR_DELAY state due to the short_detected
         variable being TRUE, per IEEE Std 802.3, Figure 33-9.";
      reference
        "IEEE Std 802.3, 30.9.1.1.10 aPSEShortCounter";
    }

    leaf cumulative-energy {
      type yang:counter64;
      units millijoules;
      description
        "The cumulative energy supplied by the PSE as measured at
         the MDI in millijoules.";

      reference
        "IEEE Std 802.3, 30.9.1.1.14 aPSECumulativeEnergy";
    }

  }

  leaf actual-power {
    type decimal64 {
      fraction-digits 4;
    }

    units milliwatts;
    config false;
    description
      "The actual power drawn by a PD over the port.";
    reference
      "IEEE Std 802.3, 30.9.1.1.12 aPSEActualPower";
  }

  leaf power-accuracy {
    type int64;
    units milliwatts;
    config false;
    description
      "An integer value indicating the accuracy
       associated with power-accuracy in +/- milliwatts.";
```

```
      reference
        "IEEE Std 802.3, 30.9.1.1.13 aPSEPowerAccuracy";
    }

  }

  container single-pair {
    presence "PSE port working in PoDL.";

    description
      "PoDL PSE configuration as defined in
       IEEE Std 802.3, 30.15.1.";

    leaf pse-enable {
      type boolean;
      default false;
      description
        "When true enables the PSE function on the interface,
         when false disables the PSE function on the interface.";
      reference
        "IEEE Std 802.3, 30.15.1.1.2 aPoDLPSEAdminState";
    }

    leaf detection-status {
      type single-pair-detection-state;
      config false;
      description
        "Indicates the current status of the PoDL PSE.";
      reference
        "IEEE Std 802.3, 30.15.1.1.3 aPoDLPSEPowerDetectionStatus";
    }

    leaf podl-type {
      type enumeration {
        enum unknown {
          description "Unknown PSE type.";
        }
        enum typeA {
          description "TypeA";
        }
        enum typeB {
          description "TypeB";
        }
        enum typeC {
          description "TypeC";
        }
        enum typeD {
          description "TypeD";
        }
      }
      config false;
      description "PSE type specified in IEEE Std 802.3, 104.4.1.";
    }
```

```
leaf detected-pd-type {
  when "../detection-status = 'deliveringPower'" {
    description
      "This node only applies when the detection status is
       delivering power.";
  }

  type enumeration {
    enum unknown {
      description "Unknown";
    }
    enum typeA {
      description "TypeA";
    }
    enum typeB {
      description "TypeB";
    }
    enum typeC {
      description "TypeC";
    }
    enum typeD {
      description "TypeD";
    }
  }
  config false;
  description
    "Indicates the Type of the detected PoDL PD as specified in
     IEEE Std 802.3, 104.5.1.";

  reference
    "IEEE Std 802.3, 30.15.1.1.5 aPoDLPSEDetectedPDType";
}

leaf pd-power-class {

  when "../detection-status = 'deliveringPower'" {
    description
      "This node only applies when the detection status is
       delivering power.";
  }

  type power-class;
  config false;
  description
    "Power class of the PD detected on the PSE port.";
  reference
    "IEEE Std 802.3, 30.15.1.1.6 aPoDLPSEDetectedPDPowerClass";
}

container statistics {
  config false;
  description "Statistics information of the single-pair PSE";

  leaf power-denied {
```

```
  type yang:counter64;
  description
     "This counter is incremented when the PoDL PSE state
      diagram variable power_available transitions from true
      to false (see IEEE Std 802.3, 104.4.3.3).";
  reference
    "IEEE Std 802.3, 30.15.1.1.9 aPoDLPSEPowerDeniedCounter";
}

leaf invalid-signature {
  type yang:counter64;
  description
    "This counter is incremented when the PSE state diagram
     enters the SIGNATURE_INVALID state per
     IEEE Std 802.3, Figure 33-9.";
  reference
    "IEEE Std 802.3, 30.15.1.1.7
     aPoDLPSEInvalidSignatureCounter";
}

leaf invalid-class {
  type yang:counter64;
  description
    "This counter is incremented when the PoDL PSE state
     diagram variable tclass_timer_done transitions from false
     to true or when the valid_class variable transitions from
     true to false (see IEEE Std 802.3, 104.4.3.3).";

  reference
    "IEEE Std 802.3, 30.15.1.1.8 aPoDLPSEInvalidClassCounter";
}

leaf overload {
  type yang:counter64;
  description
    "This counter is incremented when the PSE state diagram
     variable overload_held transitions from false to true
     (see IEEE Std 802.3, 104.4.3.3).";
  reference
    "IEEE Std 802.3, 30.15.1.1.10 aPoDLPSEOverLoadCounter";
}

leaf fvs-absence {
  type yang:counter64;
  description
    "Maintain Full Voltage Signature absent counter.
     This counter is incremented when the PoDL PSE state
     diagram variable mfvs_timeout transitions from false to
     true (see IEEE Std 802.3, 104.4.3.3).";

  reference
    "IEEE Std 802.3, 30.15.1.1.11
     aPoDLPSEMaintainFullVoltageSignatureAbsentCounter";
}
```

```
          leaf cumulative-energy {
            type yang:counter64;
            units millijoules;
            description
              "A count of the cumulative energy supplied by the PoDL
               PSE, measured at the MDI, and expressed in units of
               millijoules.";
            reference
              "IEEE Std 802.3, 30.15.1.1.14 aPoDLPSECumulativeEnergy";
          }

        }

        leaf actual-power {
          type decimal64 {
            fraction-digits 4 ;
          }

          units milliwatts;
          config false;
          description
            "An integer value indicating present (actual) power being
             supplied by the PoDL PSE as measured at the MDI in
             milliwatts.";
          reference
            "IEEE Std 802.3, 30.15.1.1.12 aPoDLPSEActualPower";
        }

