

IEEE P802.3.1b™/D0.1

Draft Standard for Ethernet Structure of Management Information version 2 (SMIPv2) Data Model Definitions

Sponsor
LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved XX XXXX 202X
IEEE-SA Standards Board

Prepared by the
LAN/MAN Standards Committee
of the
IEEE Computer Society

This draft is a revision of IEEE Std 802.3.1-2013. Draft D0.1 is prepared for the Task Force review. This draft expires 6 months after the date of publication or when the next version is published, whichever comes first.

Copyright © 2018 by The Institute of Electrical and Electronics Engineers, Inc.
Three Park Avenue
New York, New York 10016-5997, USA

1 All rights reserved.
2

3 This document is an unapproved draft of a proposed IEEE Standard. As such, this document is subject to
4 change. USE AT YOUR OWN RISK! IEEE copyright statements SHALL NOT BE REMOVED from draft
5 or approved IEEE standards, or modified in any way. Because this is an unapproved draft, this document
6 must not be utilized for any conformance/compliance purposes. Permission is hereby granted for officers
7 from each IEEE Standards Working Group or Committee to reproduce the draft document developed by that
8 Working Group for purposes of international standardization consideration. IEEE Standards Department
9 must be informed of the submission for consideration prior to any reproduction for international
10 standardization consideration (stds-ipr@ieee.org). Prior to adoption of this document, in whole or in part, by
11 another standards development organization, permission must first be obtained from the IEEE Standards
12 Department (stds-ipr@ieee.org). When requesting permission, IEEE Standards Department will require a
13 copy of the standard development organization's document highlighting the use of IEEE content. Other
14 entities seeking permission to reproduce this document, in whole or in part, must also obtain permission
15 from the IEEE Standards Department.
16
17
18
19

20 IEEE Standards Department
21 445 Hoes Lane
22 Piscataway, NJ 08854, USA
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 Permissions have been granted as follows^a:

2
3 ***Editor's Note (to be removed prior to publication):***

4
5 Confirm permissions are still applicable.
6

7
8 Portions of this standard reprinted with permission from Avaya, IETF RFC 2108 Definitions of Managed
9 Objects for IEEE 802.3 Repeater Devices using SMIPv2, © 1997.

10
11 Portions of this standard reprinted with permission from Microsemi Corporation, IETF RFC 3621 Definitions
12 Power Ethernet MIB, © 2003.

13
14 Portions of this standard reprinted with permission from Avaya, IETF RFC 3621 Definitions Power Ethernet
15 MIB, © 2003.

16
17 Portions of this standard reprinted with permission from the Hewlett-Packard Company, IETF RFC 3635
18 Definitions of Managed Objects for the Ethernet-like Interface Types, © 2003.

19
20 Material from IETF RFC 4836 Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units
21 (MAUs) reprinted with permission from the IETF Trust and the persons identified as the document authors,
22 © 2007.

23
24 Portions of this standard reprinted with permission from Edward Beili, IETF RFC 4836 Definitions of
25 Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs), © 2007.

26
27 Material from IETF RFC 4837 Managed Objects of Ethernet Passive Optical Networks (EPON) reprinted with
28 permission from the IETF Trust and the persons identified as the document authors, © 2007.

29
30 Portions of this standard reprinted with permission from PMC-Sierra, IETF RFC 4837 Managed Objects of
31 Ethernet Passive Optical Networks (EPON), © 2007.

32
33 Material from IETF RFC 4878 Definitions and Managed Objects for Operations, Administration, and
34 Maintenance (OAM) Functions on Ethernet-Like Interfaces reprinted with permission from the IETF Trust
35 and the persons identified as the document authors, © 2007.

36
37 Material from IETF RFC 5066 Ethernet in the First Mile Copper (EFMCu) Interfaces MIB reprinted with
38 permission from the IETF Trust and the persons identified as the document authors.

39
40 Portions of this standard reprinted with permission from Edward Beili, IETF RFC 5066 Ethernet in the First
41 Mile Copper (EFMCu) Interfaces MIB, © 2007.

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64 ^aEvery effort has been made to secure permission to reprint borrowed material contained in this document. If omissions have been made,
65 please bring them to the attention of the IEEE P802.3.1 working group.

1 **Abstract:** This standard defines Structure of Management Information version 2 (SMIv2) MIB
2 module specifications for IEEE Std 802.3 Ethernet and associated managed object branch and
3 leaf assignments used in the variable descriptors in IEEE Std 802.3 Variable Request operations,
4 administration, and maintenance protocol data unit (OAMPUD).
5

6 **Keywords:** Ethernet, IEEE 802.3.1™, Management Information Base (MIB), network management,
7 Simple Network Management Protocol (SNMP), Structure of Management Information Version 2
8 (SMIv2)
9

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53 The Institute of Electrical and Electronics Engineers, Inc.
54 3 Park Avenue, New York, NY 10016-5997, USA

55 Copyright © 2013 by the Institute of Electrical and Electronics Engineers, Inc.
56 All rights reserved. Published 2 August 2013. Printed in the United States of America.

57 IEEE and 802 are registered trademarks in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics
58 Engineers, Incorporated.

59 PDF: ISBN 978-0-7381-8432-6 STD98241
60 Print: ISBN 978-0-7381-8433-3 STDPD98241

61 *IEEE prohibits discrimination, harassment, and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.*
62

63 *No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission*
64 *of the publisher.*
65

Editor's Note (to be removed prior to publication):

This front matter is provided for comment only. Front matter is not part of a published standard and is therefore, not part of the draft standard. You are invited to review and comment on it as it will be included in the published standard after approval.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

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

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA 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 makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. 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: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

1 WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
2 OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE
3 UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND
4 REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.
5

6 7 **Translations**

8
9
10 The IEEE consensus development process involves the review of documents in English only. In the event
11 that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE
12 standard.
13

14 15 **Official statements**

16
17 A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board
18 Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its
19 committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures,
20 symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall
21 make it clear that the presenter's views should be considered the personal views of that individual rather
22 than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements
23 made by volunteers may not represent the formal position of their employer(s) or affiliation(s).
24
25

26 27 **Comments on standards**

28
29
30 Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of
31 membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations,**
32 **consulting information, or advice pertaining to IEEE Standards documents.**
33

34
35 Suggestions for changes in documents should be in the form of a proposed change of text, together with
36 appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is
37 important that any responses to comments and questions also receive the concurrence of a balance of interests.
38 For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of
39 Governors are not able to provide an instant response to comments, or questions except in those cases where
40 the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation
41 requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard
42 is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the
43 Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#).^b An IEEE Account is
44 needed to access the application.
45
46

47
48 Comments on standards should be submitted using the [Contact Us](#) form.^c
49

50 51 **Laws and regulations**

52
53 Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the
54 provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory
55 requirements. Implementers of the standard are responsible for observing or referring to the applicable
56 regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not
57 in compliance with applicable laws, and these documents may not be construed as doing so.
58
59
60

61
62
63 ^b Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

64 ^c Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.
65

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US 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, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, 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; <https://www.copyright.com/>. 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 10 years. When a document is more than 10 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 [IEEE Xplore](#) or [contact IEEE](#).^d For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).^e Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

^d Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

^e Available at: <https://standards.ieee.org/standard/index.html>.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).^f

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 <https://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.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

^f Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

Participants

The following individuals were officers and members of the IEEE 802.3 Working Group at the beginning of the IEEE P802.3.2a Working Group ballot.

David J. Law, *IEEE 802.3 Working Group Chair*

Adam Healey, *IEEE 802.3 Working Group Vice-Chair*

Jon Lewis, *IEEE 802.3 Working Group Secretary*

Steven B. Carlson, *IEEE 802.3 Working Group Executive Secretary*

Valerie Maguire, *IEEE 802.3 Working Group Treasurer*

Marek Hajduczenia, *IEEE P802.3.2a Task Force Chair and Editor*

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

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

1 When the IEEE-SA Standards Board approved this standard on XX Month 202X, it had the following
2 membership:
3

4 **FirstName SecondName, Chair**

5 **FirstName SecondName, Vice Chair**

6 **FirstName SecondName, Past Chair**

7 **Konstantinos Karachalios, Secretary**

8 [to be supplied at publication]
9

10 *Member Emeritus
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Introduction

This introduction is not part of IEEE Std 802.3.1b-202x, Standard for Ethernet Structure of Management Information version 2 (SMIv2) Data Model Definitions.

The initial version of this standard was based on the managed object definitions provided in IEEE Std 802.3TM-2008, which subsumed and superseded IEEE Std 802.3anTM-2006, IEEE Std 802.3apTM-2007, IEEE Std 802.3aqTM-2006, and IEEE Std 802.3asTM-2006. It also includes the Logical Link Discovery Protocol Ethernet extensions provided in Annex F of IEEE Std 802.1ABTM-2009.^g In addition, the initial version of this standard incorporated and updated the MIB module definitions formerly defined in IETF RFC 2108 [B20],^h IETF RFC 3621 [B27], IETF RFC 3635 [B29], IETF RFC 3637 [B30], IETF RFC 4836 [B35], IETF RFC 4837 [B36], IETF RFC 4878 [B37], and IETF RFC 5066 [B38].

The first revision of this standard updated the MIB module definitions to reflect the managed object definitions provided in IEEE Std 802.3-2012, which subsumed and superseded IEEE Std 802.3-2008, IEEE Std 802.3atTM, IEEE Std 802.3avTM, IEEE Std 802.3azTM, IEEE Std 802.3baTM, IEEE Std 802.3bcTM, IEEE Std 802.3bdTM, IEEE Std 802.3bfTM, and IEEE Std 802.3bgTM.

The second revision of this standard ... <<<TBD>>>

^gInformation on references can be found in Clause 2.

^hThe numbers in brackets correspond to those of the bibliography in Annex A.

Contents

1.	Overview.....	1
1.1	Scope.....	2
1.2	Purpose.....	2
1.3	Internet-Standard Management Framework	2
1.4	Security considerations	2
1.5	Conformance.....	3
2.	Normative references	5
3.	Definitions	7
4.	Abbreviations.....	9
5.	Ethernet logical link discovery protocol (LLDP) extension MIB module	11
5.1	Structure of the IEEE 802.3 LLDP extension MIB	11
5.2	Relationship to other MIBs.....	11
5.3	Security considerations for IEEE 802.3 LLDP extension MIB module.....	14
5.4	MIB module definition	15
6.	Ethernet operations, administration, and maintenance (OAM) MIB module.....	41
6.1	Introduction.....	41
6.2	Overview.....	41
6.2.1	Remote fault indication.....	41
6.2.2	Link monitoring	41
6.2.3	Remote loopback	42
6.2.4	Ethernet OAM protocol data units.....	42
6.3	Relation to other MIB modules	42
6.3.1	Relation to other EFM MIB modules	42
6.3.2	Mapping of IEEE 802.3 managed objects	42
6.4	MIB structure	45
6.5	Security considerations for Ethernet operations, administration, and maintenance (OAM) MIB module.....	45
6.6	MIB module definition	46
7.	Ethernet repeater device MIB module	83
7.1	Overview.....	83
7.1.1	Repeater management.....	83
7.1.2	Structure of the MIB	83
7.1.3	Relationship to MIB-II.....	83
7.2	Topology mapping	84
7.3	MIB module definition	84
8.	Ethernet data terminal equipment (DTE) power via medium dependent interface (MDI) MIB module.....	129
8.1	Introduction.....	129
8.2	Overview.....	129
8.3	MIB structure	129

1	8.4 Security considerations for Ethernet data terminal equipment (DTE) power via medium	
2	dependent interface (MDI) MIB module	129
3	8.5 MIB module definition	130
4		
5		
6	9. Ethernet passive optical networks (EPON) MIB module	143
7		
8	9.1 Overview	143
9	9.1.1 EPON architecture highlights	143
10	9.1.2 Management architecture	149
11	9.2 MIB structure	150
12	9.3 Relationship to other MIB modules	154
13	9.3.1 Relation to the Interfaces Group MIB and Ethernet-like interface MIB	154
14	9.3.2 Relation to the IEEE 802.3 MAU MIBs	160
15	9.3.3 Relation to the Ethernet OAM MIB	160
16	9.3.4 Relation to the bridge MIB	160
17	9.4 Mapping of IEEE 802.3 managed objects	160
18	9.5 Security considerations for Ethernet passive optical network (EPON) MIB module	163
19	9.5.1 dot3MpcpAdminState	163
20	9.5.2 dot3EponFecMode	163
21	9.5.3 dot3ExtPkgObjectReset	163
22	9.5.4 dot3ExtPkgObjectPowerDown	163
23	9.5.5 dot3ExtPkgObjectFecEnabled	163
24	9.5.6 dot3ExtPkgObjectRegisterAction	163
25	9.5.7 dot3ExtPkgObjectReportNumThreshold	163
26	9.5.8 dot3ExtPkgObjectReportThreshold	163
27	9.5.9 dot3ExtPkgOptIfLowerInputPowerThreshold	164
28	9.5.10 dot3ExtPkgOptIfUpperInputPowerThreshold	164
29	9.5.11 dot3ExtPkgOptIfLowerOutputPowerThreshold	164
30	9.5.12 dot3ExtPkgOptIfUpperOutputPowerThreshold	164
31	9.5.13 dot3ExtPkgOptIfTransmitEnable	164
32	9.6 MIB module definition	164
33		
34	10. Ethernet-like interface MIB module	209
35		
36	10.1 Introduction	209
37	10.2 Overview	209
38	10.2.1 Relation to MIB-2	209
39	10.2.2 Relation to the Interfaces Group MIB	209
40	10.2.3 Relation to the IEEE 802.3 MAU-MIB module	215
41	10.2.4 Mapping of IEEE 802.3 managed objects	216
42	10.3 Security considerations for Ethernet-like interface MIB module	218
43	10.4 MIB module definition	219
44		
45	11. Ethernet in the First Mile copper (EFMCu) interfaces MIB module	253
46		
47	11.1 Introduction	253
48	11.2 Relation to other MIB modules	253
49	11.2.1 Relation to Interfaces Group MIB module	253
50	11.2.2 Relation to SHDSL MIB module	259
51	11.2.3 Relation to VDSL MIB module	259
52	11.2.4 Relation to Ethernet-Like and MAU MIB modules	259
53	11.3 MIB structure	260
54	11.3.1 EFM copper MIB overview	260
55	11.3.2 PME profiles	260
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		

1	11.3.3 Mapping of IEEE 802.3 managed objects	261
2	11.4 Security considerations for Ethernet in the First Mile copper interfaces MIB module	262
3	11.5 MIB module definition	262
4		
5		
6	12. Ethernet wide area network (WAN) interface sublayer (WIS) MIB module	313
7		
8	12.1 Overview	313
9	12.1.1 Relationship to the SONET/SDH interface MIB	313
10	12.1.2 Relationship to the Ethernet-like interface MIB	313
11	12.1.3 Relationship to the IEEE 802.3 MAU MIB	314
12	12.1.4 Use of the ifTable	314
13	12.1.5 SONET/SDH terminology	315
14	12.1.6 Mapping of IEEE 802.3 managed objects	315
15	12.1.7 Mapping of SNMP objects to WIS station management registers	319
16	12.1.8 Structure of the MIB module	322
17	12.2 Security considerations for Ethernet wide area network (WAN) interface sublayer (WIS)	
18	MIB module	323
19	12.3 MIB module definition	324
20		
21		
22		
23		
24	Annex 12A (informative) Collection of performance data using WIS MDIO registers	337
25		
26	13. Ethernet medium attachment units (MAUs) MIB module	339
27		
28	13.1 Introduction	339
29	13.2 Overview	339
30	13.2.1 Relationship to IETF RFC 3636 and IETF RFC 4836	339
31	13.2.2 Relationship to other MIBs	339
32	13.2.3 Management of internal MAUs	340
33	13.2.4 Mapping of IEEE 802.3 managed objects	340
34	13.2.5 Addition of new MAU types	343
35	13.3 Security considerations for Ethernet medium attachment units (MAUs) MIB module	344
36	13.4 IANA considerations	344
37	13.5 MIB module definition	344
38		
39		
40		
41		
42	Annex A (informative) Bibliography	375
43		
44	Annex B (normative) Branch and leaf assignments for IEEE 802.3 and IEEE 802.3.1	
45	managed objects	379
46		
47		
48	B.1 Branch and leaf table	379
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		

Standard for Ethernet Structure of Management Information version 2 (SMIPv2) Data Model Definitions

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

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

1. Overview

Editor's Note (to be removed prior to publication):

Text of the Overview will need to be aligned with the new scope of the project and perhaps keep the original Overview in historic description of the project. A new scope aligned with PAR scope needs to be provided.

This document supersedes and makes obsolete Annex 30A and Annex 30B of IEEE Std 802.3™-2008, Annex F of IEEE Std 802.1AB™-2009,¹ IETF RFC 2108 [B20],² IETF RFC 3621 [B27], IETF RFC 3635 [B29], IETF RFC 3637 [B30], IETF RFC 4836 [B35], IETF RFC 4837 [B36], IETF RFC 4878 [B37], and IETF RFC 5066 [B38].

¹Information on references can be found in Clause 2.

²The numbers in brackets correspond to those of the bibliography in Annex A.

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

The term “Ethernet-like interfaces” was historically used because the interfaces defined by the IEEE 802.3 Working Group were not considered “Ethernet” per se, but “Ethernet-like,” because “Ethernet” was taken to mean “Ethernet version 2” according to the (DEC, Intel, Xerox) DIX “blue book.” Today and in the context of SNMP management and SMIPv2 MIB modules, “Ethernet,” “Ethernet-like,” and “IEEE 802.3” are synonymous and interchangeable in the marketplace. The term “Ethernet-like” is retained in this document because of its common usage in the SNMP-based network management community.

1.1 Scope

This standard contains the Management Information Base (MIB) module specifications for IEEE Std 802.3, also known as Ethernet. It includes the Structure of Management Information Version 2 (SMIPv2) MIB module specifications formerly produced and published by the Internet Engineering Task Force (IETF), and the managed object branch and leaf assignments provided in the Guidelines for the Definition of Managed Objects (GDMO) MIB modules formerly specified within IEEE Std 802.3, as well as extensions resulting from recent amendments to IEEE Std 802.3. The SMIPv2 MIB modules are intended for use with the Simple Network Management Protocol (SNMP), commonly used to manage Ethernet.

1.2 Purpose

The purpose of the standard is to publish the SMIPv2 MIB module specifications in a single document that is separate from IEEE Std 802.3, and that can be published in a machine-readable format. Future amendments and revisions to IEEE Std 802.3.1 will be performed to update the MIB specifications as required to track future amendments and revisions to IEEE Std 802.3.

1.3 Internet-Standard Management Framework

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

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the SNMP.

Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This standard specifies MIB modules that are compliant to the SMIPv2, which is described in IETF STD 58 (RFC 2578), IETF STD 58 (RFC 2579), and IETF STD 58 (RFC 2580).

1.4 Security considerations

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in a MIB module.

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

SNMPv3 should be deployed, rather than previous versions of SNMP, and cryptographic security should be enabled. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

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

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

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

1.5 Conformance

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

2. Normative references

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

ANSI T1.231-1997, Layer 1 In-Service Digital Transmission Performance Monitoring.³

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

ETSI TS1 101 270-1; (1999), Transmission and Multiplexing (TM); Access transmission systems on metallic access cables; Very high speed Digital Subscriber Line (VDSL); Part 1: Functional requirements.⁴

IEEE Std 802®, IEEE Standard for Local and Metropolitan Area Networks—Architecture and Overview.^{5, 6}

Editor's Note (to be removed prior to publication):

Reference to IEEE Std 802.1D was replaced with IEEE Std 802.1Q per Maintenance Request 1383 (see https://www.ieee802.org/3/maint/requests/maint_1383.pdf)

IEEE Std 802.1Q™, Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks.

IEEE Std 802.1AB™-2009, IEEE Standard for Local and Metropolitan Area Networks—Station and Media Access Control Discovery.

IEEE Std 802.3™, IEEE Standard for Ethernet.

IEEE Std 802.9a™-1995, IEEE Standards for Local and Metropolitan Area Networks—Supplement to Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers—Specification of ISLAN16-T.

IETF RFC 1213, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, McCloghrie, K., and M. Rose, M. Mareh, 1991.⁷

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

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

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

³ANSI publications are available from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (<http://www.ansi.org/>). ANSI publications are available from the American National Standards Institute (<http://www.ansi.org/>).

⁴ETSI standards are available from the European Telecommunications Standards Institute at 650, Route des Lucioles, 06921 Sophia-Antipolis Cedex, France (<http://www.etsi.org/>). ETSI publications are available from the European Telecommunications Standards Institute (<http://www.etsi.org/>).

⁵IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (<http://standards.ieee.org/>). IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁶The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

⁷IETF RFCs are available from the Internet Engineering Task Force website at <http://www.ietf.org/rfc.html>. IETF documents (i.e., RFCs) are available for download at <http://www.rfc-archive.org/>.

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

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

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

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

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

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

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

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

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

ITU-T Recommendation G.983.1, 1998—Optical line systems for local and access networks—Broadband optical access systems based on Passive Optical Networks (PON).⁸

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

ITU-T Recommendation G.993.1, 2004—Very high speed digital subscriber line transceivers.

⁸ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (<http://www.itu.int/>) ITU-T publications are available from the International Telecommunications Union (<http://www.itu.int/>).

3. Definitions

For the purposes of this document, the following terms and definitions apply. *The IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.⁹

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

group: Within the context of the repeater management ~~MIB~~ Management Information Base (MIB) module defined in Clause 7 of IEEE Std 802.3.1-2013: A recommended, but optional, entity defined in Clause 30 of IEEE Std 802.3, Clause 30, in order to support a modular numbering scheme. The classical example allows an implementor to represent field-replaceable units as groups of ports, with the port numbering matching the modular hardware implementation.

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

Loss of Codegroup Delineation: See 50.3.5.3 of IEEE Std 802.3-50.3.5.3.

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

managed repeater: A repeater as defined by IEEE Std 802.3 incorporating a management entity that complies with the MIB Management Information Base (MIB) module definition contained in Clause 7 of this document/IEEE Std 802.3.1-2013.

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

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

Path Coding violations: In IEEE Std 802.3, the path layer coding violations count is based on block errors and not on BIP-8 errors; i.e., it is incremented only once for each B3 byte that indicates incorrect parity, regardless of the number of bits in error. Note that Section 8.4.5.1 of ANSI T1.231-1997 allows either path BIP-8 errors or path block errors to be used for the path layer error count.

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

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

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

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

⁹The *IEEE Standards Dictionary Online* subscription is available at http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

STS-Path Remote Defect Indication: IEEE Std 802.3 mandates the use of ERDI-P (Enhanced Remote Defect Indication-Path) defined in ANSI T1.231-1997 to signal remote server defects (triggered by path AIS or path LOP) and remote payload defects (triggered by Payload Label Mismatch or Loss of Codegroup Delineation). IETF RFC 3592 defines the one-bit RDI-P (Remote Defect Indication-Path), which ~~signals~~ signals remote server defects (i.e., path AIS and path LOP) only. An implementation of the MIB module defined in ~~Clause 12~~ Clause 12 of IEEE Std 802.3.1-2013 sets the sonetPathSTSRDI bit in the appropriate instance of sonetPathCurrentStatus when it receives an ERDI-P server defect indication from the remote end. Both ERDI-P payload defects and ERDI-P server defects are reported in the object etherWisFarEndPathCurrentStatus.

system: An entity compliant with one or more ~~MIB~~ Management Information Base (MIB) modules of this standard.

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

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

4. Abbreviations

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

1	OAM	operations, administration, and maintenance
2	OAMPDU	operations, administration, and maintenance protocol data unit
3		
4	OID	object identifier
5	OLT	optical line terminal
6	OMP	optical multipoint
7	ONU	optical network unit
8		
9	OSI	Open Systems Interconnection
10		
11	P2MP	point-to-multipoint
12	P2PE	point-to-point emulation
13	PAF	PME aggregation function
14	PBO	power back-off
15	PCS	physical coding sublayer
16	PD	powered device
17	PDU	protocol data unit
18	PHY	Physical Layer entity
19	PLM	Payload Label Mismatch
20	PMA	physical medium attachment
21	PMD	physical medium dependent
22	PME	physical medium entity
23	PON	passive optical network
24	PSD	power spectral density
25	PSE	power sourcing equipment
26	RFC	Request for Comments
27	ROM	read-only-memory
28	RS	reconciliation sublayer
29	RTT	round-trip time
30	SDH	Synchronous Digital Hierarchy
31	SLA	service level agreement
32	SLD	start of LLID delimiter
33	SMIv2	structure of management information version 2
34	SNMP	simple network management protocol
35	SNR	signal-to-noise ratio
36	SONET	Synchronous Optical Network
37	TCPAM	trellis coded pulse amplitude modulation
38	TDM	time division multiplexing
39	TDMA	time division multiple access
40	TLV	type/length/value
41	TQ	time quanta
42	WAN	wide area network
43	WDM	wavelength division multiplexing
44	WIS	WAN interface sublayer
45		
46		
47		
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		

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

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

5.1 Structure of the IEEE 802.3 LLDP extension MIB

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

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

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

^aM = Mandatory.

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

5.2 Relationship to other MIBs

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

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

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

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

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

5.3 Security considerations for IEEE 802.3 LLDP extension MIB module

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write.¹⁰ Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

Setting the object, `lldpV2Xdot3PortConfigTLVsTxEnable`, to incorrect values can result in improper operation of LLDP.

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

a) Objects that are associated with the transmit mode are as follows:

- 1) `lldpV2Xdot3LocPortAutoNegSupported`
- 2) `lldpV2Xdot3LocPortAutoNegEnabled`
- 3) `lldpV2Xdot3LocPortAutoNegAdvertisedCap`
- 4) `lldpV2Xdot3LocPortOperMauType`
- 5) `lldpV2Xdot3LocPowerPortClass`
- 6) `lldpV2Xdot3LocPowerMDISupported`
- 7) `lldpV2Xdot3LocPowerMDIEnabled`
- 8) `lldpV2Xdot3LocPowerPairControlable`
- 9) `lldpV2Xdot3LocPowerPairs`
- 10) `lldpV2Xdot3LocPowerClass`
- 11) `lldpV2Xdot3LocMaxFrameSize`

b) Objects that are associated with the receive mode are as follows:

- 1) `lldpV2Xdot3RemPortAutoNegSupported`
- 2) `lldpV2Xdot3RemPortAutoNegEnabled`
- 3) `lldpV2Xdot3RemPortAutoNegAdvertisedCap`
- 4) `lldpV2Xdot3RemPortOperMauType`
- 5) `lldpV2Xdot3RemPowerPortClass`
- 6) `lldpV2Xdot3RemPowerMDISupported`
- 7) `lldpV2Xdot3RemPowerMDIEnabled`
- 8) `lldpV2Xdot3RemPowerPairControlable`
- 9) `lldpV2Xdot3RemPowerPairs`
- 10) `lldpV2Xdot3RemPowerClass`
- 11) `lldpV2Xdot3RemMaxFrameSize`

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

¹⁰In IETF MIB definitions, the MAX-ACCESS clause defines the type of access that is allowed for particular data elements in the MIB. An explanation of the MAX-ACCESS mapping is given in section 7.3 of IETF STD 58, RFC 2578.

5.4 MIB module definition

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

An ASCII text version of the MIB definition can be found at the following URL¹¹:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C5mib.txt

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

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

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

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```
1  IEEE8023-DOT3-LLDP-EXT-V2-MIB DEFINITIONS ::= BEGIN
2
3  IMPORTS
4      MODULE-IDENTITY,
5      OBJECT-TYPE,
6      Unsigned32,
7      Integer32,
8      org
9
10     FROM SNMPv2-SMI
11     TruthValue
12     FROM SNMPv2-TC
13     MODULE-COMPLIANCE,
14     OBJECT-GROUP
15     FROM SNMPv2-CONF
16     ifGeneralInformationGroup
17     FROM IF-MIB
18     lldpV2LocPortIfIndex,
19     lldpV2RemLocalIfIndex,
20     lldpV2RemTimeMark,
21     lldpV2RemLocalDestMACAddress,
22     lldpV2RemIndex,
23     lldpV2PortConfigEntry
24     FROM LLDP-V2-MIB
25
26     -- http://www.ieee802.org/1/files/public/MIBs/LLDP-V2-MIB-200906080000Z.txt
27     LldpV2PowerPortClass
28     FROM LLDP-V2-TC-MIB
29
30     -- http://www.ieee802.org/1/files/public/MIBs/LLDP-V2-TC-MIB-200906080000Z.txt
31 ;
32
33 ieee8023lldpV2Xdot3MIB MODULE-IDENTITY
34     LAST-UPDATED "201304110000Z" -- April 11, 2013
35     ORGANIZATION "IEEE 802.3 Working Group"
36     CONTACT-INFO
37         "WG-URL: http://www.ieee802.org/3/index.html
38         WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
39
40         Contact: Howard Frazier
41         Postal: 3151 Zanker Road
42                 San Jose, CA 95134
43                 USA
44         Tel: +1.408.922.8164
45         E-mail: hfrazier@broadcom.com"
46
47     DESCRIPTION
48         "The LLDP Management Information Base extension module for
49         IEEE 802.3 organizationally defined discovery information."
50
51
52     REVISION "201304110000Z" -- April 11, 2013
53     DESCRIPTION
54         "Revision, based on an earlier version in IEEE Std 802.3.1-2011."
55
56
57     REVISION "201102020000Z" -- February 2, 2011
58     DESCRIPTION
59         "This revision incorporated changes to the MIB module to
60         add objects to support management of Energy Efficient
61         Ethernet (EEE) and Enhanced DTE Power via the MDI (PoE+)."
```



```
1      ::= { org ieee(111)
2          standards-association-numbers-series-standards(2)
3          lan-man-stds(802)ieee802dot3(3) ieee802dot3dot1mibs(1) 5 }
4
5  -----
6  -----
7  --
8  -- Organizationally Defined Information Extension - IEEE 802.3
9  --
10 -----
11 -----
12 -----
13
14 lldpV2Xdot3Objects      OBJECT IDENTIFIER ::= { ieee8023lldpV2Xdot3MIB 1 }
15
16 -- LLDP IEEE 802.3 extension MIB groups
17 lldpV2Xdot3Config      OBJECT IDENTIFIER ::= { lldpV2Xdot3Objects 1 }
18 lldpV2Xdot3LocalData   OBJECT IDENTIFIER ::= { lldpV2Xdot3Objects 2 }
19 lldpV2Xdot3RemoteData  OBJECT IDENTIFIER ::= { lldpV2Xdot3Objects 3 }
20
21
22
23
24 -----
25 -- IEEE 802.3 - Configuration
26 -----
27
28
29
30 --
31 -- Version 2 of lldpV2Xdot3PortConfigTable
32 -- supports use of multiple destination MAC addresses
33 --
34
35
36 lldpV2Xdot3PortConfigTable OBJECT-TYPE
37     SYNTAX      SEQUENCE OF LldpV2Xdot3PortConfigEntry
38     MAX-ACCESS  not-accessible
39     STATUS      current
40     DESCRIPTION
41         "A table that controls selection of LLDP TLVs to be transmitted
42         on individual ports."
43     ::= { lldpV2Xdot3Config 1 }
44
45
46 lldpV2Xdot3PortConfigEntry OBJECT-TYPE
47     SYNTAX      LldpV2Xdot3PortConfigEntry
48     MAX-ACCESS  not-accessible
49     STATUS      current
50     DESCRIPTION
51         "LLDP configuration information that controls the
52         transmission of IEEE 802.3 organizationally defined TLVs on
53         LLDP transmission capable ports.
54
55         This configuration object augments the lldpV2PortConfigEntry of
56         the LLDP-MIB, therefore it is only present along with the port
57         configuration defined by the associated lldpV2PortConfigEntry
58         entry.
59
60         Each active lldpV2Xdot3PortConfigEntry is restored from non-volatile
61         storage (along with the corresponding lldpV2PortConfigEntry)
62         after a re-initialization of the management system."
63     AUGMENTS { lldpV2PortConfigEntry }
64
65
```

```

1      ::= { lldpV2Xdot3PortConfigTable 1 }
2
3      LldpV2Xdot3PortConfigEntry ::= SEQUENCE {
4          lldpV2Xdot3PortConfigTLVsTxEnable  BITS
5      }
6
7      lldpV2Xdot3PortConfigTLVsTxEnable  OBJECT-TYPE
8          SYNTAX      BITS {
9              macPhyConfigStatus(0),
10             powerViaMDI(1),
11             unused(2), --avoids re-use of the old link agg bit number
12             maxFrameSize(3)
13         }
14
15      MAX-ACCESS      read-write
16      STATUS          current
17      DESCRIPTION
18          "The lldpV2Xdot3PortConfigTLVsTxEnable, defined as a bitmap,
19          includes the IEEE 802.3 organizationally defined set of LLDP
20          TLVs whose transmission is allowed by the local LLDP agent by
21          the network management. Each bit in the bitmap corresponds
22          to an IEEE 802.3 subtype associated with a specific IEEE
23          802.3 optional TLV.
24
25          The bit 'macPhyConfigStatus(0)' indicates that the LLDP agent
26          should transmit 'MAC/PHY configuration/status TLV'.
27
28          The bit 'powerViaMDI(1)' indicates that the LLDP agent should
29          transmit 'Power via MDI TLV'.
30
31          The bit 'unused(2)' is no longer used; this was used for
32          the 'Link Aggregation TLV' in the previous version.
33
34          The bit 'maxFrameSize(3)' indicates that the LLDP agent should
35          transmit 'Maximum-frame-size TLV'.
36
37          The default value for lldpV2Xdot3PortConfigTLVsTxEnable object
38          is an empty set, which means no enumerated values are set.
39
40          The value of this object is restored from non-volatile
41          storage after a re-initialization of the management system."
42
43      REFERENCE
44          "IEEE Std 802.3 30.12.1.1.1"
45
46      DEFVAL { { } }
47
48      ::= { lldpV2Xdot3PortConfigEntry 1 }
49
50
51
52
53      -----
54      -- IEEE 802.3 - Local Device Information
55      -----
56
57      ---
58      --- lldpV2Xdot3LocPortTable: Ethernet Port AutoNeg/Speed/Duplex
59          Information Table
60      --- V2 modified to be indexed by ifIndex.
61      ---
62
63      lldpV2Xdot3LocPortTable OBJECT-TYPE
64          SYNTAX      SEQUENCE OF LldpV2Xdot3LocPortEntry
65

```

```
1      MAX-ACCESS    not-accessible
2      STATUS        current
3      DESCRIPTION
4          "This table contains one row per port of Ethernet port
5          information (as a part of the LLDP 802.3 organizational
6          extension) on the local system known to this agent."
7      ::= { lldpV2Xdot3LocalData 1 }
8
9
10     lldpV2Xdot3LocPortEntry OBJECT-TYPE
11         SYNTAX      LldpV2Xdot3LocPortEntry
12         MAX-ACCESS  not-accessible
13         STATUS      current
14         DESCRIPTION
15             "Information about a particular port component."
16         INDEX       { lldpV2LocPortIfIndex }
17         ::= { lldpV2Xdot3LocPortTable 1 }
18
19
20     LldpV2Xdot3LocPortEntry ::= SEQUENCE {
21         lldpV2Xdot3LocPortAutoNegSupported    TruthValue,
22         lldpV2Xdot3LocPortAutoNegEnabled      TruthValue,
23         lldpV2Xdot3LocPortAutoNegAdvertisedCap OCTET STRING,
24         lldpV2Xdot3LocPortOperMauType         Unsigned32
25     }
26
27     lldpV2Xdot3LocPortAutoNegSupported OBJECT-TYPE
28         SYNTAX      TruthValue
29         MAX-ACCESS  read-only
30         STATUS      current
31         DESCRIPTION
32             "The truth value used to indicate whether the given port
33             (associated with the local system) supports Auto-negotiation."
34         REFERENCE
35             "IEEE Std 802.3 30.12.2.1.1"
36         ::= { lldpV2Xdot3LocPortEntry 1 }
37
38
39     lldpV2Xdot3LocPortAutoNegEnabled OBJECT-TYPE
40         SYNTAX      TruthValue
41         MAX-ACCESS  read-only
42         STATUS      current
43         DESCRIPTION
44             "The truth value used to indicate whether port
45             Auto-negotiation is enabled on the given port associated
46             with the local system."
47         REFERENCE
48             "IEEE Std 802.3 30.12.2.1.2"
49         ::= { lldpV2Xdot3LocPortEntry 2 }
50
51
52     lldpV2Xdot3LocPortAutoNegAdvertisedCap OBJECT-TYPE
53         SYNTAX      OCTET STRING(SIZE(2))
54         MAX-ACCESS  read-only
55         STATUS      current
56         DESCRIPTION
57             "This object contains the value (bitmap) of the
58             ifMauAutoNegCapAdvertisedBits object (defined in IETF RFC
59             3636) which is associated with the given port on the
60             local system."
61         REFERENCE
62             "IEEE Std 802.3 30.12.2.1.3"
63         ::= { lldpV2Xdot3LocPortEntry 3 }
64
65
```