        leaf power-accuracy {
          type int64;
          units milliwatts;
          config false;
          description
            "A signed integer value indicating the accuracy associated
             with power-accuracy in milliwatts.";
          reference
            "IEEE Std 802.3, 30.15.1.1.13 aPoDLPSEPowerAccuracy";
        }
      }
    }
  }
}
```

7. YANG module for Ethernet Passive Optical Network (EPON)

7.1 Introduction

This clause defines a YANG module to manage Ethernet Passive Optical Network (EPON).

7.2 YANG module structure

The *ieee802-ethernet-pon* YANG module of this clause is focused on the configuration and monitoring of EPON.

7.2.1 Introduction

EPON is defined in IEEE Std 802.3, covering Physical Layers and Media Access Control sublayers. 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 7–1. Individual branches of the PON are terminated with the Optical Line Terminal (OLT) in the Central Office or at remote optical nodes, 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 7–1 presents an example PON topology.

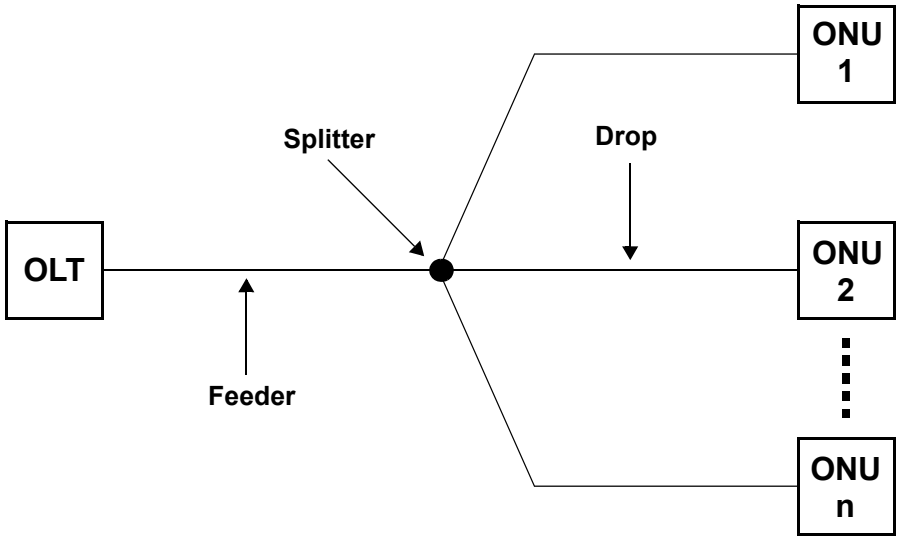


Figure 7–1—PON topology example

The following clauses in IEEE Std 802.3 define 1G-EPON:

- Clause 60: Physical Medium Dependent (PMD) sublayer for 1G-EPON
- Clause 64: MPCP (Multipoint Control Protocol) for 1G-EPON
- Clause 65: Reconciliation Sublayer (RS), Physical Coding Sublayer (PCS), and Physical Media Attachment (PMA) sublayers for 1G-EPON

The following clauses in IEEE Std 802.3 define 10G-EPON:

- Clause 75: PMD sublayer for 10G-EPON
- Clause 76: RS, PCS, and PMA sublayers for 10G-EPON

— Clause 77: MPCP for 10G-EPON

Additionally, IEEE Std 802.3, Clause 30 and Clause 45 are also applicable to EPON.

7.2.2 Principles of operation

The EPON specification extended the specification of Gigabit Ethernet (in case of 1G-EPON) or 10 Gigabit Ethernet (in case of 10G-EPON) as described in IEEE Std 802.3. The Ethernet MAC operates at the data rate of 1 Gb/s (in 1G-EPON) or 10 Gb/s (in 10G-EPON) and it is connected to a media dependent interface through the GMII (in 1G-EPON) or XGMII (in 10G-EPON) interface. The EPON PCS layer extended the Ethernet PCS, adding burst-mode operation capabilities and EPON-specific forward error correction (FEC). The following new, EPON-specific layers were added:

— MPCP is placed in the MAC control layer, providing EPON media access, station discovery, and registration protocol.

— Functionality of the reconciliation sublayer (RS) was extended, creating logical links over shared passive optical medium, providing private transmission channels to each of the connected ONU.

— FEC functionality (optional in 1G-EPON, mandatory in 10G-EPON) located in the PCS was added, extending the Ethernet PCS layer, enhancing reach and split performance of the EPON optical link.

Figure 7–2 presents the EPON layering model.

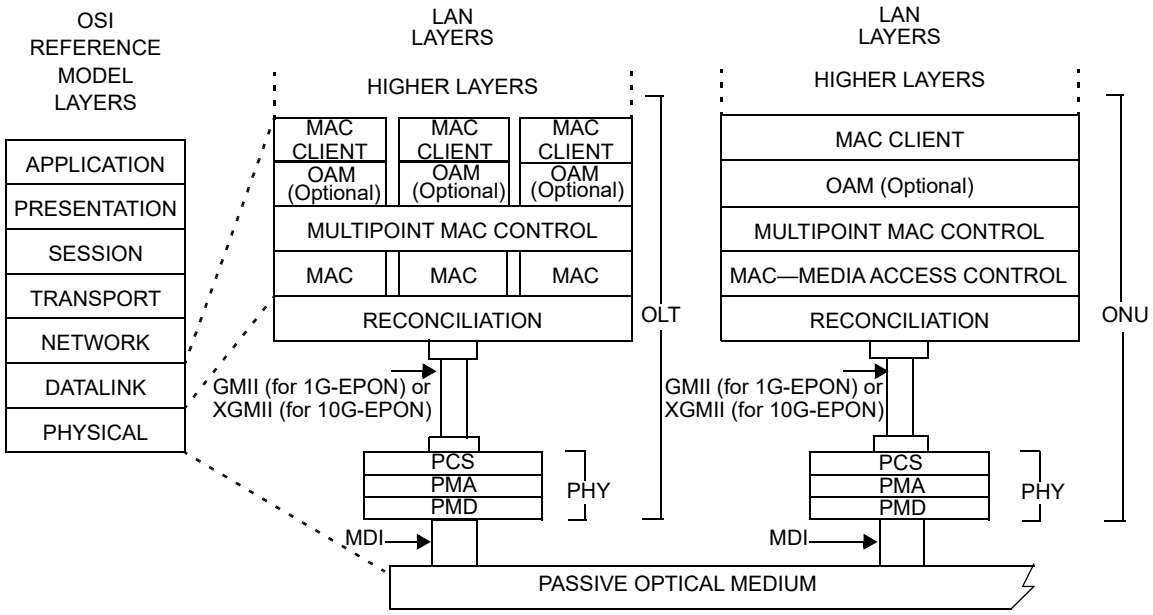


Figure 7–2—Relationship of Multipoint MAC Control and the OSI protocol stack

7.2.3 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 continuously 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 the 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).

7.2.4 PMD specifications

The EPON PMD specifications are based on a wavelength plan 1defined in IEEE Std 802.3, Clause 60 (1G-EPON) and Clause 75 (10G-EPON). The OLT and ONU optical parameters were derived in part from applicable 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.

7.2.5 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. 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 a 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 the master device clock. This coordinates upstream transmissions from individual

ONUs so that the transmissions arrive at the OLT at the precisely anticipated time, and thus data from different ONUs do 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 IEEE Std 802.3, Clause 64 (for 1G-EPON) and Clause 77 (for 10G-EPON).

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 service level agreement. 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.

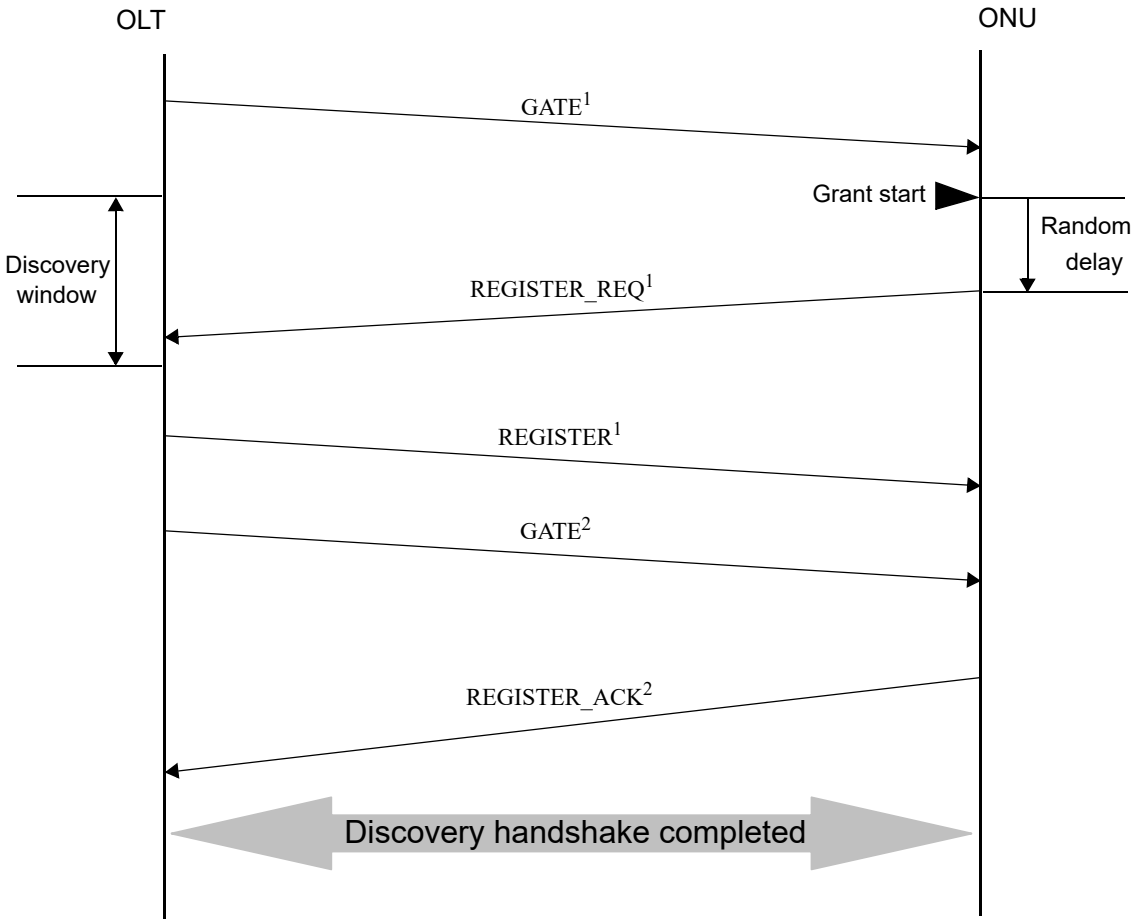
In addition, the MPCP defines a protocol of auto-discovery and registration of ONUs.

The MPCP registration process is presented in Figure 7–3, while details are described in IEEE Std 802.3, Clause 64 (for 1G-EPON) and Clause 77 (for 10G-EPON). Note that MPCP for 10G-EPON supports the coexistence mode, i.e., simultaneous operation of 1G-EPON and 10G-EPON devices on the same fiber plant, through time sharing the upstream transmission channel.

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 a 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, a decision a non-overlapping 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 and the measured RTT so 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.



[1] Messages sent on broadcast channel
[2] Messages sent on unicast channels

Figure 7–3—Discovery handshake message exchange

7.2.6 Forward error correction (FEC)

The FEC mechanism is optional for 1G-EPON and is defined to enhance the 1G-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 1G-EPON is frame-based, meaning that parity information is added at the end of each Ethernet packet. Extra space between individual Ethernet packets is provided 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 7–4 presents the structure of an FEC-protected 1G-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.

Figure 7–4—1G-EPON FEC-protected frame

The FEC mechanism is mandatory for 10G-EPON, and similarly to 1G-EPON – defined to enhance the EPON link budget. The mandatory FEC mechanism increases the available link budget by improving the link BER from $2x10^{-3}$ 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 mandatory FEC used in 10G-EPON is stream-based, meaning that 32 parity symbols are inserted into the bit stream at regular intervals (every 223 information symbols). Details of the FEC encoding process in 10G-EPON are described in IEEE Std 802.3, 76.3.2.4, including the structure of the resulting frame and the resulting bit ordering shown in IEEE Std 802.3, Figure 76-12.

7.2.7 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 modules follows this separation.

Figure 7–5 presents the relation of the 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 7–5, 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.

Each row in the tables is indexed according to the ifIndex; specifically, there is a row for each logical link. There are some control objects that are shared and are the same for the logical 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 YANG 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 7–5). However, from a management perspective, it is more convenient to partition the management of the layers for the logical links, as the atomic managed entity is the logical link. It is also convenient to use the interface index of the logical link for that purpose, as it is already used to index the rows of the logical links at the Interface, MAU, and Ethernet-like interface YANG module.

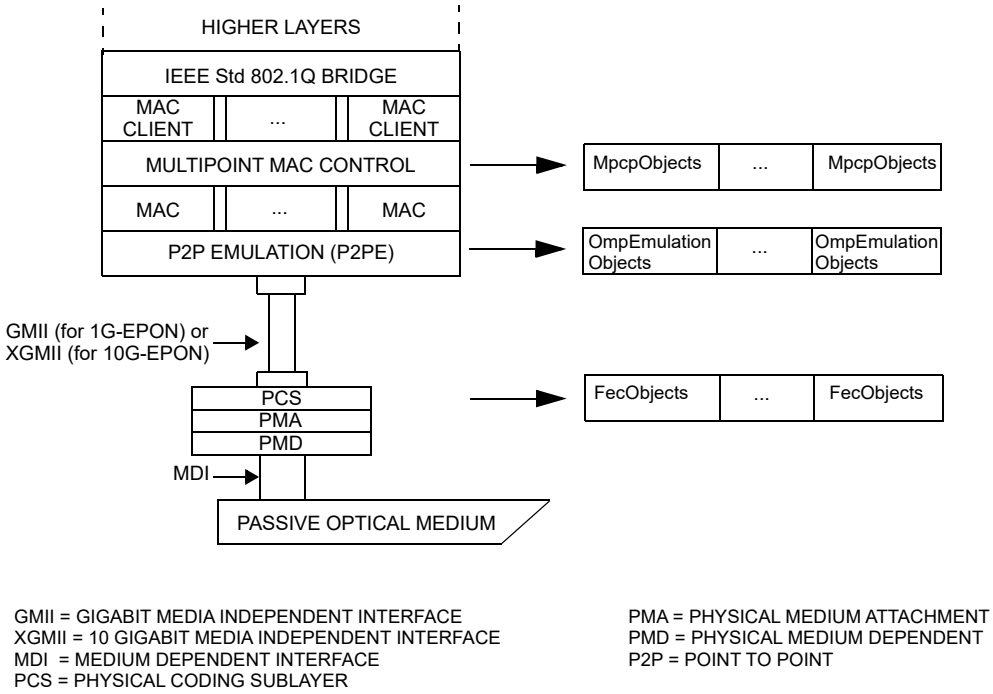


Figure 7–5—Relationship of the YANG module to the EPON sublayers

7.3 Mapping of IEEE Std 802.3, Clause 30 managed objects

This sub-clause contains the mapping between YANG data nodes included in *ieee802-ethernet-pon* (see Table 7–1) YANG module, managed objects, and attributes defined in IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB.

Table 7–1—Mapping between IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB managed objects and *ieee802-ethernet-pon* YANG data nodes

IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB		Corresponding *ieee802-ethernet-pon* YANG data nodes		
Managed object(s)	**Attribute(s)**	**Container(s)**	**Data node(s)**	**R/W**
dot3EponFecTable	dot3EponFecMode		fec-mode	R/W
	dot3EponFecPCSCodingViolation	statistics-pon-fec	fec-code-group-violations	R
	dot3EponFecAbility		fec-capability	R
	dot3EponFecCorrectedBlocks	statistics-pon-fec	fec-code-word-corrected-errors	R
	dot3EponFecUncorrectableBlocks	statistics-pon-fec	fec-code-word-uncorrected-errors	R
	dot3EponFecBufferHeadCodingViolation	statistics-pon-fec	fec-buffer-head-coding-violation	R
dot3MpcpControl Table	dot3MpcpAdminState		mpcp-admin-state	R/W
	dot3MpcpMode		mpcp-mode	R
	dot3MpcpLinkID		mpcp-logical-link-id	R
	dot3MpcpRemoteMACAddress		mpcp-remote-mac-address	R
	dot3MpcpRegistrationState		mpcp-logical-link-state	R
	dot3MpcpSyncTime		mpcp-sync-time	R
	dot3MpcpTransmitElapsed		mpcp-elapsed-time-out	R
	dot3MpcpReceiveElapsed		mpcp-elapsed-time-in	R
	dot3MpcpRoundTripTime		mpcp-round-trip-time	R
	dot3MpcpMaximumPendingGrants		mpcp-maximum-grant-count	R

Table 7–1—Mapping between IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB managed objects and *ieee802-ethernet-pon* YANG data nodes *(continued)*

IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB		Corresponding *ieee802-ethernet-pon* YANG data nodes		
Managed object(s)	**Attribute(s)**	**Container(s)**	**Data node(s)**	**R/W**
dot3ExtPkgQueueSets Table	dot3QueueSetIndex	mpcp-queue-thresholds	mpcp-queue-set-index	R/W
	dot3ExtPkgObjectReportThreshold		mpcp-queue-set-threshold	R/W
	dot3QueueIndex	mpcp-queues	mpcp-queue-index	R/W
	dot3ExtPkgObjectReportNumThreshold		mpcp-queue-threshold-count	R/W
	dot3ExtPkgObjectReportMaximumNumThreshold		mpcp-queue-threshold-count-max	R
	dot3ExtPkgStatTxFramesQueue		in-mpcp-queue-frames	R
	dot3ExtPkgStatRxFramesQueue		out-mpcp-queue-frames	R
	dot3ExtPkgStatDroppedFramesQueue		mpcp-queue-frames-drop	R
dot3ExtPkgControl Table	dot3ExtPkgObjectReset dot3MpcpOperStatus		mpcp-logical-link-admin-state	R/W
	dot3ExtPkgObjectNumberOfLLIDs		mpcp-logical-link-count	R
	dot3ExtPkgObjectReportMaximumNumQueues		mpcp-maximum-queue-count-per-report	R
dot3RecognizedMulticast-IDs Table	dot3RecognizedMulticastID	multicast-IDs	multicast-ID	R/W

Table 7–1—Mapping between IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB managed objects and *ieee802-ethernet-pon* YANG data nodes *(continued)*

IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB		Corresponding *ieee802-ethernet-pon* YANG data nodes		
Managed object(s)	**Attribute(s)**	**Container(s)**	**Data node(s)**	**R/W**
dot3OmpEmulation Table	dot3OmpEmulationType	statistics-ompe	ompe-mode	R
	dot3OmpEmulationSLDErrors		in-ompe-frames-errored-sld	R
	dot3OmpEmulationCRC8Errors		in-ompe-frames-errored-crc8	R
	dot3OmpEmulationBadLLID		in-ompe-frames-with-bad-llid	R
	dot3OmpEmulationGoodLLID		in-ompe-frames-with-good-llid	R
	dot3OmpEmulationBroadcastBitNotOnuLlid		in-ompe-frames-not-match-onu-llid-broadcast	R
	dot3OmpEmulationOnuLLIDNotBroadcast		in-ompe-frames-match-onu-llid-not-broadcast	R
	dot3OmpEmulationBroadcastBitPlusOnuLlid		in-ompe-frames-match-onu-llid-broadcast	R
	dot3OmpEmulationNotBroadcastBitNotOnuLlid		in-ompe-frames-not-match-onu-llid-not-broadcast	R
			in-ompe-frames	R
			ompe-onu-frames-with-good-llid-good-crc8	
			ompe-olt-frames-with-good-llid-good-crc8	

Table 7–1—Mapping between IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB managed objects and *ieee802-ethernet-pon* YANG data nodes *(continued)*

IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB		Corresponding *ieee802-ethernet-pon* YANG data nodes		
Managed object(s)	**Attribute(s)**	**Container(s)**	**Data node(s)**	**R/W**
dot3MpcpStat Table	dot3MpcpMACCtrlFramesTransmitted	statistics-mpcp	out-mpcp-mac-ctrl-frames	R
	dot3MpcpMACCtrlFramesReceived		in-mpcp-mac-ctrl-frames	R
	dot3MpcpDiscoveryWindowsSent		mpcp-discovery-window-count	R
	dot3MpcpDiscoveryTimeout		mpcp-discovery-timeout-count	R
	dot3MpcpTxRegRequest		out-mpcp-register-req	R
	dot3MpcpRxRegRequest		in-mpcp-register-req	R
	dot3MpcpTxRegAck		out-mpcp-register-ack	R
	dot3MpcpRxRegAck		in-mpcp-register-ack	R
	dot3MpcpTxReport		out-mpcp-report	R
	dot3MpcpRxReport		in-mpcp-report	R
	dot3MpcpTxGate		out-mpcp-gate	R
	dot3MpcpRxGate		in-mpcp-gate	R
	dot3MpcpTxRegister		out-mpcp-register	R
	dot3MpcpRxRegister		in-mpcp-register	R

Table 7–1—Mapping between IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB managed objects and *ieee802-ethernet-pon* YANG data nodes *(continued)*

IEEE Std 802.3.1, IEEE8023-DOT3-EPON-MIB		Corresponding *ieee802-ethernet-pon* YANG data nodes		
Managed object(s)	**Attribute(s)**	**Container(s)**	**Data node(s)**	**R/W**
dot3ExtPkgOptIf Table	dot3ExtPkgOptIfLowerInputPowerThreshold	thresholds-trx	in-trx-power-low-threshold	R/W
	dot3ExtPkgOptIfUpperInputPowerThreshold		in-trx-power-high-threshold	R/W
	dot3ExtPkgOptIfLowerOutputPowerThreshold		out-trx-power-low-threshold	R/W
	dot3ExtPkgOptIfUpperOutputPowerThreshold		out-trx-power-high-threshold	R/W
	dot3ExtPkgOptIfSignalDetect		in-trx-power-signal-detect	R
	dot3ExtPkgOptIfInputPower		in-trx-power	R
	dot3ExtPkgOptIfLowInputPower		in-trx-power-low-15-minutes-bin	R
	dot3ExtPkgOptIfHighInputPower		in-trx-power-high-15-minutes-bin	R
	dot3ExtPkgOptIfTransmitEnable		out-trx-power-signal-detect	R/W
	dot3ExtPkgOptIfOutputPower		out-trx-power	R
	dot3ExtPkgOptIfLowOutputPower		out-trx-power-low-15-minutes-bin	R
	dot3ExtPkgOptIfHighOutputPower		out-trx-power-high-15-minutes-bin	R
	dot3ExtPkgOptIfSuspectedFlag		trx-data-reliable	R

7.4 YANG module definition[9]

The YANG module tree hierarchy uses terms defined in IETF RFC 8407.

7.4.1 Tree hierarchy

```
module: ieee802-ethernet-pon
  augment /if:interfaces/if:interface/ieee802-eth-if:ethernet:
    +--rw fec-mode?                                fec-mode {fec-supported}?
    +--rw mpcp-admin-state?                        mpcp-admin-state
    +--ro mpcp-logical-link-admin-state?           mpcp-logical-link-admin-state
    +--rw trx-transmit-admin-state?                trx-admin-state {trx-power-level-reporting-supported}?
    +--ro capabilities
    |  +--ro mpcp-supported?   mpcp-supported
    +--ro statistics-mpcp
    |  +--ro out-mpcp-mac-ctrl-frames?       yang:counter64
    |  +--ro in-mpcp-mac-ctrl-frames?        yang:counter64
    |  +--ro mpcp-discovery-window-count?    yang:counter64
    |  +--ro mpcp-discovery-timeout-count?   yang:counter64
    |  +--ro out-mpcp-register-req?          yang:counter64
    |  +--ro in-mpcp-register-req?           yang:counter64
    |  +--ro out-mpcp-register-ack?          yang:counter64
    |  +--ro in-mpcp-register-ack?           yang:counter64
    |  +--ro out-mpcp-report?                yang:counter64
    |  +--ro in-mpcp-report?                 yang:counter64
    |  +--ro out-mpcp-gate?                  yang:counter64
    |  +--ro in-mpcp-gate?                   yang:counter64
    |  +--ro out-mpcp-register?              yang:counter64
    |  +--ro in-mpcp-register?               yang:counter64
    +--rw statistics-ompe
    |  +--ro in-ompe-frames-errored-sld?                      yang:counter64
    |  +--ro in-ompe-frames-errored-crc8?                     yang:counter64
    |  +--ro ompe-onu-frames-with-good-llid-good-crc8?        yang:counter64
```

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

```
|  +--ro in-ompe-frames-with-bad-llid?                       yang:counter64
|  +--ro in-ompe-frames-with-good-llid?                      yang:counter64
|  +--ro in-ompe-frames?                                     yang:counter64
|  +--ro in-ompe-frames-not-match-onu-llid-broadcast?        yang:counter64
|  +--ro in-ompe-frames-match-onu-llid-not-broadcast?        yang:counter64
|  +--ro in-ompe-frames-match-onu-llid-broadcast?            yang:counter64
|  +--ro in-ompe-frames-not-match-onu-llid-not-broadcast?    yang:counter64
+--rw thresholds-trx {trx-power-level-reporting-supported}?
|  +--rw in-trx-power-low-threshold?     int32 {trx-power-level-reporting-supported}?
|  +--rw in-trx-power-high-threshold?    int32 {trx-power-level-reporting-supported}?
|  +--rw out-trx-power-low-threshold?    int32 {trx-power-level-reporting-supported}?
|  +--rw out-trx-power-high-threshold?   int32 {trx-power-level-reporting-supported}?
+--rw statistics-trx {trx-power-level-reporting-supported}?
|  +--ro in-trx-power-signal-detect?          boolean
|  +--ro in-trx-power?                        int32
|  +--ro in-trx-power-low-15-minutes-bin?     int32
|  +--ro in-trx-power-high-15-minutes-bin?    int32
|  +--ro out-trx-power-signal-detect?         boolean
|  +--ro out-trx-power?                       int32
|  +--ro out-trx-power-low-15-minutes-bin?    int32
|  +--ro out-trx-power-high-15-minutes-bin?   int32
|  +--ro trx-data-reliable?                   boolean {trx-power-level-reporting-supported}?
+--ro statistics-pon-fec {fec-supported}?
|  +--ro fec-code-group-violations?           yang:counter64
|  +--ro fec-buffer-head-coding-violations?   yang:counter64
|  +--ro fec-code-word-corrected-errors?      yang:counter64
|  +--ro fec-code-word-uncorrected-errors?    yang:counter64
+--rw mpcp-logical-link-admin-actions
|  +---x state-change-action-type
|  |  +---w input
|  |     +---w state-change-action-type?   identityref
|  +---x reset-action-type
|  |  +---w input
|  |     +---w reset-action-type?   identityref
|  +---x register-type
|     +---w input
```

```
|        +---w register-type?   identityref
+--rw mpcp-queues* [mpcp-queue-index]
|  +--rw mpcp-queue-index                  uint8
|  +--rw mpcp-queue-threshold-count?       uint8
|  +--ro mpcp-queue-threshold-count-max?   uint8
|  +--rw mpcp-queue-thresholds* [mpcp-queue-set-index]
|  |  +--rw mpcp-queue-set-index        uint8
|  |  +--rw mpcp-queue-set-threshold?   uint64
|  +--ro in-mpcp-queue-frames?             yang:counter64
|  +--ro out-mpcp-queue-frames?            yang:counter64
|  +--ro mpcp-queue-frames-drop?           yang:counter64
+--rw multicast-IDs* [multicast-ID]
|  +--rw multicast-ID    uint32
+--ro fec-capability?                        fec-capability
+--ro mpcp-mode?                             mpcp-mode
+--ro mpcp-sync-time?                        uint64
+--ro mpcp-logical-link-id?                  mpcp-supported
+--ro mpcp-remote-mac-address?               yang:mac-address
+--ro mpcp-logical-link-state?               mpcp-logical-link-state
+--ro mpcp-elapsed-time-out?                 uint64
+--ro mpcp-elapsed-time-in?                  uint64
+--ro mpcp-round-trip-time?                  uint16
+--ro mpcp-maximum-grant-count?              uint8
+--ro mpcp-logical-link-count?               mpcp-llid-count
+--ro mpcp-maximum-queue-count-per-report?   mpcp-maximum-queue-count-per-report
+--ro ompe-mode?                             ompe-mode
```

7.4.2 YANG module

In the following YANG module definition, should any discrepancy between the text of the description for individual YANG nodes and the corresponding definition in 7.2 through 7.4 of this clause occur, the definitions and mappings in 7.4 shall take precedence.

An ASCII text version of the YANG module can be found at the following URL:[10] https://github.com/YangModels/yang/tree/master/standard/ieee/published/802.3.