```
1
2  lldpV2Xdot3LocPortOperMauType OBJECT-TYPE
3      SYNTAX      Unsigned32(0..2147483647)
4      MAX-ACCESS  read-only
5      STATUS      current
6      DESCRIPTION
7          "An integer value that indicates the operational MAU type
8          of the given port on the local system.
9
10         This object contains the integer value derived from the
11         list position of the corresponding dot3MauType as listed
12         in Clause 13 and is equal to the last number in the
13         respective dot3MauType OID.
14
15         For example, if the ifMauType object is dot3MauType1000BaseTHD
16         which corresponds to {dot3MauType 29}, the numerical value of
17         this field is 29. For MAU types not listed in Clause 13,
18         the value of this field shall be set to zero."
19
20  REFERENCE
21      "IEEE Std 802.3 30.12.2.1.4"
22  ::= { lldpV2Xdot3LocPortEntry 4 }
23
24
25
26
27  ---
28  ---
29  --- lldpV2Xdot3LocPowerTable: Power Ethernet Information Table
30  --- V2 modified to be indexed by ifIndex.
31  ---
32  ---
33  ---
34
35  lldpV2Xdot3LocPowerTable OBJECT-TYPE
36      SYNTAX      SEQUENCE OF LldpV2Xdot3LocPowerEntry
37      MAX-ACCESS  not-accessible
38      STATUS      current
39      DESCRIPTION
40          "This table contains one row per port of power Ethernet
41          information (as a part of the LLDP IEEE 802.3 organizational
42          extension) on the local system known to this agent."
43  ::= { lldpV2Xdot3LocalData 2 }
44
45
46  lldpV2Xdot3LocPowerEntry OBJECT-TYPE
47      SYNTAX      LldpV2Xdot3LocPowerEntry
48      MAX-ACCESS  not-accessible
49      STATUS      current
50      DESCRIPTION
51          "Information about a particular port component."
52  INDEX  { lldpV2LocPortIfIndex }
53  ::= { lldpV2Xdot3LocPowerTable 1 }
54
55
56  LldpV2Xdot3LocPowerEntry ::= SEQUENCE {
57      lldpV2Xdot3LocPowerPortClass      LldpV2PowerPortClass,
58      lldpV2Xdot3LocPowerMDISupported   TruthValue,
59      lldpV2Xdot3LocPowerMDIEnabled     TruthValue,
60      lldpV2Xdot3LocPowerPairControlable TruthValue,
61      lldpV2Xdot3LocPowerPairs          Unsigned32,
62      lldpV2Xdot3LocPowerClass          Unsigned32,
63      lldpV2Xdot3LocPowerType           INTEGER,
64      lldpV2Xdot3LocPowerSource         INTEGER,
```

```
1         lldpV2Xdot3LocPowerPriority          INTEGER,
2         lldpV2Xdot3LocPDRequestedPowerValue  Integer32,
3         lldpV2Xdot3LocPSEAllocatedPowerValue Integer32,
4         lldpV2Xdot3LocResponseTime           Integer32,
5         lldpV2Xdot3LocReady                  TruthValue,
6         lldpV2Xdot3LocReducedOperationPowerValue Integer32
7     }
8
9
10
11 lldpV2Xdot3LocPowerPortClass OBJECT-TYPE
12     SYNTAX      LldpV2PowerPortClass
13     MAX-ACCESS  read-only
14     STATUS      current
15     DESCRIPTION
16         "The value that identifies the port Class of the given port
17         associated with the local system."
18     REFERENCE
19         "IEEE Std 802.3 30.12.2.1.5"
20     ::= { lldpV2Xdot3LocPowerEntry 1 }
21
22
23 lldpV2Xdot3LocPowerMDISupported OBJECT-TYPE
24     SYNTAX      TruthValue
25     MAX-ACCESS  read-only
26     STATUS      current
27     DESCRIPTION
28         "The truth value used to indicate whether the MDI power is
29         supported on the given port associated with the local system."
30     REFERENCE
31         "IEEE Std 802.3 30.12.2.1.6"
32     ::= { lldpV2Xdot3LocPowerEntry 2 }
33
34
35 lldpV2Xdot3LocPowerMDIEnabled OBJECT-TYPE
36     SYNTAX      TruthValue
37     MAX-ACCESS  read-only
38     STATUS      current
39     DESCRIPTION
40         "The truth value used to identify whether MDI power is
41         enabled on the given port associated with the local system."
42     REFERENCE
43         "IEEE Std 802.3 30.12.2.1.7"
44     ::= { lldpV2Xdot3LocPowerEntry 3 }
45
46
47 lldpV2Xdot3LocPowerPairControlable OBJECT-TYPE
48     SYNTAX      TruthValue
49     MAX-ACCESS  read-only
50     STATUS      current
51     DESCRIPTION
52         "The truth value is derived from the value of
53         pethPsePortPowerPairsControlAbility object (defined in
54         Clause 8) and is used to indicate whether the pair selection
55         can be controlled on the given port associated with the
56         local system."
57     REFERENCE
58         "IEEE Std 802.3 30.12.2.1.8"
59     ::= { lldpV2Xdot3LocPowerEntry 4 }
60
61
62 lldpV2Xdot3LocPowerPairs OBJECT-TYPE
63     SYNTAX      Unsigned32(1|2)
64     MAX-ACCESS  read-only
65
```

```
1      STATUS      current
2      DESCRIPTION
3          "This object contains the value of the pethPsePortPowerPairs
4          object (defined in Clause 8) which is associated with
5          the given port on the local system."
6      REFERENCE
7          "IEEE Std 802.3 30.12.2.1.9"
8      ::= { lldpV2Xdot3LocPowerEntry 5 }
9
10
11 lldpV2Xdot3LocPowerClass OBJECT-TYPE
12     SYNTAX      Unsigned32(1|2|3|4|5)
13     MAX-ACCESS  read-only
14     STATUS      current
15     DESCRIPTION
16         "This object contains the value of the
17         pethPsePortPowerClassifications object (defined in
18         Clause 8) which is associated with the given port on the
19         local system."
20     REFERENCE
21         "IEEE Std 802.3 30.12.2.1.10"
22     ::= { lldpV2Xdot3LocPowerEntry 6 }
23
24
25 lldpV2Xdot3LocPowerType OBJECT-TYPE
26     SYNTAX      INTEGER {
27         psetype1(0),
28         psetype2(1),
29         pdtype(2),
30         pdtype2(3)
31     }
32     MAX-ACCESS  read-only
33     STATUS      current
34     DESCRIPTION
35         "A GET returns an integer indicating whether the local
36         system is a PSE or a PD and whether it is Type 1 or Type 2."
37     REFERENCE
38         "IEEE Std 802.3 30.12.2.1.14"
39     ::= { lldpV2Xdot3LocPowerEntry 7 }
40
41
42
43 lldpV2Xdot3LocPowerSource OBJECT-TYPE
44     SYNTAX      INTEGER {
45         pseprimary(0),
46         psebackup(1),
47         pseunknown(2),
48         pdpseandlocal(3),
49         pdpseonly(4),
50         pdunknown(5)
51     }
52     MAX-ACCESS  read-only
53     STATUS      current
54     DESCRIPTION
55         "A GET returns an integer indicating the power sources of the
56         local system. A PSE indicates whether it is being powered by
57         a primary power source; a backup power source; or unknown. A PD
58         indicates whether it is being powered by a PSE and locally;
59         by a PSE only; or unknown."
60     REFERENCE
61         "IEEE Std 802.3 30.12.2.1.15"
62     ::= { lldpV2Xdot3LocPowerEntry 8 }
63
64
65
```

```
1  lldpV2Xdot3LocPowerPriority  OBJECT-TYPE
2      SYNTAX          INTEGER {
3          low(0),
4          high(1),
5          critical(2),
6          unknown(3)
7      }
8
9      MAX-ACCESS      read-write
10     STATUS          current
11     DESCRIPTION
12         "A GET returns the priority of a PD system. For a PSE, this
13         is the priority that the PSE assigns to the PD. For a PD, this
14         is the priority that the PD requests from the PSE. A SET
15         operation changes the priority of the PD system to the indicated
16         value."
17
18     REFERENCE
19         "IEEE Std 802.3 30.12.2.1.16"
20     ::= { lldpV2Xdot3LocPowerEntry 9 }
21
22  lldpV2Xdot3LocPDRequestedPowerValue  OBJECT-TYPE
23      SYNTAX          Integer32
24      MAX-ACCESS      read-only
25      STATUS          current
26      DESCRIPTION
27          "A GET returns the PD requested power value.
28          For a PD, it is the power value that the PD has currently
29          requested from the remote system. PD requested power value
30          is the maximum input average power the PD ever draws under
31          this power allocation if accepted. For a PSE, it is the power
32          value that the PSE mirrors back to the remote system. This is
33          the PD requested power value that was used by the PSE to compute
34          the power it has currently allocated to the remote system.
35          The PD requested power value is encoded according to
36          IEEE Std 802.3 Equation (79-1), where X is the decimal value of
37          aLldpXdot3LocPDRequestedPowerValue."
38
39      REFERENCE
40          "IEEE Std 802.3 30.12.2.1.17"
41      ::= { lldpV2Xdot3LocPowerEntry 10 }
42
43
44  lldpV2Xdot3LocPSEAllocatedPowerValue  OBJECT-TYPE
45      SYNTAX          Integer32
46      MAX-ACCESS      read-only
47      STATUS          current
48      DESCRIPTION
49          "A GET returns the PSE allocated power value.
50          For a PSE, it is the power value that the PSE has currently
51          allocated to the remote system. The PSE allocated power value
52          is the maximum input average power that the PSE wants the PD
53          to ever draw under this allocation if it is accepted. For a PD,
54          it is the power value that the PD mirrors back to the remote
55          system. This is the PSE allocated power value that was used by
56          the PD to compute the power that it has currently requested from
57          the remote system. The PSE allocated power value is encoded
58          according to IEEE Std 802.3 Equation (79-2), where X is the
59          decimal value of aLldpXdot3LocPSEAllocatedPowerValue."
60
61      REFERENCE
62          "IEEE Std 802.3 30.12.2.1.18"
63      ::= { lldpV2Xdot3LocPowerEntry 11 }
64
65
```

```
1  lldpV2Xdot3LocResponseTime  OBJECT-TYPE
2      SYNTAX      Integer32
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "A GET returns the response time in seconds of the local system.
7          For a PD, it is the maximum time required to update the value of
8          lldpV2Xdot3LocPDRequestedPowerValue when the remote system
9          requests the PD to change its max power draw. For a PSE, it is
10         the maximum time required to update the value of
11         lldpV2Xdot3LocPDRequestedPowerValue when the remote system
12         requests of the PSE a new power value."
13
14     REFERENCE
15         "IEEE Std 802.3 30.12.2.1.19"
16     ::= { lldpV2Xdot3LocPowerEntry 12 }
17
18 lldpV2Xdot3LocReady  OBJECT-TYPE
19     SYNTAX      TruthValue
20     MAX-ACCESS  read-only
21     STATUS      current
22     DESCRIPTION
23         "The truth value used to identify whether the local Data Link Layer
24         classification engine has completed initialization and is ready to
25         receive and transmit LLDPDUs."
26
27     REFERENCE
28         "IEEE Std 802.3 30.12.2.1.20"
29     ::= { lldpV2Xdot3LocPowerEntry 13 }
30
31 lldpV2Xdot3LocReducedOperationPowerValue  OBJECT-TYPE
32     SYNTAX      Integer32
33     MAX-ACCESS  read-only
34     STATUS      current
35     DESCRIPTION
36         "A GET returns the reduced operation power value. For a PD, it
37         is a power value that is lower than the currently requested
38         power value. This reduced operation power value represents a
39         power state in which the PD could continue to operate, but with
40         less functionality than at the current PD requested power value.
41         The PSE could optionally use this information in the event that
42         the PSE subsequently requests a lower PD power value than the
43         PD requested power value. For a PSE, it is a power value that the
44         PSE could ask the PD to move to if the PSE wants the PD to move
45         to a lower power state. The definition and encoding of PD
46         requested power value is the same as described in
47         lldpV2Xdot3LocPDRequestedPowerValue. The default value for this
48         field is the hexadecimal value FFFF"
49
50     REFERENCE
51         "IEEE Std 802.3 30.12.2.1.21"
52     ::= { lldpV2Xdot3LocPowerEntry 14 }
53
54 ---
55 ---
56 --- lldpV2Xdot3LocMaxFrameSizeTable: Maximum Frame Size information
57 --- V2 modified to be indexed by ifIndex.
58 ---
59
60 lldpV2Xdot3LocMaxFrameSizeTable  OBJECT-TYPE
61     SYNTAX      SEQUENCE OF LldpV2Xdot3LocMaxFrameSizeEntry
62     MAX-ACCESS  not-accessible
```



```
1      STATUS      current
2      DESCRIPTION
3          "This table contains one row per port of maximum frame
4          size information (as a part of the LLDP IEEE 802.3 organizational
5          extension) on the local system known to this agent."
6      ::= { lldpV2Xdot3LocalData 3 }
7
8
9      lldpV2Xdot3LocMaxFrameSizeEntry OBJECT-TYPE
10     SYNTAX      LldpV2Xdot3LocMaxFrameSizeEntry
11     MAX-ACCESS  not-accessible
12     STATUS      current
13     DESCRIPTION
14         "Maximum Frame Size information about a particular port
15         component."
16     INDEX      { lldpV2LocPortIfIndex }
17     ::= { lldpV2Xdot3LocMaxFrameSizeTable 1 }
18
19
20     LldpV2Xdot3LocMaxFrameSizeEntry ::= SEQUENCE {
21         lldpV2Xdot3LocMaxFrameSize      Unsigned32
22     }
23
24     lldpV2Xdot3LocMaxFrameSize OBJECT-TYPE
25     SYNTAX      Unsigned32(0..65535)
26     MAX-ACCESS  read-only
27     STATUS      current
28     DESCRIPTION
29         "An integer value indicating the maximum supported frame
30         size in octets on the given port of the local system."
31
32     REFERENCE
33         "IEEE Std 802.3 30.12.2.1.13"
34     ::= { lldpV2Xdot3LocMaxFrameSizeEntry 1 }
35
36     ---
37     ---
38     --- lldpV2Xdot3LocEEETable: Energy Efficient Ethernet Information Table
39     --- V2 modified to be indexed by ifIndex.
40     ---
41     ---
42
43
44     lldpV2Xdot3LocEEETable OBJECT-TYPE
45     SYNTAX      SEQUENCE OF LldpV2Xdot3LocEEEEEntry
46     MAX-ACCESS  not-accessible
47     STATUS      current
48     DESCRIPTION
49         "This table contains one row per port of Energy Efficient Ethernet
50         information (as a part of the LLDP IEEE 802.3 organizational
51         extension) on the local system known to this agent."
52     ::= { lldpV2Xdot3LocalData 4 }
53
54
55     lldpV2Xdot3LocEEEEEntry OBJECT-TYPE
56     SYNTAX      LldpV2Xdot3LocEEEEEntry
57     MAX-ACCESS  not-accessible
58     STATUS      current
59     DESCRIPTION
60         "Information about a particular port component."
61     INDEX      { lldpV2LocPortIfIndex }
62     ::= { lldpV2Xdot3LocEEETable 1 }
63
64
65     LldpV2Xdot3LocEEEEEntry ::= SEQUENCE {
```

```

1         lldpV2Xdot3LocTxTwSys                Integer32,
2         lldpV2Xdot3LocTxTwSysEcho            Integer32,
3         lldpV2Xdot3LocRxTwSys                Integer32,
4         lldpV2Xdot3LocRxTwSysEcho            Integer32,
5         lldpV2Xdot3LocFbTwSys                Integer32,
6         lldpV2Xdot3TxDllReady                TruthValue,
7         lldpV2Xdot3RxDllReady                TruthValue,
8         lldpV2Xdot3LocDllEnabled              TruthValue
9     }
10
11
12     lldpV2Xdot3LocTxTwSys      OBJECT-TYPE
13         SYNTAX      Integer32
14         MAX-ACCESS   read-only
15         STATUS      current
16         DESCRIPTION
17             "A GET returns the value of Tw_sys_tx that the local system
18             can support in the transmit direction.
19             This object maps to the variable LocTxSystemValue as defined
20             in IEEE Std 802.3 78.4.2.3."
21         REFERENCE
22             "IEEE Std 802.3 30.12.2.1.22"
23         ::= {lldpV2Xdot3LocEEEEEntry 1 }
24
25
26     lldpV2Xdot3LocTxTwSysEcho    OBJECT-TYPE
27         SYNTAX      Integer32
28         MAX-ACCESS   read-only
29         STATUS      current
30         DESCRIPTION
31             "A GET returns the value of Tw_sys_tx that the remote system is
32             advertising that it can support in the transmit direction and is
33             echoed by the local system under the control of the EEE DLL receiver
34             state diagram. This object maps to the variable
35             LocTxSystemValueEcho as defined in IEEE Std 802.3 78.4.2.3"
36         REFERENCE
37             "IEEE Std 802.3 30.12.2.1.23"
38         ::= {lldpV2Xdot3LocEEEEEntry 2 }
39
40
41
42     lldpV2Xdot3LocRxTwSys      OBJECT-TYPE
43         SYNTAX      Integer32
44         MAX-ACCESS   read-only
45         STATUS      current
46         DESCRIPTION
47             "A GET returns the value of Tw_sys_tx that
48             the local system is requesting in the receive direction.
49             This object maps to the variable LocRxSystemValue as
50             defined in IEEE Std 802.3 78.4.2.3."
51         REFERENCE
52             "IEEE Std 802.3 30.12.2.1.24"
53         ::= {lldpV2Xdot3LocEEEEEntry 3 }
54
55
56     lldpV2Xdot3LocRxTwSysEcho    OBJECT-TYPE
57         SYNTAX      Integer32
58         MAX-ACCESS   read-only
59         STATUS      current
60         DESCRIPTION
61             "A GET returns the value of Tw_sys_tx that
62             the remote system is advertising that it is requesting in the
63             receive direction and is echoed by the local system under the
64             control of the EEE DLL transmitter state diagram. This object
65

```

```
1         maps to the variable LocRxSystemValueEcho as defined in
2         IEEE Std 802.3 78.4.2.3."
3     REFERENCE
4         "IEEE Std 802.3 30.12.2.1.25"
5         ::= {lldpV2Xdot3LocEEEEEntry 4 }
6
7     lldpV2Xdot3LocFbTwSys          OBJECT-TYPE
8         SYNTAX          Integer32
9         MAX-ACCESS      read-only
10        STATUS          current
11        DESCRIPTION
12            "A GET returns the value of the fallback Tw_sys_tx
13            that the local system is advertising to the remote system.
14            This object maps to the variable LocFbSystemValue as defined
15            in IEEE Std 802.3 78.4.2.3."
16        REFERENCE
17            "IEEE Std 802.3 30.12.2.1.26"
18            ::= {lldpV2Xdot3LocEEEEEntry 5 }
19
20    lldpV2Xdot3TxDllReady          OBJECT-TYPE
21        SYNTAX          TruthValue
22        MAX-ACCESS      read-only
23        STATUS          current
24        DESCRIPTION
25            "The truth value used to identify whether the local Data Link Layer
26            EEE layer management function has completed initialization and
27            is ready to receive and transmit LLDPDUs."
28        REFERENCE
29            "IEEE Std 802.3 30.12.2.1.27"
30            ::= {lldpV2Xdot3LocEEEEEntry 6 }
31
32    lldpV2Xdot3RxDllReady          OBJECT-TYPE
33        SYNTAX          TruthValue
34        MAX-ACCESS      read-only
35        STATUS          current
36        DESCRIPTION
37            "The truth value used to identify whether the local Data Link Layer
38            EEE layer management function has completed initialization and
39            is ready to receive and transmit LLDPDUs."
40        REFERENCE
41            "IEEE Std 802.3 30.12.2.1.28"
42            ::= {lldpV2Xdot3LocEEEEEntry 7 }
43
44    lldpV2Xdot3LocDllEnabled       OBJECT-TYPE
45        SYNTAX          TruthValue
46        MAX-ACCESS      read-only
47        STATUS          current
48        DESCRIPTION
49            "The truth value used to identify whether the local system has
50            completed auto-negotiation with a link partner that has
51            indicated at least one EEE capability."
52        REFERENCE
53            "IEEE Std 802.3 30.12.2.1.29"
54            ::= {lldpV2Xdot3LocEEEEEntry 8 }
55
56    -----
57    -- IEEE 802.3 - Remote Devices Information
```

```
1 -----
2
3 ---
4 ---
5 --- lldpV2Xdot3RemPortTable: Ethernet Information Table
6 --- V2 modified to be indexed by ifIndex and destination MAC address.
7 ---
8 ---
9 ---
10
11 lldpV2Xdot3RemPortTable OBJECT-TYPE
12     SYNTAX      SEQUENCE OF LldpV2Xdot3RemPortEntry
13     MAX-ACCESS  not-accessible
14     STATUS      current
15     DESCRIPTION
16         "This table contains Ethernet port information (as a part
17         of the LLDP IEEE 802.3 organizational extension) of the remote
18         system."
19     ::= { lldpV2Xdot3RemoteData 1 }
20
21
22 lldpV2Xdot3RemPortEntry OBJECT-TYPE
23     SYNTAX      LldpV2Xdot3RemPortEntry
24     MAX-ACCESS  not-accessible
25     STATUS      current
26     DESCRIPTION
27         "Information about a particular physical network connection."
28     INDEX      { lldpV2RemTimeMark,
29                 lldpV2RemLocalIfIndex,
30                 lldpV2RemLocalDestMACAddress,
31                 lldpV2RemIndex }
32     ::= { lldpV2Xdot3RemPortTable 1 }
33
34
35 lldpV2Xdot3RemPortEntry ::= SEQUENCE {
36     lldpV2Xdot3RemPortAutoNegSupported      TruthValue,
37     lldpV2Xdot3RemPortAutoNegEnabled        TruthValue,
38     lldpV2Xdot3RemPortAutoNegAdvertisedCap  OCTET STRING,
39     lldpV2Xdot3RemPortOperMauType           Unsigned32
40 }
41
42
43
44 lldpV2Xdot3RemPortAutoNegSupported OBJECT-TYPE
45     SYNTAX      TruthValue
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49         "The truth value used to indicate whether the given port
50         (associated with remote system) supports Auto-negotiation."
51     REFERENCE
52         "IEEE Std 802.3 30.12.3.1.1"
53     ::= { lldpV2Xdot3RemPortEntry 1 }
54
55
56 lldpV2Xdot3RemPortAutoNegEnabled OBJECT-TYPE
57     SYNTAX      TruthValue
58     MAX-ACCESS  read-only
59     STATUS      current
60     DESCRIPTION
61         "The truth value used to indicate whether port
62         Auto-negotiation is enabled on the given port associated
63         with the remote system."
64     REFERENCE
65
```

```
1      "IEEE Std 802.3 30.12.3.1.2"
2      ::= { lldpV2Xdot3RemPortEntry 2 }
3
4  lldpV2Xdot3RemPortAutoNegAdvertisedCap OBJECT-TYPE
5      SYNTAX      OCTET STRING(SIZE(2))
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "This object contains the value (bitmap) of the
10             ifMauAutoNegCapAdvertisedBits object (defined in IETF RFC
11             3636) which is associated with the given port on the
12             remote system."
13
14      REFERENCE
15          "IEEE Std 802.3 30.12.3.1.3"
16      ::= { lldpV2Xdot3RemPortEntry 3 }
17
18  lldpV2Xdot3RemPortOperMauType OBJECT-TYPE
19      SYNTAX      Unsigned32(0..2147483647)
20      MAX-ACCESS  read-only
21      STATUS      current
22      DESCRIPTION
23          "An integer value that indicates the operational MAU type
24             of the sending device.
25
26             This object contains the integer value derived from the
27             list position of the corresponding dot3MauType as listed in
28             in Clause 13 and is equal to the last number in
29             the respective dot3MauType OID.
30
31             For example, if the ifMauType object is dot3MauType1000BaseTHD
32             which corresponds to {dot3MauType 29}, the numerical value of
33             this field is 29. For MAU types not listed in Clause 13,
34             the value of this field shall be set to zero."
35
36      REFERENCE
37          "IEEE Std 802.3 30.12.3.1.4"
38      ::= { lldpV2Xdot3RemPortEntry 4 }
39
40
41
42
43  ---
44  ---
45  --- lldpV2Xdot3RemPowerTable: Power Ethernet Information Table
46  --- V2 modified to be indexed by ifIndex and destination MAC address.
47  ---
48  ---
49
50  lldpV2Xdot3RemPowerTable OBJECT-TYPE
51      SYNTAX      SEQUENCE OF LldpV2Xdot3RemPowerEntry
52      MAX-ACCESS  not-accessible
53      STATUS      current
54      DESCRIPTION
55          "This table contains Ethernet power information (as a part
56             of the LLDP IEEE 802.3 organizational extension) of the remote
57             system."
58      ::= { lldpV2Xdot3RemoteData 2 }
59
60
61  lldpV2Xdot3RemPowerEntry OBJECT-TYPE
62      SYNTAX      LldpV2Xdot3RemPowerEntry
63      MAX-ACCESS  not-accessible
64      STATUS      current
65
```

```
1      DESCRIPTION
2          "Information about a particular physical network connection."
3      INDEX      { lldpV2RemTimeMark,
4                  lldpV2RemLocalIfIndex,
5                  lldpV2RemLocalDestMACAddress,
6                  lldpV2RemIndex }
7      ::= { lldpV2Xdot3RemPowerTable 1 }
8
9
10     LldpV2Xdot3RemPowerEntry ::= SEQUENCE {
11         lldpV2Xdot3RemPowerPortClass      LldpV2PowerPortClass,
12         lldpV2Xdot3RemPowerMDISupported    TruthValue,
13         lldpV2Xdot3RemPowerMDIEnabled      TruthValue,
14         lldpV2Xdot3RemPowerPairControlable TruthValue,
15         lldpV2Xdot3RemPowerPairs           Unsigned32,
16         lldpV2Xdot3RemPowerClass           Unsigned32,
17         lldpV2Xdot3RemPowerType            INTEGER,
18         lldpV2Xdot3RemPowerSource          INTEGER,
19         lldpV2Xdot3RemPowerPriority         INTEGER,
20         lldpV2Xdot3RemPDRRequestedPowerValue Integer32,
21         lldpV2Xdot3RemPSEAllocatedPowerValue Integer32
22     }
23
24
25
26     lldpV2Xdot3RemPowerPortClass OBJECT-TYPE
27         SYNTAX      LldpV2PowerPortClass
28         MAX-ACCESS  read-only
29         STATUS      current
30         DESCRIPTION
31             "The value that identifies the port Class of the given port
32              associated with the remote system."
33         REFERENCE
34             "IEEE Std 802.3 30.12.3.1.5"
35         ::= { lldpV2Xdot3RemPowerEntry 1 }
36
37
38     lldpV2Xdot3RemPowerMDISupported OBJECT-TYPE
39         SYNTAX      TruthValue
40         MAX-ACCESS  read-only
41         STATUS      current
42         DESCRIPTION
43             "The truth value used to indicate whether the MDI power
44              is supported on the given port associated with the remote
45              system."
46         REFERENCE
47             "IEEE Std 802.3 30.12.3.1.6"
48         ::= { lldpV2Xdot3RemPowerEntry 2 }
49
50
51
52     lldpV2Xdot3RemPowerMDIEnabled OBJECT-TYPE
53         SYNTAX      TruthValue
54         MAX-ACCESS  read-only
55         STATUS      current
56         DESCRIPTION
57             "The truth value used to identify whether MDI power is
58              enabled on the given port associated with the remote system."
59         REFERENCE
60             "IEEE Std 802.3 30.12.3.1.7"
61         ::= { lldpV2Xdot3RemPowerEntry 3 }
62
63
64     lldpV2Xdot3RemPowerPairControlable OBJECT-TYPE
65         SYNTAX      TruthValue
```

```
1      MAX-ACCESS    read-only
2      STATUS        current
3      DESCRIPTION
4          "The truth value is derived from the value of
5          pethPsePortPowerPairsControlAbility object (defined in
6          Clause 8) and is used to indicate whether the pair selection
7          can be controlled on the given port associated with the
8          remote system."
9
10     REFERENCE
11         "IEEE Std 802.3 30.12.3.1.8"
12     ::= { lldpV2Xdot3RemPowerEntry 4 }
13
14     lldpV2Xdot3RemPowerPairs OBJECT-TYPE
15         SYNTAX      Unsigned32(1|2)
16         MAX-ACCESS  read-only
17         STATUS      current
18         DESCRIPTION
19             "This object contains the value of the pethPsePortPowerPairs
20             object (defined in Clause 8) which is associated with
21             the given port on the remote system."
22
23     REFERENCE
24         "IEEE Std 802.3 30.12.3.1.9"
25     ::= { lldpV2Xdot3RemPowerEntry 5 }
26
27     lldpV2Xdot3RemPowerClass OBJECT-TYPE
28         SYNTAX      Unsigned32(1|2|3|4|5)
29         MAX-ACCESS  read-only
30         STATUS      current
31         DESCRIPTION
32             "This object contains the value of the
33             pethPsePortPowerClassifications object (defined in
34             Clause 8) which is associated with the given port on the
35             remote system."
36
37     REFERENCE
38         "IEEE Std 802.3 30.12.3.1.10"
39     ::= { lldpV2Xdot3RemPowerEntry 6 }
40
41
42
43     lldpV2Xdot3RemPowerType OBJECT-TYPE
44         SYNTAX      INTEGER {
45             psetype1(0),
46             psetype2(1),
47             pdtype(2),
48             pdtype2(3)
49         }
50         MAX-ACCESS  read-only
51         STATUS      current
52         DESCRIPTION
53             "A GET returns an integer indicating whether the remote
54             system is a PSE or a PD and whether it is Type 1 or Type 2."
55
56     REFERENCE
57         "IEEE Std 802.3 30.12.3.1.14"
58     ::= { lldpV2Xdot3RemPowerEntry 7 }
59
60     lldpV2Xdot3RemPowerSource OBJECT-TYPE
61         SYNTAX      INTEGER {
62             pseprimary(0),
63             psebackup(1),
64             pseunknown(2),
65
```

```
1          pdpseandlocal(3),
2          pdlocalonly(4),
3          pdpseonly(5),
4          pdunknown(6)
5      }
6  MAX-ACCESS  read-only
7  STATUS      current
8  DESCRIPTION
9
10     "A GET returns an integer indicating the power sources of the
11     remote system. When the remote system is a PSE, it indicates
12     whether it is being powered by a primary power source; a backup
13     power source; or unknown. When the remote system is a PD, it
14     indicates whether it is being powered by a PSE and locally;
15     locally only; by a PSE only; or unknown."
16  REFERENCE
17     "IEEE Std 802.3 30.12.3.1.15"
18  ::= { lldpV2Xdot3RemPowerEntry 8 }
19
20
21  lldpV2Xdot3RemPowerPriority  OBJECT-TYPE
22  SYNTAX      INTEGER {
23              low(0),
24              high(1),
25              critical(2),
26              unknown(3)
27          }
28  MAX-ACCESS  read-write
29  STATUS      current
30  DESCRIPTION
31
32     "A GET returns the priority of a PD system. For a PSE, this
33     is the priority that the remote system requests. For a PD, this
34     is the priority that the remote system has assigned."
35  REFERENCE
36     "IEEE Std 802.3 30.12.3.1.16"
37  ::= { lldpV2Xdot3RemPowerEntry 9 }
38
39
40  lldpV2Xdot3RemPDRequestedPowerValue  OBJECT-TYPE
41  SYNTAX      Integer32
42  MAX-ACCESS  read-only
43  STATUS      current
44  DESCRIPTION
45
46     "A GET returns the PD requested power value that was used
47     by the remote system to compute the power value that is has
48     currently allocated to the PD. For a PSE, it is the PD requested
49     power value received from the remote system. The definition and
50     encoding of PD requested power value is the same as described in
51     lldpV2Xdot3LocPDRequestedPowerValue."
52  REFERENCE
53     "IEEE Std 802.3 30.12.3.1.17"
54  ::= { lldpV2Xdot3RemPowerEntry 10 }
55
56
57  lldpV2Xdot3RemPSEAllocatedPowerValue  OBJECT-TYPE
58  SYNTAX      Integer32
59  MAX-ACCESS  read-only
60  STATUS      current
61  DESCRIPTION
62
63     "A GET returns the PSE allocated power value
64     received from the remote system. For a PSE, it is the PSE allocated
65     power value that was used by the remote system to compute the power
66     value that it has currently requested from the PSE. For a PD, it
```



```
1         is the PSE allocated power value received from the remote system.
2         The definition and encoding of PSE allocated power value is
3         the same as described in lldpV2Xdot3LocPSEAllocatedPowerValue."
4     REFERENCE
5         "IEEE Std 802.3 30.12.3.1.18"
6     ::= { lldpV2Xdot3RemPowerEntry 11 }
7
8     ---
9     ---
10    --- lldpV2Xdot3RemMaxFrameSizeTable: Maximum Frame Size information
11    --- V2 modified to be indexed by ifIndex and destination MAC address.
12    ---
13    ---
14
15    lldpV2Xdot3RemMaxFrameSizeTable OBJECT-TYPE
16        SYNTAX      SEQUENCE OF LldpV2Xdot3RemMaxFrameSizeEntry
17        MAX-ACCESS   not-accessible
18        STATUS       current
19        DESCRIPTION
20            "This table contains one row per port/destination
21             address pair of maximum frame
22             size information (as a part of the LLDP IEEE 802.3
23             organizational extension) of the remote system."
24        ::= { lldpV2Xdot3RemoteData 3 }
25
26
27    lldpV2Xdot3RemMaxFrameSizeEntry OBJECT-TYPE
28        SYNTAX      LldpV2Xdot3RemMaxFrameSizeEntry
29        MAX-ACCESS   not-accessible
30        STATUS       current
31        DESCRIPTION
32            "Maximum Frame Size information about a particular port
33             component."
34        INDEX       { lldpV2RemTimeMark,
35                     lldpV2RemLocalIfIndex,
36                     lldpV2RemLocalDestMACAddress,
37                     lldpV2RemIndex }
38        ::= { lldpV2Xdot3RemMaxFrameSizeTable 1 }
39
40
41    LldpV2Xdot3RemMaxFrameSizeEntry ::= SEQUENCE {
42        lldpV2Xdot3RemMaxFrameSize Unsigned32
43    }
44
45
46    lldpV2Xdot3RemMaxFrameSize OBJECT-TYPE
47        SYNTAX      Unsigned32(0..65535)
48        MAX-ACCESS   read-only
49        STATUS       current
50        DESCRIPTION
51            "An integer value indicating the maximum supported frame
52             size in octets on the port component associated with the
53             remote system."
54        REFERENCE
55            "IEEE Std 802.3 30.12.3.1.13"
56        ::= { lldpV2Xdot3RemMaxFrameSizeEntry 1 }
57
58
59    ---
60    ---
61    --- lldpV2Xdot3RemEEETable: Energy Efficient Ethernet Information Table
62    --- V2 modified to be indexed by ifIndex.
63    ---
64    ---
65    ---
```

```
1
2  lldpV2Xdot3RemEEETable OBJECT-TYPE
3      SYNTAX      SEQUENCE OF LldpV2Xdot3RemEEEEEntry
4      MAX-ACCESS  not-accessible
5      STATUS      current
6      DESCRIPTION
7          "This table contains one row per port of Energy Efficient Ethernet
8          information (as a part of the LLDP IEEE 802.3 organizational
9          extension) on the local system known to this agent."
10     ::= { lldpV2Xdot3RemoteData 4 }
11
12
13  lldpV2Xdot3RemEEEEEntry OBJECT-TYPE
14      SYNTAX      LldpV2Xdot3RemEEEEEntry
15      MAX-ACCESS  not-accessible
16      STATUS      current
17      DESCRIPTION
18          "Information about a particular port component."
19      INDEX       { lldpV2RemLocalIfIndex }
20      ::= { lldpV2Xdot3RemEEETable 1 }
21
22
23  LldpV2Xdot3RemEEEEEntry ::= SEQUENCE {
24      lldpV2Xdot3RemTxTwSys          Integer32,
25      lldpV2Xdot3RemTxTwSysEcho      Integer32,
26      lldpV2Xdot3RemRxTwSys          Integer32,
27      lldpV2Xdot3RemRxTwSysEcho      Integer32,
28      lldpV2Xdot3RemFbTwSys          Integer32
29  }
30
31
32
33  lldpV2Xdot3RemTxTwSys      OBJECT-TYPE
34      SYNTAX      Integer32
35      MAX-ACCESS  read-only
36      STATUS      current
37      DESCRIPTION
38          "A GET returns the value of Tw_sys_tx that the remote system
39          can support in the transmit direction.
40          This object maps to the variable RemTxSystemValue as defined
41          in IEEE Std 802.3 78.4.2.3."
42      REFERENCE
43          "IEEE Std 802.3 30.12.3.1.19"
44      ::= { lldpV2Xdot3RemEEEEEntry 1 }
45
46
47  lldpV2Xdot3RemTxTwSysEcho  OBJECT-TYPE
48      SYNTAX      Integer32
49      MAX-ACCESS  read-only
50      STATUS      current
51      DESCRIPTION
52          "A GET returns the value of Tw_sys_tx that the local system is
53          advertising that it can support in the transmit direction as
54          echoed by the remote system under the control of the EEE DLL receiver
55          state diagram. This object maps to the variable
56          RemTxSystemValueEcho as defined in IEEE Std 802.3 78.4.2.3"
57      REFERENCE
58          "IEEE Std 802.3 30.12.3.1.20"
59      ::= { lldpV2Xdot3RemEEEEEntry 2 }
60
61
62  lldpV2Xdot3RemRxTwSys      OBJECT-TYPE
63      SYNTAX      Integer32
64      MAX-ACCESS  read-only
65
```

```
1      STATUS      current
2      DESCRIPTION
3          "A GET returns the value of Tw_sys_tx that
4          the remote system is requesting in the receive direction.
5          This object maps to the variable RemRxSystemValue as
6          defined in IEEE Std 802.3 78.4.2.3."
7      REFERENCE
8          "IEEE Std 802.3 30.12.3.1.21"
9      ::= {lldpV2Xdot3RemEEEEEntry 3 }
10
11
12      lldpV2Xdot3RemRxTwSysEcho      OBJECT-TYPE
13          SYNTAX      Integer32
14          MAX-ACCESS  read-only
15          STATUS      current
16          DESCRIPTION
17              "A GET returns the value of Tw_sys_tx that
18              the local system is advertising that it is requesting in the
19              receive direction and is echoed by the remote system under the
20              control of the EEE DLL transmitter state diagram. This object
21              maps to the variable RemRxSystemValueEcho as defined in
22              IEEE Std 802.3 78.4.2.3."
23          REFERENCE
24              "IEEE Std 802.3 30.12.3.1.22"
25          ::= {lldpV2Xdot3RemEEEEEntry 4 }
26
27
28      lldpV2Xdot3RemFbTwSys          OBJECT-TYPE
29          SYNTAX      Integer32
30          MAX-ACCESS  read-only
31          STATUS      current
32          DESCRIPTION
33              "A GET returns the value of the fallback Tw_sys_tx
34              that the remote system is advertising.
35              This object maps to the variable RemFbSystemValue as defined
36              in IEEE Std 802.3 78.4.2.3."
37          REFERENCE
38              "IEEE Std 802.3 30.12.3.1.23"
39          ::= {lldpV2Xdot3RemEEEEEntry 5 }
40
41
42
43
44
45      -----
46      -- Conformance statements
47      -----
48      lldpV2Xdot3Conformance OBJECT IDENTIFIER ::= { ieee8023lldpV2Xdot3MIB 2 }
49      lldpV2Xdot3Compliances OBJECT IDENTIFIER ::= { lldpV2Xdot3Conformance 1 }
50      lldpV2Xdot3Groups      OBJECT IDENTIFIER ::= { lldpV2Xdot3Conformance 2 }
51
52
53      -- Compliance statements
54
55
56      lldpV2Xdot3TxRxCompliance MODULE-COMPLIANCE
57          STATUS      current
58          DESCRIPTION
59              "A compliance statement for SNMP entities that implement
60              the LLDP IEEE 802.3 organizational extension MIB.
61
62              This group is mandatory for all agents that implement the
63              LLDP IEEE 802.3 organizational extension in TX and/or RX mode.
64
65
```

```
1          This version defines compliance requirements for
2          V2 of the LLDP MIB."
3      MODULE -- this module
4          MANDATORY-GROUPS { lldpV2Xdot3ConfigGroup,
5                             ifGeneralInformationGroup
6          }
7      ::= { lldpV2Xdot3Compliances 1 }
8
9
10     lldpV2Xdot3TxCompliance MODULE-COMPLIANCE
11         STATUS current
12         DESCRIPTION
13             "The compliance statement for SNMP entities that implement
14             the LLDP IEEE 802.3 organizational extension MIB.
15
16             This group is mandatory for agents that implement the
17             LLDP IEEE 802.3 organizational extension in the TX mode.
18
19             This version defines compliance requirements for
20             V2 of the LLDP MIB."
21     MODULE -- this module
22         MANDATORY-GROUPS { lldpV2Xdot3LocSysGroup }
23     ::= { lldpV2Xdot3Compliances 2 }
24
25
26     lldpV2Xdot3RxCompliance MODULE-COMPLIANCE
27         STATUS current
28         DESCRIPTION
29             "The compliance statement for SNMP entities that implement
30             the LLDP IEEE 802.3 organizational extension MIB.
31
32             This group is mandatory for agents that implement the
33             LLDP IEEE 802.3 organizational extension in the RX mode.
34
35             This version defines compliance requirements for
36             V2 of the LLDP MIB."
37     MODULE -- this module
38         MANDATORY-GROUPS { lldpV2Xdot3RemSysGroup }
39     ::= { lldpV2Xdot3Compliances 3 }
40
41
42
43
44     -- MIB groupings
45
46
47     lldpV2Xdot3ConfigGroup    OBJECT-GROUP
48         OBJECTS {
49             lldpV2Xdot3PortConfigTLVsTxEnable
50         }
51     STATUS current
52     DESCRIPTION
53         "The collection of objects that are used to configure the
54         LLDP IEEE 802.3 organizational extension implementation behavior."
55     ::= { lldpV2Xdot3Groups 1 }
56
57
58     lldpV2Xdot3LocSysGroup    OBJECT-GROUP
59         OBJECTS {
60             lldpV2Xdot3LocPortAutoNegSupported,
61             lldpV2Xdot3LocPortAutoNegEnabled,
62             lldpV2Xdot3LocPortAutoNegAdvertisedCap,
63             lldpV2Xdot3LocPortOperMauType,
64             lldpV2Xdot3LocPowerPortClass,
```

```
1      lldpV2Xdot3LocPowerMDISupported,
2      lldpV2Xdot3LocPowerMDIEnabled,
3      lldpV2Xdot3LocPowerPairControlable,
4      lldpV2Xdot3LocPowerPairs,
5      lldpV2Xdot3LocPowerClass,
6      lldpV2Xdot3LocMaxFrameSize,
7      lldpV2Xdot3LocPowerType,
8      lldpV2Xdot3LocPowerSource,
9      lldpV2Xdot3LocPowerPriority,
10     lldpV2Xdot3LocPDRequestedPowerValue,
11     lldpV2Xdot3LocPSEAllocatedPowerValue,
12     lldpV2Xdot3LocResponseTime,
13     lldpV2Xdot3LocReady,
14     lldpV2Xdot3LocReducedOperationPowerValue,
15     lldpV2Xdot3LocTxTwSys,
16     lldpV2Xdot3LocTxTwSysEcho,
17     lldpV2Xdot3LocRxTwSys,
18     lldpV2Xdot3LocRxTwSysEcho,
19     lldpV2Xdot3LocFbTwSys,
20     lldpV2Xdot3TxDllReady,
21     lldpV2Xdot3RxDllReady,
22     lldpV2Xdot3LocDllEnabled
23 }
24
25 STATUS current
26 DESCRIPTION
27     "The collection of objects that are used to represent LLDP
28     IEEE 802.3 organizational extension Local Device Information."
29 ::= { lldpV2Xdot3Groups 2 }
30
31
32
33 lldpV2Xdot3RemSysGroup OBJECT-GROUP
34     OBJECTS {
35         lldpV2Xdot3RemPortAutoNegSupported,
36         lldpV2Xdot3RemPortAutoNegEnabled,
37         lldpV2Xdot3RemPortAutoNegAdvertisedCap,
38         lldpV2Xdot3RemPortOperMauType,
39         lldpV2Xdot3RemPowerPortClass,
40         lldpV2Xdot3RemPowerMDISupported,
41         lldpV2Xdot3RemPowerMDIEnabled,
42         lldpV2Xdot3RemPowerPairControlable,
43         lldpV2Xdot3RemPowerPairs,
44         lldpV2Xdot3RemPowerClass,
45         lldpV2Xdot3RemMaxFrameSize,
46         lldpV2Xdot3RemPowerType,
47         lldpV2Xdot3RemPowerSource,
48         lldpV2Xdot3RemPowerPriority,
49         lldpV2Xdot3RemPDRequestedPowerValue,
50         lldpV2Xdot3RemPSEAllocatedPowerValue,
51         lldpV2Xdot3RemTxTwSys,
52         lldpV2Xdot3RemTxTwSysEcho,
53         lldpV2Xdot3RemRxTwSys,
54         lldpV2Xdot3RemRxTwSysEcho,
55         lldpV2Xdot3RemFbTwSys
56     }
57
58 STATUS current
59 DESCRIPTION
60     "The collection of objects that are used to represent LLDP
61     IEEE 802.3 organizational extension Local Device Information."
62 ::= { lldpV2Xdot3Groups 3 }
63
64
65
```

1 END

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

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

6.1 Introduction

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

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

6.2 Overview

Ethernet OAM is composed of a core set of functions and a set of optional functional groups as described in 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. The optional functional groups are for (1) link events, (2) remote loopback, and (3) variable retrieval and response. Each optional functional group is controlled by a separate MIB table(s).

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

Ethernet OAM has three functional objectives, which are detailed in 6.2.1 through 6.2.3. The definition of a basic Ethernet OAM protocol data unit is given in 6.2.4.

6.2.1 Remote fault indication

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

6.2.2 Link monitoring

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

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

6.2.3 Remote loopback

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

6.2.4 Ethernet OAM protocol data units

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

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

6.3 Relation to other MIB modules

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

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

6.3.1 Relation to other EFM MIB modules

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

6.3.2 Mapping of IEEE 802.3 managed objects

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

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

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

IEEE 802.3 managed object	Corresponding SNMP object
oOAM	
.aOAMID	IF-MIB ifIndex
.aOAMAdminState	dot3OamAdminState
.aOAMMode	dot3OamMode
.aOAMDiscoveryState	dot3OamOperStatus
.aOAMRemoteMACAddress	dot3OamPeerMacAddress
.aOAMLocalConfiguration	dot3OamFunctionsSupported
.aOAMRemoteConfiguration	dot3OamPeerFunctionsSupported, dot3Oam-PeerMode
.aOAMLocalPDUConfiguration	dot3OamMaxOamPduSize
.aOAMRemotePDUConfiguration	dot3OamPeerMaxOamPduSize
.aOAMLocalFlagsField	dot3OamOperStatus, dot3OamEventLogEntry
.aOAMRemoteFlagsField	dot3OamOperStatus, dot3OamEventLogEntry
.aOAMLocalRevision	dot3OamConfigRevision
.aOAMRemoteRevision	dot3OamPeerConfigRevision
.aOAMLocalState	dot3OamLoopbackStatus
.aOAMRemoteState	dot3OamLoopbackStatus
.aOAMRemoteVendorOUI	dot3OamPeerVendorOui
.aOAMRemoteVendorSpecificInfo	dot3OamPeerVendorInfo
.aOAMUnsupportedCodesTx	dot3OamUnsupportedCodesTx
.aOAMUnsupportedCodesRx	dot3OamUnsupportedCodesRx
.aOAMInformationTx	dot3OamInformationTx
.aOAMInformationRx	dot3OamInformationRx
.aOAMUniqueEventNotificationTx	dot3OamUniqueEventNotificationTx
.aOAMUniqueEventNotificationRx	dot3OamUniqueEventNotificationRx
.aOAMDuplicateEventNotificationTx	dot3OamDuplicateEventNotificationTx
.aOAMDuplicateEventNotificationRx	dot3OamDuplicateEventNotificationRx
.aOAMLoopbackControlTx	dot3OamLoopbackControlTx

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

IEEE 802.3 managed object	Corresponding SNMP object
.aOAMLoopbackControlRx	dot3OamLoopbackControlRx
.aOAMVariableRequestTx	dot3OamVariableRequestTx
.aOAMVariableRequestRx	dot3OamVariableRequestRx
.aOAMVariableResponseTx	dot3OamVariableResponseTx
.aOAMVariableResponseRx	dot3OamVariableResponseRx
.aOAMOrganizationSpecificTx	dot3OamOrgSpecificTx
.aOAMOrganizationSpecificRx	dot3OamOrgSpecificTx
.aOAMLocalErrSymPeriodConfig	dot3OamErrSymPeriodWindow, dot3OamErrSymPeriodThreshold
.aOAMLocalErrSymPeriodEvent	dot3OamEventLogEntry
.aOAMLocalErrFrameConfig	dot3OamErrFrameWindow, dot3OamErr- FrameThreshold
.aOAMLocalErrFrameEvent	dot3OamEventLogEntry
.aOAMLocalErrFramePeriodConfig	dot3OamErrFramePeriodWindow, dot3OamEr- rFramePeriodThreshold
.aOAMLocalErrFramePeriodEvent	dot3OamEventLogEntry
.aOAMLocalErrFrameSecsSummaryConfig	dot3OamErrFrameSecsSummaryWindow, dot3OamErrFrameSecsummaryThreshold
.aOAMLocalErrFrameSecsSummaryEvent	dot3OamEventLogEntry
.aOAMRemoteErrSymPeriodEvent	dot3OamEventLogEntry
.aOAMRemoteErrFrameEvent	dot3OamEventLogEntry
.aOAMRemoteErrFramePeriodEvent	dot3OamEventLogEntry
.aOAMRemoteErrFrameSecsSummaryEvent	dot3OamEventLogEntry
.aFramesLostDueToOAMError	dot3OamFramesLostDueToOam
.acOAMAdminControl	dot3OamAdminState

6.4 MIB structure

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

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

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

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

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

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

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

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

The readable objects in this module can provide information about network traffic, and therefore, they may be considered sensitive. In particular, OAM provides mechanisms for reading the Clause 30 IEEE 802.3 MIB attributes from a link partner via a specialized layer 2 protocol. Unlike SNMP, IEEE 802.3 OAM does not include encryption or authentication mechanisms. It should not be used in environments where this interface information is considered sensitive, and where the facility terminations are unprotected. By default, OAM is disabled on Ethernet-like interfaces and is therefore not a risk.

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

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

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

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

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

6.6 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL¹²:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C6mib.txt

¹²Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
1  IEEE8023-DOT3-OAM-MIB DEFINITIONS ::= BEGIN
2      IMPORTS
3          MODULE-IDENTITY, OBJECT-TYPE, Counter32, Unsigned32,
4              Integer32, NOTIFICATION-TYPE, org
5              FROM SNMPv2-SMI
6              -- from [RFC2578]
7          TEXTUAL-CONVENTION, MacAddress, TimeStamp, TruthValue
8              FROM SNMPv2-TC
9              -- from [RFC2579]
10         CounterBasedGauge64
11         FROM HCNUM-TC
12         -- from [RFC2856]
13         ifIndex
14         FROM IF-MIB
15         -- from [RFC2863]
16     MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
17     FROM SNMPv2-CONF;
18     -- from [RFC2580]
19     ieee8023Dot3OamMIB MODULE-IDENTITY
20     LAST-UPDATED "201304110000Z" -- April 11, 2013
21     ORGANIZATION
22         "IEEE 802.3 working group"
23     CONTACT-INFO
24         "WG-URL: http://www.ieee802.org/3/index.html
25         WG-E-Mail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
26
27         Contact: Howard Frazier
28         Postal: 3151 Zanker Road
29                 San Jose, CA 95134
30                 USA
31         Tel:    +1.408.922.8164
32         E-mail: hfrazier@broadcom.com"
33     DESCRIPTION
34         "The MIB module for managing the new Ethernet OAM features
35         introduced by the Ethernet in the First Mile Task Force (IEEE
36         802.3ah). The functionality presented here is based on IEEE
37         Std 802.3ah, released in October, 2004, which was prepared as
38         an addendum to IEEE Std 802.3. Since then, IEEE Std 802.3ah
39         has been merged into the base IEEE 802.3 standard.
40
41         In particular, this MIB focuses on the new OAM functions
42         introduced in Clause 57 of IEEE Std 802.3. The OAM functionality
43         of Clause 57 is controlled by new management attributes
44         introduced in Clause 30 of IEEE Std 802.3. The OAM functions are
45         not specific to any particular Ethernet Physical Layer, and
46         can be generically applied to any Ethernet interface.
47
48         An Ethernet OAM protocol data unit is a valid Ethernet frame
49         with a destination MAC address equal to the reserved MAC
50         address for Slow Protocols (See Annex 57A of IEEE Std 802.3), a
51         lengthOrType field equal to the reserved type for Slow
52         Protocols, and a Slow Protocols subtype equal to that of the
53         subtype reserved for Ethernet OAM. OAMPDU is used throughout
54         this document as an abbreviation for Ethernet OAM protocol
55         data unit."
56
57     REVISION      "201304110000Z" -- April 11, 2013
58     DESCRIPTION   "Revision, based on an earlier version in
59                   IEEE Std 802.3.1-2011."
```

```
1
2     REVISION      "201102020000Z" -- February 2, 2011
3     DESCRIPTION   "Initial version, based on an earlier version in RFC 4878."
4
5
6     ::= { org ieee(111)
7           standards-association-numbers-series-standards(2)
8           lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 6 }
9
10
11  --
12  -- Sections of the Ethernet OAM MIB
13  --
14     dot3OamNotifications OBJECT IDENTIFIER ::= { ieee8023Dot3OamMIB 0 }
15     dot3OamObjects       OBJECT IDENTIFIER ::= { ieee8023Dot3OamMIB 1 }
16     dot3OamConformance   OBJECT IDENTIFIER ::= { ieee8023Dot3OamMIB 2 }
17
18  --
19  -- Textual conventions for the OAM MIB
20  --
21
22  EightOTwoOui ::= TEXTUAL-CONVENTION
23      DISPLAY-HINT "3x:"
24      STATUS      current
25      DESCRIPTION
26          "24-bit Organizationally Unique Identifier. Information on
27          OUIs can be found in IEEE 802-2001 [802-2001], Clause 9."
28      SYNTAX      OCTET STRING(SIZE(3))
29
30  -- *****
31  --
32  -- Ethernet OAM Control group
33  --
34
35
36  dot3OamTable OBJECT-TYPE
37      SYNTAX      SEQUENCE OF Dot3OamEntry
38      MAX-ACCESS  not-accessible
39      STATUS      current
40      DESCRIPTION
41          "This table contains the primary controls and status for the
42          OAM capabilities of an Ethernet-like interface. There will be
43          one row in this table for each Ethernet-like interface in the
44          system that supports the OAM functions defined in IEEE Std 802.3."
45      ::= { dot3OamObjects 1 }
46
47
48  dot3OamEntry OBJECT-TYPE
49      SYNTAX      Dot3OamEntry
50      MAX-ACCESS  not-accessible
51      STATUS      current
52      DESCRIPTION
53          "An entry in the table that contains information on the
54          Ethernet OAM function for a single Ethernet like interface.
55          Entries in the table are created automatically for each
56          interface supporting Ethernet OAM. The status of the row
57          entry can be determined from dot3OamOperStatus.
58
59          A dot3OamEntry is indexed in the dot3OamTable by the ifIndex
60          object of the Interfaces Group MIB.
61          "
62      INDEX      { ifIndex }
63      ::= { dot3OamTable 1 }
```

```
1
2   Dot3OamEntry ::=
3       SEQUENCE {
4           dot3OamAdminState          INTEGER,
5           dot3OamOperStatus          INTEGER,
6           dot3OamMode                INTEGER,
7           dot3OamMaxOamPduSize       Unsigned32,
8           dot3OamConfigRevision      Unsigned32,
9           dot3OamFunctionsSupported  BITS
10      }
11
12
13   dot3OamAdminState OBJECT-TYPE
14       SYNTAX      INTEGER {
15           enabled(1),
16           disabled(2)
17       }
18       MAX-ACCESS  read-write
19       STATUS      current
20       DESCRIPTION
21           "This object is used to provision the default administrative
22           OAM mode for this interface. This object represents the
23           desired state of OAM for this interface.
24
25           The dot3OamAdminState starts in the disabled(2) state
26           until an explicit management action or configuration
27           information retained by the system causes a transition to the
28           enabled(1) state. When enabled(1), Ethernet OAM will attempt
29           to operate over this interface."
30
31       REFERENCE   "IEEE Std 802.3, 30.3.6.1.2"
32       ::= { dot3OamEntry 1 }
33
34
35   dot3OamOperStatus OBJECT-TYPE
36       SYNTAX      INTEGER {
37           disabled(1),
38           linkFault(2),
39           passiveWait(3),
40           activeSendLocal(4),
41           sendLocalAndRemote(5),
42           sendLocalAndRemoteOk(6),
43           oamPeeringLocallyRejected(7),
44           oamPeeringRemotelyRejected(8),
45           operational(9),
46           nonOperHalfDuplex(10)
47       }
48       MAX-ACCESS  read-only
49       STATUS      current
50       DESCRIPTION
51           "At initialization and failure conditions, two OAM entities on
52           the same full-duplex Ethernet link begin a discovery phase to
53           determine what OAM capabilities may be used on that link. The
54           progress of this initialization is controlled by the OAM
55           sublayer.
56
57           This value is disabled(1) if OAM is disabled on this
58           interface via the dot3OamAdminState.
59
60           If the link has detected a fault and is transmitting OAMPDUs
61           with a link fault indication, the value is linkFault(2).
62
63
64
65
```


Also, if the interface is not operational (ifOperStatus is not up(1)), linkFault(2) is returned. Note that the object ifOperStatus may not be up(1) as a result of link failure or administrative action (ifAdminState being down(2) or testing(3)).

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

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

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

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

Since Ethernet OAM functions are not designed to work completely over half-duplex interfaces, the value nonOperHalfDuplex(10) is returned whenever Ethernet OAM is enabled (dot3OamAdminState is enabled(1)), but the interface is in half-duplex operation."

REFERENCE "IEEE Std 802.3, 30.3.6.1.4, 30.3.6.1.10, 30.3.6.1.11"
::= { dot3OamEntry 2 }

dot3OamMode OBJECT-TYPE
SYNTAX INTEGER {
passive(1),

```
1         active(2)
2     }
3     MAX-ACCESS read-write
4     STATUS current
5     DESCRIPTION
6         "This object configures the mode of OAM operation for this
7         Ethernet-like interface. OAM on Ethernet interfaces may be in
8         'active' mode or 'passive' mode. These two modes differ in
9         that active mode provides additional capabilities to initiate
10        monitoring activities with the remote OAM peer entity, while
11        passive mode generally waits for the peer to initiate OAM
12        actions with it. As an example, an active OAM entity can put
13        the remote OAM entity in a loopback state, where a passive OAM
14        entity cannot.
15
16        The default value of dot3OamMode is dependent on the type of
17        system on which this Ethernet-like interface resides. The
18        default value should be 'active(2)' unless it is known that
19        this system should take on a subservient role to the other
20        device connected over this interface.
21
22        Changing this value results in incrementing the configuration
23        revision field of locally generated OAMPDUs (IEEE Std 802.3
24        30.3.6.1.12) and potentially rerunning the OAM discovery process
25        if the dot3OamOperStatus was already operational(9)."
```

```
29     REFERENCE "IEEE Std 802.3, 30.3.6.1.3"
30     ::= { dot3OamEntry 3 }
31
32 dot3OamMaxOampPduSize OBJECT-TYPE
33     SYNTAX      Unsigned32 (64..1518)
34     UNITS       "octets"
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION
38         "The largest OAMPDU that the OAM entity supports. OAM
39         entities exchange maximum OAMPDU sizes and negotiate to use
40         the smaller of the two maximum OAMPDU sizes between the peers.
41         This value is determined by the local implementation."
```

```
45     REFERENCE "IEEE Std 802.3, 30.3.6.1.8"
46     ::= { dot3OamEntry 4 }
47
48 dot3OamConfigRevision OBJECT-TYPE
49     SYNTAX      Unsigned32(0..65535)
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "The configuration revision of the OAM entity as reflected in
54         the latest OAMPDU sent by the OAM entity. The config revision
55         is used by OAM entities to indicate that configuration changes
56         have occurred, which might require the peer OAM entity to
57         re-evaluate whether OAM peering is allowed."
```

```
60     REFERENCE "IEEE Std 802.3, 30.3.6.1.12"
61     ::= { dot3OamEntry 5 }
62
63 dot3OamFunctionsSupported OBJECT-TYPE
64     SYNTAX      BITS {
65
```

```

1          unidirectionalSupport (0),
2          loopbackSupport(1),
3          eventSupport(2),
4          variableSupport(3)
5      }
6  MAX-ACCESS  read-only
7  STATUS      current
8  DESCRIPTION
9
10     "The OAM functions supported on this Ethernet-like interface.
11     OAM consists of separate functional sets beyond the basic
12     discovery process that is required. These functional
13     groups can be supported independently by any implementation.
14     These values are communicated to the peer via the local
15     configuration field of Information OAMPDUs.
16
17     Setting 'unidirectionalSupport(0)' indicates that the OAM
18     entity supports the transmission of OAMPDUs on links that are
19     operating in unidirectional mode (traffic flowing in one
20     direction only). Setting 'loopbackSupport(1)' indicates that
21     the OAM entity can initiate and respond to loopback commands.
22     Setting 'eventSupport(2)' indicates that the OAM entity can
23     send and receive Event Notification OAMPDUs. Setting
24     'variableSupport(3)' indicates that the OAM entity can send
25     and receive Variable Request and Response OAMPDUs."
26
27
28  REFERENCE   "IEEE Std 802.3, 30.3.6.1.6"
29  ::= { dot3OamEntry 6 }
30
31  -- *****
32  --
33  -- Ethernet OAM Peer group
34  --
35
36
37  dot3OamPeerTable OBJECT-TYPE
38      SYNTAX      SEQUENCE OF Dot3OamPeerEntry
39      MAX-ACCESS  not-accessible
40      STATUS      current
41      DESCRIPTION
42          "This table contains information about the OAM peer for a
43          particular Ethernet-like interface. OAM entities communicate
44          with a single OAM peer entity on Ethernet links on which OAM
45          is enabled and operating properly. There is one entry in this
46          table for each entry in the dot3OamTable for which information
47          on the peer OAM entity is available."
48
49
50  ::= { dot3OamObjects 2 }
51
52
53  dot3OamPeerEntry OBJECT-TYPE
54      SYNTAX      Dot3OamPeerEntry
55      MAX-ACCESS  not-accessible
56      STATUS      current
57      DESCRIPTION
58          "An entry in the table containing information on the peer OAM
59          entity for a single Ethernet-like interface.
60
61          Note that there is at most one OAM peer for each Ethernet-like
62          interface. Entries are automatically created when information
63          about the OAM peer entity becomes available, and automatically
64          deleted when the OAM peer entity is no longer in
65

```

```
1      communication. Peer information is not available when
2      dot3OamOperStatus is disabled(1), linkFault(2),
3      passiveWait(3), activeSendLocal(4), or nonOperHalfDuplex(10).
4
5      INDEX          { ifIndex }
6      ::= { dot3OamPeerTable 1 }
7
8
9      Dot3OamPeerEntry ::=
10     SEQUENCE {
11         dot3OamPeerMacAddress      MacAddress,
12         dot3OamPeerVendorOui      EightOTwoOui,
13         dot3OamPeerVendorInfo     Unsigned32,
14         dot3OamPeerMode           INTEGER,
15         dot3OamPeerMaxOamPduSize   Unsigned32,
16         dot3OamPeerConfigRevision Unsigned32,
17         dot3OamPeerFunctionsSupported BITS
18     }
19
20
21     dot3OamPeerMacAddress OBJECT-TYPE
22     SYNTAX      MacAddress
23     MAX-ACCESS  read-only
24     STATUS      current
25     DESCRIPTION
26         "The MAC address of the peer OAM entity. The MAC address is
27         derived from the most recently received OAMPDU."
28
29
30     REFERENCE   "IEEE Std 802.3, 30.3.6.1.5."
31     ::= { dot3OamPeerEntry 1 }
32
33     dot3OamPeerVendorOui OBJECT-TYPE
34     SYNTAX      EightOTwoOui
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION
38         "The OUI of the OAM peer as reflected in the latest
39         Information OAMPDU received with a Local Information TLV. The
40         OUI can be used to identify the vendor of the remote OAM
41         entity. This value is initialized to three octets of zero
42         before any Local Information TLV is received."
43
44
45     REFERENCE   "IEEE Std 802.3, 30.3.6.1.16."
46     ::= { dot3OamPeerEntry 2 }
47
48     dot3OamPeerVendorInfo OBJECT-TYPE
49     SYNTAX      Unsigned32
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "The Vendor Info of the OAM peer as reflected in the latest
54         Information OAMPDU received with a Local Information TLV.
55         The semantics of the Vendor Information field is proprietary
56         and specific to the vendor (identified by the
57         dot3OamPeerVendorOui). This information could, for example,
58         be used to identify a specific product or product family.
59         This value is initialized to zero before any Local
60         Information TLV is received."
61
62
63     REFERENCE   "IEEE Std 802.3, 30.3.6.1.17."
64     ::= { dot3OamPeerEntry 3 }
65
```

```
1
2      dot3OamPeerMode OBJECT-TYPE
3          SYNTAX      INTEGER {
4              passive(1),
5              active(2),
6              unknown(3)
7          }
8
9      MAX-ACCESS      read-only
10     STATUS          current
11     DESCRIPTION
12         "The mode of the OAM peer as reflected in the latest
13         Information OAMPDU received with a Local Information TLV. The
14         mode of the peer can be determined from the Configuration
15         field in the Local Information TLV of the last Information
16         OAMPDU received from the peer. The value is unknown(3)
17         whenever no Local Information TLV has been received. The
18         values of active(2) and passive(1) are returned when a Local
19         Information TLV has been received indicating that the peer is
20         in active or passive mode, respectively."
21
22
23     REFERENCE      "IEEE Std 802.3, 30.3.6.1.7."
24     ::= { dot3OamPeerEntry 4 }
25
26     dot3OamPeerMaxOamPduSize OBJECT-TYPE
27         SYNTAX      Unsigned32 (0 | 64..1518)
28         UNITS       "octets"
29         MAX-ACCESS  read-only
30         STATUS      current
31         DESCRIPTION
32             "The maximum size of OAMPDU supported by the peer as reflected
33             in the latest Information OAMPDU received with a Local
34             Information TLV. Ethernet OAM on this interface shall not use
35             OAMPDUs that exceed this size. The maximum OAMPDU size can be
36             determined from the PDU Configuration field of the Local
37             Information TLV of the last Information OAMPDU received from
38             the peer. A value of zero is returned if no Local Information
39             TLV has been received. Otherwise, the value of the OAM peer's
40             maximum OAMPDU size is returned in this value."
41
42
43
44     REFERENCE      "IEEE Std 802.3, 30.3.6.1.9."
45     ::= { dot3OamPeerEntry 5 }
46
47     dot3OamPeerConfigRevision OBJECT-TYPE
48         SYNTAX      Unsigned32(0..65535)
49         MAX-ACCESS  read-only
50         STATUS      current
51         DESCRIPTION
52             "The configuration revision of the OAM peer as reflected in
53             the latest OAMPDU. This attribute is changed by the peer
54             whenever it has a local configuration change for Ethernet OAM
55             on this interface. The configuration revision can be
56             determined from the Revision field of the Local Information
57             TLV of the most recently received Information OAMPDU with
58             a Local Information TLV. A value of zero is returned if
59             no Local Information TLV has been received."
60
61
62     REFERENCE      "IEEE Std 802.3, 30.3.6.1.13."
63     ::= { dot3OamPeerEntry 6 }
64
65     dot3OamPeerFunctionsSupported OBJECT-TYPE
```

```
1      SYNTAX      BITS {
2                  unidirectionalSupport (0),
3                  loopbackSupport(1),
4                  eventSupport(2),
5                  variableSupport(3)
6              }
7
8      MAX-ACCESS   read-only
9      STATUS      current
10     DESCRIPTION
11         "The OAM functions supported on this Ethernet-like interface.
12         OAM consists of separate functionality sets above the basic
13         discovery process. This value indicates the capabilities of
14         the peer OAM entity with respect to these functions. This
15         value is initialized so all bits are clear.
16
17         If unidirectionalSupport(0) is set, then the peer OAM entity
18         supports sending OAM frames on Ethernet interfaces when the
19         receive path is known to be inoperable. If
20         loopbackSupport(1) is set, then the peer OAM entity can send
21         and receive OAM loopback commands. If eventSupport(2) is set,
22         then the peer OAM entity can send and receive event OAMPDUs to
23         signal various error conditions. If variableSupport(3) is
24         set, then the peer OAM entity can send and receive variable
25         requests to monitor the attribute value as described in Clause
26         57 of IEEE Std 802.3.
27
28         The capabilities of the OAM peer can be determined from the
29         configuration field of the Local Information TLV of the most
30         recently received Information OAMPDU with a Local Information
31         TLV. All zeros are returned if no Local Information TLV has
32         yet been received."
33
34     REFERENCE    "IEEE Std 802.3 30.3.6.1.7."
35     ::= { dot3OamPeerEntry 7 }
36
37     -- *****
38     --
39     -- Ethernet OAM Loopback group
40     --
41
42     dot3OamLoopbackTable OBJECT-TYPE
43     SYNTAX      SEQUENCE OF Dot3OamLoopbackEntry
44     MAX-ACCESS   not-accessible
45     STATUS      current
46     DESCRIPTION
47         "This table contains controls for the loopback state of the
48         local link as well as indicates the status of the loopback
49         function. There is one entry in this table for each entry in
50         dot3OamTable that supports loopback functionality (where
51         dot3OamFunctionsSupported includes the loopbackSupport bit
52         set).
53
54         Loopback can be used to place the remote OAM entity in a state
55         where every received frame (except OAMPDUs) is echoed back
56         over the same interface on which they were received. In this
57         state, at the remote entity, 'normal' traffic is disabled as
58         only the looped back frames are transmitted on the interface.
59         Loopback is thus an intrusive operation that prohibits normal
60         data flow and should be used accordingly."
61
62
63
64
65
```

```

1      ::= { dot3OamObjects 3 }
2
3
4      dot3OamLoopbackEntry OBJECT-TYPE
5          SYNTAX      Dot3OamLoopbackEntry
6          MAX-ACCESS   not-accessible
7          STATUS       current
8          DESCRIPTION
9              "An entry in the table, containing information on the loopback
10             status for a single Ethernet-like interface. Entries in the
11             table are automatically created whenever the local OAM entity
12             supports loopback capabilities. The loopback status on the
13             interface can be determined from the dot3OamLoopbackStatus
14             object."
15
16
17          INDEX          { ifIndex }
18      ::= { dot3OamLoopbackTable 1 }
19
20
21      Dot3OamLoopbackEntry ::=
22          SEQUENCE {
23              dot3OamLoopbackStatus      INTEGER,
24              dot3OamLoopbackIgnoreRx    INTEGER
25          }
26
27      dot3OamLoopbackStatus OBJECT-TYPE
28          SYNTAX      INTEGER {
29              -- all values, except where noted, can be read
30              -- but cannot be written
31              noLoopback (1),
32
33              -- initiatingLoopback can be read or written
34              initiatingLoopback (2),
35              remoteLoopback (3),
36
37              -- terminatingLoopback can be read or written
38              terminatingLoopback (4),
39              localLoopback (5),
40              unknown (6)
41          }
42
43      MAX-ACCESS   read-write
44      STATUS       current
45      DESCRIPTION
46          "The loopback status of the OAM entity. This status is
47          determined by a combination of the local parser and
48          multiplexer states, the remote parser and multiplexer states,
49          as well as by the actions of the local OAM client. When
50          operating in normal mode with no loopback in progress, the
51          status reads noLoopback(1).
52
53          The values initiatingLoopback(2) and terminatingLoopback(4)
54          can be read or written. The other values can only be read -
55          they can never be written. Writing initiatingLoopback causes
56          the local OAM entity to start the loopback process with its
57          peer. This value can only be written when the status is
58          noLoopback(1). Writing the value initiatingLoopback(2) in any
59          other state has no effect. When in remoteLoopback(3), writing
60          terminatingLoopback(4) causes the local OAM entity to initiate
61          the termination of the loopback state. Writing
62          terminatingLoopback(4) in any other state has no effect.