```
module ieee802-ethernet-pon {
  yang-version 1.1;

  namespace "urn:ieee:std:802.3:yang:ieee802-ethernet-pon";

  prefix "ieee802-eth-pon";

  revision 2019-06-21 {
    description "Initial revision.";
  }

  import ietf-yang-types {
    prefix "yang";
    reference "IETF RFC 6991";
  }

  import ietf-interfaces {
    prefix "if";
    reference "IETF RFC 8343";
  }

  import ieee802-ethernet-interface {
    prefix "ieee802-eth-if";
  }

  organization
    "IEEE 802.3 Ethernet Working Group
     Web URL: http://www.ieee802.org/3/";

  contact
    "Web URL: http://www.ieee802.org/3/";

  description
    "This module contains a collection of YANG definitions for
     managing the Multi Point Control Protocol for Ethernet PON (EPON),
     as defined in IEEE Std 802.3, Clause 64 and Clause 77.

     This YANG module augments the 'ethernet' module.";

  reference
    "IEEE Std 802.3-2018, Clause 64 and Clause 77, unless dated explicitly
     IEEE Std 802.3.1-2013, Clause 9, unless dated explicitly";

  typedef mpcp-supported {
    type boolean;
```

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

```
  description
    "This object indicates that the given interface supports MPCP,
     i.e., it is an Ethernet PON (EPON) interface.";
}

typedef mpcp-llid {
  type uint64 {
    range "0 .. 32767";
  }

  description
    "Logical Link Identifiers (LLIDs) are used to identify a single
     MAC from a number of MACs which may be present in the
     EPON OLT or ONU. LLIDs between the value of 0x07FFE
     and 0x7FFF are assigned for ONU discovery and registration.
     Other LLIDs are dynamically assigned by the OLT during the
     registration process. For a complete description of how the
     LLID is used in an EPON device, see IEEE Std 802.3, Clause 65
     for 1G-EPON and Clause 76 for 10G-EPON.";

  reference
    "IEEE Std 802.3, 65.1.3.3 for 1G-EPON and 76.2.6.1.3 for 10G-EPON";
}

typedef mpcp-maximum-queue-count-per-report {
  type uint8 {
    range "0..7";
  }

  default "0";

  description
    "Defines the maximum number of queues (0-7) in the REPORT
     MPCPDU as defined in IEEE Std 802.3, Clause 64 and Clause 77.";
}

typedef mpcp-llid-count {
  type uint64 {
    range "0 .. 32767";
  }

  description
    "Indicates the number of registered LLIDs. The initialization
     value is 0. This is applicable for an OLT with the same value
     for all logical interfaces and for an ONU.";

  reference
    "IEEE Std 802.3, 65.1.3.3 for 1G-EPON and 76.2.6.1.3 for 10G-EPON";

}

typedef mpcp-admin-state {
  type enumeration {

    enum enabled {
      description
        "When selecting the value of 'enabled', the MultiPoint
         Control Protocol sublayer on the OLT / ONU is enabled.";
    }
```

```
    enum disabled {
      description
        "When selecting the value of 'disabled', the MultiPoint
         Control Protocol sublayer on the OLT / ONU is disabled.";
    }
  }

  description
    "Enumeration of valid administrative states for a MultiPoint MAC
     Control sublayer on the OLT or ONU.";

  reference
    "IEEE Std 802.3, 30.3.5.2.1";
}

typedef mpcp-mode {
  type enumeration {
    enum olt {
      description
        "MPCP mode: olt";
    }

    enum onu {
      description
        "MPCP mode: onu";
    }
  }

  description
    "Enumeration of valid MPCP modes for EPON interfaces.";

  reference
    "IEEE Std 802.3, 30.3.5.1.3";
}

typedef mpcp-logical-link-state {
  type enumeration {
    enum unregistered {
      description
        "MPCP registration state: logical link is NOT registered.";
    }

    enum registering {
      description
        "MPCP registration state: logical link is currently in the
         process of registering.";
    }

    enum registered {
      description
        "MPCP registration state: logical link is currently
         registered.";
    }
  }

  description
    "Enumeration of valid MPCP registration states for EPON
     interfaces.";
```

```
  reference
    "IEEE Std 802.3, 30.3.5.1.6";
}

typedef mpcp-logical-link-admin-state {
  type enumeration {
    enum reset {
      description
        "When read, the value of 'reset' indicates that the given
         logical link on the OLT / ONU has been reset.";
    }
    enum operate {
      description
        "When read, the value of 'operate' indicates that the
         given logical link on the OLT / ONU has moved into
         operating mode.";
    }
    enum unknown {
      description
        "When read, the value of 'unknown' indicates that the
         status of the given logical link on the OLT / ONU is
         currently not known.";
    }
    enum registered {
      description
        "When read, the value of 'registered' indicates that the
         given logical link on the OLT / ONU has been registered.";
    }
    enum deregistered {
      description
        "When read, the value of 'deregistered' indicates that the
         given logical link on the OLT / ONU has been deregistered.";
    }
    enum reregistered {
      description
        "When read, the value of 'reregistered' indicates that the
         given logical link on the OLT / ONU has been reregistered.";
    }
  }
  description
    "Enumeration of valid administrative states for a logical link
     on the OLT or ONU.";
}

typedef ompe-mode {
  type enumeration {
    enum unknown {
      description
        "omp-emulation mode: unknown = system is initializing";
    }

    enum olt {
      description
        "omp-emulation mode: olt";
    }

    enum onu {
      description
```

```
        "omp-emulation mode: onu";
      }
    }

    description
      "Enumeration of valid OMP-Emulation modes for EPON
       interfaces.";

    reference
      "IEEE Std 802.3, 30.3.7.1.2";
  }

  typedef fec-capability {
    type enumeration {
      enum unknown {
        description
          "FEC capability: unknown = system is initializing.";
      }

      enum supported {
        description
          "FEC capability: supported.";
      }

      enum "not supported" {
        description
          "FEC capability: not supported.";
      }
    }

    description
      "Enumeration of valid FEC capability values for EPON
       interfaces with enabled MPCP.";

    reference
      "IEEE Std 802.3, 30.5.1.1.15";
  }

  typedef fec-mode {
    type enumeration {
      enum unknown {
        description
          "FEC mode: unknown = system is initializing.";
      }

      enum disabled {
        description
          "FEC mode: disabled = FEC is disabled for the given
           logical link (both Tx and Rx directions).";
      }

      enum enabled-Tx-Rx {
        description
          "FEC mode: enabled-Tx-Rx = FEC is enabled for the given
           logical link in both Tx and Rx directions.";
      }

      enum enabled-Tx-only {
        description
```

```
        "FEC mode: enabled-Tx-only = FEC is enabled for
         the given logical link but only in Tx direction.";
    }

    enum enabled-Rx-only {
      description
        "FEC mode: enabled-Rx-only = FEC is enabled for
         the given logical link but only in Rx direction.";
    }
  }

  description
    "Enumeration of valid FEC modes for EPON interfaces.";

  reference
    "IEEE Std 802.3, 30.5.1.1.16";
}

typedef trx-admin-state {
  type enumeration {
    enum enabled {
      description
        "When read as 'enabled', the transmitter is enabled and
         operating under the control of the logical control protocol.
         When set to 'enabled', the transmitter is enabled to
         operate under the control of the logical control protocol.";
    }

    enum disabled {
      description
        "When read as 'disabled', the transmitter is currently
         disabled (not transmitting). When set to 'disabled', the
         transmitter is expected to be disabled (to stop transmitting).";
    }
  }

  description
    "Enumeration of valid administrative states for an optical
     transceiver.";

  reference
    "IEEE Std 802.3.1, dot3ExtPkgOptIfTransmitEnable";
}

feature trx-power-level-reporting-supported {

  description
    "This object indicates the support for optical transceiver power
     level monitoring and reporting capability. When 'true', the
     given interface supports the optical power level monitoring
     and reporting function. Otherwise, the value is 'false'.";
}

feature
fec-supported {
  description
    "This object indicates the support of operation of the optional
     FEC sublayer of the 1G-EPON PHY specified in IEEE Std 802.3,
     65.2. The value of 'unknown' is reported in the initialization,
```

```
      for non FEC support state or type not yet known. The value of
      'not supported' is reported when the sublayer is not supported.
      The value of 'supported' is reported when the sublayer is
      supported. This object is applicable for an OLT, with the
      same value for all logical links, and for an ONU.";

    reference
      "IEEE Std 802.3, 30.5.1.1.15";
  }

  identity state-change-action-type {
    description
      "Type of interface state change requested.";
  }
  identity power-down {
    base state-change-action-type;
    description
      "Power down the EPON logical interface.
       Power-down actions are applicable for the OLT and ONU.  A
       power down of a specific logical interface affects only
       the logical interface (and not the physical interface).
       the logical interface will be unavailable while the
       power-down occurs and data may be lost. Other logical
       interface are unaffected by power-down.

       This action is relevant when the admin state is active.";

  }
  identity power-up {
    base state-change-action-type;
    description
      "Exit EPON logical interface power-down state.";
  }

  identity reset-action-type {
    description
      "Type of reset action requested.";
  }
  identity reset-interface {
    base reset-action-type;
    description
      "Reset the EPON logical interface. Resetting an interface
       can lead 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 logical interface.
       A reset for a specific logical interface resets only
       this logical interface and not the physical interface.

       Thus, a logical link that is malfunctioning can be
       reset without affecting the operation of other logical
       interfaces.

       The reset can cause Discontinuities in the values of the
       counters of the interface, similar to re-initialization
       of the management system.";
  }
  identity register-type {
```

```
  description
    "Type of registration requested.";
}
identity register {
  base register-type;
  description
    "Register indicates a request to register an LLID.
     This action applies to an OLT or ONU logical interface.";
}
identity reregister {
  base register-type;
  description
    "Re-register indicates an request to re-register an LLID.
     This action applies to an OLT or ONU logical interface.";
 }
identity deregister {
  base register-type;
  description
    "De-register indicates an request to de-register an LLID.
     This action applies to an OLT or ONU logical interface.
     Deregister may result in an interruption of service to
     users connected to the respective EPON interface.";
}

augment "/if:interfaces/if:interface/ieee802-eth-if:ethernet" {
  description
    "Augments the definition of Ethernet interface (/if:interfaces/
     if:interface/ieee802-eth-if:ethernet) module with nodes
     specific to Ethernet PON (EPON).";


  leaf fec-mode {
    if-feature "fec-supported";

    type fec-mode;

    description
      "This object reflects the current administrative state of the
       FEC function for the given logical link on an ONU or OLT.

       When reading the value of 'disabled', the FEC function on the
       given logical link is disabled.

       When reading the value of 'enabled-Tx-Rx', the FEC function on
       the given logical link is enabled in both Tx and Rx directions.

       When reading the value of 'enabled-Tx-only', the FEC function
       on the given logical link is enabled in Tx direction only.

       When reading the value of 'enabled-Rx-only', the FEC function
       on the given logical link is enabled in Rx direction only.

       When reading the value of 'unknown', the state of the FEC
       function on the given logical link is unknown or the FEC
       function is currently initializing.

       This object is applicable for an OLT and an ONU. This object has
       the same value for each logical link.";
```

```
    reference
      "IEEE Std 802.3, 30.5.1.1.16";
  }

  leaf mpcp-admin-state {
    type mpcp-admin-state;

    description
      "This object reflects the current administrative state of the
       MultiPoint MAC Control sublayer, as defined in IEEE Std 802.3,
       Clause 64 and Clause 77, for the OLT / ONU.

       When reading the value of 'enabled', the MultiPoint Control
       Protocol on the OLT / ONU is enabled.

       When reading the value of 'disabled', the MultiPoint Control
       Protocol on the OLT / ONU is disabled.

       This object is applicable for an OLT and an ONU. It has the
       same value for all logical links.";

    reference
      "IEEE Std 802.3, 30.3.5.1.2";
  }

  leaf mpcp-logical-link-admin-state {
    type mpcp-logical-link-admin-state;
    config false;
    description
      "This object reflects the current administrative state of a
       logical link on an ONU or OLT.

       When reading the value of 'reset', the given logical link is
       undergoing a reset.

       When reading the value of 'unknown', the current status of the
       given logical link is unknown and the link might be undergoing
       initialization.

       When reading the value of 'operate', the given logical link is
       operating  normally.

       When reading the value of 'registered', the given logical link
       was requested to perform registration.

       When reading the value of 'deregistered', the given logical
       link was requested to perform deregistration.

       When reading the value of 'reregistered', the given logical
       link was requested to perform reregistration.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgObjectRegisterAction";
  }

  leaf trx-transmit-admin-state {
```

```
    when
      "../../ieee802-eth-if:ethernet/
           ieee802-eth-pon:mpcp-admin-state = 'enabled'";

    if-feature "trx-power-level-reporting-supported";

    type trx-admin-state;

    description
      "This object reflects the current status of the transmitter in
       the optical transceiver.

       When read as 'enabled', the optical transmitter is enabled and
       operating under the control of the logical control protocol.

       When read as 'disabled', the optical transmitter is disabled.

       This object is applicable for an OLT and an ONU. At the OLT, this
       object has a distinct value for each logical link.

       The value of this object is only reliable when
       /if:interfaces-state/if:interface/ieee802-eth-if:ethernet/
       'mpcp-admin-state' is equal to 'enabled'.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfTransmitEnable";
  }

  container capabilities {

    config false;
    description
      "This container includes all EPON interface-specific capabilities.";

    leaf mpcp-supported {
      type mpcp-supported;

      default
        "true";
      description
        "This object indicates that the given interface supports MPCP,
         i.e., it is an Ethernet PON (EPON) interface.";
    }
  }

  container statistics-mpcp {
    config false;
    description
      "This container defines a set of MPCP-related statistics
       counters of an EPON interface, as defined in
       IEEE Std 802.3, Clause 64 and Clause 77.";

    leaf out-mpcp-mac-ctrl-frames {
      type yang:counter64;

      units frames;

      config false;
```

```
    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. It has a
       distinct value for each logical link.

       Discontinuities of this counter can occur at
       re-initialization of the management system, and at other
       times, as indicated by the value of the 'discontinuity-time'
       leaf defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.7";
  }

  leaf in-mpcp-mac-ctrl-frames {
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of MPCP frames passed by the MAC sublayer to the
       MAC Control sublayer.

       This counter is incremented when a frame is received at the
       interface which is an MPCP frame or has a Length/Type Ethernet
       header field value equal to the Type assigned for
       802.3_MAC_Control as specified in IEEE Std 802.3, 31.4.1.3.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.8";
  }

  leaf mpcp-discovery-window-count {
    when "../../ompe-mode = 'olt'";
    type yang:counter64;

    units "discovery windows";

    config false;

    description
      "A count of discovery windows generated by the OLT.
```

```
        The counter is incremented by one for each generated
        discovery window.

        This object is applicable for an OLT and has the same value
        for each logical link.

        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 the 'discontinuity-time' leaf
        defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.22";
  }

  leaf mpcp-discovery-timeout-count {
    when "../../ompe-mode = 'olt'";
    type yang:counter64;

    units "discovery timeouts";

    config false;

    description
      "A count of the number of times a discovery timeout occurs.

       This counter is incremented by one for each discovery
       processing state-machine reset resulting from timeout
       waiting for message arrival.

       This object is applicable for an OLT and has the same value
       for each logical link.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.23";
  }

  leaf out-mpcp-register-req {
    when "../../ompe-mode = 'onu'";
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of the number of times a REGISTER_REQ MPCP frame
       transmission occurs.

       This counter is incremented by one for each REGISTER_REQ MPCP
       frame transmitted as defined in IEEE Std 802.3,
       Clause 64 and Clause 77.

       This object is applicable for an ONU and has the same value
```

```
      for each logical link.

      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 the 'discontinuity-time' leaf
      defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.12";
  }

  leaf in-mpcp-register-req {
    when "../../ompe-mode = 'olt'";
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of the number of times a REGISTER_REQ MPCP frame
       reception occurs.

       This counter is incremented by one for each REGISTER_REQ MPCP
       frame received as defined in IEEE Std 802.3,
       Clause 64 and Clause 77.

       This object is applicable for an OLT and has the same value
       for each logical link.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.17";
  }

  leaf out-mpcp-register-ack {
    when "../../ompe-mode = 'onu'";
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of the number of times a REGISTER_ACK MPCP frame
       transmission occurs.

       This counter is incremented by one for each REGISTER_ACK MPCP
       frame transmitted as defined in IEEE Std 802.3,
       Clause 64 and Clause 77.

       This object is applicable for an ONU and has a distinct value
       for each logical link.

       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 the 'discontinuity-time' leaf
      defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.10";
  }

  leaf in-mpcp-register-ack {
    when "../../ompe-mode = 'olt'";
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of the number of times a REGISTER_ACK MPCP frame
       reception occurs.

       This counter is incremented by one for each REGISTER_ACK MPCP
       frame received as defined in IEEE Std 802.3,
       Clause 64 and Clause 77.

       This object is applicable for an OLT and has a distinct
       value for each logical link.

       Discontinuities of this counter can occur at
       re-initialization of the management system and at other
       times, as indicated by the value of the 'discontinuity-time'
       leaf defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.15";
  }

  leaf out-mpcp-report {
    when "../../ompe-mode = 'onu'";
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of the number of times a REPORT MPCP frame
       transmission occurs.

       This counter is incremented by one for each REPORT MPCP frame
       transmitted as defined in IEEE Std 802.3,
       Clause 64 and Clause 77.

       This object is applicable for an ONU and has a distinct value for
       each logical link.

       Discontinuities of this counter can occur at
       re-initialization of the management system and at other
       times, as indicated by the value of the 'discontinuity-time'
       leaf defined in the ietf-interfaces YANG module (IETF RFC 8343).";
```

```
    reference
      "IEEE Std 802.3, 30.3.5.1.13";
  }

  leaf in-mpcp-report {
    when "../../ompe-mode = 'olt'";
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of the number of times a REPORT MPCP frame
       reception occurs.

       This counter is incremented by one for each REPORT MPCP frame
       received as defined in IEEE Std 802.3,
       Clause 64 and Clause 77.

       This object is applicable for an OLT and has a distinct
       value for each logical link.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.18";
  }

  leaf out-mpcp-gate {
    when "../../ompe-mode = 'olt'";
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of the number of times a GATE MPCP frame
       transmission occurs.

       This counter is incremented by one for each GATE MPCP frame
       transmitted as defined in IEEE Std 802.3,
       Clause 64 and Clause 77.

       This object is applicable for an OLT and has a distinct
       value for each logical link.

       Discontinuities of this counter can occur at
       re-initialization of the management system and at other
       times, as indicated by the value of the 'discontinuity-time'
       leaf defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.5.1.9";
```

```
}

leaf in-mpcp-gate {
  when "../../ompe-mode = 'onu'";
  type yang:counter64;

  units frames;

  config false;

  description
    "A count of the number of times a GATE MPCP frame reception
     occurs.

     This counter is incremented by one for each GATE MPCP frame
     received as defined in IEEE Std 802.3,
     Clause 64 and Clause 77.

     This object is applicable for an ONU and has a distinct value
     for each logical link.

     Discontinuities of this counter can occur at
     re-initialization of the management system and at other
     times, as indicated by the value of the 'discontinuity-time'
     leaf defined in the ietf-interfaces YANG module (IETF RFC 8343).";

  reference
    "IEEE Std 802.3, 30.3.5.1.14";
}

leaf out-mpcp-register {
  when "../../ompe-mode = 'olt'";
  type yang:counter64;

  units frames;

  config false;

  description
    "A count of the number of times a REGISTER MPCP frame
     transmission occurs.

     This counter is incremented by one for each REGISTER MPCP
     frame transmitted as defined in IEEE Std 802.3,
     Clause 64 and Clause 77.

     This object is applicable for an OLT and has a distinct
     value for each logical link.

     Discontinuities of this counter can occur at
     re-initialization of the management system and at other
     times, as indicated by the value of the 'discontinuity-time'
     leaf defined in the ietf-interfaces YANG module (IETF RFC 8343).";

  reference
    "IEEE Std 802.3, 30.3.5.1.11";
}

leaf in-mpcp-register {
```

```
      when "../../ompe-mode = 'onu'";
      type yang:counter64;

      units frames;

      config false;

      description
        "A count of the number of times a REGISTER MPCP frame
         reception occurs.

         This counter is incremented by one for each REGISTER MPCP
         frame received as defined in IEEE Std 802.3,
         Clause 64 and Clause 77.

         This object is applicable for an ONU and has a distinct value
         for each logical link.

         Discontinuities of this counter can occur at
         re-initialization of the management system and at other
         times, as indicated by the value of the 'discontinuity-time'
         leaf defined in the ietf-interfaces YANG module (IETF RFC 8343).";

      reference
        "IEEE Std 802.3, 30.3.5.1.16";
    }
  }

  container statistics-ompe {
    description
      "This container defines a set of OMP-Emulation-related
       statistics counters of an EPON interface, as defined in
       IEEE Std 802.3, Clause 65 and Clause 76.";

    reference
      "IEEE Std 802.3.1, dot3OmpEmulationStatEntry";

    leaf in-ompe-frames-errored-sld {
      type yang:counter64;

      units frames;

      config false;

      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 or
         76.2.6.1.3.1, as appropriate.

         This object is applicable for an OLT and an ONU. It has a
         distinct value for each logical link.

         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 the 'discontinuity-time' leaf
         defined in the ietf-interfaces YANG module (IETF RFC 8343).";

      reference
        "IEEE Std 802.3, 30.3.7.1.3";
```

```
}

leaf in-ompe-frames-errored-crc8 {
  type yang:counter64;

  units frames;

  config false;

  description
    "A count of frames received that contain a valid SLD field,
     as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1 as
     appropriate, but do not pass the CRC-8 check as defined in
     IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3 as appropriate.

     This object is applicable for an OLT and an ONU. It has a
     distinct value for each logical link.

     Discontinuities of this counter can occur at
     re-initialization of the management system and at other
     times, as indicated by the value of the 'discontinuity-time'
     leaf defined in the ietf-interfaces YANG module (IETF RFC 8343).";

  reference
    "IEEE Std 802.3, 30.3.7.1.4";
}

leaf ompe-onu-frames-with-good-llid-good-crc8 {
  when "../../ompe-mode = 'onu'";
  type yang:counter64;
  units frames;
  config false;
  description
    "A count of frames received that 1) contain a valid SLD field
     in an ONU, 2) meet the rule for frame acceptance, and
     3) pass the CRC-8 check.

     The SLD is defined in IEEE Std 802.3, 65.1.3.3.1 or
     76.2.6.1.3.1, as appropriate.

     The rules for LLID acceptance are defined in IEEE Std 802.3,
     65.1.3.3.2 or 76.2.6.1.3.2, as appropriate.

     The CRC-8 check is defined in IEEE Std 802.3,
     65.1.3.3.3 or 76.2.6.1.3.3, as appropriate.

     This object is applicable for an ONU and has a distinct value
     for each logical link.

     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 the
     'discontinuity-time' leaf defined in the ietf-interfaces
     YANG module (IETF RFC 8343).";
   reference
    "IEEE Std 802.3, 30.3.7.1.6";
}

leaf ompe-olt-frames-with-good-llid-good-crc8 {
```

```
  when "../../ompe-mode = 'olt'";
  type yang:counter64;
  units frames;
  config false;
  description
    "A count of frames received that 1) contain a valid SLD field
     in an OLT, and 2) pass the CRC-8 check.

     The SLD is defined in IEEE Std 802.3, 65.1.3.3.1 or
     76.2.6.1.3.1, as appropriate.

     The frame acceptance are defined in IEEE Std 802.3,
     65.1.3.3.2 or 76.2.6.1.3.2, as appropriate.

     The CRC-8 check is defined in IEEE Std 802.3,
     65.1.3.3.3 or 76.2.6.1.3.3, as appropriate.

     This object is applicable for an OLT and has a distinct
     value for each logical link.

     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 the
     'discontinuity-time' leaf defined in the ietf-interfaces
     YANG module (IETF RFC 8343).";

   reference
    "IEEE Std 802.3, 30.3.7.1.6";
}

leaf in-ompe-frames-with-bad-llid {
  when "../../ompe-mode = 'olt'";
  type yang:counter64;

  units frames;

  config false;

  description
    "A count of frames received that contain a valid SLD field, as
     defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as
     appropriate, and pass the CRC-8 check as defined in IEEE Std
     802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate, but are
     discarded due to the LLID check.

     This object is applicable for an OLT and has a distinct value
     for each logical link.

     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 the 'discontinuity-time' leaf
     defined in the ietf-interfaces YANG module (IETF RFC 8343).";

  reference
    "IEEE Std 802.3, 30.3.7.1.8";
}

leaf in-ompe-frames-with-good-llid {
  type yang:counter64;
```

```
    units frames;

    config false;

    description
      "A count of frames received that contain a valid SLD field,
       as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1 as
       appropriate, but do not pass the CRC-8 check as defined in
       IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3 as appropriate.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.7.1.4";
  }

  leaf in-ompe-frames {
    type yang:counter64;

    units frames;

    config false;

    description
      "A count of frames received that contain a valid SLD field,
       as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as
       appropriate, and pass the CRC-8 check as defined in
       IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.3.7.1.6 (ONU) and 30.3.7.1.7 (OLT)";
  }

  leaf in-ompe-frames-not-match-onu-llid-broadcast {
    when "../../ompe-mode = 'onu'";

    type yang:counter64;

    units frames;

    config false;

    description
      "A count of frames received that contain a valid SLD field,
```

```
      as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as
      appropriate, pass the CRC-8 check, as defined in
      IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate,
      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 and Clause 76, as appropriate.

      This object is applicable for an ONU only.

      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 the 'discontinuity-time' leaf
      defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3.1, dot3OmpEmulationBroadcastBitNotOnuLlid";
  }

  leaf in-ompe-frames-match-onu-llid-not-broadcast {
    when "../../ompe-mode = 'onu'";

    type yang:counter64;

    units frames;

    config false;

    description
      "A count of frames received that contain a valid SLD field,
       as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as
       appropriate, pass the CRC-8 check, as defined in
       IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate,
       and contain the ONU's LLID (frame accepted) as defined in
       IEEE Std 802.3, Clause 65 and Clause 76, as appropriate.

       This object is applicable for an ONU only.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3.1, dot3OmpEmulationOnuLLIDNotBroadcast";
  }

  leaf in-ompe-frames-match-onu-llid-broadcast {
    when "../../ompe-mode = 'onu'";

    type yang:counter64;

    units frames;

    config false;

    description
      "A count of frames received that contain a valid SLD field,
       as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as
       appropriate, pass the CRC-8 check, as defined in
```

```
        IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate,
        and contain the broadcast bit in the LLID and the ONU's LLID
        (frame accepted) as defined in IEEE Std 802.3, Clause 65 and
        Clause 76, as appropriate.

        This object is applicable for an ONU only.

        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 the 'discontinuity-time' leaf
        defined in the ietf-interfaces YANG module (IETF RFC 8343).";

      reference
        "IEEE Std 802.3.1, dot3OmpEmulationBroadcastBitPlusOnuLlid";
    }

    leaf in-ompe-frames-not-match-onu-llid-not-broadcast {
      when "../../ompe-mode = 'onu'";

      type yang:counter64;

      units frames;

      config false;

      description
        "A count of frames received that contain a valid SLD field,
         as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as
         appropriate, pass the CRC-8 check, as defined in
         IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate,
         do not contain the broadcast bit in the LLID and do not
         contain the ONU's LLID (frame is NOT accepted) as defined in
         IEEE Std 802.3, Clause 65 and Clause 76, as appropriate.

         This object is applicable for an ONU only.

         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 the 'discontinuity-time' leaf
         defined in the ietf-interfaces YANG module (IETF RFC 8343).";

      reference
        "IEEE Std 802.3.1, dot3OmpEmulationNotBroadcastBitNotOnuLlid";
    }
  }
  container thresholds-trx {
    if-feature "trx-power-level-reporting-supported";

    description
      "This container defines a set of optical transceiver
       thresholds of an EPON interface as defined in
       IEEE Std 802.3, Clause 60 and Clause 75.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfEntry";

    leaf in-trx-power-low-threshold {
      if-feature "trx-power-level-reporting-supported";
```

```
    type int32;

    units "0.1 dBm";

    description
      "This object reflects the current setting of low alarm
       threshold for the input power into the optical receiver.
       If the value reported in 'in-trx-power' object drops below
       the value set in 'in-trx-power-low-threshold', a
       'in-trx-power-low-threshold-crossing' event is generated.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfLowerInputPowerThreshold";
  }

  leaf in-trx-power-high-threshold {
    if-feature "trx-power-level-reporting-supported";

    type int32;

    units "0.1 dBm";

    description
      "This object reflects the current setting of high alarm
       threshold for the input power into the optical receiver. If
       the value reported in 'in-trx-power' object exceeds the
       value set in 'in-trx-power-high-threshold', a
       'in-trx-power-high-threshold-crossing' event is generated.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfUpperInputPowerThreshold";
  }

  leaf out-trx-power-low-threshold {
    if-feature "trx-power-level-reporting-supported";

    type int32;

    units "0.1 dBm";

    description
      "This object reflects the current setting of low alarm
       threshold for the output power out of the optical
       transmitter. If the value reported in 'out-trx-power' object
       drops below the value set in 'out-trx-power-low-threshold',
       a 'out-trx-power-low-threshold-crossing' event is generated.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfLowerOutputPowerThreshold";
  }
```

```
    leaf out-trx-power-high-threshold {
      if-feature "trx-power-level-reporting-supported";

      type int32;

      units "0.1 dBm";

      description
        "This object reflects the current setting of high alarm
         threshold for the output power out of the optical
         transmitter. If the value reported in 'out-trx-power' object
         exceeds the value set in 'out-trx-power-high-threshold', a
         'out-trx-power-high-threshold-crossing' event is generated.

         This object is applicable for an OLT and an ONU. It has a
         distinct value for each logical link.";

      reference
        "IEEE Std 802.3.1, dot3ExtPkgOptIfUpperOutputPowerThreshold";
    }
  }
  container statistics-trx {
    if-feature "trx-power-level-reporting-supported";

    description
      "This container defines a set of optical transceiver
       statistics counters of an EPON interface as defined in
       IEEE Std 802.3, Clause 60 and Clause 75.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfEntry";

    leaf in-trx-power-signal-detect {
      type boolean;

      config false;

      description
        "This object indicates whether a valid optical signal was
         detected (when read as 'true') or not (when read as 'false')
         at the input to the optical transceiver.

         This object is applicable for an OLT and an ONU. It has a
         distinct value for each logical link.";

      reference
        "IEEE Std 802.3.1, dot3ExtPkgOptIfSignalDetect";
    }

    leaf in-trx-power {
      type int32;

      units "0.1 dBm";

      config false;

      description
        "This object reflects the value of the input power, as
```

```
        measured at the optical transceiver, expressed in units of
        0.1 dBm.

        At the ONU, the measurement is performed in a continuous
        manner.

        At the OLT, the measurement is performed in a burst-mode
        manner, for each incoming data burst.

        This object is applicable for an OLT and an ONU. It has a
        distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfInputPower";
  }

  leaf in-trx-power-low-15-minutes-bin {
    type int32;

    units "0.1 dBm";

    config false;

    description
      "This object reflects the lowest value of the input power
       during the period of the last 15 minutes, as measured at the
       optical transceiver, and expressed in units of 0.1 dBm.

       At the ONU, the measurement is performed in a continuous
       manner and stored in a rolling 15-minutes' long observation
       bin.