```

1
2 If the OAM client initiates a loopback and has sent a
3 Loopback OAMPDU and is waiting for a response, where the local
4 parser and multiplexer states are DISCARD (see IEEE Std 802.3,
5 57.2.11.1), the status is 'initiatingLoopback'. In this
6 case, the local OAM entity has yet to receive any
7 acknowledgment that the remote OAM entity has received its
8 loopback command request.
9
10 If the local OAM client knows that the remote OAM entity is in
11 loopback mode (via the remote state information as described
12 in IEEE Std 802.3, 57.2.11.1, 30.3.6.1.15), the status is
13 remoteLoopback(3). If the local OAM client is in the process
14 of terminating the remote loopback (see IEEE Std 802.3, 57.2.11.3,
15 30.3.6.1.14) with its local multiplexer and parser states in
16 DISCARD, the status is terminatingLoopback(4). If the remote
17 OAM client has put the local OAM entity in loopback mode as
18 indicated by its local parser state, the status is
19 localLoopback(5).
20

21
22 The unknown(6) status indicates that the parser and
23 multiplexer combination is unexpected. This status may be
24 returned if the OAM loopback is in a transition state but
25 should not persist.
26

27 The values of this attribute correspond to the following
28 values of the local and remote parser and multiplexer states.
29

value	LclPrsr	LclMux	RmtPrsr	RmtMux
noLoopback	FWD	FWD	FWD	FWD
initLoopback	DISCARD	DISCARD	FWD	FWD
rmtLoopback	DISCARD	FWD	LPBK	DISCARD
tmtngLoopback	DISCARD	DISCARD	LPBK	DISCARD
lclLoopback	LPBK	DISCARD	DISCARD	FWD
unknown	***	any other combination	***	***

30
31
32
33
34
35
36
37
38
39
40 REFERENCE "IEEE Std 802.3, 57.2.11, 30.3.6.1.14, 30.3.6.1.15"
41 ::= { dot3OamLoopbackEntry 1 }
42

43 dot3OamLoopbackIgnoreRx OBJECT-TYPE
44 SYNTAX INTEGER { ignore(1), process(2) }
45 MAX-ACCESS read-write
46 STATUS current
47 DESCRIPTION
48 "Since OAM loopback is a disruptive operation (user traffic
49 does not pass), this attribute provides a mechanism to provide
50 controls over whether received OAM loopback commands are
51 processed or ignored. When the value is ignore(1), received
52 loopback commands are ignored. When the value is process(2),
53 OAM loopback commands are processed. The default value is to
54 ignore loopback commands (ignore(1))."
55
56

57 REFERENCE "IEEE Std 802.3, 57.2.11, 30.3.6.1.14, 30.3.6.1.15"
58 ::= { dot3OamLoopbackEntry 2 }
59 -- *****
60 --
61 -- Ethernet OAM Statistics group
62 --
63
64

65 dot3OamStatsTable OBJECT-TYPE


```

1      SYNTAX      SEQUENCE OF Dot3OamStatsEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "This table contains statistics for the OAM function on a
6          particular Ethernet-like interface. There is an entry in the
7          table for every entry in the dot3OamTable.
8
9
10         The counters in this table are defined as 32-bit entries to
11         match the counter size as defined in IEEE Std 802.3. Given that
12         the OAM protocol is a slow protocol, the counters increment at
13         a slow rate."
14
15         ::= { dot3OamObjects 4 }
16
17  dot3OamStatsEntry OBJECT-TYPE
18      SYNTAX      Dot3OamStatsEntry
19      MAX-ACCESS  not-accessible
20      STATUS      current
21      DESCRIPTION
22          "An entry in the table containing statistics information on
23          the Ethernet OAM function for a single Ethernet-like
24          interface. Entries are automatically created for every entry
25          in the dot3OamTable. Counters are maintained across
26          transitions in dot3OamOperStatus."
27
28
29      INDEX      { ifIndex }
30      ::= { dot3OamStatsTable 1 }
31
32
33  Dot3OamStatsEntry ::=
34      SEQUENCE {
35          dot3OamInformationTx      Counter32,
36          dot3OamInformationRx      Counter32,
37          dot3OamUniqueEventNotificationTx  Counter32,
38          dot3OamUniqueEventNotificationRx  Counter32,
39          dot3OamDuplicateEventNotificationTx Counter32,
40          dot3OamDuplicateEventNotificationRx Counter32,
41          dot3OamLoopbackControlTx   Counter32,
42          dot3OamLoopbackControlRx   Counter32,
43          dot3OamVariableRequestTx   Counter32,
44          dot3OamVariableRequestRx   Counter32,
45          dot3OamVariableResponseTx  Counter32,
46          dot3OamVariableResponseRx  Counter32,
47          dot3OamOrgSpecificTx       Counter32,
48          dot3OamOrgSpecificRx       Counter32,
49          dot3OamUnsupportedCodesTx  Counter32,
50          dot3OamUnsupportedCodesRx  Counter32,
51          dot3OamFramesLostDueToOam  Counter32
52      }
53
54
55
56  dot3OamInformationTx OBJECT-TYPE
57      SYNTAX      Counter32
58      UNITS       "frames"
59      MAX-ACCESS  read-only
60      STATUS      current
61      DESCRIPTION
62          "A count of the number of Information OAMPDUs transmitted on
63          this interface.
64
65

```

1 Discontinuities of this counter can occur at re-initialization
2 of the management system, and at other times as indicated by
3 the value of the ifCounterDiscontinuityTime."
4
5 REFERENCE "IEEE Std 802.3, 30.3.6.1.20."
6 ::= { dot3OamStatsEntry 1 }
7
8
9 dot3OamInformationRx OBJECT-TYPE
10 SYNTAX Counter32
11 UNITS "frames"
12 MAX-ACCESS read-only
13 STATUS current
14 DESCRIPTION
15 "A count of the number of Information OAMPDUs received on this
16 interface."
17
18 Discontinuities of this counter can occur at re-initialization
19 of the management system, and at other times as indicated by
20 the value of the ifCounterDiscontinuityTime."
21
22
23 REFERENCE "IEEE Std 802.3, 30.3.6.1.21."
24 ::= { dot3OamStatsEntry 2 }
25
26 dot3OamUniqueEventNotificationTx OBJECT-TYPE
27 SYNTAX Counter32
28 UNITS "frames"
29 MAX-ACCESS read-only
30 STATUS current
31 DESCRIPTION
32 "A count of the number of unique Event OAMPDUs transmitted on
33 this interface. Event Notifications may be sent in duplicate
34 to increase the probability of successfully being received,
35 given the possibility that a frame may be lost in transit.
36 Duplicate Event Notification transmissions are counted by
37 dot3OamDuplicateEventNotificationTx."
38
39
40 A unique Event Notification OAMPDU is indicated as an Event
41 Notification OAMPDU with a Sequence Number field that is
42 distinct from the previously transmitted Event Notification
43 OAMPDU Sequence Number.
44
45
46 Discontinuities of this counter can occur at re-initialization
47 of the management system, and at other times as indicated by
48 the value of the ifCounterDiscontinuityTime."
49
50
51 REFERENCE "IEEE Std 802.3, 30.3.6.1.22."
52 ::= { dot3OamStatsEntry 3 }
53
54 dot3OamUniqueEventNotificationRx OBJECT-TYPE
55 SYNTAX Counter32
56 UNITS "frames"
57 MAX-ACCESS read-only
58 STATUS current
59 DESCRIPTION
60 "A count of the number of unique Event OAMPDUs received on
61 this interface. Event Notification OAMPDUs may be sent in
62 duplicate to increase the probability of successfully being
63 received, given the possibility that a frame may be lost in
64 transit. Duplicate Event Notification receptions are counted
65

1 by dot3OamDuplicateEventNotificationRx.
2
3 A unique Event Notification OAMPDU is indicated as an Event
4 Notification OAMPDU with a Sequence Number field that is
5 distinct from the previously received Event Notification
6 OAMPDU Sequence Number.
7
8
9 Discontinuities of this counter can occur at re-initialization
10 of the management system, and at other times as indicated by
11 the value of the ifCounterDiscontinuityTime."
12
13 REFERENCE "IEEE Std 802.3, 30.3.6.1.24."
14 ::= { dot3OamStatsEntry 4 }
15
16 dot3OamDuplicateEventNotificationTx OBJECT-TYPE
17 SYNTAX Counter32
18 UNITS "frames"
19 MAX-ACCESS read-only
20 STATUS current
21 DESCRIPTION
22 "A count of the number of duplicate Event OAMPDUs transmitted
23 on this interface. Event Notification OAMPDUs may be sent in
24 duplicate to increase the probability of successfully being
25 received, given the possibility that a frame may be lost in
26 transit."
27
28
29 A duplicate Event Notification OAMPDU is indicated as an Event
30 Notification OAMPDU with a Sequence Number field that is
31 identical to the previously transmitted Event Notification
32 OAMPDU Sequence Number.
33
34
35 Discontinuities of this counter can occur at re-initialization
36 of the management system, and at other times as indicated by
37 the value of the ifCounterDiscontinuityTime."
38
39 REFERENCE "IEEE Std 802.3, 30.3.6.1.23."
40 ::= { dot3OamStatsEntry 5 }
41
42 dot3OamDuplicateEventNotificationRx OBJECT-TYPE
43 SYNTAX Counter32
44 UNITS "frames"
45 MAX-ACCESS read-only
46 STATUS current
47 DESCRIPTION
48 "A count of the number of duplicate Event OAMPDUs received on
49 this interface. Event Notification OAMPDUs may be sent in
50 duplicate to increase the probability of successfully being
51 received, given the possibility that a frame may be lost in
52 transit."
53
54
55 A duplicate Event Notification OAMPDU is indicated as an Event
56 Notification OAMPDU with a Sequence Number field that is
57 identical to the previously received Event Notification OAMPDU
58 Sequence Number.
59
60
61 Discontinuities of this counter can occur at re-initialization
62 of the management system, and at other times as indicated by
63 the value of the ifCounterDiscontinuityTime."
64
65

```
1      REFERENCE    "IEEE Std 802.3, 30.3.6.1.25."
2      ::= { dot3OamStatsEntry 6 }
3
4      dot3OamLoopbackControlTx OBJECT-TYPE
5          SYNTAX      Counter32
6          UNITS        "frames"
7          MAX-ACCESS   read-only
8          STATUS       current
9          DESCRIPTION
10             "A count of the number of Loopback Control OAMPDUs transmitted
11             on this interface.
12
13             Discontinuities of this counter can occur at re-initialization
14             of the management system, and at other times as indicated by
15             the value of the ifCounterDiscontinuityTime."
16
17      REFERENCE    "IEEE Std 802.3, 30.3.6.1.26."
18      ::= { dot3OamStatsEntry 7 }
19
20      dot3OamLoopbackControlRx OBJECT-TYPE
21          SYNTAX      Counter32
22          UNITS        "frames"
23          MAX-ACCESS   read-only
24          STATUS       current
25          DESCRIPTION
26             "A count of the number of Loopback Control OAMPDUs received
27             on this interface.
28
29             Discontinuities of this counter can occur at re-initialization
30             of the management system, and at other times as indicated by
31             the value of the ifCounterDiscontinuityTime."
32
33      REFERENCE    "IEEE Std 802.3, 30.3.6.1.27."
34      ::= { dot3OamStatsEntry 8 }
35
36      dot3OamVariableRequestTx OBJECT-TYPE
37          SYNTAX      Counter32
38          UNITS        "frames"
39          MAX-ACCESS   read-only
40          STATUS       current
41          DESCRIPTION
42             "A count of the number of Variable Request OAMPDUs transmitted
43             on this interface.
44
45             Discontinuities of this counter can occur at re-initialization
46             of the management system, and at other times as indicated by
47             the value of the ifCounterDiscontinuityTime."
48
49      REFERENCE    "IEEE Std 802.3, 30.3.6.1.28."
50      ::= { dot3OamStatsEntry 9 }
51
52      dot3OamVariableRequestRx OBJECT-TYPE
53          SYNTAX      Counter32
54          UNITS        "frames"
55          MAX-ACCESS   read-only
56          STATUS       current
57          DESCRIPTION
58             "A count of the number of Variable Request OAMPDUs received on
59             this interface.
```

```
1
2      Discontinuities of this counter can occur at re-initialization
3      of the management system, and at other times as indicated by
4      the value of the ifCounterDiscontinuityTime."
5
6      REFERENCE      "IEEE Std 802.3, 30.3.6.1.29."
7      ::= { dot3OamStatsEntry 10 }
8
9
10     dot3OamVariableResponseTx OBJECT-TYPE
11         SYNTAX      Counter32
12         UNITS        "frames"
13         MAX-ACCESS   read-only
14         STATUS       current
15         DESCRIPTION
16             "A count of the number of Variable Response OAMPDUs
17             transmitted on this interface.
18
19             Discontinuities of this counter can occur at re-initialization
20             of the management system, and at other times as indicated by
21             the value of the ifCounterDiscontinuityTime."
22
23     REFERENCE      "IEEE Std 802.3, 30.3.6.1.30."
24     ::= { dot3OamStatsEntry 11 }
25
26
27     dot3OamVariableResponseRx OBJECT-TYPE
28         SYNTAX      Counter32
29         UNITS        "frames"
30         MAX-ACCESS   read-only
31         STATUS       current
32         DESCRIPTION
33             "A count of the number of Variable Response OAMPDUs received
34             on this interface.
35
36             Discontinuities of this counter can occur at re-initialization
37             of the management system, and at other times as indicated by
38             the value of the ifCounterDiscontinuityTime."
39
40     REFERENCE      "IEEE Std 802.3, 30.3.6.1.31."
41     ::= { dot3OamStatsEntry 12 }
42
43
44     dot3OamOrgSpecificTx OBJECT-TYPE
45         SYNTAX      Counter32
46         UNITS        "frames"
47         MAX-ACCESS   read-only
48         STATUS       current
49         DESCRIPTION
50             "A count of the number of Organization Specific OAMPDUs
51             transmitted on this interface.
52
53             Discontinuities of this counter can occur at re-initialization
54             of the management system, and at other times as indicated by
55             the value of the ifCounterDiscontinuityTime."
56
57     REFERENCE      "IEEE Std 802.3, 30.3.6.1.32."
58     ::= { dot3OamStatsEntry 13 }
59
60
61     dot3OamOrgSpecificRx OBJECT-TYPE
62         SYNTAX      Counter32
63         UNITS        "frames"
```

```
1      MAX-ACCESS    read-only
2      STATUS        current
3      DESCRIPTION
4          "A count of the number of Organization Specific OAMPDUs
5          received on this interface.
6
7          Discontinuities of this counter can occur at re-initialization
8          of the management system, and at other times as indicated by
9          the value of the ifCounterDiscontinuityTime."
10
11
12      REFERENCE     "IEEE Std 802.3, 30.3.6.1.33."
13      ::= { dot3OamStatsEntry 14 }
14
15  dot3OamUnsupportedCodesTx OBJECT-TYPE
16      SYNTAX        Counter32
17      UNITS          "frames"
18      MAX-ACCESS    read-only
19      STATUS        current
20      DESCRIPTION
21          "A count of the number of OAMPDUs transmitted on this
22          interface with an unsupported op-code.
23
24          Discontinuities of this counter can occur at re-initialization
25          of the management system, and at other times as indicated by
26          the value of the ifCounterDiscontinuityTime."
27
28
29      REFERENCE     "IEEE Std 802.3, 30.3.6.1.18."
30      ::= { dot3OamStatsEntry 15 }
31
32
33  dot3OamUnsupportedCodesRx OBJECT-TYPE
34      SYNTAX        Counter32
35      UNITS          "frames"
36      MAX-ACCESS    read-only
37      STATUS        current
38      DESCRIPTION
39          "A count of the number of OAMPDUs received on this interface
40          with an unsupported op-code.
41
42          Discontinuities of this counter can occur at re-initialization
43          of the management system, and at other times as indicated by
44          the value of the ifCounterDiscontinuityTime."
45
46
47      REFERENCE     "IEEE Std 802.3, 30.3.6.1.19."
48      ::= { dot3OamStatsEntry 16 }
49
50
51  dot3OamFramesLostDueToOam OBJECT-TYPE
52      SYNTAX        Counter32
53      UNITS          "frames"
54      MAX-ACCESS    read-only
55      STATUS        current
56      DESCRIPTION
57          "A count of the number of frames that were dropped by the OAM
58          multiplexer. Since the OAM multiplexer has multiple inputs
59          and a single output, there may be cases where frames are
60          dropped due to transmit resource contention. This counter is
61          incremented whenever a frame is dropped by the OAM layer.
62          Note that any Ethernet frame, not just OAMPDUs, may be dropped
63          by the OAM layer. This can occur when an OAMPDU takes
64          precedence over a 'normal' frame resulting in the 'normal'
65          frame being dropped."
```

```
1         frame being dropped.
2
3         When this counter is incremented, no other counters in this
4         MIB are incremented.
5
6         Discontinuities of this counter can occur at re-initialization
7         of the management system, and at other times as indicated by
8         the value of the ifCounterDiscontinuityTime."
9
10
11     REFERENCE      "IEEE Std 802.3, 30.3.6.1.46."
12     ::= { dot3OamStatsEntry 17 }
13
14     -- *****
15     --
16     -- Ethernet OAM Event Configuration group
17     --
18
19     dot3OamEventConfigTable OBJECT-TYPE
20     SYNTAX          SEQUENCE OF Dot3OamEventConfigEntry
21     MAX-ACCESS      not-accessible
22     STATUS          current
23     DESCRIPTION
24         "Ethernet OAM includes the ability to generate and receive
25         Event Notification OAMPDUs to indicate various link problems.
26         This table contains the mechanisms to enable Event
27         Notifications and configure the thresholds to generate the
28         standard Ethernet OAM events. There is one entry in the table
29         for every entry in dot3OamTable that supports OAM events
30         (where dot3OamFunctionsSupported includes the eventSupport
31         bit set). The values in the table are maintained across
32         changes to dot3OamOperStatus.
33
34         The standard threshold crossing events are:
35
36         - Errored Symbol Period Event. Generated when the number of
37         symbol errors exceeds a threshold within a given window
38         defined by a number of symbols (for example, 1,000 symbols
39         out of 1,000,000 had errors).
40
41         - Errored Frame Period Event. Generated when the number of
42         frame errors exceeds a threshold within a given window
43         defined by a number of frames (for example, 10 frames out
44         of 1000 had errors).
45
46         - Errored Frame Event. Generated when the number of frame
47         errors exceeds a threshold within a given window defined
48         by a period of time (for example, 10 frames in 1 second
49         had errors).
50
51         - Errored Frame Seconds Summary Event. Generated when the
52         number of errored frame seconds exceeds a threshold within
53         a given time period (for example, 10 errored frame seconds
54         within the last 100 seconds). An errored frame second is
55         defined as a 1 second interval which had >0 frame errors.
56         There are other events (dying gasp, critical events) that are
57         not threshold crossing events but that can be
58         enabled/disabled via this table."
59
60     ::= { dot3OamObjects 5 }
61
62     dot3OamEventConfigEntry OBJECT-TYPE
63     SYNTAX          Dot3OamEventConfigEntry
64     MAX-ACCESS      not-accessible
65
```

```

1      STATUS      current
2      DESCRIPTION
3          "Entries are automatically created and deleted from this
4          table, and exist whenever the OAM entity supports Ethernet OAM
5          events (as indicated by the eventSupport bit in
6          dot3OamFunctionsSupported). Values in the table are
7          maintained across changes to the value of dot3OamOperStatus.
8
9
10         Event configuration controls when the local management entity
11         sends Event Notification OAMPDUs to its OAM peer, and when
12         certain event flags are set or cleared in OAMPDUs."
13
14     INDEX          { ifIndex }
15     ::= { dot3OamEventConfigTable 1 }
16
17 Dot3OamEventConfigEntry ::=
18     SEQUENCE {
19         dot3OamErrSymPeriodWindowHi      Unsigned32,
20         dot3OamErrSymPeriodWindowLo      Unsigned32,
21         dot3OamErrSymPeriodThresholdHi    Unsigned32,
22         dot3OamErrSymPeriodThresholdLo    Unsigned32,
23         dot3OamErrSymPeriodEvNotifEnable  TruthValue,
24         dot3OamErrFramePeriodWindow      Unsigned32,
25         dot3OamErrFramePeriodThreshold    Unsigned32,
26         dot3OamErrFramePeriodEvNotifEnable TruthValue,
27         dot3OamErrFrameWindow            Unsigned32,
28         dot3OamErrFrameThreshold          Unsigned32,
29         dot3OamErrFrameEvNotifEnable      TruthValue,
30         dot3OamErrFrameSecsSummaryWindow  Integer32,
31         dot3OamErrFrameSecsSummaryThreshold Integer32,
32         dot3OamErrFrameSecsEvNotifEnable  TruthValue,
33         dot3OamDyingGaspEnable            TruthValue,
34         dot3OamCriticalEventEnable        TruthValue
35     }
36
37
38
39 dot3OamErrSymPeriodWindowHi OBJECT-TYPE
40     SYNTAX      Unsigned32
41     UNITS       "2^32 symbols"
42     MAX-ACCESS  read-write
43     STATUS      current
44     DESCRIPTION
45         "The two objects dot3OamErrSymPeriodWindowHi and
46         dot3OamErrSymPeriodLo together form an unsigned 64-bit
47         integer representing the number of symbols over which this
48         threshold event is defined. This is defined as
49         dot3OamErrSymPeriodWindow = ((2^32)*dot3OamErrSymPeriodWindowHi)
50                                     + dot3OamErrSymPeriodWindowLo
51
52
53         If dot3OamErrSymPeriodThreshold symbol errors occur within a
54         window of dot3OamErrSymPeriodWindow symbols, an Event
55         Notification OAMPDU should be generated with an Errorred Symbol
56         Period Event TLV indicating that the threshold has been
57         crossed in this window.
58
59
60         The default value for dot3OamErrSymPeriodWindow is the number
61         of symbols in one second for the underlying Physical Layer."
62
63
64     REFERENCE   "IEEE Std 802.3, 30.3.6.1.34"
65     ::= { dot3OamEventConfigEntry 1 }

```


dot3OamErrSymPeriodWindowLo OBJECT-TYPE

SYNTAX Unsigned32
UNITS "symbols"
MAX-ACCESS read-write
STATUS current

DESCRIPTION

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

$$\text{dot3OamErrSymPeriodWindow} = ((2^{32}) * \text{dot3OamErrSymPeriodWindowHi}) + \text{dot3OamErrSymPeriodWindowLo}$$

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

The default value for dot3OamErrSymPeriodWindow is the number of symbols in one second for the underlying Physical Layer."

REFERENCE "IEEE Std 802.3, 30.3.6.1.34"
::= { dot3OamEventConfigEntry 2 }

dot3OamErrSymPeriodThresholdHi OBJECT-TYPE

SYNTAX Unsigned32
UNITS "2^32 symbols"
MAX-ACCESS read-write
STATUS current

DESCRIPTION

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

This is defined as

$$\text{dot3OamErrSymPeriodThreshold} = ((2^{32}) * \text{dot3OamErrSymPeriodThresholdHi}) + \text{dot3OamErrSymPeriodThresholdLo}$$

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

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

REFERENCE "IEEE Std 802.3, 30.3.6.1.34"
::= { dot3OamEventConfigEntry 3 }

```
1
2      dot3OamErrSymPeriodThresholdLo OBJECT-TYPE
3          SYNTAX      Unsigned32
4          UNITS       "symbols"
5          MAX-ACCESS  read-write
6          STATUS      current
7          DESCRIPTION
8              "The two objects dot3OamErrSymPeriodThresholdHi and
9              dot3OamErrSymPeriodThresholdLo together form an unsigned
10             64-bit integer representing the minimum number of symbol errors
11             occurring within a given window to cause an Errored Symbol Period Event.
12
13             This is defined as
14
15             dot3OamErrSymPeriodThreshold =
16                 ((2^32) * dot3OamErrSymPeriodThresholdHi)
17                 + dot3OamErrSymPeriodThresholdLo
18
19             If dot3OamErrSymPeriodThreshold symbol errors occur within a
20             window of dot3OamErrSymPeriodWindow symbols, an Event
21             Notification OAMPDU is generated with an Errored Symbol
22             Period Event TLV indicating that the threshold has been
23             crossed in this window.
24
25             The default value for dot3OamErrSymPeriodThreshold is one
26             symbol error. If the threshold value is zero, then an Event
27             Notification OAMPDU is sent periodically (at the end of every
28             window). This can be used as an asynchronous notification to
29             the peer OAM entity of the statistics related to this
30             threshold crossing alarm."
31
32             REFERENCE      "IEEE Std 802.3, 30.3.6.1.34"
33             ::= { dot3OamEventConfigEntry 4 }
34
35      dot3OamErrSymPeriodEvNotifEnable OBJECT-TYPE
36          SYNTAX      TruthValue
37          MAX-ACCESS  read-write
38          STATUS      current
39          DESCRIPTION
40              "If true, the OAM entity sends an Event Notification
41              OAMPDU when an Errored Symbol Period Event occurs.
42
43              The default value for this object is true for
44              Ethernet-like interfaces that support OAM. If the OAM layer
45              does not support Event Notifications (as indicated via the
46              dot3OamFunctionsSupported attribute), this value is ignored."
47
48             ::= { dot3OamEventConfigEntry 5 }
49
50      dot3OamErrFramePeriodWindow OBJECT-TYPE
51          SYNTAX      Unsigned32
52          UNITS       "frames"
53          MAX-ACCESS  read-write
54          STATUS      current
55          DESCRIPTION
56              "The number of frames over which the threshold is defined.
57              The default value of the window is the number of minimum size
58              Ethernet frames that can be received over the Physical Layer
59              in one second.
```

1
2 If dot3OamErrFramePeriodThreshold frame errors occur within a
3 window of dot3OamErrFramePeriodWindow frames, an Event
4 Notification OAMPDU should be generated with an Errored Frame
5 Period Event TLV indicating that the threshold has been
6 crossed in this window."
7
8
9 REFERENCE "IEEE Std 802.3, 30.3.6.1.38"
10 ::= { dot3OamEventConfigEntry 6 }
11
12 dot3OamErrFramePeriodThreshold OBJECT-TYPE
13 SYNTAX Unsigned32
14 UNITS "frames"
15 MAX-ACCESS read-write
16 STATUS current
17 DESCRIPTION
18 "The minimum number of frame errors that cause an Errored Frame
19 Period Event. The default value is one frame error. If the
20 threshold value is zero, then an Event Notification OAMPDU is
21 sent periodically (at the end of every window). This can be
22 used as an asynchronous notification to the peer OAM entity of
23 the statistics related to this threshold crossing alarm."
24
25
26 If dot3OamErrFramePeriodThreshold frame errors occur within a
27 window of dot3OamErrFramePeriodWindow frames, an Event
28 Notification OAMPDU is generated with an Errored Frame
29 Period Event TLV indicating that the threshold has been
30 crossed in this window."
31
32
33 REFERENCE "IEEE Std 802.3, 30.3.6.1.38"
34 ::= { dot3OamEventConfigEntry 7 }
35
36 dot3OamErrFramePeriodEvNotifEnable OBJECT-TYPE
37 SYNTAX TruthValue
38 MAX-ACCESS read-write
39 STATUS current
40 DESCRIPTION
41 "If true, the OAM entity should send an Event Notification
42 OAMPDU when an Errored Frame Period Event occurs."
43
44
45 By default, this object should have the value true for
46 Ethernet-like interfaces that support OAM. If the OAM layer
47 does not support Event Notifications (as indicated via the
48 dot3OamFunctionsSupported attribute), this value is ignored."
49
50 ::= { dot3OamEventConfigEntry 8 }
51
52
53 dot3OamErrFrameWindow OBJECT-TYPE
54 SYNTAX Unsigned32
55 UNITS "tenths of a second"
56 MAX-ACCESS read-write
57 STATUS current
58 DESCRIPTION
59 "The amount of time (in 100 ms increments) over which the
60 threshold is defined. The default value is 10 (1 second)."
61
62
63 If dot3OamErrFrameThreshold frame errors occur within a window
64 of dot3OamErrFrameWindow seconds (measured in tenths of
65 seconds), an Event Notification OAMPDU should be generated

```
1         with an Errored Frame Event TLV indicating that the threshold
2         has been crossed in this window."
3
4     REFERENCE    "IEEE Std 802.3, 30.3.6.1.36"
5     DEFVAL { 10 }
6     ::= { dot3OamEventConfigEntry 9 }
7
8
9     dot3OamErrFrameThreshold OBJECT-TYPE
10     SYNTAX      Unsigned32
11     UNITS       "frames"
12     MAX-ACCESS  read-write
13     STATUS      current
14     DESCRIPTION
15         "The minimum number of frame errors that cause an Errored Frame
16         Event. The default value is one frame error. If the
17         threshold value is zero, then an Event Notification OAMPDU is
18         sent periodically (at the end of every window). This can be
19         used as an asynchronous notification to the peer OAM entity of
20         the statistics related to this threshold crossing alarm.
21
22         If dot3OamErrFrameThreshold frame errors occur within a window
23         of dot3OamErrFrameWindow (in tenths of seconds), an Event
24         Notification OAMPDU is generated with an Errored Frame
25         Event TLV indicating the threshold has been crossed in this
26         window."
27
28
29     REFERENCE    "IEEE Std 802.3, 30.3.6.1.36"
30     DEFVAL { 1 }
31     ::= { dot3OamEventConfigEntry 10 }
32
33
34     dot3OamErrFrameEvNotifEnable OBJECT-TYPE
35     SYNTAX      TruthValue
36     MAX-ACCESS  read-write
37     STATUS      current
38     DESCRIPTION
39         "If true, the OAM entity should send an Event Notification
40         OAMPDU when an Errored Frame Event occurs.
41
42         By default, this object should have the value true for
43         Ethernet-like interfaces that support OAM. If the OAM layer
44         does not support Event Notifications (as indicated via the
45         dot3OamFunctionsSupported attribute), this value is ignored."
46
47
48     DEFVAL { true }
49     ::= { dot3OamEventConfigEntry 11 }
50
51
52     dot3OamErrFrameSecsSummaryWindow OBJECT-TYPE
53     SYNTAX      Integer32 (100..9000)
54     UNITS       "tenths of a second"
55     MAX-ACCESS  read-write
56     STATUS      current
57     DESCRIPTION
58         "The amount of time (in 100 ms intervals) over which the
59         threshold is defined. The default value is 100 (10 seconds).
60
61         If dot3OamErrFrameSecsSummaryThreshold frame errors occur
62         within a window of dot3OamErrFrameSecsSummaryWindow (in tenths
63         of seconds), an Event Notification OAMPDU should be generated
64         with an Errored Frame Seconds Summary Event TLV indicating
65
```

```
1         that the threshold has been crossed in this window."
2
3     REFERENCE    "IEEE Std 802.3, 30.3.6.1.40"
4     DEFVAL { 100 }
5     ::= { dot3OamEventConfigEntry 12 }
6
7 dot3OamErrFrameSecsSummaryThreshold OBJECT-TYPE
8     SYNTAX      Integer32 (1..900)
9     UNITS       "errored frame seconds"
10    MAX-ACCESS   read-write
11    STATUS       current
12    DESCRIPTION
13        "The minimum number of errored frame seconds that cause an Errored
14        Frame Seconds Summary Event. The default value is one errored frame
15        second. If the threshold value is zero, then an Event
16        Notification OAMPDU is sent periodically (at the end of every
17        window). This can be used as an asynchronous notification to
18        the peer OAM entity of the statistics related to this
19        threshold crossing alarm."
20
21        If dot3OamErrFrameSecsSummaryThreshold frame errors occur
22        within a window of dot3OamErrFrameSecsSummaryWindow (in tenths
23        of seconds), an Event Notification OAMPDU is generated
24        with an Errored Frame Seconds Summary Event TLV indicating
25        that the threshold has been crossed in this window."
26
27    REFERENCE    "IEEE Std 802.3, 30.3.6.1.40"
28    DEFVAL { 1 }
29    ::= { dot3OamEventConfigEntry 13 }
30
31 dot3OamErrFrameSecsEvNotifEnable OBJECT-TYPE
32     SYNTAX      TruthValue
33     MAX-ACCESS   read-write
34     STATUS       current
35     DESCRIPTION
36         "If true, the local OAM entity sends an Event Notification
37         OAMPDU when an Errored Frame Seconds Event occurs."
38
39         The default value for this object is true for
40         Ethernet-like interfaces that support OAM. If the OAM layer
41         does not support Event Notifications (as indicated via the
42         dot3OamFunctionsSupported attribute), this value is ignored."
43
44     DEFVAL { true }
45     ::= { dot3OamEventConfigEntry 14 }
46
47 dot3OamDyingGaspEnable OBJECT-TYPE
48     SYNTAX      TruthValue
49     MAX-ACCESS   read-write
50     STATUS       current
51     DESCRIPTION
52         "If true, the local OAM entity should attempt to indicate a
53         dying gasp via the OAMPDU flags field to its peer OAM entity
54         when a dying gasp event occurs. The exact definition of a
55         dying gasp event is implementation dependent. If the system
56         does not support dying gasp capability, setting this object
57         has no effect, and reading the object returns 'false'."
58
59         The default value for this object is true for
```

```
1      Ethernet-like interfaces that support OAM. If the OAM layer
2      does not support Event Notifications (as indicated via the
3      dot3OamFunctionsSupported attribute), this value is ignored."
4
5      DEFVAL { true }
6      ::= { dot3OamEventConfigEntry 15 }
7
8
9      dot3OamCriticalEventEnable OBJECT-TYPE
10     SYNTAX      TruthValue
11     MAX-ACCESS  read-write
12     STATUS      current
13     DESCRIPTION
14         "If true, the local OAM entity should attempt to indicate a
15         critical event via the OAMPDU flags to its peer OAM entity
16         when a critical event occurs. The exact definition of a
17         critical event is implementation dependent. If the system
18         does not support critical event capability, setting this
19         object has no effect, and reading the object should
20         result in 'false'."
21
22
23     By default, this object should have the value true for
24     Ethernet-like interfaces that support OAM. If the OAM layer
25     does not support Event Notifications (as indicated via the
26     dot3OamFunctionsSupported attribute), this value is ignored."
27
28
29     DEFVAL { true }
30     ::= { dot3OamEventConfigEntry 16 }
31
32     -- *****
33     --
34     -- Ethernet OAM Event Log group
35     --
36
37     dot3OamEventLogTable OBJECT-TYPE
38     SYNTAX      SEQUENCE OF Dot3OamEventLogEntry
39     MAX-ACCESS  not-accessible
40     STATUS      current
41     DESCRIPTION
42         "This table records a history of the events that have occurred
43         at the Ethernet OAM level. These events can include locally
44         detected events, which may result in locally generated
45         OAMPDUs, and remotely detected events, which are detected by
46         the OAM peer entity and signaled to the local entity via
47         Ethernet OAM. Ethernet OAM events can be signaled by Event
48         Notification OAMPDUs or by the flags field in any OAMPDU."
49
50
51
52     This table contains both threshold crossing events and
53     non-threshold crossing events. The parameters for the
54     threshold window, threshold value, and actual value
55     (dot3OamEventLogWindowXX, dot3OamEventLogThresholdXX,
56     dot3OamEventLogValue) are only applicable to threshold
57     crossing events, and are returned as all F's (2^32 - 1) for
58     non-threshold crossing events.
59
60
61     Entries in the table are automatically created when such
62     events are detected. The size of the table is implementation
63     dependent. When the table reaches its maximum size, older
64     entries are automatically deleted to make room for newer
65     entries."
```

```

1      ::= { dot3OamObjects 6 }
2
3
4      dot3OamEventLogEntry OBJECT-TYPE
5          SYNTAX      Dot3OamEventLogEntry
6          MAX-ACCESS   not-accessible
7          STATUS       current
8          DESCRIPTION
9              "An entry in the dot3OamEventLogTable. Entries are
10             automatically created whenever Ethernet OAM events occur at
11             the local OAM entity, and when Event Notification OAMPDUs are
12             received at the local OAM entity (indicating that events have
13             occurred at the peer OAM entity). The size of the table is
14             implementation dependent, but when the table becomes full,
15             older events are automatically deleted to make room for newer
16             events. The table index dot3OamEventLogIndex increments for
17             each new entry, and when the maximum value is reached, the
18             value restarts at zero."
19
20
21
22      INDEX          { ifIndex, dot3OamEventLogIndex }
23      ::= { dot3OamEventLogTable 1 }
24
25      Dot3OamEventLogEntry ::=
26      SEQUENCE {
27          dot3OamEventLogIndex      Unsigned32,
28          dot3OamEventLogTimestamp  TimeStamp,
29          dot3OamEventLogOui        EightOTwoOui,
30          dot3OamEventLogType       Unsigned32,
31          dot3OamEventLogLocation   INTEGER,
32          dot3OamEventLogWindowHi   Unsigned32,
33          dot3OamEventLogWindowLo   Unsigned32,
34          dot3OamEventLogThresholdHi Unsigned32,
35          dot3OamEventLogThresholdLo Unsigned32,
36          dot3OamEventLogValue       CounterBasedGauge64,
37          dot3OamEventLogRunningTotal CounterBasedGauge64,
38          dot3OamEventLogEventTotal  Unsigned32
39      }
40
41
42
43      dot3OamEventLogIndex OBJECT-TYPE
44          SYNTAX      Unsigned32(1..4294967295)
45          MAX-ACCESS   not-accessible
46          STATUS       current
47          DESCRIPTION
48              "An arbitrary integer for identifying individual events
49              within the event log."
50      ::= { dot3OamEventLogEntry 1 }
51
52
53      dot3OamEventLogTimestamp OBJECT-TYPE
54          SYNTAX      TimeStamp
55          MAX-ACCESS   read-only
56          STATUS       current
57          DESCRIPTION
58              "The value of sysUpTime at the time of the logged event. For
59              locally generated events, the time of the event can be
60              accurately retrieved from sysUpTime. For remotely generated
61              events, the time of the event is indicated by the reception of
62              the Event Notification OAMPDU indicating that the event
63              occurred on the peer. A system may attempt to adjust the
64              timestamp value to more accurately reflect the time of the
65

```

```
1         event at the peer OAM entity by using other information, such
2         as that found in the timestamp found of the Event Notification
3         TLVs, which provides an indication of the relative time
4         between events at the peer entity."
5         ::= { dot3OamEventLogEntry 2 }
6
7     dot3OamEventLogOui OBJECT-TYPE
8     SYNTAX      EightOTwoOui
9     MAX-ACCESS  read-only
10    STATUS      current
11    DESCRIPTION
12        "The OUI of the entity defining the object type. All IEEE
13        802.3 defined events (as appearing in IEEE Std 802.3 except for the
14        Organizationally Unique Event TLVs) use the IEEE 802.3 OUI of
15        0x0180C2. Organizations defining their own Event Notification
16        TLVs include their OUI in the Event Notification TLV that
17        gets reflected here."
18        ::= { dot3OamEventLogEntry 3 }
19
20    dot3OamEventLogType OBJECT-TYPE
21    SYNTAX      Unsigned32
22    MAX-ACCESS  read-only
23    STATUS      current
24    DESCRIPTION
25        "The type of event that generated this entry in the event log.
26        When the OUI is the IEEE 802.3 OUI of 0x0180C2, the following
27        event types are defined:
28            erroredSymbolEvent(1),
29            erroredFramePeriodEvent(2),
30            erroredFrameEvent(3),
31            erroredFrameSecondsEvent(4),
32            linkFault(256),
33            dyingGaspEvent(257),
34            criticalLinkEvent(258)
35        The first four are considered threshold crossing events, as
36        they are generated when a metric exceeds a given value within
37        a specified window. The other three are not threshold
38        crossing events.
39
40        When the OUI is not 71874 (0x0180C2 in hex), then some other
41        organization has defined the event space. If event subtyping
42        is known to the implementation, it may be reflected here.
43        Otherwise, this value should return all F's (2^32 - 1)."
```

```
44    REFERENCE  "IEEE Std 802.3, 30.3.6.1.10 and 57.5.3."
45    ::= { dot3OamEventLogEntry 4 }
46
47    dot3OamEventLogLocation OBJECT-TYPE
48    SYNTAX      INTEGER { local(1), remote(2) }
49    MAX-ACCESS  read-only
50    STATUS      current
51    DESCRIPTION
52        "Whether this event occurred locally (local(1)), or was
53        received from the OAM peer via Ethernet OAM (remote(2))."
54
55    ::= { dot3OamEventLogEntry 5 }
56
57    dot3OamEventLogWindowHi OBJECT-TYPE
58    SYNTAX      Unsigned32
```



```
1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "If the event represents a threshold crossing event, the two
5          objects dot3OamEventWindowHi and dot3OamEventWindowLo, form
6          an unsigned 64-bit integer yielding the window over which the
7          value was measured for the threshold crossing event (for
8          example, 5, when 11 occurrences happened in 5 seconds while
9          the threshold was 10). The two objects are combined as:
10         dot3OamEventLogWindow = ((2^32) * dot3OamEventLogWindowHi)
11                                     + dot3OamEventLogWindowLo
12
13         Otherwise, this value is returned as all F's (2^32 - 1) and
14         adds no useful information."
15
16
17     REFERENCE  "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
18     ::= { dot3OamEventLogEntry 6 }
19
20
21 dot3OamEventLogWindowLo      OBJECT-TYPE
22     SYNTAX      Unsigned32
23     MAX-ACCESS  read-only
24     STATUS      current
25     DESCRIPTION
26         "If the event represents a threshold crossing event, the two
27         objects dot3OamEventWindowHi and dot3OamEventWindowLo form an
28         unsigned 64-bit integer yielding the window over which the
29         value was measured for the threshold crossing event (for
30         example, 5, when 11 occurrences happened in 5 seconds while
31         the threshold was 10). The two objects are combined as:
32
33         dot3OamEventLogWindow = ((2^32) * dot3OamEventLogWindowHi)
34                                     + dot3OamEventLogWindowLo
35
36         Otherwise, this value is returned as all F's (2^32 - 1) and
37         adds no useful information."
38
39
40     REFERENCE  "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
41     ::= { dot3OamEventLogEntry 7 }
42
43
44 dot3OamEventLogThresholdHi   OBJECT-TYPE
45     SYNTAX      Unsigned32
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49         "If the event represents a threshold crossing event, the two
50         objects dot3OamEventThresholdHi and dot3OamEventThresholdLo
51         form an unsigned 64-bit integer yielding the value that was
52         crossed for the threshold crossing event (for example, 10,
53         when 11 occurrences happened in 5 seconds while the threshold
54         was 10). The two objects are combined as:
55
56         dot3OamEventLogThreshold = ((2^32) * dot3OamEventLogThresholdHi)
57                                     + dot3OamEventLogThresholdLo
58
59         Otherwise, this value is returned as all F's (2^32 -1) and
60         adds no useful information."
61
62
63     REFERENCE  "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
64     ::= { dot3OamEventLogEntry 8 }
```

dot3OamEventLogThresholdLo OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"If the event represents a threshold crossing event, the two
objects dot3OamEventThresholdHi and dot3OamEventThresholdLo
form an unsigned 64-bit integer yielding the value that was
crossed for the threshold crossing event (for example, 10,
when 11 occurrences happened in 5 seconds while the threshold
was 10). The two objects are combined as:
$$\text{dot3OamEventLogThreshold} = ((2^{32}) * \text{dot3OamEventLogThresholdHi}) + \text{dot3OamEventLogThresholdLo}$$

Otherwise, this value is returned as all F's ($2^{32} - 1$) and
adds no useful information."
REFERENCE "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
 ::= { dot3OamEventLogEntry 9 }
dot3OamEventLogValue OBJECT-TYPE
SYNTAX CounterBasedGauge64
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"If the event represents a threshold crossing event, this
value indicates the value of the parameter within the given
window that generated this event (for example, 11, when 11
occurrences happened in 5 seconds while the threshold was 10).
Otherwise, this value is returned as all F's
($2^{64} - 1$) and adds no useful information."
REFERENCE "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
 ::= { dot3OamEventLogEntry 10 }
dot3OamEventLogRunningTotal OBJECT-TYPE
SYNTAX CounterBasedGauge64
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Each Event Notification TLV contains a running total of the
number of times an event has occurred, as well as the number
of times an Event Notification for the event has been
transmitted. For non-threshold crossing events, the number of
events (dot3OamLogRunningTotal) and the number of resultant
Event Notifications (dot3OamLogEventTotal) should be
identical.
For threshold crossing events, since multiple occurrences may
be required to cross the threshold, these values are likely
different. This value represents the total number of times
this event has happened since the last reset (for example,
3253, when 3253 symbol errors have occurred since the last
reset, which has resulted in 51 symbol error threshold
crossing events since the last reset)."

```
1      REFERENCE    "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
2      ::= { dot3OamEventLogEntry 11 }
3
4      dot3OamEventLogEventTotal      OBJECT-TYPE
5      SYNTAX      Unsigned32
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9
10     "Each Event Notification TLV contains a running total of the
11     number of times an event has occurred, as well as the number
12     of times an Event Notification for the event has been
13     transmitted. For non-threshold crossing events, the number of
14     events (dot3OamLogRunningTotal) and the number of resultant
15     Event Notifications (dot3OamLogEventTotal) should be
16     identical.
17
18     For threshold crossing events, since multiple occurrences may
19     be required to cross the threshold, these values are likely
20     different. This value represents the total number of times
21     one or more of these occurrences have resulted in an Event
22     Notification (for example, 51 when 3253 symbol errors have
23     occurred since the last reset, which has resulted in 51 symbol
24     error threshold crossing events since the last reset)."
```

```
27     REFERENCE    "IEEE Std 802.3, 30.3.6.1.37 and 57.5.3.2."
28     ::= { dot3OamEventLogEntry 12 }
29
30     -- *****
31     --
32     -- Ethernet OAM Notifications
33     --
34
35
36     dot3OamThresholdEvent NOTIFICATION-TYPE
37     OBJECTS { dot3OamEventLogTimestamp,
38               dot3OamEventLogOui,
39               dot3OamEventLogType,
40               dot3OamEventLogLocation,
41               dot3OamEventLogWindowHi,
42               dot3OamEventLogWindowLo,
43               dot3OamEventLogThresholdHi,
44               dot3OamEventLogThresholdLo,
45               dot3OamEventLogValue,
46               dot3OamEventLogRunningTotal,
47               dot3OamEventLogEventTotal
48             }
49     STATUS      current
50     DESCRIPTION
51
52     "A dot3OamThresholdEvent notification is sent when a local or
53     remote threshold crossing event is detected. A local
54     threshold crossing event is detected by the local entity,
55     while a remote threshold crossing event is detected by the
56     reception of an Ethernet OAM Event Notification OAMPDU
57     that indicates a threshold event.
58
59     This notification should not be sent more than once per
60     second.
61
62     The OAM entity can be derived from extracting the ifIndex from
63     the variable bindings. The objects in the notification
```

```
1      correspond to the values in a row instance in the
2      dot3OamEventLogTable.
3
4      The management entity should periodically check
5      dot3OamEventLogTable to detect any missed events."
6      ::= { dot3OamNotifications 1 }
7
8
9      dot3OamNonThresholdEvent NOTIFICATION-TYPE
10     OBJECTS { dot3OamEventLogTimestamp,
11               dot3OamEventLogOui,
12               dot3OamEventLogType,
13               dot3OamEventLogLocation,
14               dot3OamEventLogEventTotal
15             }
16     STATUS current
17     DESCRIPTION
18         "A dot3OamNonThresholdEvent notification is sent when a local
19         or remote non-threshold crossing event is detected. A local
20         event is detected by the local entity, while a remote event is
21         detected by the reception of an Ethernet OAM Event
22         Notification OAMPDU that indicates a non-threshold crossing
23         event.
24
25         This notification should not be sent more than once per
26         second.
27
28         The OAM entity can be derived from extracting the ifIndex from
29         the variable bindings. The objects in the notification
30         correspond to the values in a row instance of the
31         dot3OamEventLogTable.
32
33         The management entity should periodically check
34         dot3OamEventLogTable to detect any missed events."
35         ::= { dot3OamNotifications 2 }
36
37
38     -- *****
39     --
40     -- Conformance statements
41     --
42
43     dot3OamGroups OBJECT IDENTIFIER ::= { dot3OamConformance 1 }
44     dot3OamCompliances OBJECT IDENTIFIER ::= { dot3OamConformance 2 }
45
46     -- Compliance statements
47
48     dot3OamCompliance MODULE-COMPLIANCE
49     STATUS current
50     DESCRIPTION "The compliance statement for managed entities
51                 supporting OAM on Ethernet-like interfaces."
52
53     MODULE -- this module
54     MANDATORY-GROUPS { dot3OamControlGroup,
55                       dot3OamPeerGroup,
56                       dot3OamStatsBaseGroup
57                     }
58
59     GROUP dot3OamLoopbackGroup
60     DESCRIPTION
61         "This group is mandatory for all IEEE 802.3 OAM
```

```
1         implementations that support loopback functionality."
2
3     GROUP          dot3OamErrSymbolPeriodEventGroup
4     DESCRIPTION
5         "This group is mandatory for all IEEE 802.3 OAM
6         implementations that support event functionality."
7
8     GROUP          dot3OamErrFramePeriodEventGroup
9     DESCRIPTION
10        "This group is mandatory for all IEEE 802.3 OAM
11        implementations that support event functionality."
12
13
14     GROUP          dot3OamErrFrameEventGroup
15     DESCRIPTION
16        "This group is mandatory for all IEEE 802.3 OAM
17        implementations that support event functionality."
18
19
20     GROUP          dot3OamErrFrameSecsSummaryEventGroup
21     DESCRIPTION
22        "This group is mandatory for all IEEE 802.3 OAM
23        implementations that support event functionality."
24
25
26     GROUP          dot3OamFlagEventGroup
27     DESCRIPTION
28        "This group is optional for all IEEE 802.3 OAM
29        implementations. The ability to send critical events or dying
30        gasp events is not required in any system."
31
32     GROUP          dot3OamEventLogGroup
33     DESCRIPTION
34        "This group is optional for all IEEE 802.3 OAM
35        implementations. Entries in this table are dependent on what
36        event functionality is supported in the local OAM
37        implementation. At least one type of event shall be supported
38        for entries to appear in this table."
39
40
41     GROUP          dot3OamNotificationGroup
42     DESCRIPTION
43        "This group is optional for all IEEE 802.3 OAM
44        implementations. Since the information in the notifications
45        is dependent on the dot3OamEventLogTable, that table shall be
46        implemented for notifications."
47
48     ::= { dot3OamCompliances 1 }
49
50 dot3OamControlGroup OBJECT-GROUP
51     OBJECTS          {
52         dot3OamAdminState,
53         dot3OamOperStatus,
54         dot3OamMode,
55         dot3OamMaxOamPduSize,
56         dot3OamConfigRevision,
57         dot3OamFunctionsSupported
58     }
59     STATUS            current
60     DESCRIPTION
61         "A collection of objects providing the abilities,
62         configuration, and status of an Ethernet OAM entity."
63     ::= { dot3OamGroups 1 }
64
65
```

```
1      dot3OamPeerGroup OBJECT-GROUP
2          OBJECTS      {      dot3OamPeerMacAddress,
3                              dot3OamPeerVendorOui,
4                              dot3OamPeerVendorInfo,
5                              dot3OamPeerMode,
6                              dot3OamPeerFunctionsSupported,
7                              dot3OamPeerMaxOamPduSize,
8                              dot3OamPeerConfigRevision
9                              }
10
11      STATUS          current
12      DESCRIPTION
13          "A collection of objects providing the abilities,
14          configuration, and status of a peer Ethernet OAM entity."
15      ::= { dot3OamGroups 2 }
16
17      dot3OamStatsBaseGroup OBJECT-GROUP
18          OBJECTS      {      dot3OamInformationTx,
19                              dot3OamInformationRx,
20                              dot3OamUniqueEventNotificationTx,
21                              dot3OamUniqueEventNotificationRx,
22                              dot3OamDuplicateEventNotificationTx,
23                              dot3OamDuplicateEventNotificationRx,
24                              dot3OamLoopbackControlTx,
25                              dot3OamLoopbackControlRx,
26                              dot3OamVariableRequestTx,
27                              dot3OamVariableRequestRx,
28                              dot3OamVariableResponseTx,
29                              dot3OamVariableResponseRx,
30                              dot3OamOrgSpecificTx,
31                              dot3OamOrgSpecificRx,
32                              dot3OamUnsupportedCodesTx,
33                              dot3OamUnsupportedCodesRx,
34                              dot3OamFramesLostDueToOam
35                              }
36
37      STATUS          current
38      DESCRIPTION
39          "A collection of objects providing the statistics for the
40          number of various transmit and receive events for OAM on an
41          Ethernet-like interface. Note that all of these counters shall
42          be supported even if the related function (as described in
43          dot3OamFunctionsSupported) is not supported."
44      ::= { dot3OamGroups 3 }
45
46      dot3OamLoopbackGroup OBJECT-GROUP
47          OBJECTS      {      dot3OamLoopbackStatus,
48                              dot3OamLoopbackIgnoreRx
49                              }
50
51      STATUS          current
52      DESCRIPTION
53          "A collection of objects for controlling the OAM remote
54          loopback function."
55      ::= { dot3OamGroups 4 }
56
57      dot3OamErrSymbolPeriodEventGroup OBJECT-GROUP
58          OBJECTS      {      dot3OamErrSymPeriodWindowHi,
59                              dot3OamErrSymPeriodWindowLo,
60                              dot3OamErrSymPeriodThresholdHi,
61                              dot3OamErrSymPeriodThresholdLo,
62                              dot3OamErrSymPeriodEvNotifEnable
63                              }
```

```
1         }
2     STATUS      current
3     DESCRIPTION
4         "A collection of objects for configuring the thresholds for an
5         Errored Symbol Period Event.
6
7         Each IEEE Std 802.3 defined Event Notification TLV has its own
8         conformance group because each event can be implemented
9         independently of any other."
10    ::= { dot3OamGroups 5 }
11
12
13    dot3OamErrFramePeriodEventGroup OBJECT-GROUP
14    OBJECTS      {      dot3OamErrFramePeriodWindow,
15                        dot3OamErrFramePeriodThreshold,
16                        dot3OamErrFramePeriodEvNotifEnable
17                    }
18    STATUS      current
19    DESCRIPTION
20        "A collection of objects for configuring the thresholds for an
21        Errored Frame Period Event.
22
23        Each IEEE Std 802.3 defined Event Notification TLV has its own
24        conformance group because each event can be implemented
25        independently of any other."
26    ::= { dot3OamGroups 6 }
27
28
29
30    dot3OamErrFrameEventGroup OBJECT-GROUP
31    OBJECTS      {      dot3OamErrFrameWindow,
32                        dot3OamErrFrameThreshold,
33                        dot3OamErrFrameEvNotifEnable
34                    }
35    STATUS      current
36    DESCRIPTION
37        "A collection of objects for configuring the thresholds for an
38        Errored Frame Event.
39
40        Each IEEE Std 802.3 defined Event Notification TLV has its own
41        conformance group because each event can be implemented
42        independently of any other."
43    ::= { dot3OamGroups 7 }
44
45
46
47    dot3OamErrFrameSecsSummaryEventGroup OBJECT-GROUP
48    OBJECTS      {      dot3OamErrFrameSecsSummaryWindow,
49                        dot3OamErrFrameSecsSummaryThreshold,
50                        dot3OamErrFrameSecsEvNotifEnable
51                    }
52    STATUS      current
53    DESCRIPTION
54        "A collection of objects for configuring the thresholds for an
55        Errored Frame Seconds Summary Event.
56
57        Each IEEE Std 802.3 defined Event Notification TLV has its own
58        conformance group because each event can be implemented
59        independently of any other."
60    ::= { dot3OamGroups 8 }
61
62
63
64    dot3OamFlagEventGroup OBJECT-GROUP
65    OBJECTS      {      dot3OamDyingGaspEnable,
66                        dot3OamCriticalEventEnable
```

```
1          }
2      STATUS      current
3      DESCRIPTION
4          "A collection of objects for configuring the sending OAMPDUs
5          with the critical event flag or dying gasp flag enabled."
6      ::= { dot3OamGroups 9 }
7
8
9      dot3OamEventLogGroup OBJECT-GROUP
10     OBJECTS { dot3OamEventLogTimestamp,
11               dot3OamEventLogOui,
12               dot3OamEventLogType,
13               dot3OamEventLogLocation,
14               dot3OamEventLogWindowHi,
15               dot3OamEventLogWindowLo,
16               dot3OamEventLogThresholdHi,
17               dot3OamEventLogThresholdLo,
18               dot3OamEventLogValue,
19               dot3OamEventLogRunningTotal,
20               dot3OamEventLogEventTotal
21           }
22
23     STATUS      current
24     DESCRIPTION
25         "A collection of objects for configuring the thresholds for an
26         Errored Frame Seconds Summary Event and maintaining the event
27         information."
28     ::= { dot3OamGroups 10 }
29
30
31     dot3OamNotificationGroup NOTIFICATION-GROUP
32     NOTIFICATIONS {
33         dot3OamThresholdEvent,
34         dot3OamNonThresholdEvent
35     }
36
37     STATUS      current
38     DESCRIPTION
39         "A collection of notifications used by Ethernet OAM to signal
40         to a management entity that local or remote events have
41         occurred on a specified Ethernet link."
42     ::= { dot3OamGroups 11 }
43
44
45     END
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

7. Ethernet repeater device MIB module

7.1 Overview

This clause defines a portion of the MIB for use with SNMP. In particular, it defines objects for managing IEEE 802.3 repeaters.

7.1.1 Repeater management

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

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

7.1.2 Structure of the MIB

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

7.1.2.1 Basic definitions

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

7.1.2.2 Monitor definitions

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

7.1.2.3 Address tracking definitions

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

7.1.2.4 Top N definitions

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

7.1.3 Relationship to MIB-II

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

7.1.3.1 Relationship to the “system” group

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

7.1.3.2 Relationship to the “interfaces” group

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

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

NOTE—When a network management entity is observing a repeater, it may appear as though the repeater is passing packets to a higher level protocol entity. However, this is only a means of implementing management, and this passing of management information is not part of the repeater functionality.¹³

7.2 Topology mapping

Network topology mapping is described in section 4 of IETF RFC 2108 [B20].

7.3 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL¹⁴:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C7mib.txt

¹³Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

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

```
1      IEEE8023-SNMP-REPEATER-MIB DEFINITIONS ::= BEGIN
2
3      IMPORTS
4          Counter32, Counter64, Integer32, Gauge32,
5          OBJECT-TYPE, MODULE-IDENTITY, NOTIFICATION-TYPE, org
6              FROM SNMPv2-SMI
7          TimeStamp, MacAddress, TEXTUAL-CONVENTION,
8          RowStatus, TestAndIncr
9              FROM SNMPv2-TC
10         OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
11             FROM SNMPv2-CONF
12         OwnerString
13             FROM RFC1271-MIB;
14
15
16
17     ieee8023snmpRptrMIB MODULE-IDENTITY
18         LAST-UPDATED "201304110000Z" -- April 11, 2013
19         ORGANIZATION
20             "IEEE 802.3 working group"
21         CONTACT-INFO
22             "WG-URL: http://www.ieee802.org/3/index.html
23             WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
24
25             Contact: Howard Frazier
26             Postal: 3151 Zanker Road
27                 San Jose, CA 95134
28                 USA
29             Tel: +1.408.922.8164
30             E-mail: hfrazier@broadcom.com"
31
32     DESCRIPTION
33         "Management information for IEEE 802.3 repeaters."
34
35
36     REVISION "201304110000Z" -- April 11, 2013
37     DESCRIPTION
38         "Revision, based on an earlier version in IEEE Std 802.3.1-2011."
39
40
41     REVISION "201102020000Z" -- February 2, 2011
42     DESCRIPTION
43         "Initial revision, based on an earlier version in RFC 2108"
44
45
46     ::= { org ieee(111) standards-association-numbers-series-standards(2)
47         lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 7 }
48
49     ieee8023snmpDot3RptrMgt OBJECT IDENTIFIER ::= { ieee8023snmpRptrMIB 1}
50
51
52     OptMacAddr ::= TEXTUAL-CONVENTION
53         DISPLAY-HINT "1x:"
54         STATUS current
55         DESCRIPTION
56             "Either a 6 octet address in the 'canonical'
57             order defined by IEEE Std 802.1a, i.e., as if it
58             were transmitted least significant bit first
59             if a value is available or a zero length string."
60         REFERENCE
61             "See MacAddress in SNMPv2-TC. The only difference
62             is that a zero length string is allowed as a value
63             for OptMacAddr and not for MacAddress."
64         SYNTAX OCTET STRING (SIZE (0 | 6))
65
```

```

1
2
3
4  -- Basic information at the repeater, group, and port level.
5
6  rpPtrBasicPackage
7      OBJECT IDENTIFIER ::= { ieee8023snmpDot3RptrMgt 1 }
8      rpPtrGroupInfo
9          OBJECT IDENTIFIER ::= { rpPtrBasicPackage 1 }
10     rpPtrPortInfo
11         OBJECT IDENTIFIER ::= { rpPtrBasicPackage 2 }
12     rpPtrAllRptrInfo
13         OBJECT IDENTIFIER ::= { rpPtrBasicPackage 3 }
14
15
16  -- Monitoring information at the repeater, group, and port level.
17  rpPtrMonitorPackage
18      OBJECT IDENTIFIER ::= { ieee8023snmpDot3RptrMgt 2 }
19      rpPtrMonitorRptrInfo
20          OBJECT IDENTIFIER ::= { rpPtrMonitorPackage 1 }
21      rpPtrMonitorGroupInfo
22          OBJECT IDENTIFIER ::= { rpPtrMonitorPackage 2 }
23      rpPtrMonitorPortInfo
24          OBJECT IDENTIFIER ::= { rpPtrMonitorPackage 3 }
25      rpPtrMonitorAllRptrInfo
26          OBJECT IDENTIFIER ::= { rpPtrMonitorPackage 4 }
27
28
29  -- Address tracking information at the repeater, group,
30  -- and port level.
31
32  rpPtrAddrTrackPackage
33      OBJECT IDENTIFIER ::= { ieee8023snmpDot3RptrMgt 3 }
34      rpPtrAddrTrackRptrInfo
35          OBJECT IDENTIFIER ::= { rpPtrAddrTrackPackage 1 }
36      rpPtrAddrTrackGroupInfo
37          -- this subtree is currently unused
38          OBJECT IDENTIFIER ::= { rpPtrAddrTrackPackage 2 }
39      rpPtrAddrTrackPortInfo
40          OBJECT IDENTIFIER ::= { rpPtrAddrTrackPackage 3 }
41
42
43  -- TopN information.
44  rpPtrTopNPackage
45      OBJECT IDENTIFIER ::= { ieee8023snmpDot3RptrMgt 4 }
46      rpPtrTopNRptrInfo
47          -- this subtree is currently unused
48          OBJECT IDENTIFIER ::= { rpPtrTopNPackage 1 }
49      rpPtrTopNGroupInfo
50          -- this subtree is currently unused
51          OBJECT IDENTIFIER ::= { rpPtrTopNPackage 2 }
52      rpPtrTopNPortInfo
53          OBJECT IDENTIFIER ::= { rpPtrTopNPackage 3 }
54
55
56  -- Basic information at the group level.
57  --
58  -- Configuration and status objects for each
59  -- managed group in the repeater system, independent
60  -- of whether there is one or more managed
61  -- repeater-units in the repeater system.
62
63
64  rpPtrGroupTable OBJECT-TYPE
65      SYNTAX          SEQUENCE OF RptrGroupEntry

```

```
1      MAX-ACCESS    not-accessible
2      STATUS        current
3      DESCRIPTION
4          "Table of descriptive and status information about
5          the groups of ports."
6      ::= { rpPtrGroupInfo 1 }
7
8
9      rpPtrGroupEntry OBJECT-TYPE
10     SYNTAX          RpPtrGroupEntry
11     MAX-ACCESS      not-accessible
12     STATUS          current
13     DESCRIPTION
14         "An entry in the table, containing information
15         about a single group of ports."
16     INDEX            { rpPtrGroupIndex }
17     ::= { rpPtrGroupTable 1 }
18
19
20     RpPtrGroupEntry ::=
21     SEQUENCE {
22         rpPtrGroupIndex
23             Integer32,
24         rpPtrGroupObjectID
25             OBJECT IDENTIFIER,
26         rpPtrGroupOperStatus
27             INTEGER,
28         rpPtrGroupPortCapacity
29             Integer32
30     }
31
32
33     rpPtrGroupIndex OBJECT-TYPE
34     SYNTAX          Integer32 (1..2147483647)
35     MAX-ACCESS      not-accessible
36     STATUS          current
37     DESCRIPTION
38         "This object identifies the group within the
39         repeater system for which this entry contains
40         information."
41     REFERENCE
42         "IEEE Std 802.3, 30.4.2.1.1, aGroupID."
43     ::= { rpPtrGroupEntry 1 }
44
45
46     rpPtrGroupObjectID OBJECT-TYPE
47     SYNTAX          OBJECT IDENTIFIER
48     MAX-ACCESS      read-only
49     STATUS          current
50     DESCRIPTION
51         "The vendor's authoritative identification of the
52         group. This value may be allocated within the SMI
53         enterprises subtree (1.3.6.1.4.1) and provides a
54         straight-forward and unambiguous means for
55         determining what kind of group is being managed.
56
57         For example, this object could take the value
58         1.3.6.1.4.1.4242.1.2.14 if vendor 'Flintstones,
59         Inc.' was assigned the subtree 1.3.6.1.4.1.4242,
60         and had assigned the identifier
61         1.3.6.1.4.1.4242.1.2.14 to its 'Wilma Flintstone
62         6-Port FOIRL Plug-in module.'"
63     ::= { rpPtrGroupEntry 2 }
```

```
1
2  rptrGroupOperStatus OBJECT-TYPE
3      SYNTAX      INTEGER {
4          other(1),
5          operational(2),
6          malfunctioning(3),
7          notPresent(4),
8          underTest(5),
9          resetInProgress(6)
10         }
11
12  MAX-ACCESS  read-only
13  STATUS      current
14  DESCRIPTION
15      "An object that indicates the operational status
16      of the group.
17
18      A status of notPresent(4) indicates that the group
19      is temporarily or permanently physically and/or
20      logically not a part of the repeater. It is an
21      implementation-specific matter as to whether the
22      agent effectively removes notPresent entries from
23      the table.
24
25      A status of operational(2) indicates that the
26      group is functioning, and a status of
27      malfunctioning(3) indicates that the group is
28      malfunctioning in some way."
29
30  ::= { rptrGroupEntry 3 }
31
32
33  rptrGroupPortCapacity OBJECT-TYPE
34      SYNTAX      Integer32 (1..2147483647)
35      MAX-ACCESS  read-only
36      STATUS      current
37      DESCRIPTION
38          "The rptrGroupPortCapacity is the number of ports
39          that can be contained within the group. Valid
40          range is 1-2147483647. Within each group, the
41          ports are uniquely numbered in the range from 1 to
42          rptrGroupPortCapacity.
43
44          Some ports may not be present in the repeater system, in
45          which case the actual number of ports present
46          will be less than the value of rptrGroupPortCapacity.
47          The number of ports present in the group will never
48          be greater than the value of rptrGroupPortCapacity.
49
50          Note: In practice, this will generally be the
51          number of ports on a module, card, or board, and
52          the port numbers will correspond to numbers marked
53          on the physical embodiment."
54
55  REFERENCE
56      "IEEE Std 802.3, 30.4.2.1.2, aGroupPortCapacity."
57
58  ::= { rptrGroupEntry 4 }
59
60
61  -- Basic information at the port level.
62  --
63  -- Configuration and status objects for
64  -- each managed repeater port in the repeater system,
```

```
1      -- independent of whether there is one or more
2      -- managed repeater-units in the repeater system.
3
4      rpPtrPortTable OBJECT-TYPE
5          SYNTAX      SEQUENCE OF RptrPortEntry
6          MAX-ACCESS   not-accessible
7          STATUS       current
8          DESCRIPTION
9              "Table of descriptive and status information about
10             the repeater ports in the repeater system. The number of
11             entries is independent of the number of repeaters
12             in the managed repeater system."
13             ::= { rpPtrPortInfo 1 }
14
15
16      rpPtrPortEntry OBJECT-TYPE
17          SYNTAX      RptrPortEntry
18          MAX-ACCESS   not-accessible
19          STATUS       current
20          DESCRIPTION
21              "An entry in the table, containing information
22             about a single port."
23
24
25          INDEX      { rpPtrPortGroupIndex, rpPtrPortIndex }
26          ::= { rpPtrPortTable 1 }
27
28
29      RptrPortEntry ::=
30          SEQUENCE {
31              rpPtrPortGroupIndex
32                  Integer32,
33              rpPtrPortIndex
34                  Integer32,
35              rpPtrPortAdminStatus
36                  INTEGER,
37              rpPtrPortAutoPartitionState
38                  INTEGER,
39              rpPtrPortOperStatus
40                  INTEGER,
41              rpPtrPortRptrId
42                  Integer32
43          }
44
45
46      rpPtrPortGroupIndex OBJECT-TYPE
47          SYNTAX      Integer32 (1..2147483647)
48          MAX-ACCESS   not-accessible
49          STATUS       current
50          DESCRIPTION
51              "This object identifies the group containing the
52             port for which this entry contains information."
53             ::= { rpPtrPortEntry 1 }
54
55
56      rpPtrPortIndex OBJECT-TYPE
57          SYNTAX      Integer32 (1..2147483647)
58          MAX-ACCESS   not-accessible
59          STATUS       current
60          DESCRIPTION
61              "This object identifies the port within the group
62             for which this entry contains information. This
63             identifies the port independently from the repeater
64             to which it may be attached. The numbering scheme for
65
```



```
1           ports is implementation specific; however, this
2           value can never be greater than
3           rpPtrGroupPortCapacity for the associated group."
4
5   REFERENCE
6       "IEEE Std 802.3, 30.4.3.1.1, aPortID."
7   ::= { rpPtrPortEntry 2 }
8
9   rpPtrPortAdminStatus OBJECT-TYPE
10      SYNTAX      INTEGER {
11                  enabled(1),
12                  disabled(2)
13              }
14      MAX-ACCESS   read-write
15      STATUS       current
16      DESCRIPTION
17          "Setting this object to disabled(2) disables the
18          port. A disabled port neither transmits nor
19          receives. Once disabled, a port shall be
20          explicitly enabled to restore operation. A port
21          that is disabled when power is lost or when a
22          reset is exerted shall remain disabled when normal
23          operation resumes.
24
25          The admin status takes precedence over auto-
26          partition and functionally operates between the
27          auto-partition mechanism and the AUI/PMA.
28
29          Setting this object to enabled(1) enables the port
30          and exerts a BEGIN on the port's auto-partition
31          state machine.
32
33          (In effect, when a port is disabled, the value of
34          rpPtrPortAutoPartitionState for that port is frozen
35          until the port is next enabled. When the port
36          becomes enabled, the rpPtrPortAutoPartitionState
37          becomes notAutoPartitioned(1), regardless of its
38          pre-disabling state.)"
39
40      REFERENCE
41          "IEEE Std 802.3, 30.4.3.1.2, aPortAdminState
42          and 30.4.3.2.1, acPortAdminControl."
43      ::= { rpPtrPortEntry 3 }
44
45   rpPtrPortAutoPartitionState OBJECT-TYPE
46      SYNTAX      INTEGER {
47                  notAutoPartitioned(1),
48                  autoPartitioned(2)
49              }
50      MAX-ACCESS   read-only
51      STATUS       current
52      DESCRIPTION
53          "The autoPartitionState flag indicates whether the
54          port is currently partitioned by the repeater's
55          auto-partition protection.
56
57          The conditions that cause port partitioning are
58          specified in partition state machine in Clauses
59          9 and 27 of IEEE Std 802.3. They are not
60          differentiated here."
61
62      REFERENCE
```