       At the OLT, the measurement is the average power for each
       incoming data burst, and stored in a rolling 15-minutes'
       long observation bin.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfLowInputPower";
  }

  leaf in-trx-power-high-15-minutes-bin {
    type int32;

    units "0.1 dBm";

    config false;

    description
      "This object reflects the highest value of the input power
       during the period of the last 15 minutes, as measured at the
       optical transceiver, and expressed in units of 0.1 dBm.

       At the ONU, the measurement is performed in a continuous
       manner and stored in a rolling 15-minutes' long observation
       bin.
```

```
      At the OLT, the measurement is the average power for each
      incoming data burst, and stored in a rolling 15-minutes'
      long observation bin.

      This object is applicable for an OLT and an ONU. It has a
      distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfHighInputPower";
  }

  leaf out-trx-power-signal-detect {
    type boolean;

    config false;

    description
      "This object indicates whether a valid optical signal was
       detected (when read as 'true') or not (when read as 'false')
       at the output from the optical transceiver.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfTransmitAlarm";
  }

  leaf out-trx-power {
    type int32;

    units "0.1 dBm";

    config false;

    description
      "This object reflects the value of the output power, as
       measured at the optical transceiver, expressed in units of
       0.1 dBm.

       At the ONU, the measurement is performed in a burst-mode
       manner for each outgoing data burst.

       At the OLT, the measurement is performed in a continuous
       manner.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfOutputPower";
  }

  leaf out-trx-power-low-15-minutes-bin {
    type int32;

    units "0.1 dBm";

    config false;
```

```
    description
      "This object reflects the lowest value of the output power
       during the period of the last 15 minutes, as measured at the
       optical transceiver, and expressed in units of 0.1 dBm.

       At the ONU, the measurement is performed in a burst-mode
       manner and stored in a rolling 15-minutes' long observation
       bin.

       At the OLT, the measurement is the average power for each
       incoming data burst, and stored in a rolling 15-minutes'
       long observation bin.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfLowOutputPower";
  }

  leaf out-trx-power-high-15-minutes-bin {
    type int32;

    units "0.1 dBm";

    config false;

    description
      "This object reflects the highest value of the output power
       during the period of the last 15 minutes, as measured at the
       optical transceiver, and expressed in units of 0.1 dBm.

       At the ONU, the measurement is performed in a burst-mode
       manner and stored in a rolling 15-minutes' long observation
       bin.

       At the OLT, the measurement is the average power for each
       incoming data burst, and stored in a rolling 15-minutes'
       long observation bin.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgOptIfHighOutputPower";
  }

  leaf trx-data-reliable {
    if-feature "trx-power-level-reporting-supported";

    type boolean;

    config false;

    description
      "This object indicates whether data contained in individual
       counters in 'statistics-trx' container are reliable
       (when read as 'true') or not (when read as 'false').
```

```
        This object is applicable for an OLT and an ONU. It has a
        distinct value for each logical link.";

      reference
        "IEEE Std 802.3.1, dot3ExtPkgOptIfSuspectedFlag";
    }

  }

  container statistics-pon-fec {
    when
      "(../fec-capability = 'supported') and
       (../fec-mode = 'enabled-Tx-Rx')";

    if-feature "fec-supported";

    config false;

    description
      "This container defines a set of FEC-related statistics
       counters of an EPON interface, as defined in
       IEEE Std 802.3, Clause 65 and Clause 76.";

    reference
      "IEEE Std 802.3.1, dot3OmpEmulationStatEntry";

    leaf fec-code-group-violations {
      type yang:counter64;

      units code-group;

      config false;

      description
        "For 1G-EPON this 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 35-1). The contents of this counter is undefined when
         FEC is operating. For 10G-EPON this object is not applicable.

         This object is applicable for an OLT and an ONU. At the
         OLT, it has a distinct value for each logical link.

         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 the 'discontinuity-time' leaf
         defined in the ietf-interfaces YANG module (IETF RFC 8343).";

      reference
        "IEEE Std 802.3, 30.5.1.1.14";
    }

    leaf fec-buffer-head-coding-violations {
      type yang:counter64;

      units code-group;

      config false;
```

```
    description
      "For 1G-EPON PHY, this object represents the count of the
       number of invalid code-groups received directly from the
       link when FEC is enabled. When FEC is disabled this
       counter stops counting.

       For 10G-EPON PHYs, this object is set to zero.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3.1, dot3EponFecBufferHeadCodingViolation";
  }

  leaf fec-code-word-corrected-errors {
    type yang:counter64;

    units code-group;

    config false;

    description
      "For 1G-EPON or 10G-EPON PHYs, this object represents a count
       of corrected FEC blocks.

       This counter increments by one for each received FEC block
       that contained detected errors and was corrected by the FEC
       function in the PHY.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.

       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 the 'discontinuity-time' leaf
       defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.5.1.1.17";
  }

  leaf fec-code-word-uncorrected-errors {
    type yang:counter64;

    units code-group;

    config false;

    description
      "For 1G-EPON or 10G-EPON PHYs, this object represents a count of
       uncorrectable FEC blocks.
```

```
        This counter increments by one for each received FEC block
        that contained detected errors and was not corrected by the
        FEC function in the PHY.

        This object is applicable for an OLT and an ONU. It has a
        distinct value for each logical link.

        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 the 'discontinuity-time' leaf
        defined in the ietf-interfaces YANG module (IETF RFC 8343).";

    reference
      "IEEE Std 802.3, 30.5.1.1.18";
  }
}

container mpcp-logical-link-admin-actions {
  description
    "Container of actions.";
  action state-change-action-type {
    description
      "Request a state change on the interface.";
    input {
      leaf state-change-action-type {
        type identityref {
          base state-change-action-type;
        }
      description
        "Type of interface state change requested.";
      }
    }
  }
  action reset-action-type {
    description
      "Request a reset-action of the interface.";
    input {
      leaf reset-action-type {
        type identityref {
          base reset-action-type;
      }
      description
        "Type of reset action requested of the interface.";
      }
    }
  }
  action register-type {
    description
      "Request a registration action.";
    input {
      leaf register-type {
        type identityref {
          base register-type;
      }
      description
        "Type of registration action requested of the interface.";
      }
    }
  }
```

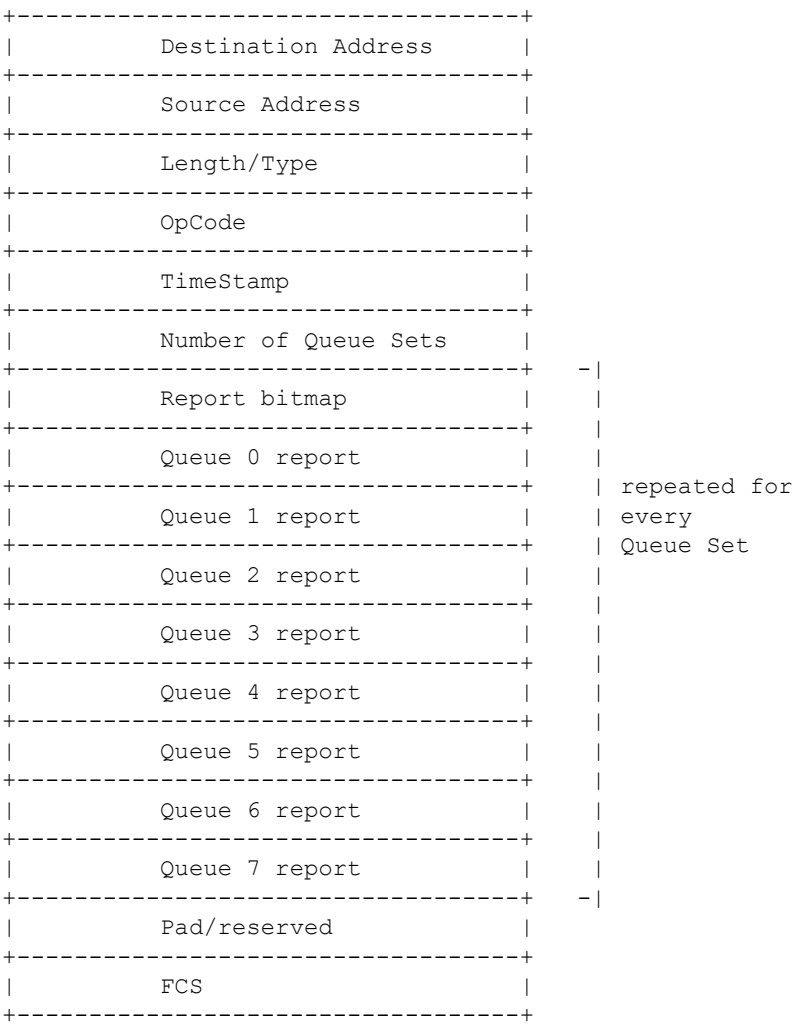
```
}

list mpcp-queues {
  key mpcp-queue-index;

  description
    "An instance of this object for each value of
     'mpcp-queue-index' is created when a new logical link is
     registered and deleted when the logical link is deregistered.

     All instances of this object in the ONU associated with the
     given logical link are then mapped to a REPORT MPCPDU, when
     generated.

     +-----------------------------------+
     |          Destination Address      |
     +-----------------------------------+
     |          Source Address           |
     +-----------------------------------+
     |          Length/Type              |
     +-----------------------------------+
     |          OpCode                   |
     +-----------------------------------+
     |          TimeStamp                |
     +-----------------------------------+
     |          Number of Queue Sets     |
     +-----------------------------------+   -|
     |          Report bitmap            |    |
     +-----------------------------------+    |
     |          Queue 0 report           |    |
     +-----------------------------------+    | repeated
     |          Queue 1 report           |    | for 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 N report' field reports the current occupancy of
     each upstream transmission queue associated with the given
     logical link.

     The 'Number of Queue Sets' field defines the number of
     reported 'Queue N report' sets.

     For each Queue Set, the 'Report bitmap' field defines which
     upstream transmission queues are present in the REPORT MPCPDU.
```

```
   Although the REPORT MPCPDU can report current occupation for
   up to 8 upstream transmission queues in a single REPORT MPCPDU
   , the actual number is flexible. The 'mpcp-queue-group'
   grouping has a variable size that is limited by value of
   'mpcp-maximum-queue-count-per-report' object, allowing ONUs
   report the occupancy of fewer upstream transmission queues, as
   needed.

   This object is applicable for an OLT and an ONU. At the OLT,
   this object has a distinct value for each logical link and
   every queue. At the ONU, it has a distinct value for
   every queue.";

reference
  "IEEE Std 802.3.1, dot3ExtPkgQueueEntry";

leaf mpcp-queue-index {
  type uint8 {
    range "0 .. 7" {
      description
        "This object indicates the identity (index) of a queue in
         the ONU. It can have a value between 0 and 7, limited by
         the value stored in the
         'mpcp-maximum-queue-count-per-report' object.";

      reference
        "See 'mpcp-maximum-queue-count-per-report' object";
    }
  }

  description
    "An object represents the index of an upstream transmission
     queue storing subscriber packets. The size (occupancy) of
     the upstream transmission queue identified by this object is
     then reported within REPORT MPCPDU, defined in
     IEEE Std 802.3, Clause 64 and Clause 77.

     This object indicates the identity (index) of a queue in the
     ONU. It can have a value between 0 and 7, limited by the value
     stored in the 'mpcp-maximum-queue-count-per-report' object.

     This object is applicable for an OLT and an ONU. It has a
     distinct value for each logical link and each queue.
     At the ONU, it has a distinct value for each queue.";

  reference
    "IEEE Std 802.3.1, dot3QueueIndex";
}

leaf mpcp-queue-threshold-count {
  type uint8 {
    range "0 .. 7" {
      description
        "This object indicates the identity (index) of a queue in
         the ONU. It can have a value between 0 and 7, limited by
         the value stored in the
         'mpcp-maximum-queue-count-per-report' object.";

      reference
```

```
        "See 'mpcp-queue-threshold-count-max' object";
      }
    }

    description
      "This object reflects the number of reporting thresholds for
       the specific upstream transmission queue, reflected in the
       REPORT MPCPDU, as defined in IEEE Std 802.3,
       Clause 64 and Clause 77.

       Each 'Queue set' provides information for the specific
       upstream transmission queue occupancy of frames below the
       matching reporting threshold.

       A read of this object reflects the number of reporting
       thresholds for the specific upstream transmission queue.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link and each queue.
       At the ONU, it has a distinct value for each queue.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgObjectReportNumThreshold";
  }

  leaf mpcp-queue-threshold-count-max {
    type uint8 {
      range "0 .. 7" {
        description
          "This object can have a value between 0 and 7.";
      }
    }
    config false;
    description
      "This object reflects the maximum number of reporting
       thresholds for the specific upstream transmission queue,
       reflected in the REPORT MPCPDU, as defined in
       IEEE Std 802.3, Clause 64 and Clause 77.

       A read of this object reflects the maximum number of
       reporting thresholds for the specific upstream transmission
       queue.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link and each queue.
       At the ONU, it has a distinct value for each queue.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgObjectReportMaximumNumThreshold";
  }

  list mpcp-queue-thresholds {
    when "../mpcp-queue-threshold-count > 0";

    key mpcp-queue-set-index;

    max-elements 7;

    description
```

"An instance of this object for each value of
'mpcp-queue-index' is created when a new logical link is
registered and deleted when the logical link is
deregistered.

All instances of this object in the ONU associated with the
given logical link are then mapped to a REPORT MPCPDU,
when generated.



The 'Queue N report' field reports the current occupancy of
each upstream transmission queue associated with the given
logical link.

The 'Number of Queue Sets' field defines the number of
reported 'Queue N report' sets.

For each Queue Set, the 'Report bitmap' field defines which
upstream transmission queues are present in the
REPORT MPCPDU. Although the REPORT MPCPDU can report current
occupation for up to 8 upstream transmission queues in a
single REPORT MPCPDU, the actual number is flexible.

The 'mpcp-queue-group' grouping has a variable size that is

```
   limited by value of 'mpcp-maximum-queue-count-per-report'
   object, allowing ONUs to report the occupancy of fewer
   upstream transmission queues, as needed.

   This object is applicable for an OLT and an ONU. It has a
   distinct value for each logical link and every queue.
   At the ONU, it has a distinct value for every queue.";

reference
  "IEEE Std 802.3.1, dot3ExtPkgQueueSetsEntry";

leaf mpcp-queue-set-index {
  type uint8 {
    range "0 .. 7" {
      description
        "This object indicates the identity (index) of a queue
         in the ONU. It can have a value between 0 and 7,
         limited by the value stored in the
         'mpcp-maximum-queue-count-per-report' object.";

      reference
        "See 'mpcp-maximum-queue-count-per-report' object";
    }
  }

  description
    "This object represents the index of the Queue Set for the
     'mpcp-queue-set-group' grouping. The size (occupancy) of
     the upstream transmission queues belonging to the given
     Queue Set is then reported within REPORT MPCPDU, defined
     in IEEE Std 802.3, Clause 64 and Clause 77.

     This object can have a value between 0 and 7, limited by
     the value stored in the 'mpcp-queue-threshold-count-max'
     object.";

  reference
    "IEEE Std 802.3.1, dot3QueueSetIndex";
}

leaf mpcp-queue-set-threshold {
  type uint64;

  units "TQ";

  default "0";

  description
    "This object defines the value of a reporting threshold
     for each Queue Set stored in REPORT MPCPDU defined in
     IEEE Std 802.3, Clause 64 and Clause 77.

     The number of Queue Sets for each upstream transmission
     queue is defined in the 'mpcp-queue-threshold-count'
     object.

     Within REPORT MPCPDU, each Queue Set provides information
     on the current upstream transmission queue occupancy for
     frames below the matching threshold.
```

```
        The value stored in this object is expressed in the units
        of Time quanta (TQ), where 1 TQ = 16 ns.

        A read of this object provides the current threshold value
        for the specific upstream transmission queue.

        This object is applicable for an OLT and an ONU. At the
        OLT, it has a distinct value for each logical link, each
        queue, and each Queue Set.

        At the ONU, it has a distinct value for each queue and
        each Queue Set.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgObjectReportThreshold";
  }
}

leaf in-mpcp-queue-frames {
  type yang:counter64;

  config false;

  description
    "A count of the number of times a frame reception event
     results in a frame being queued in (for ONUs) or received
     from (for OLTs) the corresponding queue. This object is
     incremented by one for each frame written to (in the case
     of the ONU) or received for (in case of the OLT) the
     associated queue.

     The queue index matches the queue number in REPORT MPCPDU,
     as defined in IEEE Std 802.3, Clause 64 and Clause 77.

     This object is applicable for an OLT and an ONU. At the OLT,
     it has a distinct value for each logical link and each queue.
     At the ONU, it has a distinct value for each queue.