```
1         "IEEE Std 802.3, 30.4.3.1.3, aAutoPartitionState."
2     ::= { rpPtrPortEntry 4 }
3
4     rpPtrPortOperStatus OBJECT-TYPE
5         SYNTAX      INTEGER {
6             operational(1),
7             notOperational(2),
8             notPresent(3)
9         }
10
11     MAX-ACCESS read-only
12     STATUS current
13     DESCRIPTION
14         "This object indicates the port's operational
15         status. The notPresent(3) status indicates the
16         port is physically removed (note this may or may
17         not be possible depending on the type of port.)
18         The operational(1) status indicates that the port
19         is enabled (see rpPtrPortAdminStatus) and working,
20         even though it might be auto-partitioned (see
21         rpPtrPortAutoPartitionState).
22
23         If this object has the value operational(1) and
24         rpPtrPortAdminStatus is set to disabled(2), it is
25         expected that this object's value will soon change
26         to notOperational(2)."
```

```
28     ::= { rpPtrPortEntry 5 }
29
30     rpPtrPortRPtrId OBJECT-TYPE
31         SYNTAX      Integer32 (0..2147483647)
32         MAX-ACCESS read-only
33         STATUS current
34         DESCRIPTION
35             "This object identifies the repeater to
36             which this port belongs. The repeater
37             identified by a particular value of this object
38             is the same as that identified by the same
39             value of rpPtrInfoId. A value of zero
40             indicates that this port currently is not
41             a member of any repeater."
42
43     ::= { rpPtrPortEntry 6 }
44
45
46
47     -- New version of basic information at the repeater level.
48     --
49     -- Configuration, status, and control objects for
50     -- each managed repeater in the repeater system.
51
52     rpPtrInfoTable OBJECT-TYPE
53         SYNTAX      SEQUENCE OF RPtrInfoEntry
54         MAX-ACCESS not-accessible
55         STATUS current
56         DESCRIPTION
57             "A table of information about each
58             non-trivial repeater. The number of entries
59             depends on the physical configuration of the
60             managed repeater system."
61
62     ::= { rpPtrAllRPtrInfo 1 }
63
64     rpPtrInfoEntry OBJECT-TYPE
```

```
1      SYNTAX      RptrInfoEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "An entry in the table, containing information
6          about a single non-trivial repeater."
7      INDEX      { rptrInfoId }
8      ::= { rptrInfoTable 1 }
9
10
11  RptrInfoEntry ::=
12      SEQUENCE {
13          rptrInfoId
14              Integer32,
15          rptrInfoRptrType
16              INTEGER,
17          rptrInfoOperStatus
18              INTEGER,
19          rptrInfoReset
20              INTEGER,
21          rptrInfoPartitionedPorts
22              Gauge32,
23          rptrInfoLastChange
24              TimeStamp
25      }
26
27
28  rptrInfoId OBJECT-TYPE
29      SYNTAX      Integer32 (1..2147483647)
30      MAX-ACCESS  not-accessible
31      STATUS      current
32      DESCRIPTION
33          "This object identifies the repeater for which
34          this entry contains information."
35      ::= { rptrInfoEntry 1 }
36
37
38  rptrInfoRptrType OBJECT-TYPE
39      SYNTAX      INTEGER {
40          other(1),          -- undefined or unknown
41          tenMb(2),
42          onehundredMbClassI(3),
43          onehundredMbClassII(4)
44      }
45      MAX-ACCESS  read-only
46      STATUS      current
47      DESCRIPTION
48          "The rptrInfoRptrType returns a value that identifies
49          the CSMA/CD repeater type."
50      REFERENCE
51          "IEEE Std 802.3, 30.4.1.1.2, aRepeaterType."
52      ::= { rptrInfoEntry 2 }
53
54
55  rptrInfoOperStatus OBJECT-TYPE
56      SYNTAX      INTEGER {
57          other(1),
58          ok(2),
59          failure(3)
60      }
61      MAX-ACCESS  read-only
62      STATUS      current
63      DESCRIPTION
```

```
1           "The rptrInfoOperStatus object indicates the
2           operational state of the repeater."
3
4   REFERENCE
5       "IEEE Std 802.3, 30.4.1.1.5, aRepeaterHealthState."
6   ::= { rptrInfoEntry 3 }
7
8   rptrInfoReset OBJECT-TYPE
9       SYNTAX      INTEGER {
10                noReset(1),
11                reset(2)
12            }
13       MAX-ACCESS  read-write
14       STATUS      current
15       DESCRIPTION
16           "Setting this object to reset(2) causes a
17           transition to the START state of Figure 9-2 in
18           Clause 9 IEEE Std 802.3 for a 10 Mb/s repeater,
19           and to the START state of Figure 27-2 in Clause 27
20           of that standard for a 100 Mb/s repeater.
21
22           Setting this object to noReset(1) has no effect.
23           The agent will return the value noReset(1)
24           when this object is read.
25
26           After receiving a request to set this variable to
27           reset(2), the agent is allowed to delay the reset
28           for a short period. For example, the implementor
29           may choose to delay the reset long enough to allow
30           the SNMP response to be transmitted. In any
31           event, SNMP requires that a response be transmitted.
32
33           This action does not reset the management counters
34           defined in this document nor does it affect the
35           portAdminStatus parameters. Included in this
36           action is the execution of a disruptive Self-Test
37           with the following characteristics: a) The nature
38           of the tests is not specified. b) The test resets
39           the repeater but without affecting management
40           information about the repeater. c) The test does
41           not inject packets onto any segment. d) Packets
42           received during the test may or may not be
43           transferred. e) The test does not interfere with
44           management functions.
45
46           After performing this self-test, the agent will
47           update the repeater health information (including
48           rptrInfoOperStatus), and send a rptrInfoResetEvent
49           notification."
50
51   REFERENCE
52       "IEEE Std 802.3, 30.4.1.2.1, acResetRepeater."
53   ::= { rptrInfoEntry 4 }
54
55   rptrInfoPartitionedPorts OBJECT-TYPE
56       SYNTAX      Gauge32
57       MAX-ACCESS  read-only
58       STATUS      current
59       DESCRIPTION
60           "This object returns the total number of ports in
61           the repeater whose current state meets all three
```

```
1           of the following criteria: rpPtrPortOperStatus
2           does not have the value notPresent(3),
3           rpPtrPortAdminStatus is enabled(1), and
4           rpPtrPortAutoPartitionState is autoPartitioned(2).
5       ::= { rpPtrInfoEntry 5 }
6
7   rpPtrInfoLastChange OBJECT-TYPE
8       SYNTAX      TimeStamp
9       MAX-ACCESS  read-only
10      STATUS      current
11      DESCRIPTION
12          "The value of sysUpTime when any of the following
13          conditions occurred:
14              1) agent cold- or warm-started;
15              2) this instance of repeater was created
16                 (such as when a device or module was
17                 added to the repeater system);
18              3) a change in the value of rpPtrInfoOperStatus;
19              4) ports were added or removed as members of
20                 the repeater; or
21              5) any of the counters associated with this
22                 repeater had a discontinuity."
23      ::= { rpPtrInfoEntry 6 }
24
25  -- Statistics at the port level.
26  --
27
28  rpPtrMonitorPortTable OBJECT-TYPE
29      SYNTAX      SEQUENCE OF RptrMonitorPortEntry
30      MAX-ACCESS  not-accessible
31      STATUS      current
32      DESCRIPTION
33          "Table of performance and error statistics for the
34          ports. The number of entries is the same as that
35          in the rpPtrPortTable.
36
37          The columnar object rpPtrMonitorPortLastChange
38          is used to indicate possible discontinuities
39          of counter type columnar objects in the table."
40      ::= { rpPtrMonitorPortInfo 1 }
41
42  rpPtrMonitorPortEntry OBJECT-TYPE
43      SYNTAX      RptrMonitorPortEntry
44      MAX-ACCESS  not-accessible
45      STATUS      current
46      DESCRIPTION
47          "An entry in the table, containing performance and
48          error statistics for a single port."
49      INDEX      { rpPtrMonitorPortGroupIndex, rpPtrMonitorPortIndex }
50      ::= { rpPtrMonitorPortTable 1 }
51
52  RptrMonitorPortEntry ::=
53      SEQUENCE {
54          rpPtrMonitorPortGroupIndex
55              Integer32,
56          rpPtrMonitorPortIndex
57              Integer32,
58          rpPtrMonitorPortReadableFrames
59              Counter32,
```

```
1      rpPtrMonitorPortReadableOctets
2          Counter32,
3      rpPtrMonitorPortFCSErrors
4          Counter32,
5      rpPtrMonitorPortAlignmentErrors
6          Counter32,
7      rpPtrMonitorPortFrameTooLongs
8          Counter32,
9      rpPtrMonitorPortShortEvents
10         Counter32,
11      rpPtrMonitorPortRunt
12         Counter32,
13      rpPtrMonitorPortCollisions
14         Counter32,
15      rpPtrMonitorPortLateEvents
16         Counter32,
17      rpPtrMonitorPortVeryLongEvents
18         Counter32,
19      rpPtrMonitorPortDataRateMismatches
20         Counter32,
21      rpPtrMonitorPortAutoPartitions
22         Counter32,
23      rpPtrMonitorPortTotalErrors
24         Counter32,
25      rpPtrMonitorPortLastChange
26         TimeStamp
27  }
28
29  rpPtrMonitorPortGroupIndex OBJECT-TYPE
30      SYNTAX      Integer32 (1..2147483647)
31      MAX-ACCESS  not-accessible
32      STATUS      current
33      DESCRIPTION
34          "This object identifies the group containing the
35          port for which this entry contains information."
36      ::= { rpPtrMonitorPortEntry 1 }
37
38  rpPtrMonitorPortIndex OBJECT-TYPE
39      SYNTAX      Integer32 (1..2147483647)
40      MAX-ACCESS  not-accessible
41      STATUS      current
42      DESCRIPTION
43          "This object identifies the port within the group
44          for which this entry contains information."
45      REFERENCE
46          "IEEE Std 802.3, 30.4.3.1.1, aPortID."
47      ::= { rpPtrMonitorPortEntry 2 }
48
49  rpPtrMonitorPortReadableFrames OBJECT-TYPE
50      SYNTAX      Counter32
51      MAX-ACCESS  read-only
52      STATUS      current
53      DESCRIPTION
54          "This object is the number of frames of valid
55          frame length that have been received on this port.
56          This counter is incremented by one for each frame
57          received on this port whose OctetCount is greater
58          than or equal to minFrameSize and less than or
59          equal to maxFrameSize (Ref: IEEE 802.3 Std,
```

1 4.4.2.1) and for which the FCSError and
2 CollisionEvent signals are not asserted.
3
4 A discontinuity may occur in the value
5 when the value of object
6 rpPtrMonitorPortLastChange changes.
7
8
9 This statistic provides one of the parameters
10 necessary for obtaining the packet error ratio.
11 The approximate minimum time for rollover of this
12 counter is 80 hours at 10 Mb/s."
13 REFERENCE
14 "IEEE Std 802.3, 30.4.3.1.4, aReadableFrames."
15 ::= { rpPtrMonitorPortEntry 3 }
16
17 rpPtrMonitorPortReadableOctets OBJECT-TYPE
18 SYNTAX Counter32
19 MAX-ACCESS read-only
20 STATUS current
21 DESCRIPTION
22 "This object is the number of octets contained in
23 valid frames that have been received on this port.
24 This counter is incremented by OctetCount for each
25 frame received on this port that has been
26 determined to be a readable frame (i.e., including
27 FCS octets but excluding framing bits and dribble
28 bits).
29
30 A discontinuity may occur in the value
31 when the value of object
32 rpPtrMonitorPortLastChange changes.
33
34 This statistic provides an indicator of the total
35 data transferred. The approximate minimum time
36 for rollover of this counter in a 10 Mb/s repeater
37 is 58 minutes.
38
39 For ports receiving traffic at a maximum rate in
40 a 100 Mb/s repeater, this counter can roll over
41 in less than 6 minutes. Since that amount of time
42 could be less than a management station's poll cycle
43 time, in order to avoid a loss of information a
44 management station is advised to also poll the
45 rpPtrMonitorPortUpper32Octets object, or to use the
46 64-bit counter defined by
47 rpPtrMonitorPortHCReadableOctets instead of the
48 two 32-bit counters."
49
50 REFERENCE
51 "IEEE Std 802.3, 30.4.3.1.5, aReadableOctets."
52 ::= { rpPtrMonitorPortEntry 4 }
53
54 rpPtrMonitorPortFCSErrors OBJECT-TYPE
55 SYNTAX Counter32
56 MAX-ACCESS read-only
57 STATUS current
58 DESCRIPTION
59 "This counter is incremented by one for each frame
60 received on this port with the FCSError signal
61 asserted and the FramingError and CollisionEvent
62 signals asserted."
63
64
65

```
1          signals deasserted and whose OctetCount is greater
2          than or equal to minFrameSize and less than or
3          equal to maxFrameSizeLimit (See IEEE Std 802.3 4.2.7.1).
4
5          A discontinuity may occur in the value
6          when the value of object
7          rpPtrMonitorPortLastChange changes.
8
9
10         The approximate minimum time for rollover of this
11         counter is 80 hours at 10 Mb/s."
12     REFERENCE
13         "IEEE Std 802.3, 30.4.3.1.6,
14         aFrameCheckSequenceErrors."
15     ::= { rpPtrMonitorPortEntry 5 }
16
17 rpPtrMonitorPortAlignmentErrors OBJECT-TYPE
18     SYNTAX      Counter32
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "This counter is incremented by one for each frame
23         received on this port with the FCSError and
24         FramingError signals asserted and CollisionEvent
25         signal deasserted and whose OctetCount is greater
26         than or equal to minFrameSize and less than or
27         equal to maxFrameSizeLimit (See IEEE Std 802.3, 4.2.7.1).
28         If rpPtrMonitorPortAlignmentErrors is
29         incremented then the rpPtrMonitorPortFCSErrors
30         Counter shall not be incremented for the same
31         frame.
32
33         A discontinuity may occur in the value
34         when the value of object
35         rpPtrMonitorPortLastChange changes.
36
37         The approximate minimum time for rollover of this
38         counter is 80 hours at 10 Mb/s."
39     REFERENCE
40         "IEEE Std 802.3, 30.4.3.1.7, aAlignmentErrors."
41     ::= { rpPtrMonitorPortEntry 6 }
42
43 rpPtrMonitorPortFrameTooLongs OBJECT-TYPE
44     SYNTAX      Counter32
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         "This counter is incremented by one for each frame
49         received on this port whose OctetCount is greater
50         than maxFrameSizeLimit (See IEEE Std 802.3, 4.2.7.1).
51         If rpPtrMonitorPortFrameTooLongs is incremented
52         then neither the rpPtrMonitorPortAlignmentErrors
53         nor the rpPtrMonitorPortFCSErrors counter shall be
54         incremented for the frame.
55
56         A discontinuity may occur in the value
57         when the value of object
58         rpPtrMonitorPortLastChange changes.
59
60         The approximate minimum time for rollover of this
```



```
1           counter is 61 days in a 10 Mb/s repeater."
2 REFERENCE
3     "IEEE Std 802.3, 30.4.3.1.8, aFramesTooLong."
4 ::= { rpPtrMonitorPortEntry 7 }
5
6 rpPtrMonitorPortShortEvents OBJECT-TYPE
7     SYNTAX      Counter32
8     MAX-ACCESS  read-only
9     STATUS      current
10    DESCRIPTION
11        "This counter is incremented by one for each
12         CarrierEvent on this port with ActivityDuration
13         less than ShortEventMaxTime. ShortEventMaxTime is
14         greater than 74 bit times and less than 82 bit
15         times. ShortEventMaxTime has tolerances included
16         to provide for circuit losses between a
17         conformance test point at the AUI and the
18         measurement point within the state machine.
19
20         Notes:
21
22         ShortEvents may indicate externally
23         generated noise hits that will cause the repeater
24         to transmit Runts to its other ports, or propagate
25         a collision (which may be late) back to the
26         transmitting DTE and damaged frames to the rest of
27         the network.
28
29         Implementors may wish to consider selecting the
30         ShortEventMaxTime towards the lower end of the
31         allowed tolerance range to accommodate bit losses
32         suffered through physical channel devices not
33         budgeted for within this standard.
34
35         The significance of this attribute is different
36         in 10 and 100 Mb/s collision domains. Clause 9
37         repeaters perform fragment extension of short
38         events which would be counted as runts on the
39         interconnect ports of other repeaters. Clause
40         27 repeaters do not perform fragment extension.
41
42         A discontinuity may occur in the value
43         when the value of object
44         rpPtrMonitorPortLastChange changes.
45
46         The approximate minimum time for rollover of this
47         counter is 16 hours in a 10 Mb/s repeater."
48 REFERENCE
49     "IEEE Std 802.3, 30.4.3.1.9, aShortEvents."
50 ::= { rpPtrMonitorPortEntry 8 }
51
52 rpPtrMonitorPortRunts OBJECT-TYPE
53     SYNTAX      Counter32
54     MAX-ACCESS  read-only
55     STATUS      current
56     DESCRIPTION
57        "This counter is incremented by one for each
58         CarrierEvent on this port that meets one of the
59         following two conditions. Only one test need be
```

1 made. a) The ActivityDuration is greater than
2 ShortEventMaxTime and less than ValidPacketMinTime
3 and the CollisionEvent signal is deasserted. b)
4 The OctetCount is less than 64, the
5 ActivityDuration is greater than ShortEventMaxTime
6 and the CollisionEvent signal is deasserted.
7 ValidPacketMinTime is greater than or equal to 552
8 bit times and less than 565 bit times.
9
10
11 An event whose length is greater than 74 bit times
12 but less than 82 bit times shall increment either
13 the shortEvents counter or the runts counter but
14 not both. A CarrierEvent greater than or equal to
15 552 bit times but less than 565 bit times may or
16 may not be counted as a runt.
17
18 ValidPacketMinTime has tolerances included to
19 provide for circuit losses between a conformance
20 test point at the AUI and the measurement point
21 within the state machine.
22
23 Runts usually indicate collision fragments, a
24 normal network event. In certain situations
25 associated with large diameter networks a
26 percentage of collision fragments may exceed
27 ValidPacketMinTime.
28 A discontinuity may occur in the value
29 when the value of object
30 rpPtrMonitorPortLastChange changes.
31
32 The approximate minimum time for rollover of this
33 counter is 16 hours in a 10 Mb/s repeater."
34
35 REFERENCE
36 "IEEE Std 802.3, 30.4.3.1.10, aRunts."
37 ::= { rpPtrMonitorPortEntry 9 }
38
39
40 rpPtrMonitorPortCollisions OBJECT-TYPE
41 SYNTAX Counter32
42 MAX-ACCESS read-only
43 STATUS current
44 DESCRIPTION
45 "For a Clause 9 repeater, this counter is
46 incremented by one for any CarrierEvent signal
47 on any port for which the CollisionEvent signal
48 on this port is asserted. For a Clause 27
49 repeater port the counter increments on entering
50 the Collision Count Increment state of the
51 partition state diagram (Figure 27-8 of
52 IEEE Std 802.3).
53 A discontinuity may occur in the value
54 when the value of object
55 rpPtrMonitorPortLastChange changes.
56
57 The approximate minimum time for rollover of this
58 counter is 16 hours in a 10 Mb/s repeater."
59 REFERENCE
60 "IEEE Std 802.3, 30.4.3.1.11, aCollisions."
61 ::= { rpPtrMonitorPortEntry 10 }
62
63
64
65

```
1
2   rptrMonitorPortLateEvents OBJECT-TYPE
3       SYNTAX      Counter32
4       MAX-ACCESS  read-only
5       STATUS      current
6       DESCRIPTION
7           "For a Clause 9 repeater port, this counter is
8           incremented by one for each CarrierEvent
9           on this port in which the CollIn(X)
10          variable transitions to the value SQE (see
11          9.6.6.2, IEEE Std 802.3) while the
12          ActivityDuration is greater than the
13          LateEventThreshold. For a Clause 27 repeater
14          port, this counter is incremented by one on
15          entering the Collision Count Increment state
16          of the partition state diagram (Figure 27-8)
17          while the ActivityDuration is greater than
18          the LateEvent- Threshold. Such a CarrierEvent
19          is counted twice, as both a collision and as a
20          lateEvent.
21
22          The LateEventThreshold is greater than 480 bit
23          times and less than 565 bit times.
24          LateEventThreshold has tolerances included to
25          permit an implementation to build a single
26          threshold to serve as both the LateEventThreshold
27          and ValidPacketMinTime threshold.
28
29          A discontinuity may occur in the value
30          when the value of object
31          rptrMonitorPortLastChange changes.
32
33          The approximate minimum time for rollover of this
34          counter is 81 hours in a 10 Mb/s repeater."
35
36  REFERENCE
37      "IEEE Std 802.3, 30.4.3.1.12, aLateEvents."
38  ::= { rptrMonitorPortEntry 11 }
39
40  rptrMonitorPortVeryLongEvents OBJECT-TYPE
41      SYNTAX      Counter32
42      MAX-ACCESS  read-only
43      STATUS      current
44      DESCRIPTION
45          "For a Clause 9 repeater port, this counter
46          is incremented by one for each CarrierEvent
47          whose ActivityDuration is greater than the
48          MAU Jabber Lockup Protection timer TW3
49          (See IEEE Std 802.3 9.6.1 and 9.6.5).
50
51          For a Clause 27 repeater port, this counter
52          is incremented by one on entry to the
53          Rx Jabber state of the receiver timer state
54          diagram (Figure 27-7). Other counters may
55          be incremented as appropriate.
56
57          A discontinuity may occur in the value
58          when the value of object
59          rptrMonitorPortLastChange changes."
60
61  REFERENCE
```

```
1          "IEEE Std 802.3, 30.4.3.1.13, aVeryLongEvents."
2      ::= { rpPtrMonitorPortEntry 12 }
3
4  rpPtrMonitorPortDataRateMismatches OBJECT-TYPE
5      SYNTAX      Counter32
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "This counter is incremented by one for each
10         frame received by this port that meets all
11         of the conditions required by only one of the
12         following two measurement methods:
13
14         Measurement method A: 1) The CollisionEvent
15         signal is not asserted (10 Mb/s operation) or
16         the Collision Count Increment state of the
17         partition state diagram (Figure 27-8 of
18         IEEE Std 802.3) has not been entered
19         (100 Mb/s operation). 2) The ActivityDuration
20         is greater than ValidPacketMinTime. 3) The
21         frequency (data rate) is detectably mismatched
22         from the local transmit frequency.
23
24         Measurement method B: 1) The CollisionEvent
25         signal is not asserted (10 Mb/s operation)
26         or the Collision Count Increment state of the
27         partition state diagram (Figure 27-8 of
28         IEEE Std 802.3) has not been entered
29         (100 Mb/s operation). 2) The OctetCount is
30         greater than 63. 3) The frequency (data
31         rate) is detectably mismatched from the local
32         transmit frequency. The exact degree of
33         mismatch is vendor specific and is to be
34         defined by the vendor for conformance testing.
35
36         When this event occurs, other counters whose
37         increment conditions were satisfied may or may not
38         also be incremented, at the implementor's
39         discretion. Whether or not the repeater was able
40         to maintain data integrity is beyond the scope of
41         this standard.
42
43         A discontinuity may occur in the value
44         when the value of object
45         rpPtrMonitorPortLastChange changes."
46
47  REFERENCE
48      "IEEE Std 802.3, 30.4.3.1.14, aDataRateMismatches."
49  ::= { rpPtrMonitorPortEntry 13 }
50
51  rpPtrMonitorPortAutoPartitions OBJECT-TYPE
52      SYNTAX      Counter32
53      MAX-ACCESS  read-only
54      STATUS      current
55      DESCRIPTION
56          "This counter is incremented by one for
57          each time the repeater has automatically
58          partitioned this port.
59
60          The conditions that cause a Clause 9
```

1 repeater port to partition are specified in
2 the partition state diagram in Clause 9 of
3 IEEE Std 802.3. They are not differentiated
4 here. A Clause 27 repeater port partitions
5 on entry to the Partition Wait state of the
6 partition state diagram (Figure 27-8 in
7 IEEE Std 802.3).
8
9
10 A discontinuity may occur in the value
11 when the value of object
12 rpPtrMonitorPortLastChange changes."
13 REFERENCE
14 "IEEE Std 802.3, 30.4.3.1.15, aAutoPartitions."
15 ::= { rpPtrMonitorPortEntry 14 }
16
17 rpPtrMonitorPortTotalErrors OBJECT-TYPE
18 SYNTAX Counter32
19 MAX-ACCESS read-only
20 STATUS current
21 DESCRIPTION
22 "The total number of errors which have occurred on
23 this port. This counter is the summation of the
24 values of other error counters (for the same
25 port), namely:
26
27 rpPtrMonitorPortFCSErrors,
28 rpPtrMonitorPortAlignmentErrors,
29 rpPtrMonitorPortFrameTooLongs,
30 rpPtrMonitorPortShortEvents,
31 rpPtrMonitorPortLateEvents,
32 rpPtrMonitorPortVeryLongEvents,
33 rpPtrMonitorPortDataRateMismatches, and
34 rpPtrMonitorPortSymbolErrors.
35
36
37
38 This counter is redundant in the sense that it is
39 the summation of information already available
40 through other objects. However, it is included
41 specifically because the regular retrieval of this
42 object as a means of tracking the health of a port
43 provides a considerable optimization of network
44 management traffic over the otherwise necessary
45 retrieval of the summed counters.
46
47
48 Note that rpPtrMonitorPortRunts is not included
49 in this total; this is because runts usually
50 indicate collision fragments, a normal network
51 event.
52
53
54 A discontinuity may occur in the value
55 when the value of object
56 rpPtrMonitorPortLastChange changes."
57 ::= { rpPtrMonitorPortEntry 15 }
58
59 rpPtrMonitorPortLastChange OBJECT-TYPE
60 SYNTAX TimeStamp
61 MAX-ACCESS read-only
62 STATUS current
63 DESCRIPTION
64 "The value of sysUpTime when the last of
65

```
1         the following occurred:
2         1) the agent cold- or warm-started;
3         2) the row for the port was created
4           (such as when a device or module was added
5            to the repeater system); or
6         3) any condition that would cause one of
7            the counters for the row to experience
8            a discontinuity."
9
10        ::= { rpPtrMonitorPortEntry 16 }
11
12    rpPtrMonitor100PortTable OBJECT-TYPE
13        SYNTAX      SEQUENCE OF RptrMonitor100PortEntry
14        MAX-ACCESS   not-accessible
15        STATUS       current
16        DESCRIPTION
17            "Table of additional performance and error
18             statistics for 100 Mb/s ports, above and
19             beyond those parameters that apply to both
20             10 and 100 Mb/s ports. Entries exist only for
21             ports attached to 100 Mb/s repeaters.
22
23             The columnar object rpPtrMonitorPortLastChange
24             is used to indicate possible discontinuities
25             of counter type columnar objects in this table."
26
27        ::= { rpPtrMonitorPortInfo 2 }
28
29    rpPtrMonitor100PortEntry OBJECT-TYPE
30        SYNTAX      RptrMonitor100PortEntry
31        MAX-ACCESS   not-accessible
32        STATUS       current
33        DESCRIPTION
34            "An entry in the table, containing performance
35             and error statistics for a single 100 Mb/s port."
36
37        INDEX      { rpPtrMonitorPortGroupIndex, rpPtrMonitorPortIndex }
38
39        ::= { rpPtrMonitor100PortTable 1 }
40
41    RptrMonitor100PortEntry ::=
42        SEQUENCE {
43            rpPtrMonitorPortIsolates
44                Counter32,
45            rpPtrMonitorPortSymbolErrors
46                Counter32,
47            rpPtrMonitorPortUpper32Octets
48                Counter32,
49            rpPtrMonitorPortHCReadableOctets
50                Counter64
51        }
52
53    rpPtrMonitorPortIsolates OBJECT-TYPE
54        SYNTAX      Counter32
55        MAX-ACCESS   read-only
56        STATUS       current
57        DESCRIPTION
58            "This counter is incremented by one each time that
59             the repeater port automatically isolates as a
60             consequence of false carrier events. The conditions
61             which cause a port to automatically isolate are
62             defined by the transition from the False Carrier
63             state to the Link Unstable state of the carrier
64
65
```

1 integrity state diagram (Figure 27-9 of
2 IEEE Std 802.3).
3
4 Note: Isolates do not affect the value of
5 the PortOperStatus object.
6
7 A discontinuity may occur in the value
8 when the value of object
9 rpPtrMonitorPortLastChange changes."
10
11 REFERENCE
12 "IEEE Std 802.3, 30.4.3.1.16, aIsolates."
13 ::= { rpPtrMonitor100PortEntry 1 }
14
15 rpPtrMonitorPortSymbolErrors OBJECT-TYPE
16 SYNTAX Counter32
17 MAX-ACCESS read-only
18 STATUS current
19 DESCRIPTION
20 "This counter is incremented by one each time when
21 valid length packet was received at the port and
22 there was at least one occurrence of an invalid
23 data symbol. This can increment only once per valid
24 carrier event. A collision presence at any port of
25 the repeater containing port N, will not cause this
26 attribute to increment.
27
28 A discontinuity may occur in the value
29 when the value of object
30 rpPtrMonitorPortLastChange changes.
31
32 The approximate minimum time for rollover of this
33 counter is 7.4 hours at 100 Mb/s."
34
35 REFERENCE
36 "IEEE Std 802.3, 30.4.3.1.17,
37 aSymbolErrorDuringPacket."
38 ::= { rpPtrMonitor100PortEntry 2 }
39
40
41 rpPtrMonitorPortUpper32Octets OBJECT-TYPE
42 SYNTAX Counter32
43 MAX-ACCESS read-only
44 STATUS current
45 DESCRIPTION
46 "This object is the number of octets contained in
47 valid frames that have been received on this port,
48 modulo 2**32. That is, it contains the upper 32
49 bits of a 64-bit octets counter, of which the
50 lower 32 bits are contained in the
51 rpPtrMonitorPortReadableOctets object.
52
53 This two-counter mechanism is provided for those
54 network management protocols that do not support
55 64-bit counters (e.g. SNMP V1) and are used to
56 manage a repeater type of 100 Mb/s.
57
58 Conformance clauses for this MIB are defined such
59 that implementation of this object is not required
60 in a repeater system which does not support 100 Mb/s.
61 However, repeater systems with mixed 10 and 100 Mb/s ports
62 may implement this object across all ports,
63
64
65

```
1           including 10 Mb/s. If this object is implemented, the
2           value shall be a valid count as defined
3           in the first paragraph of this description.
4
5           A discontinuity may occur in the value
6           when the value of object
7           rpPtrMonitorPortLastChange changes."
8       ::= { rpPtrMonitor100PortEntry 3 }
9
10
11 rpPtrMonitorPortHCReadableOctets OBJECT-TYPE
12     SYNTAX      Counter64
13     MAX-ACCESS  read-only
14     STATUS      current
15     DESCRIPTION
16         "This object is the number of octets contained in
17         valid frames that have been received on this port.
18         This counter is incremented by OctetCount for each
19         frame received on this port which has been
20         determined to be a readable frame (i.e., including
21         FCS octets but excluding framing bits and dribble
22         bits).
23
24         This statistic provides an indicator of the total
25         data transferred.
26
27         This counter is a 64-bit version of rpPtrMonitor-
28         PortReadableOctets. It should be used by network
29         management protocols which support 64-bit counters
30         (e.g., SNMPv2).
31
32         Conformance clauses for this MIB are defined such
33         that implementation of this object is not required
34         in a repeater system which does not support 100 Mb/s.
35         However, repeater systems with mixed 10 and 100 Mb/s ports
36         may implement this object across all ports,
37         including 10 Mb/s. If this object is implemented, the
38         value shall be a valid count as defined
39         in the first paragraph of this description.
40
41         A discontinuity may occur in the value
42         when the value of object
43         rpPtrMonitorPortLastChange changes."
44     REFERENCE
45         "IEEE Std 802.3, 30.4.3.1.5, aReadableOctets."
46     ::= { rpPtrMonitor100PortEntry 4 }
47
48
49
50
51
52
53 -- New version of statistics at the repeater level.
54 --
55 -- Statistics objects for each managed repeater
56 -- in the repeater system.
57
58 rpPtrMonTable OBJECT-TYPE
59     SYNTAX      SEQUENCE OF RptrMonEntry
60     MAX-ACCESS  not-accessible
61     STATUS      current
62     DESCRIPTION
63         "A table of information about each
64         non-trivial repeater. The number of entries
```



```
1          in this table is the same as the number of
2          entries in the rpPtrInfoTable.
3
4          The columnar object rpPtrInfoLastChange is
5          used to indicate possible discontinuities of
6          counter type columnar objects in this table."
7      ::= { rpPtrMonitorAllRpPtrInfo 1 }
8
9
10     rpPtrMonEntry OBJECT-TYPE
11         SYNTAX      RpPtrMonEntry
12         MAX-ACCESS   not-accessible
13         STATUS       current
14         DESCRIPTION
15             "An entry in the table, containing information
16             about a single non-trivial repeater."
17         INDEX        { rpPtrInfoId }
18         ::= { rpPtrMonTable 1 }
19
20
21     RpPtrMonEntry ::=
22         SEQUENCE {
23             rpPtrMonTxCollisions
24                 Counter32,
25             rpPtrMonTotalFrames
26                 Counter32,
27             rpPtrMonTotalErrors
28                 Counter32,
29             rpPtrMonTotalOctets
30                 Counter32
31         }
32
33
34     rpPtrMonTxCollisions OBJECT-TYPE
35         SYNTAX      Counter32
36         MAX-ACCESS   read-only
37         STATUS       current
38         DESCRIPTION
39             "For a Clause 9 (10 Mb/s) repeater, this counter
40             is incremented every time the repeater state
41             machine enters the TRANSMIT COLLISION state
42             from any state other than ONE PORT LEFT
43             (see Figure 9-2 IEEE Std 802.3).
44
45             For a Clause 27 repeater, this counter is
46             incremented every time the repeater core state
47             diagram enters the Jam state as a result of
48             Activity(ALL) > 1 (see Figure 27-2 IEEE Std 802.3).
49
50             The approximate minimum time for rollover of this
51             counter is 16 hours in a 10 Mb/s repeater and 1.6
52             hours in a 100 Mb/s repeater."
53         REFERENCE
54             "IEEE Std 802.3, 30.4.1.1.8, aTransmitCollisions"
55         ::= { rpPtrMonEntry 1 }
56
57
58
59     rpPtrMonTotalFrames OBJECT-TYPE
60         SYNTAX      Counter32
61         MAX-ACCESS   read-only
62         STATUS       current
63         DESCRIPTION
64             "The number of frames of valid frame length
65
```

1 that have been received on the ports in this repeater
2 and for which the FCSError and CollisionEvent
3 signals were not asserted. If an implementation
4 can not obtain a count of frames as seen by
5 the repeater itself, this counter may be
6 implemented as the summation of the values of the
7 rpPtrMonitorPortReadableFrames counters for all of
8 the ports in the repeater.
9
10
11 This statistic provides one of the parameters
12 necessary for obtaining the packet error ratio.
13 The approximate minimum time for rollover of this
14 counter is 80 hours in a 10 Mb/s repeater."
15 ::= { rpPtrMonEntry 3 }
16
17 rpPtrMonTotalErrors OBJECT-TYPE
18 SYNTAX Counter32
19 MAX-ACCESS read-only
20 STATUS current
21 DESCRIPTION
22 "The total number of errors which have occurred on
23 all of the ports in this repeater. The errors
24 included in this count are the same as those listed
25 for the rpPtrMonitorPortTotalErrors counter. If an
26 implementation can not obtain a count of these
27 errors as seen by the repeater itself, this counter
28 may be implemented as the summation of the values of
29 the rpPtrMonitorPortTotalErrors counters for all of
30 the ports in the repeater."
31 ::= { rpPtrMonEntry 4 }
32
33
34
35 rpPtrMonTotalOctets OBJECT-TYPE
36 SYNTAX Counter32
37 MAX-ACCESS read-only
38 STATUS current
39 DESCRIPTION
40 "The total number of octets contained in the valid
41 frames that have been received on the ports in
42 this group. If an implementation can not obtain
43 a count of octets as seen by the repeater itself,
44 this counter may be the summation of the
45 values of the rpPtrMonitorPortReadableOctets
46 counters for all of the ports in the group.
47
48
49 This statistic provides an indicator of the total
50 data transferred. The approximate minimum time
51 for rollover of this counter in a 10 Mb/s repeater
52 is 58 minutes divided by the number of ports in
53 the repeater.
54
55
56 For 100 Mb/s repeaters processing traffic at a
57 maximum rate, this counter can roll over in less
58 than 6 minutes divided by the number of ports in
59 the repeater. Since that amount of time could
60 be less than a management station's poll cycle
61 time, in order to avoid a loss of information a
62 management station is advised to also poll the
63 rpPtrMonUpper32TotalOctets object, or to use the
64 64-bit counter defined by rpPtrMonHCTotalOctets
65