     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 the 'discontinuity-time' leaf
     defined in the ietf-interfaces YANG module (IETF RFC 8343).";

  reference
    "IEEE Std 802.3.1, dot3ExtPkgStatRxFramesQueue";
}

leaf out-mpcp-queue-frames {
  when "../../mpcp-mode = 'onu'";

  type yang:counter64;

  config false;

  description
    "This object reflects the number of frame transmission
     events from the corresponding upstream transmission queue.
     This object is incremented by one for each frame transmitted
```

```
        , when it is output from the associated queue.

        The queue index matches the queue number in REPORT MPCPDU,
        as defined in IEEE Std 802.3, Clause 64 and Clause 77.

        This object is applicable for an ONU only. At the ONU, it
        has a distinct value for each queue.

        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 the 'discontinuity-time' leaf
        defined in the ietf-interfaces YANG module (IETF RFC 8343).";

      reference
        "IEEE Std 802.3.1, dot3ExtPkgStatTxFramesQueue";
    }

    leaf mpcp-queue-frames-drop {
      when "../../mpcp-mode = 'onu'";

      type yang:counter64;

      config false;

      description
        "This object reflects the number of frame drop events from
         the corresponding upstream transmission queue. This object
         is incremented by one for each frame dropped in the
         associated queue.

         The queue index matches the queue number in REPORT MPCPDU,
         as defined in IEEE Std 802.3, Clause 64 and Clause 77.

         This object is applicable for an ONU only. At the ONU, it
         has a distinct value for each queue.

         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 the 'discontinuity-time' leaf
         defined in the ietf-interfaces YANG module (IETF RFC 8343).";

      reference
        "IEEE Std 802.3.1, dot3ExtPkgStatDroppedFramesQueue";
    }
  }

  list multicast-IDs {
   key multicast-ID;
    description
      "Multicast-IDs list of multicast IDs
       to be recognized by the device.";
    leaf multicast-ID {
      type uint32;
      description
        "Multicast-IDs to be recognized by the device.";
    reference
      "IEEE Std 802.3, 30.3.5.1.25";
    }
  }
```

```
leaf fec-capability {
  type fec-capability;

  config false;

  description
    "This object is used to identify whether the given interface
     is capable of supporting FEC or not.";
}

leaf mpcp-mode {
  type mpcp-mode;

  config false;

  description
    "This object is used to identify the operational state of the
     MultiPoint MAC Control sublayer as defined in
     IEEE Std 802.3, Clause 64 and Clause 77.

     Reading 'olt' for an OLT (controller) mode and 'onu' for an ONU
     (client) mode.

     This object is used to identify the operational mode for the
     MPCP objects.

     This object is applicable for an OLT, with the same value for
     all logical links, and for an ONU.";

  reference
    "IEEE Std 802.3, 30.3.5.1.3";
}

leaf mpcp-sync-time {
  type uint64;

  units "TQ (16ns)";

  config false;

  description
    "This object reports the 'sync lock time' of the OLT
     receiver in units of Time Quanta (TQ; 1 TQ = 16 ns; see
     IEEE Std 802.3, Clause 64 and Clause 77).

     The value returned is equal to [sync lock time ns]/16,
     rounded up to the nearest TQ. If this value exceeds
     4,294,967,295 TQ, the value 4,294,967,295 TQ is returned.

     This object is applicable for an OLT, with distinct values for
     all logical links, and for an ONU.";

  reference
    "IEEE Std 802.3.1, dot3MpcpSyncTime";
}

leaf mpcp-logical-link-id {
  type mpcp-supported;
```

```
    config false;

    description
      "This object is used to identify the operational state of the
       MultiPoint MAC Control sublayer as defined in
       IEEE Std 802.3, Clause 64 and Clause 77.

       Reading 'olt' for an OLT (controller) mode and 'onu' for an ONU
       (client) mode.

       This object is used to identify the operational mode for the
       MPCP objects.

       This object is applicable for an OLT, with the same value for
       all logical links, and for an ONU.";

    reference
      "IEEE Std 802.3, 30.3.5.1.3";
  }

  leaf mpcp-remote-mac-address {
    type yang:mac-address;

    config false;

    description
      "This object 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 multicast address assigned
       for MAC Control as specified in IEEE Std 802.3, Annex 31A;

       2) the lengthOrType field value equal to the Type assigned for
       MAC Control as specified in IEEE Std 802.3, Annex 31A;

       3) an MPCP Control opcode value equal to the subtype assigned
       for MPCP as specified in IEEE Std 802.3, Annex 31A.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.

       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";
  }

  leaf mpcp-logical-link-state {
    type mpcp-logical-link-state;

    config false;

    description
```

```
      "This object identifies the registration state of the
       MultiPoint MAC Control sublayer as defined in
       IEEE Std 802.3, Clause 64 and Clause 77.

       When this object has the enumeration 'unregistered', the
       interface is unregistered and may be used for registering
       a link partner.

       When this object has the enumeration 'registering',
       the interface is in the process of registering a link-partner.

       When this object has the enumeration 'registered', the
       interface has an established link-partner.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3, 30.3.5.1.6";
  }

  leaf mpcp-elapsed-time-out {
    type uint64;

    units "TQ (16ns)";

    config false;

    description
      "This object reports the interval from the last MPCP frame
       transmission in increments of Time Quanta (TQ; 1 TQ = 16 ns;
       see IEEE Std 802.3, Clause 64 and Clause 77).

       The value returned is equal to [interval from last MPCP
       frame transmission on this EPON interface, expressed
       in ns]/16. If this value exceeds 4,294,967,295 TQ, the value
       4,294,967,295 TQ is returned.

       This object is applicable for an OLT and an ONU. It has a
       distinct value for each logical link.";

    reference
      "IEEE Std 802.3, 30.3.5.1.19";
  }

  leaf mpcp-elapsed-time-in {
    type uint64;

    units "TQ (16ns)";

    config false;

    description
      "This object reports the interval from the last MPCP frame
       reception in increments of Time Quanta (TQ; 1 TQ = 16 ns; see
       IEEE Std 802.3, Clause 64 and Clause 77).

       The value returned is equal to [interval from last MPCP
       frame reception on this EPON interface, expressed in
```

```
      ns]/16. If this value exceeds 4,294,967,295 TQ, the value
      4,294,967,295 TQ is returned.

      This object is applicable for an OLT and an ONU. It has a
      distinct value for each logical link.";

    reference
      "IEEE Std 802.3, 30.3.5.1.20";
  }

  leaf mpcp-round-trip-time {
    when "../ompe-mode = 'olt'";
    type uint16;

    units "TQ (16ns)";

    config false;

    description
      "This object reports the MPCP round trip time in increments
       of Time Quanta (TQ; 1 TQ = 16 ns; see IEEE Std 802.3,
       Clause 64 and Clause 77).

       The value returned is equal to [round trip time in ns]/16.
       If this value exceeds 65,535 TQ, the value 65,535 TQ is
       returned.

       This object is applicable for an OLT. It has a distinct value
       for each logical link.";

    reference
      "IEEE Std 802.3, 30.3.5.1.21";
  }

  leaf mpcp-maximum-grant-count {
    when "../ompe-mode = 'onu'";
    type uint8;

    config false;

    description
      "This object 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 ONU and has a distinct value
       for each logical link.";

    reference
      "IEEE Std 802.3, 30.3.5.1.24";
  }

  leaf mpcp-logical-link-count {
    type mpcp-llid-count;

    units LLID;

    config false;
```

```
    description
      "This object reflects the number of logical links registered
       on the OLT / ONU. The LLID field, as defined in the
       IEEE Std 802.3, Clause 65 and Clause 76, is a 2-byte register
       (15-bit field and a broadcast bit) limiting the number of
       logical links to 32,768.

       This object is initialized to the value of 0 when the
       OLT / ONU is powered up.

       This object is applicable for an OLT and an ONU. It has the same
       value for all logical links.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgObjectNumberOfLLIDs";
  }

  leaf mpcp-maximum-queue-count-per-report {
    when "../ompe-mode = olt'";
    type mpcp-maximum-queue-count-per-report;

    config false;

    description
      "This object reflects the maximum number of queues (0-7) that can
       be accepted by the OLT in a single REPORT MPCPDU, as defined in
       IEEE Std 802.3, Clause 64 and Clause 77.

       This object is applicable for an OLT and has a distinct value
       for each logical link.";

    reference
      "IEEE Std 802.3.1, dot3ExtPkgObjectReportMaximumNumQueues";
  }

  leaf ompe-mode {
    type ompe-mode;

    config false;

    description
      "This object indicates the mode of operation of the
       Reconciliation Sublayer for Point-to-Point Emulation (see
       IEEE Std 802.3, 65.1 or 76.2 as appropriate).

       The value of 'unknown' is assigned in initialization; true
       state or type is not yet known.

       The value of 'olt' is assigned when the sublayer is operating
       in OLT mode.

       The value of 'onu' is assigned when the sublayer is operating
       in ONU mode.

       This object is applicable for an OLT and an ONU. It has the same
       value for each logical link.";

    reference
      "IEEE Std 802.3, 30.3.7.1.2";
```

```
        }
      }
    }
```

8. YANG module for Ethernet Link OAM (ELO)

8.1 Introduction

IEEE Std 802.3, Clause 57 includes management capabilities for 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 portion of the YANG module for use with NETCONF or RESTCONF to manage these Ethernet-like interface capabilities.

8.2 Overview

Ethernet OAM is composed of a core set of functions and a set of optional functional groups as described in Clause 57 of IEEE Std 802.3. 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.

Ethernet OAM provides single-hop functionality in that it works only between two directly connected Ethernet stations. Ethernet OAM has three functional objectives, which are detailed in 8.2.1 through 8.2.3. The definition of a basic Ethernet OAM protocol data unit is given in 8.2.4.

8.2.1 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 data nodes and notifications within the NETCONF management framework.

8.2.2 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 events, at both the local end and the far end of the link, into the YANG-based 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.

8.2.3 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 data nodes controlling loopback operation and reading the status of the loopback state.

8.2.4 Ethernet OAM protocol data units

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

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

The readable data nodes in this module can provide information about network traffic, and therefore, they may be considered sensitive. In particular, OAM provides mechanisms for reading the Clause 30 IEEE 802.3 management attributes from a link partner via a layer 3 protocol. IEEE Std 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 Std 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 Std 802.3 OAM, this protection from remote access is accomplished by configuring the CPE-side in passive mode using the mode leaf. 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 NETCONF/RESTCONF.

The administrative state and mode are also configuration nodes. Disabling OAM can interrupt management activities between peer devices, potentially causing serious problems. Setting the mode node 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 configuration nodes. These nodes 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 admin and mode nodes. As mentioned previously, inadequate protection of these variables can result in access to link information and functions.

8.4 Mapping of IEEE 802.3 managed objects

This subclause contains the mapping between the YANG data nodes defined in this clause and the attributes defined in IEEE Std 802.3, Clause 30. Table 8–1 provides the mapping between the *ieee802-ethernet-link-oam* module data nodes and the OAM attributes of IEEE Std 802.3, Clause 30.

Table 8–1—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-link-oam* YANG data nodes

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-link-oam* YANG data nodes		
Managed object(s)	**Attribute(s)**		**Container(s)**	**Data node(s)**	**R/W**
oOAM	aOAMAdminState	30.3.6.1.2	interfaces/interface/ethernet/link-oam	admin	R/W
	dot3OamOperStatus aOAMDiscoveryState aOAMLocalFlagsField aOAMRemoteFlagsField	IEEE Std 802.3.1 30.3.6.1.4 30.3.6.1.10 30.3.6.1.11	interfaces/interface/ethernet/link-oam/discovery-info/local	operational-status	R
	aOAMLocalState	30.3.6.1.14		loopback-mode	R
	aOAMMode	30.3.6.1.3		mode	RW
	aOAMLocalRevision	30.3.6.1.12		revision	R
	aOAMLocalPDUConfiguration	30.3.6.1.8		oammtu	R
	aOAMLocalConfiguration	30.3.6.1.6	interfaces/interface/ethernet/link-oam/discovery-info/remote/functions-supported	uni-directional-link-fault	R/W
				loopback	R/W
				mib-retrieval	R/W
		aOAMLocalConfiguration	interfaces/interface/ethernet/link-oam/discovery-info/local/functions-supported/link-monitor	link-monitoring	R/W
	aOAMLocalErrSymPeriodConfig aOAMLocalErrFrameConfig aOAMLocalErrFramePeriodConfig aOAMLocalErrFrameSecsSummaryConfig aOAMLocalErrSymPeriodConfig	30.3.6.1.34 30.3.6.1.36 30.3.6.1.38 30.3.6.1.40 30.3.6.1.42	interfaces/interface/ethernet/link-oam/link-monitor/event-type	threshold-type window threshold	R/W R/W R/W
	aOAMRemoteMACAddress	30.3.6.1.5	interfaces/interface/ethernet/link-oam/discovery-info/remote	mac-address	R
	aOAMRemoteVendorOUI	30.3.6.1.16		vendor-oui	R
	aOAMRemoteVendorSpecificInfo	30.3.6.1.17		vendor-info	R

Table 8–1—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-link-oam* YANG data nodes *(continued)*

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-link-oam* YANG data nodes		
Managed object(s)	**Attribute(s)**		**Container(s)**	**Data node(s)**	**R/W**
OAM	aOAMLocalConfiguration	30.3.6.1.6	interfaces/interface/ethernet/link-oam/discovery-info/local/functions-supported	uni-directional-link-fault	R
				loopback	R
				link-monitoring	R
				mib-retrieval	R
	dot3OamEventLogEntry	IEEE Std 802.3.1	interfaces-state/ interface/ethernet/link-oam/event-log/event-log-entry	index	R
				oui	R
				timestamp	R
				location	R
				event-type	R
				running-total	R
				event-total	R
	aOAMLocalErrSymPeriodEvent aOAMLocalErrFrameEvent aOAMLocalErrFramePeriodConfig aOAMLocalErrFrameSecsSummaryEvent aOAMRemoteErrSymPeriodEvent aOAMRemoteErrFrameEvent aOAMRemoteErrFramePeriodEvent aOAMRemoteErrFrameSecsSummaryEvent	30.3.6.1.35 30.3.6.1.37 30.3.6.1.38 30.3.6.1.41 30.3.6.1.42 30.3.6.1.43 30.3.6.1.44 30.3.6.1.45	interfaces/interface/ethernet/link-oam/event-log/event-log-entry/threshold	threshold-event-type window threshold value	R R R R
	Dot3OamStatsEntry	RFC-4878	interfaces/interface/ethernet/link-oam/statistics	out-information	R
	aOAMInformationTx	30.3.6.1.20			
	aOAMInformationRx	30.3.6.1.21		in-information	R

Table 8–1—Mapping between IEEE Std 802.3, Clause 30 managed objects and *ieee802-ethernet-link-oam* YANG data nodes *(continued)*

IEEE Std 802.3, Clause 30		Reference	Corresponding *ieee802-ethernet-link-oam* YANG data nodes		
Managed object(s)	**Attribute(s)**		**Container(s)**	**Data node(s)**	**R/W**
OAM	aOAMVariableRequestRx	30.3.6.1.29	interfaces/interface/ethernet/link-oam/statistics	variable-requeste-rx	R
	aOAMVariableResponseTx	30.3.6.1.30		out-variable-response	R
	aOAMVariableResponseRx	30.3.6.1.31		in-variable-response	R
	aOAMOrganizationSpecificTx	30.3.6.1.32		out-org-specific	R
	aOAMOrganizationSpecificRx	30.3.6.1.33		in-org-specific	R
	aOAMUnsupportedCodesTx	30.3.6.1.18		unsupported-condes-tx	R
	aOAMUnsupportedCodesRx	30.3.6.1.19		in-unsupported-codes	R
	aFramesLostDueToOAMError	30.3.6.1.46		frames-lost-due-to-oam	R
	aOAMLocalErrSymPeriodEvent, Errored Symbols	30.3.6.1.35		local-error-symbol-period-log-entries	R
	aOAMLocalErrFrameEvent, Errored Frames	30.3.6.1.37		local-error-frame-log-entries	R
	aOAMLocalErrFramePeriodEvent, Errored Frames	30.3.6.1.39		local-error-frame-period-log-entries	R
	aOAMLocalErrFrameSecsSummar-yEvent, Errored Frame Seconds Summary	30.3.6.1.41		local-error-frame-second-log-entries	R
	aOAMRemoteErrSymPeriodEvent, Errored Symbols	30.3.6.1.42		remote-error-symbol-period-log-entries	R
	aOAMRemoteErrFrameEven, Errored Frames	30.3.6.1.43		remote-error-frame-log-entries	R
	aOAMRemoteErrFramePeriodEvent, Errored Frames	30.3.6.1.44		remote-error-frame-period-log-entries	R
	aOAMRemoteErrFrameSecsSummar-yEvent, Errored Frame Seconds Summary	30.3.6.1.45		remote-error-frame-second-log-entries	R

8.5 YANG module definition[11]

The YANG module tree hierarchy uses terms defined in IETF RFC 8407.

8.5.1 Tree hierarchy