```
1         instead of the two 32-bit counters."
2     ::= { rpPtrMonEntry 5 }
3
4     rpPtrMon100Table OBJECT-TYPE
5         SYNTAX      SEQUENCE OF RpPtrMon100Entry
6         MAX-ACCESS   not-accessible
7         STATUS       current
8         DESCRIPTION
9             "A table of additional information about each
10             100 Mb/s repeater, augmenting the entries in
11             the rpPtrMonTable. Entries exist in this table
12             only for 100 Mb/s repeaters.
13
14             The columnar object rpPtrInfoLastChange is
15             used to indicate possible discontinuities of
16             counter type columnar objects in this table."
17     ::= { rpPtrMonitorAllRpPtrInfo 2 }
18
19     rpPtrMon100Entry OBJECT-TYPE
20         SYNTAX      RpPtrMon100Entry
21         MAX-ACCESS   not-accessible
22         STATUS       current
23         DESCRIPTION
24             "An entry in the table, containing information
25             about a single 100 Mb/s repeater."
26     INDEX          { rpPtrInfoId }
27     ::= { rpPtrMon100Table 1 }
28
29     RpPtrMon100Entry ::=
30     SEQUENCE {
31         rpPtrMonUpper32TotalOctets
32             Counter32,
33         rpPtrMonHCTotalOctets
34             Counter64
35     }
36
37     rpPtrMonUpper32TotalOctets OBJECT-TYPE
38         SYNTAX      Counter32
39         MAX-ACCESS   read-only
40         STATUS       current
41         DESCRIPTION
42             "The total number of octets contained in the valid
43             frames that have been received on the ports in
44             this repeater, modulo 2**32. That is, it contains
45             the upper 32 bits of a 64-bit counter, of which
46             the lower 32 bits are contained in the
47             rpPtrMonTotalOctets object. If an implementation
48             can not obtain a count of octets as seen
49             by the repeater itself, the 64-bit value
50             may be the summation of the values of the
51             rpPtrMonitorPortReadableOctets counters combined
52             with the corresponding rpPtrMonitorPortUpper32Octets
53             counters for all of the ports in the repeater.
54
55             This statistic provides an indicator of the total
56             data transferred within the repeater.
57
58             This two-counter mechanism is provided for those
59             network management protocols that do not support
```

```
1          64-bit counters (e.g., SNMP v1) and are used to
2          manage a repeater type of 100 Mb/s.
3
4          Conformance clauses for this MIB are defined such
5          that implementation of this object is not required
6          in a repeater system which does not support 100 Mb/s.
7          However, repeater systems with mixed 10 and 100 Mb/s ports
8          may implement this object across all ports,
9          including 10 Mb/s. If this object is implemented, the
10         value shall be a valid count as defined
11         in the first paragraph of this description."
12
13         ::= { rpPtrMon100Entry 1 }
14
15     rpPtrMonHCTotalOctets OBJECT-TYPE
16         SYNTAX      Counter64
17         MAX-ACCESS   read-only
18         STATUS       current
19         DESCRIPTION
20             "The total number of octets contained in the valid
21             frames that have been received on the ports in
22             this group. If a implementation can not obtain
23             a count of octets as seen by the repeater itself,
24             this counter may be the summation of the
25             values of the rpPtrMonitorPortReadableOctets
26             counters for all of the ports in the group.
27
28             This statistic provides an indicator of the total
29             data transferred.
30
31             This counter is a 64-bit (high-capacity) version
32             of rpPtrMonUpper32TotalOctets and rpPtrMonTotalOctets.
33             It should be used by network management protocols
34             which support 64-bit counters (e.g. SNMPv2).
35
36             Conformance clauses for this MIB are defined such
37             that implementation of this object is not required
38             in a repeater system which does not support 100 Mb/s.
39             However, repeater systems with mixed 10 and 100 Mb/s ports
40             may implement this object across all ports,
41             including 10 Mb/s. If this object is implemented, the
42             value shall be a valid count as defined
43             in the first paragraph of this description."
44
45         ::= { rpPtrMon100Entry 2 }
46
47
48
49
50     --
51     -- The Repeater Address Search Table
52     --
53     -- This table provides an active address tracking
54     -- capability which can be also used to collect the
55     -- necessary information for mapping the topology
56     -- of a network. Note that an NMS is required to have
57     -- read-write access to the table in order to access
58     -- this function. Section 4 "Topology Mapping" of
59     -- IETF RFC 2108 [B20] contains a description of an
60     -- algorithm that can make use of this table,
61     -- in combination with the forwarding databases
62     -- of managed bridges/switches in the network,
63     -- to map network topology. Devices may also
```

```
1  -- use the protocol and a set of managed
2  -- objects defined in IEEE Std 802.1AB Station
3  -- and Media Access Control Connectivity
4  -- Discovery to discover the physical topology
5  -- from adjacent stations.
6  --
7
8
9  rptrAddrSearchTable OBJECT-TYPE
10     SYNTAX      SEQUENCE OF RptrAddrSearchEntry
11     MAX-ACCESS  not-accessible
12     STATUS      current
13     DESCRIPTION
14         "This table contains one entry per repeater in the
15         repeater system. It defines objects that allow a network
16         management application to instruct an agent to watch
17         for a given MAC address and report which port it
18         was seen on. Only one address search can be in
19         progress on each repeater at any one time. Before
20         starting an address search, a management application
21         should obtain 'ownership' of the entry in
22         rptrAddrSearchTable for the repeater that is to
23         perform the search. This is accomplished with the
24         rptrAddrSearchLock and rptrAddrSearchStatus as
25         follows:
26
27         try_again:
28             get(rptrAddrSearchLock, rptrAddrSearchStatus)
29             while (rptrAddrSearchStatus != notInUse)
30             {
31                 /* Loop waiting for objects to be available*/
32                 short delay
33                 get(rptrAddrSearchLock, rptrAddrSearchStatus)
34             }
35
36             /* Try to claim map objects */
37             lock_value = rptrAddrSearchLock
38             if ( set(rptrAddrSearchLock = lock_value,
39                     rptrAddrSearchStatus = inUse,
40                     rptrAddrSearchOwner = 'my-IP-address')
41                 == FAILURE)
42                 /* Another manager got the lock */
43                 goto try_again
44
45             /* I have the lock */
46             set (rptrAddrSearchAddress = <search target>)
47
48             wait for rptrAddrSearchState to change from none
49
50             if (rptrAddrSearchState == single)
51                 get (rptrAddrSearchGroup, rptrAddrSearchPort)
52             /* release the lock, making sure not to overwrite
53             anyone else's lock */
54             set (rptrAddrSearchLock = lock_value+1,
55                 rptrAddrSearchStatus = notInUse,
56                 rptrAddrSearchOwner = '')
57
58         A management station first retrieves the values of
59         the appropriate instances of the rptrAddrSearchLock
60         and rptrAddrSearchStatus objects, periodically
```

repeating the retrieval if necessary, until the value of rptrAddrSearchStatus is 'notInUse'. The management station then tries to set the same instance of the rptrAddrSearchLock object to the value it just retrieved, the same instance of the rptrAddrSearchStatus object to 'inUse', and the corresponding instance of rptrAddrSearchOwner to a value indicating itself. If the set operation succeeds, then the management station has obtained ownership of the rptrAddrSearchEntry, and the value of rptrAddrSearchLock is incremented by the agent (as per the semantics of TestAndIncr). Failure of the set operation indicates that some other manager has obtained ownership of the rptrAddrSearchEntry.

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

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

```
::= { rptrAddrTrackRpPtrInfo 1 }
```

rpPtrAddrSearchEntry OBJECT-TYPE

SYNTAX RptrAddrSearchEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry containing objects for invoking an address search on a repeater."

INDEX { rpPtrInfoId }

```
::= { rpPtrAddrSearchTable 1 }
```

RptrAddrSearchEntry ::=

SEQUENCE {

rpPtrAddrSearchLock	TestAndIncr,
rpPtrAddrSearchStatus	INTEGER,
rpPtrAddrSearchAddress	MacAddress,
rpPtrAddrSearchState	INTEGER,
rpPtrAddrSearchGroup	Integer32,
rpPtrAddrSearchPort	Integer32,
rpPtrAddrSearchOwner	OwnerString

}

rpPtrAddrSearchLock OBJECT-TYPE

SYNTAX TestAndIncr

MAX-ACCESS read-write

```
1      STATUS      current
2      DESCRIPTION
3          "This object is used by a management station as an
4          advisory lock for this rptrAddrSearchEntry."
5      ::= { rptrAddrSearchEntry 1 }
6
7      rptrAddrSearchStatus OBJECT-TYPE
8      SYNTAX      INTEGER {
9          notInUse(1),
10         inUse(2)
11     }
12     MAX-ACCESS read-write
13     STATUS      current
14     DESCRIPTION
15         "This object is used to indicate that some management
16         station is currently using this rptrAddrSearchEntry.
17         Cooperating managers should set this object to
18         'notInUse' when they are finished using this entry.
19         The agent will automatically set the value of this
20         object to 'notInUse' if it has been set to 'inUse'
21         for an unusually long period of time."
22     ::= { rptrAddrSearchEntry 2 }
23
24     rptrAddrSearchAddress OBJECT-TYPE
25     SYNTAX      MacAddress
26     MAX-ACCESS read-write
27     STATUS      current
28     DESCRIPTION
29         "This object is used to search for a specified MAC
30         address. When this object is set, an address search
31         begins. This automatically sets the corresponding
32         instance of the rptrAddrSearchState object to 'none'
33         and the corresponding instances of the
34         rptrAddrSearchGroup and rptrAddrSearchPort objects to
35         0.
36
37         When a valid frame is received by this repeater with
38         a source MAC address that matches the current value
39         of rptrAddrSearchAddress, the agent will update the
40         corresponding instances of rptrAddrSearchState,
41         rptrAddrSearchGroup and rptrAddrSearchPort to reflect
42         the current status of the search, and the group and
43         port on which the frame was seen."
44     ::= { rptrAddrSearchEntry 3 }
45
46     rptrAddrSearchState OBJECT-TYPE
47     SYNTAX      INTEGER {
48         none(1),
49         single(2),
50         multiple(3)
51     }
52     MAX-ACCESS read-only
53     STATUS      current
54     DESCRIPTION
55         "The current state of the MAC address search on this
56         repeater. This object is initialized to 'none' when
57         the corresponding instance of rptrAddrSearchAddress
58         is set. If the agent detects the address on exactly
59         one port, it will set this object to 'single', and
```

```
1         set the corresponding instances of
2         rptrAddrSearchGroup and rptrAddrSearchPort to reflect
3         the group and port on which the address was heard.
4         If the agent detects the address on more than one
5         port, it will set this object to 'multiple'."
6     ::= { rptrAddrSearchEntry 4 }
7
8
9 rptrAddrSearchGroup OBJECT-TYPE
10     SYNTAX      Integer32 (0..2147483647)
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         "The group from which an error-free frame whose
15         source address is equal to the corresponding instance
16         of rptrAddrSearchAddress has been received. The
17         value of this object is undefined when the
18         corresponding instance of rptrAddrSearchState is
19         equal to 'none' or 'multiple'."
20     ::= { rptrAddrSearchEntry 5 }
21
22
23 rptrAddrSearchPort OBJECT-TYPE
24     SYNTAX      Integer32 (0..2147483647)
25     MAX-ACCESS  read-only
26     STATUS      current
27     DESCRIPTION
28         "The port from which an error-free frame whose
29         source address is equal to the corresponding instance
30         of rptrAddrSearchAddress has been received. The
31         value of this object is undefined when the
32         corresponding instance of rptrAddrSearchState is
33         equal to 'none' or 'multiple'."
34     ::= { rptrAddrSearchEntry 6 }
35
36
37 rptrAddrSearchOwner OBJECT-TYPE
38     SYNTAX      OwnerString
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "The entity that currently has 'ownership' of this
43         rptrAddrSearchEntry."
44     ::= { rptrAddrSearchEntry 7 }
45
46
47
48 --
49 -- The Port Address Tracking Table
50 --
51 -- This table provides a way for a network management
52 -- application to passively gather information (using
53 -- read-only privileges) about which network addresses
54 -- are connected to which ports of a repeater.
55 --
56 --
57
58 rptrAddrTrackTable OBJECT-TYPE
59     SYNTAX      SEQUENCE OF RptrAddrTrackEntry
60     MAX-ACCESS  not-accessible
61     STATUS      current
62     DESCRIPTION
63         "Table of address mapping information about the
64         ports."
65
```



```
1      ::= { rptrAddrTrackPortInfo 1 }
2
3  rptrAddrTrackEntry OBJECT-TYPE
4      SYNTAX      RptrAddrTrackEntry
5      MAX-ACCESS  not-accessible
6      STATUS      current
7      DESCRIPTION
8
9          "An entry in the table, containing address mapping
10         information about a single port."
11      INDEX      { rptrAddrTrackGroupIndex, rptrAddrTrackPortIndex }
12      ::= { rptrAddrTrackTable 1 }
13
14  RptrAddrTrackEntry ::=
15      SEQUENCE {
16          rptrAddrTrackGroupIndex
17              INTEGER,
18          rptrAddrTrackPortIndex
19              INTEGER,
20          rptrAddrTrackSourceAddrChanges
21              Counter32,
22          rptrAddrTrackNewLastSrcAddress
23              OptMacAddr,
24          rptrAddrTrackCapacity
25              Integer32
26      }
27
28
29  rptrAddrTrackGroupIndex OBJECT-TYPE
30      SYNTAX      Integer32 (1..2147483647)
31      MAX-ACCESS  not-accessible
32      STATUS      current
33      DESCRIPTION
34
35          "This object identifies the group containing the
36         port for which this entry contains information."
37      ::= { rptrAddrTrackEntry 1 }
38
39  rptrAddrTrackPortIndex OBJECT-TYPE
40      SYNTAX      Integer32 (1..2147483647)
41      MAX-ACCESS  not-accessible
42      STATUS      current
43      DESCRIPTION
44
45          "This object identifies the port within the group
46         for which this entry contains information."
47      REFERENCE
48
49          "IEEE Std 802.3, 30.4.3.1.1, aPortID."
50      ::= { rptrAddrTrackEntry 2 }
51
52  rptrAddrTrackSourceAddrChanges OBJECT-TYPE
53      SYNTAX      Counter32
54      MAX-ACCESS  read-only
55      STATUS      current
56      DESCRIPTION
57
58          "This counter is incremented by one for each time
59         that the rptrAddrTrackNewLastSrcAddress attribute
60         for this port has changed.
61
62         This may indicate whether a link is connected to a
63         single DTE or another multi-user segment.
64
65         A discontinuity may occur in the value when the
```

```
1           value of object rptrMonitorPortLastChange changes.
2
3           The approximate minimum time for rollover of this
4           counter is 81 hours in a 10 Mb/s repeater."
5
6   REFERENCE
7       "IEEE Std 802.3, 30.4.3.1.19, aSourceAddressChanges."
8   ::= { rptrAddrTrackEntry 3 }
9
10  rptrAddrTrackNewLastSrcAddress OBJECT-TYPE
11      SYNTAX      OptMacAddr
12      MAX-ACCESS  read-only
13      STATUS      current
14      DESCRIPTION
15          "This object is the SourceAddress of the last
16          readable frame (i.e., counted by
17          rptrMonitorPortReadableFrames) received by this
18          port. If no frames have been received by this
19          port since the agent began monitoring the port
20          activity, the agent shall return a string of
21          length zero."
22
23  REFERENCE
24      "IEEE Std 802.3, 30.4.3.1.18, aLastSourceAddress."
25  ::= { rptrAddrTrackEntry 4 }
26
27  rptrAddrTrackCapacity OBJECT-TYPE
28      SYNTAX      Integer32
29      MAX-ACCESS  read-only
30      STATUS      current
31      DESCRIPTION
32          "The maximum number of addresses that can be
33          detected on this port. This value indicates
34          to the maximum number of entries in the
35          rptrExtAddrTrackTable relative to this port.
36
37          If this object has the value of 1, the agent
38          implements only the LastSourceAddress mechanism
39          described by IETF RFC 1368 or IETF RFC 1516."
40
41  ::= { rptrAddrTrackEntry 5 }
42
43
44
45  -- Table for multiple addresses per port
46
47  rptrExtAddrTrackTable OBJECT-TYPE
48      SYNTAX      SEQUENCE OF RptrExtAddrTrackEntry
49      MAX-ACCESS  not-accessible
50      STATUS      current
51      DESCRIPTION
52          "A table to extend the address tracking table (i.e.,
53          rptrAddrTrackTable) with a list of source MAC
54          addresses that were recently received on each port.
55          The number of ports is the same as the number
56          of entries in table rptrPortTable. The number of
57          entries in this table depends on the agent/repeater
58          implementation and the number of different
59          addresses received on each port.
60
61          The first entry for each port contains
62          the same MAC address that is given by the
63          rptrAddrTrackNewLastSrcAddress for that port.
64
65
```

```
1
2         Entries in this table for a particular port are
3         retained when that port is switched from one
4         repeater to another.
5
6         The ordering of MAC addresses listed for a
7         particular port is implementation dependent."
8     ::= { rpPtrAddrTrackPortInfo 2 }
9
10
11 rpPtrExtAddrTrackEntry OBJECT-TYPE
12     SYNTAX      RpPtrExtAddrTrackEntry
13     MAX-ACCESS  not-accessible
14     STATUS      current
15     DESCRIPTION
16         "A row in the table of extended address tracking
17         information for ports. Entries cannot be directly
18         created or deleted via SNMP operations."
19     INDEX       { rpPtrAddrTrackGroupIndex,
20                 rpPtrAddrTrackPortIndex,
21                 rpPtrExtAddrTrackMacIndex }
22     ::= { rpPtrExtAddrTrackTable 1 }
23
24
25 RpPtrExtAddrTrackEntry ::= SEQUENCE {
26     rpPtrExtAddrTrackMacIndex Integer32,
27     rpPtrExtAddrTrackSourceAddress MacAddress
28 }
29
30
31 rpPtrExtAddrTrackMacIndex OBJECT-TYPE
32     SYNTAX      Integer32 (1..2147483647)
33     MAX-ACCESS  not-accessible
34     STATUS      current
35     DESCRIPTION
36         "The index of a source MAC address seen on
37         the port.
38
39         The ordering of MAC addresses listed for a
40         particular port is implementation dependent.
41
42         There is no implied relationship between a
43         particular index and a particular MAC
44         address. The index for a particular MAC
45         address may change without notice."
46     ::= { rpPtrExtAddrTrackEntry 1 }
47
48
49 rpPtrExtAddrTrackSourceAddress OBJECT-TYPE
50     SYNTAX      MacAddress
51     MAX-ACCESS  read-only
52     STATUS      current
53     DESCRIPTION
54         "The source MAC address from a readable frame
55         (i.e., counted by rpPtrMonitorPortReadableFrames)
56         recently received by the port."
57     REFERENCE
58         "IEEE Std 802.3, 30.4.3.1.18, aLastSourceAddress."
59     ::= { rpPtrExtAddrTrackEntry 2 }
60
61
62
63
64 -- The Repeater Top "N" Port Group
65 -- The Repeater Top N Port group is used to prepare reports that
```

```

1  -- describe a list of ports ordered by one of the statistics in the
2  -- Repeater Monitor Port Table. The statistic chosen by the
3  -- management station is sampled over a management
4  -- station-specified time interval, making the report rate based.
5  -- The management station also specifies the number of ports that
6  -- are reported.
7  --
8  --
9  -- The rpPtrTopNPortControlTable is used to initiate the generation
10 -- of a report. The management station may select the parameters
11 -- of such a report, such as which repeater, which statistic, how
12 -- many ports, and the start and stop times of the sampling. When
13 -- the report is prepared, entries are created in the
14 -- rpPtrTopNPortTable associated with the relevant
15 -- rpPtrTopNControlEntry. These entries are static for
16 -- each report after it has been prepared.
17
18 -- Note that counter discontinuities may appear in some
19 -- implementations if ports' assignment to repeaters changes
20 -- during the collection of data for a Top "N" report.
21 -- A management application could read the corresponding
22 -- rpPtrMonitorPortLastChange timestamp in order to check
23 -- whether a discontinuity occurred.
24
25
26
27 rpPtrTopNPortControlTable OBJECT-TYPE
28     SYNTAX      SEQUENCE OF RptrTopNPortControlEntry
29     MAX-ACCESS  not-accessible
30     STATUS      current
31     DESCRIPTION
32         "A table of control records for reports on the top 'N'
33         ports for the rate of a selected counter. The number
34         of entries depends on the configuration of the agent.
35         The maximum number of entries is implementation
36         dependent."
37     ::= { rpPtrTopNPortInfo 1 }
38
39
40
41 rpPtrTopNPortControlEntry OBJECT-TYPE
42     SYNTAX      RptrTopNPortControlEntry
43     MAX-ACCESS  not-accessible
44     STATUS      current
45     DESCRIPTION
46         "A set of parameters that control the creation of a
47         report of the top N ports according to several metrics."
48     INDEX       { rpPtrTopNPortControlIndex }
49     ::= { rpPtrTopNPortControlTable 1 }
50
51
52 RptrTopNPortControlEntry ::= SEQUENCE {
53     rpPtrTopNPortControlIndex
54         Integer32,
55     rpPtrTopNPortRepeaterId
56         Integer32,
57     rpPtrTopNPortRateBase
58         INTEGER,
59     rpPtrTopNPortTimeRemaining
60         Integer32,
61     rpPtrTopNPortDuration
62         Integer32,
63     rpPtrTopNPortRequestedSize
64         Integer32,
65

```

```
1      rpPtrTopNPortGrantedSize
2          Integer32,
3      rpPtrTopNPortStartTime
4          TimeStamp,
5      rpPtrTopNPortOwner
6          OwnerString,
7      rpPtrTopNPortRowStatus
8          RowStatus
9  }
10
11
12  rpPtrTopNPortControlIndex OBJECT-TYPE
13      SYNTAX      Integer32 (1 .. 65535)
14      MAX-ACCESS  not-accessible
15      STATUS      current
16      DESCRIPTION
17          "An index that uniquely identifies an entry in the
18          rpPtrTopNPortControl table. Each such entry defines
19          one top N report prepared for a repeater or repeater system."
20      ::= { rpPtrTopNPortControlEntry 1 }
21
22
23  rpPtrTopNPortRepeaterId OBJECT-TYPE
24      SYNTAX      Integer32 (0..2147483647)
25      MAX-ACCESS  read-create
26      STATUS      current
27      DESCRIPTION
28          "Identifies the repeater for which a top N report will
29          be prepared (see rpPtrInfoId). If the value of this
30          object is positive, only ports assigned to this repeater
31          will be used to form the list in which to order the
32          Top N table. If this value is zero, all ports will be
33          eligible for inclusion on the list.
34
35          The value of this object may not be modified if the
36          associated rpPtrTopNPortRowStatus object is equal to
37          active(1).
38          If, for a particular row in this table, the repeater
39          specified by the value of this object goes away (is
40          removed from the rpPtrInfoTable) while the associated
41          rpPtrTopNPortRowStatus object is equal to active(1),
42          the row in this table is preserved by the agent but
43          the value of rpPtrTopNPortRowStatus is changed to
44          notInService(2), and the agent may time out the row
45          if appropriate. If the specified repeater comes
46          back (reappears in the rpPtrInfoTable) before the row
47          has been timed out, the management station sets
48          the value of the rpPtrTopNPortRowStatus object back
49          to active(1) if desired (the agent doesn't do this
50          automatically)."
51      ::= { rpPtrTopNPortControlEntry 2 }
52
53
54  rpPtrTopNPortRateBase OBJECT-TYPE
55      SYNTAX      INTEGER {
56          readableFrames(1),
57          readableOctets(2),
58          fcsErrors(3),
59          alignmentErrors(4),
60          frameTooLongs(5),
61          shortEvents(6),
62          runts(7),
63          ...
64      }
```

```
1             collisions(8),
2             lateEvents(9),
3             veryLongEvents(10),
4             dataRateMismatches(11),
5             autoPartitions(12),
6             totalErrors(13),
7             isolates(14),
8             symbolErrors(15)
9         }
10
11     MAX-ACCESS    read-create
12     STATUS        current
13     DESCRIPTION
14         "The monitored variable, which the rpPtrTopNPortRate
15         variable is based upon.
16
17         The value of this object may not be modified if
18         the associated rpPtrTopNPortRowStatus object has
19         a value of active(1)."
```

::= { rpPtrTopNPortControlEntry 3 }

rpPtrTopNPortTimeRemaining OBJECT-TYPE

```
24     SYNTAX        Integer32 (0..2147483647)
25     MAX-ACCESS    read-create
26     STATUS        current
27     DESCRIPTION
28         "The number of seconds left in the report
29         currently being collected. When this object
30         is modified by the management station, a new
31         collection is started, possibly aborting a
32         currently running report. The new value is
33         used as the requested duration of this report,
34         which is loaded into the associated
35         rpPtrTopNPortDuration object.
36
37         When this object is set to a non-zero value,
38         any associated rpPtrTopNPortEntries shall be
39         made inaccessible by the agent. While the value
40         of this object is non-zero, it decrements by one
41         per second until it reaches zero. During this
42         time, all associated rpPtrTopNPortEntries shall
43         remain inaccessible. At the time that this object
44         decrements to zero, the report is made accessible
45         in the rpPtrTopNPortTable. Thus, the rpPtrTopNPort
46         table needs to be created only at the end of the
47         collection interval.
48
49         If the value of this object is set to zero
50         while the associated report is running, the
51         running report is aborted and no associated
52         rpPtrTopNPortEntries are created."
```

DEFVAL { 0 }

::= { rpPtrTopNPortControlEntry 4 }

rpPtrTopNPortDuration OBJECT-TYPE

```
60     SYNTAX        Integer32 (0..2147483647)
61     MAX-ACCESS    read-only
62     STATUS        current
63     DESCRIPTION
64         "The number of seconds that this report has
```

```
1         collected during the last sampling interval,
2         or if this report is currently being collected,
3         the number of seconds that this report is being
4         collected during this sampling interval.
5
6         When the associated rpPtrTopNPortTimeRemaining
7         object is set, this object shall be set by the
8         agent to the same value and shall not be modified
9         until the next time the rpPtrTopNPortTimeRemaining
10        is set.
11
12        This value shall be zero if no reports have been
13        requested for this rpPtrTopNPortControlEntry."
14        ::= { rpPtrTopNPortControlEntry 5 }
15
16rpPtrTopNPortRequestedSize OBJECT-TYPE
17    SYNTAX      Integer32
18    MAX-ACCESS  read-create
19    STATUS      current
20    DESCRIPTION
21        "The maximum number of repeater ports requested
22        for the Top N Table.
23
24        When this object is created or modified, the
25        agent should set rpPtrTopNPortGrantedSize as close
26        to this object as is possible for the particular
27        implementation and available resources."
28    DEFVAL { 10 }
29    ::= { rpPtrTopNPortControlEntry 6 }
30
31rpPtrTopNPortGrantedSize OBJECT-TYPE
32    SYNTAX      Integer32 (0..65535)
33    MAX-ACCESS  read-only
34    STATUS      current
35    DESCRIPTION
36        "The maximum number of repeater ports in the
37        top N table.
38
39        When the associated rpPtrTopNPortRequestedSize object is
40        created or modified, the agent should set this object as
41        closely to the requested value as is possible for the
42        particular implementation and available resources. The
43        agent shall not lower this value except as a result of a
44        set to the associated rpPtrTopNPortRequestedSize object."
45    ::= { rpPtrTopNPortControlEntry 7 }
46
47rpPtrTopNPortStartTime OBJECT-TYPE
48    SYNTAX      TimeStamp
49    MAX-ACCESS  read-only
50    STATUS      current
51    DESCRIPTION
52        "The value of sysUpTime when this top N report was
53        last started. In other words, this is the time that
54        the associated rpPtrTopNPortTimeRemaining object was
55        modified to start the requested report.
56
57        If the report has not yet been started, the value
58        of this object is zero."
59    ::= { rpPtrTopNPortControlEntry 8 }
```

```
1
2   rpPtrTopNPortOwner OBJECT-TYPE
3       SYNTAX      OwnerString
4       MAX-ACCESS   read-create
5       STATUS      current
6       DESCRIPTION
7           "The entity that configured this entry and is
8            using the resources assigned to it."
9       ::= { rpPtrTopNPortControlEntry 9 }
10
11
12   rpPtrTopNPortRowStatus OBJECT-TYPE
13       SYNTAX      RowStatus
14       MAX-ACCESS   read-create
15       STATUS      current
16       DESCRIPTION
17           "The status of this row.
18
19           If the value of this object is not equal to
20           active(1), all associated entries in the
21           rpPtrTopNPortTable shall be deleted by the
22           agent."
23       ::= { rpPtrTopNPortControlEntry 10 }
24
25
26
27   -- Top "N" reports
28
29
30   rpPtrTopNPortTable OBJECT-TYPE
31       SYNTAX      SEQUENCE OF RpPtrTopNPortEntry
32       MAX-ACCESS   not-accessible
33       STATUS      current
34       DESCRIPTION
35           "A table of reports for the top 'N' ports based on
36           setting of associated control table entries. The
37           maximum number of entries depends on the number
38           of entries in table rpPtrTopNPortControlTable and
39           the value of object rpPtrTopNPortGrantedSize for
40           each entry.
41
42           For each entry in the rpPtrTopNPortControlTable,
43           repeater ports with the highest value of
44           rpPtrTopNPortRate shall be placed in this table
45           in decreasing order of that rate until there is
46           no more room or until there are no more ports."
47       ::= { rpPtrTopNPortInfo 2 }
48
49
50
51   rpPtrTopNPortEntry OBJECT-TYPE
52       SYNTAX      RpPtrTopNPortEntry
53       MAX-ACCESS   not-accessible
54       STATUS      current
55       DESCRIPTION
56           "A set of statistics for a repeater port that is
57           part of a top N report."
58       INDEX      { rpPtrTopNPortControlIndex,
59                   rpPtrTopNPortIndex }
60       ::= { rpPtrTopNPortTable 1 }
61
62
63   RpPtrTopNPortEntry ::= SEQUENCE {
64       rpPtrTopNPortIndex
65       Integer32,
```



```
1      rpPtrTopNPortGroupIndex
2          Integer32,
3      rpPtrTopNPortPortIndex
4          Integer32,
5      rpPtrTopNPortRate
6          Gauge32
7  }
8
9
10 rpPtrTopNPortIndex OBJECT-TYPE
11     SYNTAX      Integer32 (1..65535)
12     MAX-ACCESS  not-accessible
13     STATUS      current
14     DESCRIPTION
15         "An index that uniquely identifies an entry in
16         the rpPtrTopNPort table among those in the same
17         report. This index is between 1 and N, where N
18         is the number of entries in this report. Increasing
19         values of rpPtrTopNPortIndex shall be assigned to
20         entries with decreasing values of rpPtrTopNPortRate
21         until index N is assigned to the entry with the
22         lowest value of rpPtrTopNPortRate or there are no
23         more rpPtrTopNPortEntries.
24
25         No ports are included in a report where their
26         value of rpPtrTopNPortRate would be zero."
27     ::= { rpPtrTopNPortEntry 1 }
28
29
30 rpPtrTopNPortGroupIndex OBJECT-TYPE
31     SYNTAX      Integer32 (1..2147483647)
32     MAX-ACCESS  read-only
33     STATUS      current
34     DESCRIPTION
35         "This object identifies the group containing
36         the port for this entry. (See also object
37         type rpPtrGroupIndex.)"
38     ::= { rpPtrTopNPortEntry 2 }
39
40
41 rpPtrTopNPortPortIndex OBJECT-TYPE
42     SYNTAX      Integer32 (1..2147483647)
43     MAX-ACCESS  read-only
44     STATUS      current
45     DESCRIPTION
46         "The index of the repeater port.
47         (See object type rpPtrPortIndex.)"
48     ::= { rpPtrTopNPortEntry 3 }
49
50
51 rpPtrTopNPortRate OBJECT-TYPE
52     SYNTAX      Gauge32
53     MAX-ACCESS  read-only
54     STATUS      current
55     DESCRIPTION
56         "The amount of change in the selected variable
57         during this sampling interval for the identified
58         port. The selected variable is that port's
59         instance of the object selected by
60         rpPtrTopNPortRateBase."
61     ::= { rpPtrTopNPortEntry 4 }
62
63
64
65
```

```
1
2  -- Notifications for use by Repeaters
3  -- Notifications for repeaters in a multiple-repeater implementation.
4  -- An implementation may send either the single-repeater OR
5  -- multiple-repeater version of these notifications (1 or 4; 2 or 5)
6  -- but not both.
7
8  ieee8023snmpDot3RpPtrNotifications OBJECT IDENTIFIER
9      ::= { ieee8023snmpDot3RpPtrMgt 0 }
10
11
12  rpPtrInfoHealth NOTIFICATION-TYPE
13      OBJECTS      { rpPtrInfoOperStatus }
14      STATUS      current
15      DESCRIPTION
16          "In a repeater system containing multiple managed repeaters,
17          the rpPtrInfoHealth notification conveys information
18          related to the operational status of a repeater.
19          It is sent either when the value of rpPtrInfoOperStatus
20          changes, or upon completion of a non-disruptive test.
21
22          The agent shall limit the generation of
23          consecutive rpPtrInfoHealth notifications for
24          the same repeater so that there is at least
25          a five-second gap between notifications of this type.
26          When notifications are throttled, they are dropped,
27          not queued for sending at a future time. (Note
28          that 'generating' a notification means sending
29          to all configured recipients.)"
30
31      REFERENCE
32          "IEEE Std 802.3, 30.4.1.3.1, nRepeaterHealth
33          notification."
34
35      ::= { ieee8023snmpDot3RpPtrNotifications 4 }
36
37  rpPtrInfoResetEvent NOTIFICATION-TYPE
38      OBJECTS      { rpPtrInfoOperStatus }
39      STATUS      current
40      DESCRIPTION
41          "In a repeater system containing multiple managed
42          repeaters, the rpPtrInfoResetEvent notification
43          conveys information related to the operational
44          status of a repeater. This notification is sent
45          on completion of a repeater reset action. A
46          repeater reset action is defined as a transition
47          to the START state of Figure 9-2 in Clause 9 of
48          IEEE Std 802.3, when triggered by a management
49          command (e.g., an SNMP Set on the rpPtrInfoReset
50          object).
51
52          The agent shall limit the generation of
53          consecutive rpPtrInfoResetEvent notifications for
54          a single repeater so that there is at least
55          a five-second gap between notifications of
56          this type. When notifications are throttled,
57          they are dropped, not queued for sending at
58          a future time. (Note that 'generating' a
59          notification means sending to all configured
60          recipients.)
61
62          The rpPtrInfoResetEvent is not sent when the
```