```
module: ieee802-ethernet-link-oam
  augment /if:interfaces/if:interface:
    +--rw link-oam!
       +--rw admin?                admin-state
       +--rw discovery-info
       |  +--rw local
       |  |  +--ro operational-status     operational-state
       |  |  +--ro loopback-mode          loopback-status {remote-loopback-initiate or
remote-loopback-respond}?
       |  |  +--rw mode?                  mode
       |  |  +--rw functions-supported
       |  |  |  +--rw uni-directional-link-fault?   boolean {uni-directional-link-fault}?
       |  |  |  +--rw loopback?                     boolean {remote-loopback-initiate}?
       |  |  |  +--rw link-monitor {link-monitoring-remote or
link-monitoring-local}?
       |  |  |  |  +--rw link-monitoring?   boolean
       |  |  |  |  +--rw event-type* [threshold-type] {link-monitoring-local}?
       |  |  |  |     +--rw threshold-type    threshold-event-enum
       |  |  |  |     +--rw window?           uint64
       |  |  |  |     +--rw threshold?        uint64
       |  |  |  +--rw mib-retrieval?               boolean {remote-mib-retrieval-initiate or
remote-mib-retrieval-respond}?
       |  |  +--ro revision?              uint64
       |  |  +--ro oammtu?                uint16
       |  +--ro remote
       |     +--ro mac-address?           yang:mac-address
       |     +--ro vendor-oui?            vendor-oui
       |     +--ro vendor-info?           uint64
```

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

```
       |     +--ro loopback-mode          loopback-status
       |     +--ro mode?                  mode
       |     +--ro functions-supported
       |     |  +--ro uni-directional-link-fault?   boolean
       |     |  +--ro loopback?                     boolean
       |     |  +--ro link-monitoring?              boolean
       |     |  +--ro mib-retrieval?                boolean
       |     +--ro revision?              uint64
       |     +--ro oammtu?                uint16
       +--ro event-log
       |  +--ro event-log-entry* [index]
       |     +--ro index            uint64
       |     +--ro oui              vendor-oui
       |     +--ro timestamp        uint64
       |     +--ro location         event-location
       |     +--ro event-type       identityref
       |     +--ro running-total    yang:counter64
       |     +--ro event-total      yang:counter64
       |     +--ro threshold {link-monitoring-local or
link-monitoring-remote}?
       |        +--ro threshold-event-type    threshold-event-enum
       |        +--ro window                  uint64
       |        +--ro threshold               uint64
       |        +--ro value                   uint64
       +--ro statistics
       |  +--ro out-information                         yang:counter64
       |  +--ro in-information                          yang:counter64
       |  +--ro out-unique-event-notification           yang:counter64 {link-monitoring-local}?
       |  +--ro in-unique-event-notification            yang:counter64 {link-monitoring-remote}?
       |  +--ro out-duplicate-event-notification        yang:counter64 {link-monitoring-local}?
       |  +--ro in-duplicate-event-notification         yang:counter64 {link-monitoring-remote}?
       |  +--ro out-loopback-control                    yang:counter64 {remote-loopback-initiate}?
       |  +--ro in-loopback-control                     yang:counter64 {remote-loopback-respond}?
       |  +--ro out-variable-request                    yang:counter64 {remote-mib-retrieval-initiate}?
       |  +--ro in-variable-request                     yang:counter64 {remote-mib-retrieval-respond}?
       |  +--ro out-variable-response                   yang:counter64 {remote-mib-retrieval-respond}?
```

```
     |  +--ro in-variable-response                      yang:counter64 {remote-mib-retrieval-initiate}?
     |  +--ro out-org-specific                          yang:counter64
     |  +--ro in-org-specific                           yang:counter64
     |  +--ro out-unsupported-codes                     yang:counter64
     |  +--ro in-unsupported-codes                      yang:counter64
     |  +--ro frames-lost-due-to-oam                    yang:counter64
     |  +--ro local-error-symbol-period-log-entries     yang:counter64
     |  +--ro local-error-frame-log-entries             yang:counter64
     |  +--ro local-error-frame-period-log-entries      yang:counter64
     |  +--ro local-error-frame-second-log-entries      yang:counter64
     |  +--ro remote-error-symbol-period-log-entries    yang:counter64 {link-monitoring-remote}?
     |  +--ro remote-error-frame-log-entries            yang:counter64 {link-monitoring-remote}?
     |  +--ro remote-error-frame-period-log-entries     yang:counter64 {link-monitoring-remote}?
     |  +--ro remote-error-frame-second-log-entries     yang:counter64 {link-monitoring-remote}?
     +---x remote-loopback {remote-loopback-initiate}?
     |  +---w input
     |  |  +---w enable    boolean
     |  +--ro output
     |     +--ro success          boolean
     |     +--ro error-message?   string
     +---x reset-stats
     |  +--ro output
     |     +--ro success          boolean
     |     +--ro error-message?   string
     +---n non-threshold-event
     |  +---- oui              vendor-oui
     |  +---- timestamp        uint64
     |  +---- location         event-location
     |  +---- event-type       identityref
     |  +---- running-total    yang:counter64
     |  +---- event-total      yang:counter64
     +---n threshold-event {link-monitoring-local or
link-monitoring-remote}?
        +---- oui              vendor-oui
        +---- timestamp        uint64
        +---- location         event-location
```

```
          +---- event-type        identityref
          +---- running-total     yang:counter64
          +---- event-total       yang:counter64
          +---- threshold {link-monitoring-local or
link-monitoring-remote}?
             +---- threshold-event-type    threshold-event-enum
             +---- window                  uint64
             +---- threshold               uint64
             +---- value                   uint64
```

8.5.2 YANG module

In the following YANG module definition, should any discrepancy between the text of the description for individual YANG nodes and the corresponding definition in 8.2 through 8.5 of this clause occur, the definitions in 8.2 through 8.5 shall take precedence.

An ASCII text version of the YANG module can be found at the following URL:[12] https://github.com/YangModels/yang/tree/master/standard/ieee/published/802.3.

```
module ieee802-ethernet-link-oam {

  yang-version 1.1;

  namespace "urn:ieee:std:802.3:yang:ieee802-ethernet-link-oam";

  prefix "ieee802-link-oam";

  revision 2019-06-21 {
    description "Initial revision.";
  }

  import ietf-yang-types {
    prefix yang;
    reference "IETF RFC 6991";
  }

  import iana-if-type {
    prefix ianaift;
    reference "http://www.iana.org/assignments/yang-parameters/
      iana-if-type@2018-07-03.yang";
  }

  import ietf-interfaces {
    prefix if;
    reference "IETF RFC 8343";
  }

  organization
    "IEEE 802.3 Ethernet Working Group
     Web URL: http://www.ieee802.org/3/";
  contact
    "Web URL: http://www.ieee802.org/3/";

  description
    "This module contains a collection of YANG definitions
     for managing the Ethernet Link OAM feature defined by IEEE
     802.3. It provides functionality roughly equivalent to that of
     the DOT3-OAM-MIB defined in IETF RFC 4878.";

  reference
    "IEEE Std 802.3-2018, unless dated explicitly";
```

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

```
feature uni-directional-link-fault {
  description
    "This feature means the device supports Uni Directional Link
     Fault detection.";
  reference
    "IEEE Std 802.3, 57.1.2:a, 30.3.6.1.6 aOAMLocalConfiguration and
     30.3.6.1.7 aOAMRemoteConfiguration";
}
feature remote-loopback-initiate {
  description
    "This feature means the device supports being the initiator
     of remote loopback.";
  reference
    "IEEE Std 802.3, 57.1.2:b, 30.3.6.1.6 aOAMLocalConfiguration";
}
feature remote-loopback-respond {
  description
    "This feature means the device supports responding to remote
     loopback control OAMPDUs received from the peer";
  reference
    "IEEE Std 802.3, 57.1.2:b, 30.3.6.1.7 aOAMRemoteConfiguration";
}
feature link-monitoring-local {
  description
    "This feature means the device monitors the link at the local
     side and can generate Link Event OAMPDUs to the peer device.";
  reference
    "IEEE Std 802.3, 57.1.2:c:1, 30.3.6.1.6 aOAMLocalConfiguration,
     and 30.3.6.1.7 aOAMRemoteConfiguration";
}
feature link-monitoring-remote {
  description
    "This feature means the device can process Link Event OAMPDUs
     received from the peer device and report itself about this
     event on its own management interface.";
  reference
    "IEEE Std 802.3, 57.1.2:c:1, 30.3.6.1.6 aOAMLocalConfiguration,
     and 30.3.6.1.7 aOAMRemoteConfiguration";
}
feature remote-mib-retrieval-initiate {
  description
    "This feature means the device supports data retrieval from
     the peer device. I.e. the device can send Variable Requests
     OAMPDUs to the peer side and process the received Variable
     Response OAMPDUs.";
  reference
    "IEEE Std 802.3, 57.1.2:c:2, 30.3.6.1.6 aOAMLocalConfiguration,
     and 30.3.6.1.7 aOAMRemoteConfiguration";
}
feature remote-mib-retrieval-respond {
  description
    "This feature means the device allows the peer device to
     retrieve data from the managed device. I.e. the device can
```

```
      process received Variable Requests OAMPDUs and respond with
      Variable Response OAMPDUs.";
    reference
      "IEEE Std 802.3, 57.1.2:c:2, 30.3.6.1.6 aOAMLocalConfiguration,
      and 30.3.6.1.7 aOAMRemoteConfiguration";
  }
  typedef threshold-event-enum {
    type enumeration {
      enum symbol-period-event {
        value 1;
        description
          "Errored symbol period event.";
      }
      enum frame-period-event {
        value 2;
        description
          "Errored frame period event.";
      }
      enum frame-event {
        value 3;
        description
          "Errored frame event";
      }
      enum frame-seconds-event {
        value 4;
        description
          "Errored frame seconds event.";
      }
    }
    description
      "Enumeration of the valid threshold event types.";
    reference
      "IEEE Std 802.3, 57.5.3";
  }

  identity event-type {
    description
      "Base identity for all Link OAM event types.";
  }
  identity threshold-event-type {
    base event-type;
    description
      "Event type for a Link Monitoring threshold event.";
  }
  identity link-fault-event {
    base event-type;
     if-feature "uni-directional-link-fault";
    description
      "Event type for a uni-directional link fault event.";
    reference
      "IEEE Std 802.3, 57.2.10.1";
  }
  identity dying-gasp-event {
    base event-type;
```

```
  description
    "Event type for a dying gasp event.";
  reference
    "IEEE Std 802.3, 57.2.10.1";
}
identity critical-event {
  base event-type;
  description
    "Event type for a critical event.";
  reference
    "IEEE Std 802.3, 57.2.10.1";
}

typedef mode {
  type enumeration {
    enum passive {
      value 0;
      description
        "Ethernet Link OAM Passive mode.";
    }
    enum active {
      value 1;
      description
        "Ethernet Link OAM Active mode.";
    }
  }
  description
    "Enumeration of the valid modes in which Link OAM may run.";
  reference
    "IEEE Std 802.3, 57.2.9 and 30.3.6.1.3.";
}

typedef event-location {
  type enumeration {
    enum event-location-local {
      value 1;
      description
        "A local event.";
    }
    enum event-location-remote {
      value 2;
      description
        "A remote event.";
    }
  }
  description
    "The location of the event that caused a log entry.";
}

typedef loopback-status{
  type enumeration {
    enum none {
      value 1;
      description
```

```
        "Loopback is not being performed.";
      }
      enum initiating {
        value 2;
        description
          "Initiating master loopback.";
      }
      enum master-loopback {
        value 3;
        description
          "In master loopback mode.";
      }
      enum terminating {
        value 4;
        description
          "Terminating master loopback mode.";
      }
      enum local-loopback {
        value 5;
        description
          "In slave loopback mode.";
      }
      enum unknown {
        value 6;
        description
          "Parser and multiplexer combination unexpected.";
      }
    }
    description
      "The loopback mode of an OAM interface.";
    reference
      "IEEE Std 802.3, 57.2.11";
  }
  typedef operational-state {
    type enumeration {
      enum disabled {
        value 1;
        description
          "IEEE Std 802.3 OAM is disabled.";
      }
      enum link-fault {
        value 2;
        description
          "IEEE Std 802.3 OAM has encountered a link fault.";
      }
      enum passive-wait {
        value 3;
        description
          "Passive OAM entity waiting to see if peer is
           OAM capable.";
      }
      enum active-send-local {
        value 4;
        description
```

```
          "Active OAM entity trying to determine if peer
           is OAM capable.";
      }
      enum send-local-and-remote {
        value 5;
        description
          "OAM discovered peer but still to accept or
           reject peer configuration.";
      }
      enum send-local-and-remote-ok {
        value 6;
        description
          "OAM peering is allowed by local device.";
      }
      enum peering-locally-rejected {
        value 7;
        description
          "OAM peering rejected by local device.";
      }
      enum peering-remotely-rejected {
        value 8;
        description
          "OAM peering rejected by remote device.";
      }
      enum operational {
        value 9;
        description
          "IEEE Std 802.3 OAM is operational.";
      }
      enum operational-half-duplex {
        value 10;
        description
          "IEEE Std 802.3 OAM is operating in half-duplex mode.";
      }
    }
    description
      "Operational state of an interface.";
    reference
      "IETF RFC 4878, dot3OamOperStatus; IEEE Std 802.3, 30.3.6.1.4,
       30.3.6.1.10, and 30.3.6.1.11";
  }

  typedef vendor-oui {
    type string {
      length 6;
    }
    description
      "24-bit Organizationally Unique Identifier.";
    reference
      "IEEE Std 802-2014, Clause 9";
  }

  typedef admin-state {
    type enumeration {
```

```
      enum enabled {
        value 1;
        description
          "IEEE Std 802.3, Clause 57 OAM is in the
           enabled admin state.";
      }
      enum disabled {
        value 2;
        description
          "IEEE Std 802.3, Clause 57 OAM is in the
           disabled admin state.";
      }
    }
    description
      "Admin state of the OAM function on an interface.";
    reference
      "IEEE Std 802.3, 30.3.6.1.2 and 30.3.6.2";
  }

  grouping event-details {
    description
      "Nodes describing an event, used in the event log and in
       notifications.";
    reference
      "IETF RFC 4878, Dot3OamEventLogEntry";
    leaf oui {
      type vendor-oui;
      mandatory true;
      description
        "Organizationally Unique Identifier for the device that
         generated the event.";
    }
    leaf timestamp {
      type uint64;
      units "milliseconds";
      mandatory true;
      description
        "Timestamp in milliseconds since Unix epoch for when the
         event occurred.";
    }
    leaf location {
      type event-location;
      mandatory true;
      description
        "Where the event occurred (local or remote).";
    }
    leaf event-type {
      type identityref {
        base event-type;
      }
      mandatory true;
      description
        "Type of event that occurred.";
      reference
```

```
        "IEEE Std 802.3, 30.3.6.1.10 and 30.3.6.11";
    }
    leaf running-total {
      type yang:counter64;
      mandatory true;
      description
        "The running total number of errors seen since OAM was
         enabled on the interface. For threshold events, this is the
         total number of times that particular type of error (e.g.
         symbol error) has occurred, which may be greater than the
         number of threshold-crossing event notifications of that
         type generated during that time (which is conveyed by the
         event-total leaf).";
    }
    leaf event-total {
      type yang:counter64;
      mandatory true;
      description
        "Total number of times this event has occurred since OAM was
         enabled on the interface. For threshold events this is the
         number of events generated of this type (as opposed to the
         total number of errors of that type, which may be greater,
         and is conveyed by the running-total leaf.";
    }
  }

  grouping threshold-event-details {
    description
      "Nodes describing a threshold event, used in the event
       log and in notifications";
    reference
      "IETF RFC 4878, Dot3OamEventLogEntry";

    container threshold {
      when "../event-type = 'threshold-event-type'" {
        description
          "These nodes only apply to threshold event types";
      }
      if-feature "link-monitoring-local or
        link-monitoring-remote";
      description
        "Nodes specific to threshold (link monitoring) events";

      leaf threshold-event-type {
        type threshold-event-enum;
        mandatory true;
        description
          "The type of threshold event";
        reference
          "IEEE Std 802.3, 57.5.3";
      }
      leaf window {
        type uint64;
        mandatory true;
```

```
      description
        "Size of the window in which the event was generated.
         Units are dependent on the threshold event type.";
    }
    leaf threshold {
      type uint64;
      mandatory true;
      description
        "Size of the threshold that was breached during the
         window. Units are dependent on the threshold
         event type.";
    }
    leaf value {
      type uint64;
      mandatory true;
      description
        "Breaching value. Units are dependent on the
         threshold event type, and match that
         of the threshold.";
    }
  }
}

grouping statistics-common {
  description
    "Collection of Link OAM event/packet counters.";
  reference
    "IETF RFC 4878, Dot3OamStatsEntry";

  leaf out-information {
    type yang:counter64;
    mandatory true;
    description
      "Number of information OAMPDUs transmitted.";
    reference
      "IEEE Std 802.3, 30.3.6.1.20";
  }
  leaf in-information {
    type yang:counter64;
    mandatory true;
    description
      "Number of information OAMPDUs received.";
    reference
      "IEEE Std 802.3, 30.3.6.1.21";
  }
  leaf out-unique-event-notification {
    if-feature "link-monitoring-local";
    type yang:counter64;
    mandatory true;
    description
      "Number of unique event notification OAMPDUs transmitted.";
    reference
      "IEEE Std 802.3, 30.3.6.1.22";
  }
```

```
leaf in-unique-event-notification {
  if-feature "link-monitoring-remote";
  type yang:counter64;
  mandatory true;
  description
    "Number of unique event notification OAMPDUs received.";
  reference
    "IEEE Std 802.3, 30.3.6.1.24";
}
leaf out-duplicate-event-notification {
  if-feature "link-monitoring-local";
  type yang:counter64;
  mandatory true;
  description
    "Number of duplicate event notification OAMPDUs
     transmitted.";
  reference
    "IEEE Std 802.3, 30.3.6.1.23";
}
leaf in-duplicate-event-notification {
  if-feature "link-monitoring-remote";
  type yang:counter64;
  mandatory true;
  description
    "Number of duplicate event notification OAMPDUs
     received.";
  reference
    "IEEE Std 802.3, 30.3.6.1.25";
}
leaf out-loopback-control {
  if-feature "remote-loopback-initiate";
  type yang:counter64;
  mandatory true;
  description
    "Number of loopback control OAMPDUs transmitted.";
  reference
    "IEEE Std 802.3, 30.3.6.1.26";
}
leaf in-loopback-control {
  if-feature "remote-loopback-respond";
  type yang:counter64;
  mandatory true;
  description
    "Number of loopback control OAMPDUs received.";
  reference
    "IEEE Std 802.3, 30.3.6.1.27";
}
leaf out-variable-request {
  if-feature "remote-mib-retrieval-initiate";
  type yang:counter64;
  mandatory true;
  description
    "Number of variable request OAMPDUs transmitted.";
  reference
```

```
      "IEEE Std 802.3, 30.3.6.1.28";
    }
    leaf in-variable-request {
      if-feature "remote-mib-retrieval-respond";
      type yang:counter64;
      mandatory true;
      description
        "Number of variable request OAMPDUs received.";
      reference
        "IEEE Std 802.3, 30.3.6.1.29";
    }
    leaf out-variable-response {
      if-feature "remote-mib-retrieval-respond";
      type yang:counter64;
      mandatory true;
      description
        "Number of variable response OAMPDUs transmitted.";
      reference
        "IEEE Std 802.3, 30.3.6.1.30";
    }
    leaf in-variable-response {
      if-feature "remote-mib-retrieval-initiate";
      type yang:counter64;
      mandatory true;
      description
        "Number of variable response OAMPDUs received.";
      reference
        "IEEE Std 802.3, 30.3.6.1.31";
    }
    leaf out-org-specific {
      type yang:counter64;
      mandatory true;
      description
        "Number of organization specific OAMPDUs transmitted.";
      reference
        "IEEE Std 802.3, 30.3.6.1.32";
    }
    leaf in-org-specific {
      type yang:counter64;
      mandatory true;
      description
        "Number of organization specific OAMPDUs received.";
      reference
        "IEEE Std 802.3, 30.3.6.1.33";
    }
    leaf out-unsupported-codes {
      type yang:counter64;
      mandatory true;
      description
        "Number of OAMPDUs with unsupported codes transmitted.";
      reference
        "IEEE Std 802.3, 30.3.6.1.18";
    }
    leaf in-unsupported-codes {
```

```
    type yang:counter64;
    mandatory true;
    description
      "Number of OAMPDUs with unsupported codes received.";
    reference
      "IEEE Std 802.3, 30.3.6.1.19";
  }
  leaf frames-lost-due-to-oam {
    type yang:counter64;
    mandatory true;
    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.";
    reference
      "IEEE Std 802.3, 30.3.6.1.46";
  }
}

grouping discovery-remote {
  description
    "Nodes describing the discovery process remote end of a link.";
  leaf mode {
    type mode;
    description
      "Mode (passive/active).";
    reference
      "IEEE Std 802.3, 30.3.6.1.3";
  }
  container functions-supported {
    description
      "The Link OAM functions supported by this interface.";
    reference
      "IEEE Std 802.3, 30.3.6.1.7";
    leaf uni-directional-link-fault {
      type boolean;
      description
        "Unidirectional link fault support.";
    }
    leaf loopback {
      type boolean;
      description
        "Remote Loopback support.";
    }
    leaf link-monitoring {
      type boolean;
      description
        "Link monitoring support.";
    }
    leaf mib-retrieval {
      type boolean;
```

```
      description
        "MIB variable retrieval support.";
    }
  }
  leaf revision {
    type uint64;
    config false;
    description "Configuration revision.";
    reference
      "IEEE Std 802.3, 30.3.6.1.12 and 30.3.6.1.13";
  }
  leaf oammtu {
    type uint16;
    units octets;
    config false;
    description "The maximum OAMPDU size.";
    reference
      "IEEE Std 802.3, 30.3.6.1.8 and 30.3.6.1.9";
  }
}

grouping discovery-local {
  description
    "Nodes describing the local end discovery process of a link.";
  leaf mode {
    type mode;
    description
      "Mode (passive/active)
       The default value is implementation-dependent.";
    reference
      "IEEE Std 802.3, 30.3.6.1.3";
  }
  container functions-supported {
    description
      "The Link OAM functions supported by this interface.";
    reference
      "IEEE Std 802.3, 30.3.6.1.7";
    leaf uni-directional-link-fault {
      if-feature "uni-directional-link-fault";
      type boolean;
      description
        "Unidirectional link fault support.
         This affects the setting of the 'Unidirectional Support'
         bit in the OAM configuration field put in the
         Information OAMPDU.
         This bit indicates to the peer device that it can send
         OAM PDUs on links that are operating in unidirectional
         mode (traffic flowing in one direction only).";
    }
    leaf loopback {
      if-feature "remote-loopback-initiate";
      type boolean;
      default true;
      description
```

```
    "Remote Loopback support.";
}
container link-monitor {
  if-feature "link-monitoring-remote or
    link-monitoring-local";
  description
    "Configure link monitor parameters.";
  reference
    "IEEE Std 802.3, 57.1.2:c";

  leaf link-monitoring {
    type boolean;
    default true;
    description
      "Enable or disable monitoring.
       This affects the setting of the 'Link Events' bit in the
       OAM configuration field put in the Information OAMPDU.
       This bit indicates to the peer device that the OAM
       entity can send and receive Event Notification OAMPDUs.";
  }

  list event-type {
    if-feature "link-monitoring-local";
    key threshold-type;
    description
      "A list containing at most one entry for each of the
       threshold event types. If there is no entry for a
       particular event type, the default values are used for
       both window size and threshold.";
    leaf threshold-type {
      type threshold-event-enum;
      description
        "The type of threshold event for which this list entry
         is specifying the configuration.";
      reference
        "IEEE Std 802.3, 57.5.3";
    }

    leaf window {
      type uint64;
      description
        "The size of the window to use when monitoring for
         this threshold event. The units, default and upper
         and lower bounds depend on the threshold type as
         follows:

         Symbol Period:
           Units:   number of symbols
           Default: number of symbols in one second for the
                    underlying physical layer
           Min:     number of symbols in one second for the
                    underlying physical layer
           Max:     number of symbols in one minute for the
                    underlying physical layer
```

```
          Frame:
            Units:   deciseconds
            Default: 1 second
            Min:     1 second
            Max:     1 minute

          Frame Period:
            Units:   number of frames
            Default: number of minFrameSize frames in one
                     second for the underlying physical layer
            Min:     number of minFrameSize frames in one
                     second for the underlying physical layer
            Max:     number of minFrameSize frames in one
                     minute for the underlying physical layer

          Frame Seconds:
            Units:   deciseconds
            Default: 60 seconds
            Min:     10 seconds
            Max:     900 seconds";
        reference
          "IEEE Std 802.3, 30.3.6.1.34, 30.3.6.1.36, 30.3.6.1.38,
           and 30.3.6.1.40";
      }

      leaf threshold {
        type uint64 {
          range "1..max";
        }
        default 1;
        description
          "The threshold value to use when determining whether to
           generate an event given the number of errors that
           occurred in a given window. The units depend on the
           threshold type as follows:

           Symbol Period: number of errored symbols
           Frame:         number of errored frames
           Frame Period:  number of errored frames
           Frame Seconds: number of seconds containing at least
           1 frame error";
        reference
          "IEEE Std 802.3, 30.3.6.1.34, 30.3.6.1.36, 30.3.6.1.38,
           and 30.3.6.1.40";
      }
    }
  }

  leaf mib-retrieval {
    if-feature "remote-mib-retrieval-initiate or
      remote-mib-retrieval-respond";
    type boolean;
    description
```

```
        "MIB variable retrieval support.
         This affects the setting of the 'Variable Retrieval' bit
         in the OAM configuration field put in the Information
         OAMPDU. This bit indicates to the peer device that the OAM
         entity can send and receive Variable Request and Response
         OAMPDUs.";
    }
  }

  leaf revision {
    type uint64;
    config false;
    description
      "Configuration revision.";
    reference
      "IEEE Std 802.3, 30.3.6.1.12 and 30.3.6.1.13";
  }
  leaf oammtu {
    type uint16;
    units octets;
    config false;
    description
      "The maximum OAMPDU size.";
    reference
      "IEEE Std 802.3, 30.3.6.1.8 and 30.3.6.1.9";
  }
}

grouping discovery-info {
  description
    "Information relating to the discovery process.";

  container local {
    description
      "Properties of the local device.";

    leaf operational-status {
      type operational-state;
      config false;
      mandatory true;
      description
        "Operational status.";
      reference
        "IETF RFC 4878, dot3OamOperStatus; IEEE Std 802.3,
         30.3.6.1.4, 30.3.6.1.10, and 30.3.6.1.11";
    }
    leaf loopback-mode {
      if-feature "remote-loopback-initiate or
        remote-loopback-respond";
      type loopback-status;
      config false;
      mandatory true;
      description
        "The loopback mode the interface is in.";
```

```
      reference
        "IEEE Std 802.3, 30.3.6.1.14";
    }
    uses discovery-local;
  }

  container remote {
    config false;
    description
      "Properties of the remote (peer) device.";

    leaf mac-address {
      type yang:mac-address;
      description
        "Remote MAC address.";
      reference
        "IEEE Std 802.3, 30.3.6.1.5";
    }
    leaf vendor-oui {
      type vendor-oui;
      description
        "Remote vendor OUI.";
      reference
        "IEEE Std 802.3, 30.3.6.1.16";
    }
    leaf vendor-info {
      type uint64;
      description
        "Remote vendor info. The semantics of this value are
         proprietary and specific to the vendor.";
      reference
        "IEEE Std 802.3, 30.3.6.1.17";
    }
    leaf loopback-mode {
      type loopback-status;
      mandatory true;
      description
        "The loopback mode the interface is in.";
      reference
        "IEEE Std 802.3, 30.3.6.1.15";
    }

    uses discovery-remote;
  }
}

augment '/if:interfaces/if:interface' {
  when
    "derived-from-or-self(if:type, 'ianaift:ethernetCsmacd') or
     derived-from-or-self(if:type, 'ianaift:ptm') " {
  description
    "Augments the interface model with nodes
     specific to Ethernet Link OAM.";
  }
```

```
description
  "Augments Ethernet interface model with nodes
   specific to Ethernet Link OAM.";

container link-oam {
  presence
    "Implies Link OAM is configured on the interface.";
  description
    "Interface operational state for Ethernet Link OAM.";

  leaf admin {
    type admin-state;
    default disabled;
    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.
       It starts in the disabled 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. The default value is
       implementation-dependent.";
  }

  container discovery-info {
    description
      "Information relating to the discovery process.";
    uses discovery-info;
  }

  container event-log {
    config false;
    description
      "List of Ethernet Link OAM event log entries on the
       interface.";

    list event-log-entry {
      key "index";
      description
        "Ethernet Link OAM event log entry.";
      leaf index {
        type uint64;
        description
          "Index of this event in the event log.";
      }
      uses event-details;
      uses threshold-event-details;
    }
  }

  container statistics {
    config false;
    description
```

```
    "Statistics for an 802.3 OAM interface.";

  uses statistics-common;

  leaf local-error-symbol-period-log-entries {
    type yang:counter64;
    mandatory true;
    description
      "Number of local error symbol period log entries.";
  }
  leaf local-error-frame-log-entries {
    type yang:counter64;
    mandatory true;
    description
      "Number of local error frame log entries.";
  }
  leaf local-error-frame-period-log-entries {
    type yang:counter64;
    mandatory true;
    description
      "Number of local error frame period log entries.";
  }
  leaf local-error-frame-second-log-entries {
    type yang:counter64;
    mandatory true;
    description
      "Number of local error frame second log entries.";
  }
  leaf remote-error-symbol-period-log-entries {
    if-feature "link-monitoring-remote";
    type yang:counter64;
    mandatory true;
    description
      "Number of remote error symbol period log entries.";
  }
  leaf remote-error-frame-log-entries {
    if-feature "link-monitoring-remote";
    type yang:counter64;
    mandatory true;
    description
      "Number of remote error frame log entries.";
  }
  leaf remote-error-frame-period-log-entries {
    if-feature "link-monitoring-remote";
    type yang:counter64;
    mandatory true;
    description
      "Number of remote error frame period log entries.";
  }
  leaf remote-error-frame-second-log-entries {
    if-feature "link-monitoring-remote";
    type yang:counter64;
    mandatory true;
    description
```

```
      "Number of remote error frame second log entries.";
  }
}

action remote-loopback {
  if-feature "remote-loopback-initiate";
  description
    "Start/stop remote loopback on the specified interface.";
  reference
    "IEEE Std 802.3, 57.1.2:b";
  input {
    leaf enable {
      type boolean;
      mandatory true;
      description
        "Whether to enable or disable remote loopback.";
    }
  }
  output {
    leaf success {
      type boolean;
      mandatory true;
      description
        "True if the operation was successful,
         false otherwise.";
    }
    leaf error-message {
      type string;
      description
        "If the operation failed, optionally used to
         provide extra details.";
    }
  }
}

action reset-stats {
  description
    "Reset Ethernet Link OAM statistics on this interface.";
  output {
    leaf success {
      type boolean;
      mandatory true;
      description
        "True if the operation was successful,
         false otherwise.";
    }
    leaf error-message {
      type string;
      description
        "If the operation failed, optionally used to provide
         extra details.";
    }
  }
}
```

```
      notification non-threshold-event {
        description
          "This notification is sent when a local or remote
           non-threshold crossing event is detected.";
        uses event-details {
          refine event-type {
            must ". != 'threshold-event-type'" {
              description
                "This leaf is not set to
                 'threshold-event-type'.";
            }
          }
        }
      }

      notification threshold-event {
        if-feature "link-monitoring-local or
          link-monitoring-remote";
        description
          "This notification is sent when a local or remote
           threshold crossing event is detected.";
        uses event-details {
          refine event-type {
            must ". = 'threshold-event-type'" {
              description
                "This leaf is set to 'threshold-event-type'.";
            }
          }
        }
        uses threshold-event-details;
      }
    }
  }
}
```

Consensus

WE BUILD IT.

Connect with us on:

Facebook: https://www.facebook.com/ieeesa

Twitter: @ieeesa

LinkedIn: http://www.linkedin.com/groups/IEEESA-Official-IEEE-Standards-Association-1791118

IEEE-SA Standards Insight blog: http://standardsinsight.com

YouTube: IEEE-SA Channel

IEEE
standards.ieee.org
Phone: +1 732 981 0060 Fax: +1 732 562 1571