```
1          agent restarts and sends an SNMP coldStart or
2          warmStart trap. However, it is recommended that
3          a repeater agent send the rptrInfoOperStatus
4          object as an optional object with its coldStart
5          and warmStart trap PDUs."
6      REFERENCE
7          "IEEE Std 802.3, 30.4.1.3.2, nRepeaterReset
8          notification."
9      ::= { ieee8023snmpDot3RptrNotifications 5 }
10
11
12
13  -- Conformance statements
14
15  snmpRptrModConf
16      OBJECT IDENTIFIER ::= { ieee8023snmpRptrMIB 2 }
17  snmpRptrModCompls
18      OBJECT IDENTIFIER ::= { snmpRptrModConf 1 }
19  snmpRptrModObjGrps
20      OBJECT IDENTIFIER ::= { snmpRptrModConf 2 }
21  snmpRptrModNotGrps
22      OBJECT IDENTIFIER ::= { snmpRptrModConf 3 }
23
24
25
26  -- Object groups
27
28  snmpRptrGrpBasic OBJECT-GROUP
29      OBJECTS      { rptrGroupObjectID,
30                    rptrGroupOperStatus,
31                    rptrGroupPortCapacity,
32                    rptrPortAdminStatus,
33                    rptrPortAutoPartitionState,
34                    rptrPortOperStatus,
35                    rptrPortRptrId,
36                    rptrInfoRptrType,
37                    rptrInfoOperStatus,
38                    rptrInfoReset,
39                    rptrInfoPartitionedPorts,
40                    rptrInfoLastChange }
41      STATUS      current
42      DESCRIPTION
43          "Basic group for a repeater system with one or more
44          repeater-units in multisegment (post-RFC 1516)
45          version of the MIB module."
46      ::= { snmpRptrModObjGrps 1 }
47
48
49
50  snmpRptrGrpMonitor OBJECT-GROUP
51      OBJECTS      { rptrMonitorPortReadableFrames,
52                    rptrMonitorPortReadableOctets,
53                    rptrMonitorPortFCSErrors,
54                    rptrMonitorPortAlignmentErrors,
55                    rptrMonitorPortFrameTooLongs,
56                    rptrMonitorPortShortEvents,
57                    rptrMonitorPortRunts,
58                    rptrMonitorPortCollisions,
59                    rptrMonitorPortLateEvents,
60                    rptrMonitorPortVeryLongEvents,
61                    rptrMonitorPortDataRateMismatches,
62                    rptrMonitorPortAutoPartitions,
63                    rptrMonitorPortTotalErrors,
64
65
```

```
1          rpPtrMonitorPortLastChange,
2
3          rpPtrMonTxCollisions,
4          rpPtrMonTotalFrames,
5          rpPtrMonTotalErrors,
6          rpPtrMonTotalOctets }
7      STATUS      current
8      DESCRIPTION
9
10         "Monitor group for a repeater system with one or more
11         repeater-units in multisegment (post-RFC 1516)
12         version of the MIB module."
13     ::= { snmpRptrModObjGrps 2 }
14
15 snmpRptrGrpMonitor100 OBJECT-GROUP
16     OBJECTS      { rpPtrMonitorPortIsolates,
17                   rpPtrMonitorPortSymbolErrors,
18                   rpPtrMonitorPortUpper32Octets,
19
20                   rpPtrMonUpper32TotalOctets }
21     STATUS      current
22     DESCRIPTION
23
24         "Monitor group for 100 Mb/s ports and repeaters
25         in a repeater system with one or more repeater-units in
26         multisegment (post-RFC 1516) version of the MIB
27         module. Repeater systems which support Counter64 should
28         also implement snmpRptrGrpMonitor100w64."
29     ::= { snmpRptrModObjGrps 3 }
30
31 snmpRptrGrpMonitor100w64 OBJECT-GROUP
32     OBJECTS      { rpPtrMonitorPortHCReadableOctets,
33                   rpPtrMonHCTotalOctets }
34     STATUS      current
35     DESCRIPTION
36
37         "Monitor group for 100 Mb/s ports and repeaters in a
38         repeater system with one or more repeater-units and support
39         for Counter64."
40     ::= { snmpRptrModObjGrps 4 }
41
42 snmpRptrGrpAddrTrack OBJECT-GROUP
43     OBJECTS      { rpPtrAddrTrackSourceAddrChanges,
44                   rpPtrAddrTrackNewLastSrcAddress,
45                   rpPtrAddrTrackCapacity }
46     STATUS      current
47     DESCRIPTION
48
49         "Passive address tracking group for post-RFC 1516
50         version of the MIB module."
51     ::= { snmpRptrModObjGrps 5 }
52
53 snmpRptrGrpExtAddrTrack OBJECT-GROUP
54     OBJECTS      { rpPtrExtAddrTrackSourceAddress }
55     STATUS      current
56     DESCRIPTION
57
58         "Extended passive address tracking group for
59         a repeater system with one or more repeater-units in
60         post-RFC 1516 version of the MIB module."
61     ::= { snmpRptrModObjGrps 6 }
62
63 snmpRptrGrpRptrAddrSearch OBJECT-GROUP
64     OBJECTS      { rpPtrAddrSearchLock,
```

```
1             rptrAddrSearchStatus,
2             rptrAddrSearchAddress,
3             rptrAddrSearchState,
4             rptrAddrSearchGroup,
5             rptrAddrSearchPort,
6             rptrAddrSearchOwner }
7
8     STATUS          current
9     DESCRIPTION
10        "Active MAC address search group and topology
11        mapping support for repeaters."
12        ::= { snmpRptrModObjGrps 7 }
13
14    snmpRptrGrpTopNPort OBJECT-GROUP
15        OBJECTS      { rptrTopNPortRepeaterId,
16                      rptrTopNPortRateBase,
17                      rptrTopNPortTimeRemaining,
18                      rptrTopNPortDuration,
19                      rptrTopNPortRequestedSize,
20                      rptrTopNPortGrantedSize,
21                      rptrTopNPortStartTime,
22                      rptrTopNPortOwner,
23                      rptrTopNPortRowStatus,
24                      rptrTopNPortGroupIndex,
25                      rptrTopNPortPortIndex,
26                      rptrTopNPortRate }
27
28    STATUS          current
29    DESCRIPTION
30        "Top 'N' group for repeater ports."
31        ::= { snmpRptrModObjGrps 8 }
32
33
34    ieee8023snmpDot3RptrNotGroup NOTIFICATION-GROUP
35        NOTIFICATIONS { rptrInfoHealth,
36                      rptrInfoResetEvent }
37
38    STATUS          current
39    DESCRIPTION
40        "Conformance Group for repeater notifications.
41        Formerly an empty group."
42        ::= {snmpRptrModNotGrps 1}
43
44
45    -- Compliance statements
46
47    snmpRptrModCompl MODULE-COMPLIANCE
48        STATUS          current
49        DESCRIPTION
50            "Compliance for the multisegment version of the
51            MIB module for a repeater system with one or more
52            repeater-units."
53
54
55        MODULE -- this module
56            MANDATORY-GROUPS { snmpRptrGrpBasic,
57                              snmpRptrGrpMonitor,
58                              snmpRptrGrpAddrTrack }
59
60            GROUP snmpRptrGrpMonitor100
61            DESCRIPTION
62                "Implementation of this group is
63                mandatory for managed repeater systems that
64                contain 100 Mb/s repeaters."
65
```

```
1
2      GROUP snmpRptrGrpMonitor100w64
3      DESCRIPTION
4          "Implementation of this group is
5          mandatory for managed repeater systems that
6          contain 100 Mb/s repeaters and that
7          can support Counter64."
8
9
10     GROUP snmpRptrGrpExtAddrTrack
11     DESCRIPTION
12         "Implementation of this group is
13         recommended for repeater systems that have
14         the necessary instrumentation to track
15         MAC addresses of multiple DTEs attached
16         to a single repeater port."
17
18
19     GROUP snmpRptrGrpRptrAddrSearch
20     DESCRIPTION
21         "Implementation of this group is
22         recommended for repeater systems that allow
23         read-write access and that have
24         the necessary instrumentation to
25         search all incoming data streams
26         for a particular MAC address."
27
28
29     GROUP snmpRptrGrpTopNPort
30     DESCRIPTION
31         "Implementation of this group is
32         recommended for repeater systems that have
33         the necessary resources to support
34         TopN statistics reporting."
35
36
37     GROUP ieee8023snmpDot3RptrNotGroup
38     DESCRIPTION
39         "Implementation of this group is
40         recommended for repeaters that
41         support notifications."
42
43     ::= { snmpRptrModCompls 1 }
44
45
46 END
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

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

8.1 Introduction

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

8.2 Overview

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

8.3 MIB structure

These MIB objects are categorized into three MIB groups.

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

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

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

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

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

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

- pethPsePortAdminEnable
- pethPsePortPowerPairs
- pethPsePortPowerPriority
- pethPsePortType

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

- pethMainPseUsageThreshold
- pethNotificationControlEnable

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

- pethPsePortPowerPairsControlAbility
- pethPsePortPowerPriority
- pethPsePortPowerClassifications

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

8.5 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL¹⁵:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C8mib.txt

¹⁵Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```

1  IEEE8023-POWER-ETHERNET-MIB DEFINITIONS ::= BEGIN
2
3  IMPORTS
4      MODULE-IDENTITY, OBJECT-TYPE, Integer32,
5      Gauge32, Counter32, NOTIFICATION-TYPE, org
6          FROM SNMPv2-SMI
7      TruthValue
8          FROM SNMPv2-TC
9      MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
10         FROM SNMPv2-CONF
11
12
13      SnmpAdminString
14          FROM SNMP-FRAMEWORK-MIB;
15
16  ieee8023powerEthernetMIB MODULE-IDENTITY
17
18      LAST-UPDATED "201304110000Z" -- April 11, 2013
19      ORGANIZATION
20          "IEEE 802.3 working group"
21      CONTACT-INFO
22          "WG-URL: http://www.ieee802.org/3/index.html
23          WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
24
25          Contact: Howard Frazier
26          Postal: 3151 Zanker Road
27                  San Jose, CA 95134
28                  USA
29          Tel: +1.408.922.8164
30          E-mail: hfrazier@broadcom.com"
31
32
33      DESCRIPTION
34          "The MIB module for managing Power Source Equipment
35          (PSE)specified in IEEE Std 802.3 Clause 33."
36
37
38      REVISION "201304110000Z" -- April 11, 2013
39      DESCRIPTION
40          "Revision, based on an earlier version in IEEE Std 802.3.1-2011."
41
42
43      REVISION "201102020000Z" -- February 2, 2011
44      DESCRIPTION
45          "Initial version, based on an earlier version published
46          as RFC 3621."
47
48      ::= { org ieee(111) standards-association-numbers-series-standards(2)
49          lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 8 }
50
51  pethNotifications OBJECT IDENTIFIER ::= { ieee8023powerEthernetMIB 0 }
52  pethObjects OBJECT IDENTIFIER ::= { ieee8023powerEthernetMIB 1 }
53  pethConformance OBJECT IDENTIFIER ::= { ieee8023powerEthernetMIB 2 }
54
55
56  -- PSE Objects
57
58  pethPsePortTable OBJECT-TYPE
59      SYNTAX SEQUENCE OF PethPsePortEntry
60      MAX-ACCESS not-accessible
61      STATUS current
62      DESCRIPTION
63          "A table of objects that display and control the power
64          characteristics of power Ethernet ports on a Power Source
65

```

```

1      Equipment (PSE) device. This group will be implemented in
2      managed power Ethernet switches and mid-span devices.
3      Values of all read-write objects in this table are
4      persistent at restart/reboot."
5      ::= { pethObjects 1 }
6
7      pethPsePortEntry OBJECT-TYPE
8          SYNTAX      PethPsePortEntry
9          MAX-ACCESS   not-accessible
10         STATUS      current
11         DESCRIPTION
12             "A set of objects that display and control the power
13             characteristics of a power Ethernet PSE port."
14         INDEX       { pethPsePortGroupIndex , pethPsePortIndex }
15         ::= { pethPsePortTable 1 }
16
17     PethPsePortEntry ::= SEQUENCE {
18         pethPsePortGroupIndex      Integer32,
19         pethPsePortIndex           Integer32,
20         pethPsePortAdminEnable     TruthValue,
21         pethPsePortPowerPairsControlAbility TruthValue,
22         pethPsePortPowerPairs      INTEGER,
23         pethPsePortDetectionStatus INTEGER,
24         pethPsePortPowerPriority    INTEGER,
25         pethPsePortMPSAbsentCounter Counter32,
26         pethPsePortType            SnmpAdminString,
27         pethPsePortPowerClassifications INTEGER,
28         pethPsePortInvalidSignatureCounter Counter32,
29         pethPsePortPowerDeniedCounter Counter32,
30         pethPsePortOverLoadCounter Counter32,
31         pethPsePortShortCounter    Counter32,
32         pethPsePortActualPower     Integer32,
33         pethPsePortPowerAccuracy    Integer32,
34         pethPsePortCumulativeEnergy Counter32
35     }
36
37     pethPsePortGroupIndex OBJECT-TYPE
38         SYNTAX      Integer32 (1..2147483647)
39         MAX-ACCESS   not-accessible
40         STATUS      current
41         DESCRIPTION
42             "This variable uniquely identifies the group
43             containing the port to which a power Ethernet PSE is
44             connected. Group means box in the stack, module in a
45             rack and the value 1 shall be used for non-modular devices.
46             Furthermore, the same value shall be used in this variable,
47             pethMainPseGroupIndex, and pethNotificationControlGroupIndex
48             to refer to a given box in a stack or module in a rack."
49         ::= { pethPsePortEntry 1 }
50
51     pethPsePortIndex OBJECT-TYPE
52         SYNTAX      Integer32 (1..2147483647)
53         MAX-ACCESS   not-accessible
54         STATUS      current
55         DESCRIPTION
56             "This variable uniquely identifies the power Ethernet PSE
57             port within group pethPsePortGroupIndex to which the
58             power Ethernet PSE entry is connected."
59

```

```

1      ::= { pethPsePortEntry 2 }
2
3      pethPsePortAdminEnable OBJECT-TYPE
4      SYNTAX TruthValue
5      MAX-ACCESS read-write
6      STATUS current
7      DESCRIPTION
8
9          "true (1) An interface that can provide the PSE functions.
10         false(2) The interface will act as it would if it had no PSE
11         function."
12
13      REFERENCE
14          "IEEE Std 802.3, 30.9.1.1.2 aPSEAdminState"
15      ::= { pethPsePortEntry 3 }
16
17      pethPsePortPowerPairsControlAbility OBJECT-TYPE
18      SYNTAX TruthValue
19      MAX-ACCESS read-only
20      STATUS current
21      DESCRIPTION
22
23          "Describes the capability of controlling the power pairs
24          functionality to switch pins for sourcing power.
25          The value true indicate that the device has the capability
26          to control the power pairs. When false the PSE Pinout
27          Alternative used cannot be controlled through the
28          PethPsePortAdminEnable attribute."
29      REFERENCE
30          "IEEE Std 802.3, 30.9.1.1.3
31          aPSEPowerPairsControlAbility"
32      ::= { pethPsePortEntry 4 }
33
34      pethPsePortPowerPairs OBJECT-TYPE
35      SYNTAX INTEGER {
36          signal(1),
37          spare(2)
38      }
39      MAX-ACCESS read-write
40      STATUS current
41      DESCRIPTION
42
43          "Describes or controls the pairs in use. If the value of
44          pethPsePortPowerPairsControl is true, this object is
45          writeable.
46          A value of signal(1) means that the signal pairs
47          only are in use.
48          A value of spare(2) means that the spare pairs
49          only are in use."
50      REFERENCE
51          "IEEE Std 802.3, 30.9.1.1.4 aPSEPowerPairs"
52      ::= { pethPsePortEntry 5 }
53
54      pethPsePortDetectionStatus OBJECT-TYPE
55      SYNTAX INTEGER {
56          disabled(1),
57          searching(2),
58          deliveringPower(3),
59          fault(4),
60          test(5),
61          otherFault(6)
62      }
63
64      }
65

```

```
1
2   MAX-ACCESS read-only
3   STATUS current
4   DESCRIPTION
5       "Describes the operational status of the port PD detection.
6       A value of disabled(1)- indicates that the PSE State diagram
7       is in the state DISABLED.
8       A value of deliveringPower(3) - indicates that the PSE State
9       diagram is in the state POWER_ON for a duration greater than
10      tlim max (see IEEE Std 802.3 Table 33-11).
11      A value of fault(4) - indicates that the PSE State diagram is
12      in the state TEST_ERROR.
13      A value of test(5) - indicates that the PSE State diagram is
14      in the state TEST_MODE.
15      A value of otherFault(6) - indicates that the PSE State
16      diagram is in the state IDLE due to the variable
17      error_conditions.
18      A value of searching(2)- indicates the PSE State diagram is
19      in a state other than those listed above."
20
21  REFERENCE
22      "IEEE Std 802.3, 30.9.1.1.5
23      aPSEPowerDetectionStatus"
24      ::= { pethPsePortEntry 6 }
25
26
27  pethPsePortPowerPriority OBJECT-TYPE
28  SYNTAX INTEGER {
29      critical(1),
30      high(2),
31      low(3)
32  }
33
34  MAX-ACCESS read-write
35  STATUS current
36  DESCRIPTION
37      "This object controls the priority of the port from the point
38      of view of a power management algorithm. The priority that
39      is set by this variable could be used by a control mechanism
40      that prevents over current situations by disconnecting first
41      ports with lower power priority. Ports that connect devices
42      critical to the operation of the network - like the E911
43      telephones ports - should be set to higher priority."
44      ::= { pethPsePortEntry 7 }
45
46
47  pethPsePortMPSAbsentCounter OBJECT-TYPE
48  SYNTAX Counter32
49  MAX-ACCESS read-only
50  STATUS current
51  DESCRIPTION
52      "This counter is incremented when the PSE state diagram
53      transitions directly from the state POWER_ON to the
54      state IDLE due to tmpdo_timer_done being asserted."
55
56  REFERENCE
57      "IEEE Std 802.3, 30.9.1.1.11
58      aPSEMPSAbsentCounter"
59      ::= { pethPsePortEntry 8 }
60
61
62  pethPsePortType OBJECT-TYPE
63  SYNTAX SnmpAdminString
64  MAX-ACCESS read-write
65
```

```

1      STATUS current
2      DESCRIPTION
3          "A manager will set the value of this variable to indicate
4          the type of powered device that is connected to the port.
5          The default value supplied by the agent if no value has
6          ever been set should be a zero-length octet string."
7      ::= { pethPsePortEntry 9 }
8
9
10     pethPsePortPowerClassifications OBJECT-TYPE
11         SYNTAX INTEGER {
12             class0(1),
13             class1(2),
14             class2(3),
15             class3(4),
16             class4(5)
17         }
18     MAX-ACCESS read-only
19     STATUS current
20     DESCRIPTION
21         "Classification is a way to tag different terminals on the
22         Power over LAN network according to their power consumption.
23         Devices such as IP telephones, WLAN access points and others,
24         will be classified according to their power requirements.
25
26         The meaning of the classification labels is defined in the
27         IEEE specification.
28
29         This variable is valid only while a PD is being powered,
30         that is, while the attribute pethPsePortDetectionStatus
31         is reporting the enumeration deliveringPower."
32     REFERENCE
33         "IEEE Std 802.3, 30.9.1.1.6
34         aPSEPowerClassification"
35     ::= { pethPsePortEntry 10 }
36
37
38     pethPsePortInvalidSignatureCounter OBJECT-TYPE
39         SYNTAX Counter32
40         MAX-ACCESS read-only
41         STATUS current
42
43     DESCRIPTION
44         "This counter is incremented when the PSE state diagram
45         enters the state SIGNATURE_INVALID."
46     REFERENCE
47         "IEEE Std 802.3, 30.9.1.1.7
48         aPSEInvalidSignatureCounter"
49     ::= { pethPsePortEntry 11 }
50
51
52     pethPsePortPowerDeniedCounter OBJECT-TYPE
53         SYNTAX Counter32
54         MAX-ACCESS read-only
55         STATUS current
56     DESCRIPTION
57         "This counter is incremented when the PSE state diagram
58         enters the state POWER_DENIED."
59     REFERENCE
60         "IEEE Std 802.3, 30.9.1.1.8
61         aPSEPowerDeniedCounter"
62     ::= { pethPsePortEntry 12 }

```

```
1
2   pethPsePortOverLoadCounter OBJECT-TYPE
3     SYNTAX Counter32
4     MAX-ACCESS read-only
5     STATUS current
6     DESCRIPTION
7       "This counter is incremented when the PSE state diagram
8         enters the state ERROR_DELAY_OVER."
9
10    REFERENCE
11      "IEEE Std 802.3, 30.9.1.1.9
12        aPSEOverLoadCounter"
13    ::= { pethPsePortEntry 13 }
14
15    pethPsePortShortCounter OBJECT-TYPE
16      SYNTAX Counter32
17      MAX-ACCESS read-only
18      STATUS current
19      DESCRIPTION
20        "This counter is incremented when the PSE state diagram
21          enters the state ERROR_DELAY_SHORT."
22
23      REFERENCE
24        "IEEE Std 802.3, 30.9.1.1.10
25          aPSEShortCounter"
26      ::= { pethPsePortEntry 14 }
27
28
29
30    pethPsePortActualPower      OBJECT-TYPE
31      SYNTAX Integer32
32      MAX-ACCESS read-only
33      STATUS current
34      DESCRIPTION
35        "See IEEE Std 802.3, 30.9.1.1.12 aPSEActualPower."
36      REFERENCE
37        "IEEE Std 802.3, 30.9.1.1.12 aPSEActualPower."
38      ::= { pethPsePortEntry 15 }
39
40
41
42    pethPsePortPowerAccuracy    OBJECT-TYPE
43      SYNTAX Integer32
44      MAX-ACCESS read-only
45      STATUS current
46      DESCRIPTION
47        "See IEEE Std 802.3, 30.9.1.1.13 aPSEPowerAccuracy."
48      REFERENCE
49        "IEEE Std 802.3, 30.9.1.1.13 aPSEPowerAccuracy."
50      ::= { pethPsePortEntry 16 }
51
52
53
54    pethPsePortCumulativeEnergy OBJECT-TYPE
55      SYNTAX Counter32
56      MAX-ACCESS read-only
57      STATUS current
58      DESCRIPTION
59        "See IEEE Std 802.3, 30.9.1.1.14 aPSECumulativeEnergy."
60      REFERENCE
61        "IEEE Std 802.3, 30.9.1.1.14 aPSECumulativeEnergy."
62      ::= { pethPsePortEntry 17 }
63
64
65
```

```

1  -- Main PSE Objects
2
3  pethMainPseObjects      OBJECT IDENTIFIER ::= { pethObjects 3 }
4
5  pethMainPseTable OBJECT-TYPE
6      SYNTAX      SEQUENCE OF PethMainPseEntry
7      MAX-ACCESS  not-accessible
8      STATUS      current
9      DESCRIPTION
10         "A table of objects that display and control attributes
11         of the main power source in a PSE device. Ethernet
12         switches are one example of devices that would support
13         these objects.
14         Values of all read-write objects in this table are
15         persistent at restart/reboot."
16         ::= { pethMainPseObjects 1 }
17
18
19  pethMainPseEntry OBJECT-TYPE
20      SYNTAX      PethMainPseEntry
21      MAX-ACCESS  not-accessible
22      STATUS      current
23      DESCRIPTION
24         "A set of objects that display and control the Main
25         power of a PSE."
26      INDEX      { pethMainPseGroupIndex }
27      ::= { pethMainPseTable 1 }
28
29
30  PethMainPseEntry ::= SEQUENCE {
31      pethMainPseGroupIndex
32          Integer32,
33      pethMainPsePower
34          Gauge32 ,
35      pethMainPseOperStatus
36          INTEGER,
37      pethMainPseConsumptionPower
38          Gauge32,
39      pethMainPseUsageThreshold
40          Integer32
41  }
42
43  pethMainPseGroupIndex OBJECT-TYPE
44      SYNTAX      Integer32 (1..2147483647)
45      MAX-ACCESS  not-accessible
46      STATUS      current
47      DESCRIPTION
48         "This variable uniquely identifies the group to which
49         power Ethernet PSE is connected. Group means (box in
50         the stack, module in a rack) and the value 1 shall be
51         used for non-modular devices. Furthermore, the same
52         value shall be used in this variable, pethPsePortGroupIndex,
53         and pethNotificationControlGroupIndex to refer to a
54         given box in a stack or module in a rack."
55         ::= { pethMainPseEntry 1 }
56
57
58
59  pethMainPsePower OBJECT-TYPE
60      SYNTAX      Gauge32 (1..65535)
61      UNITS      "Watts"
62      MAX-ACCESS  read-only
63      STATUS      current
64      DESCRIPTION
65

```



```
1           "The nominal power of the PSE expressed in Watts."
2       ::= { pethMainPseEntry 2 }
3
4   pethMainPseOperStatus OBJECT-TYPE
5       SYNTAX INTEGER {
6           on(1),
7           off(2),
8           faulty(3)
9       }
10
11   MAX-ACCESS read-only
12   STATUS current
13   DESCRIPTION
14       "The operational status of the main PSE."
15   ::= { pethMainPseEntry 3 }
16
17   pethMainPseConsumptionPower OBJECT-TYPE
18       SYNTAX Gauge32
19       UNITS "Watts"
20       MAX-ACCESS read-only
21       STATUS current
22       DESCRIPTION
23           "Measured usage power expressed in Watts."
24   ::= { pethMainPseEntry 4 }
25
26
27   pethMainPseUsageThreshold OBJECT-TYPE
28       SYNTAX Integer32 (1..99)
29       UNITS "%"
30       MAX-ACCESS read-write
31       STATUS current
32       DESCRIPTION
33           "The usage threshold expressed in percents for
34           comparing the measured power and initiating
35           an alarm if the threshold is exceeded."
36   ::= { pethMainPseEntry 5 }
37
38
39 -- Notification Control Objects
40
41
42   pethNotificationControl OBJECT IDENTIFIER ::= { pethObjects 4 }
43
44   pethNotificationControlTable OBJECT-TYPE
45       SYNTAX SEQUENCE OF PethNotificationControlEntry
46       MAX-ACCESS not-accessible
47
48       STATUS current
49       DESCRIPTION
50           "A table of objects that display and control the
51           Notification on a PSE device.
52           Values of all read-write objects in this table are
53           persistent at restart/reboot."
54   ::= { pethNotificationControl 1 }
55
56
57   pethNotificationControlEntry OBJECT-TYPE
58       SYNTAX PethNotificationControlEntry
59       MAX-ACCESS not-accessible
60       STATUS current
61       DESCRIPTION
62           "A set of objects that control the Notification events."
63   INDEX { pethNotificationControlGroupIndex }
64   ::= { pethNotificationControlTable 1 }
65
```

```

1
2   PethNotificationControlEntry ::= SEQUENCE {
3       pethNotificationControlGroupIndex
4           Integer32,
5       pethNotificationControlEnable
6           TruthValue
7   }
8
9   pethNotificationControlGroupIndex OBJECT-TYPE
10      SYNTAX      Integer32 (1..2147483647)
11      MAX-ACCESS  not-accessible
12      STATUS      current
13      DESCRIPTION
14          "This variable uniquely identifies the group. Group
15          means box in the stack, module in a rack and the value
16          1 shall be used for non-modular devices. Furthermore,
17          the same value shall be used in this variable,
18          pethPsePortGroupIndex, and
19          pethMainPseGroupIndex to refer to a given box in a
20          stack or module in a rack."
21      ::= { pethNotificationControlEntry 1 }
22
23
24   pethNotificationControlEnable OBJECT-TYPE
25      SYNTAX      TruthValue
26      MAX-ACCESS  read-write
27      STATUS      current
28      DESCRIPTION
29          "This object controls, on a per-group basis, whether
30          or not notifications from the agent are enabled. The
31          value true(1) means that notifications are enabled; the
32          value false(2) means that they are not."
33      ::= { pethNotificationControlEntry 2 }
34
35  --
36  -- Notifications Section
37  --
38  --
39
40
41   pethPsePortOnOffNotification NOTIFICATION-TYPE
42      OBJECTS      { pethPsePortDetectionStatus }
43      STATUS      current
44      DESCRIPTION
45          "This Notification indicates if Pse Port is delivering or
46          not power to the PD. This Notification should be sent on
47          every status change except in the searching mode.
48          At least 500 msec shall elapse between notifications
49          being emitted by the same object instance."
50      ::= { pethNotifications 1 }
51
52
53   pethMainPowerUsageOnNotification NOTIFICATION-TYPE
54      OBJECTS      { pethMainPseConsumptionPower }
55      STATUS      current
56      DESCRIPTION
57          "This Notification indicate PSE Threshold usage
58          indication is on, the usage power is above the
59          threshold. At least 500 msec shall elapse between
60          notifications being emitted by the same object
61          instance."
62      ::= { pethNotifications 2 }
63
64
65   pethMainPowerUsageOffNotification NOTIFICATION-TYPE

```

```
1      OBJECTS      { pethMainPseConsumptionPower }
2      STATUS      current
3      DESCRIPTION
4          "This Notification indicates PSE Threshold usage indication
5          off, the usage power is below the threshold.
6          At least 500 msec shall elapse between notifications being
7          emitted by the same object instance."
8      ::= { pethNotifications 3 }
9
10
11  --
12  -- Conformance statements
13  --
14  pethCompliances OBJECT IDENTIFIER ::= { pethConformance 1 }
15  pethGroups      OBJECT IDENTIFIER ::= { pethConformance 2 }
16
17  -- Compliance statements
18
19  pethCompliance MODULE-COMPLIANCE
20      STATUS current
21      DESCRIPTION
22          "Describes the requirements for conformance to the
23          Power Ethernet MIB."
24
25
26      MODULE -- this module
27          MANDATORY-GROUPS { pethPsePortGroup,
28                              pethPsePortNotificationGroup,
29                              pethNotificationControlGroup
30                              }
31
32          GROUP pethMainPseGroup
33          DESCRIPTION
34              "The pethMainPseGroup is mandatory for PSE systems
35              that implement a main power supply."
36          GROUP pethMainPowerNotificationGroup
37          DESCRIPTION
38              "The pethMainPowerNotificationGroup is mandatory for
39              PSE systems that implement a main power supply."
40      ::= { pethCompliances 1 }
41
42
43  pethPsePortGroup OBJECT-GROUP
44      OBJECTS {
45          pethPsePortAdminEnable,
46          pethPsePortPowerPairsControlAbility,
47          pethPsePortPowerPairs,
48          pethPsePortDetectionStatus,
49          pethPsePortPowerPriority,
50          pethPsePortMPSAbsentCounter,
51          pethPsePortInvalidSignatureCounter,
52          pethPsePortPowerDeniedCounter,
53          pethPsePortOverLoadCounter,
54          pethPsePortShortCounter,
55          pethPsePortType,
56          pethPsePortPowerClassifications,
57          pethPsePortActualPower,
58          pethPsePortPowerAccuracy,
59          pethPsePortCumulativeEnergy
60      }
61
62      STATUS current
63      DESCRIPTION
64          "PSE Port objects."
65
```

```

1      ::= { pethGroups 1 }
2
3  pethMainPseGroup OBJECT-GROUP
4      OBJECTS {
5          pethMainPsePower,
6          pethMainPseOperStatus,
7          pethMainPseConsumptionPower,
8          pethMainPseUsageThreshold
9      }
10
11  STATUS current
12  DESCRIPTION
13      "Main PSE Objects."
14  ::= { pethGroups 2 }
15
16  pethNotificationControlGroup OBJECT-GROUP
17
18      OBJECTS {
19          pethNotificationControlEnable
20      }
21
22  STATUS current
23  DESCRIPTION
24      "Notification Control Objects."
25  ::= { pethGroups 3 }
26
27  pethPsePortNotificationGroup NOTIFICATION-GROUP
28      NOTIFICATIONS { pethPsePortOnOffNotification}
29      STATUS current
30      DESCRIPTION "Pse Port Notifications."
31  ::= { pethGroups 4 }
32
33  pethMainPowerNotificationGroup NOTIFICATION-GROUP
34      NOTIFICATIONS { pethMainPowerUsageOnNotification,
35                      pethMainPowerUsageOffNotification}
36      STATUS current
37      DESCRIPTION "Main PSE Notifications."
38  ::= { pethGroups 5 }
39
40
41
42  END
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

9. Ethernet passive optical networks (EPON) MIB module

9.1 Overview

This clause defines a MIB module for use with SNMP to manage 1G-EPON interfaces for Ethernet Passive Optical Networks. The clause contains a list of management objects based on the attributes defined in the relevant parts of Clause 30 of IEEE Std 802.3, referring to EPON.

9.1.1 EPON architecture highlights

9.1.1.1 Introduction

The EPON standard, now part of IEEE Std 802.3, defines the Physical Layer and Media Access Control sublayer of EPON interfaces. EPON is a variant of Gigabit Ethernet used in optical access. The passive optical network (PON) comprises sections of single-mode fiber connected with passive optical splitter/coupler devices, forming a passive optical tree, as shown in Figure 9-1. Individual branches of the PON are terminated with the optical line terminal (OLT) in the central office and optical network units (ONUs) near the subscribers. ONUs can be located either in some remote location (e.g., basement in a multidwelling unit) or directly at the subscriber premises. Various types of customer premises equipment (CPE) can be connected to ONUs or even integrated with such devices. Figure 9-1 presents an example PON topology.

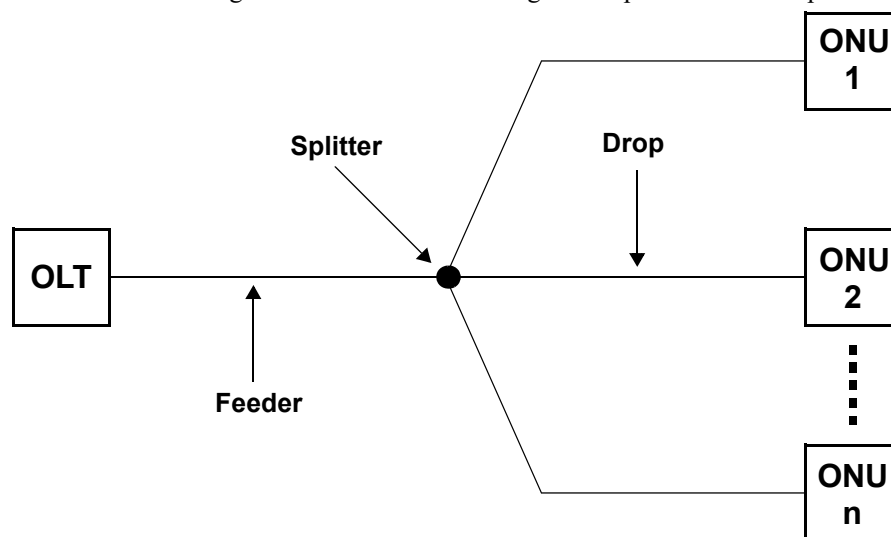


Figure 9-1—PON topology example

The IEEE layering architecture of an EPON interface is defined in the diagram of Figure 56-2 in IEEE Std 802.3. The following clauses in IEEE Std 802.3 define the corresponding layers of an EPON interface:

- Clause 30: Management
- Clause 60: PMD for EPON media (burst-mode PMD)
- Clause 64: MPCP (Multipoint Control Protocol), which defines the Multipoint architecture, and control protocol for the media access of EPON.
- Clause 65: Reconciliation Sublayer and Physical Coding Sublayer, which defines a number of extensions to standard Gigabit Ethernet PCS, i.e.:
 - a) Definition of Point-to-Point emulation function for EPON
 - b) Definition of the optional (frame-based) FEC for EPON
 - c) PMA for EPON

9.1.1.2 Principles of operation

The EPON interface specification extends the specification of Gigabit Ethernet as described in Clause 35 and Clause 36 of IEEE Std 802.3. The Ethernet MAC operates at the data rate of 1 Gb/s, and it is connected to a media-dependent interface through the GMII interface, as described in Clause 35. The EPON PCS layer extends the Gigabit Ethernet PCS as described in Clause 36. New, EPON-specific layers are added to Gigabit Ethernet layers in the following locations:

- MPCP is placed in the MAC control layer, providing EPON media access, station discovery, and registration protocol.
- Functionality of the reconciliation sublayer (RS) of Gigabit Ethernet was extended, creating logical links over shared passive optical medium, providing private transmission channels to each of the connected ONUs.
- (Optional) FEC functionality located between the PCS and PMA layers was added, extending the Gigabit Ethernet PCS layer, enhancing reach and split performance of the EPON optical link.

Figure 9-2 presents the EPON layering model.

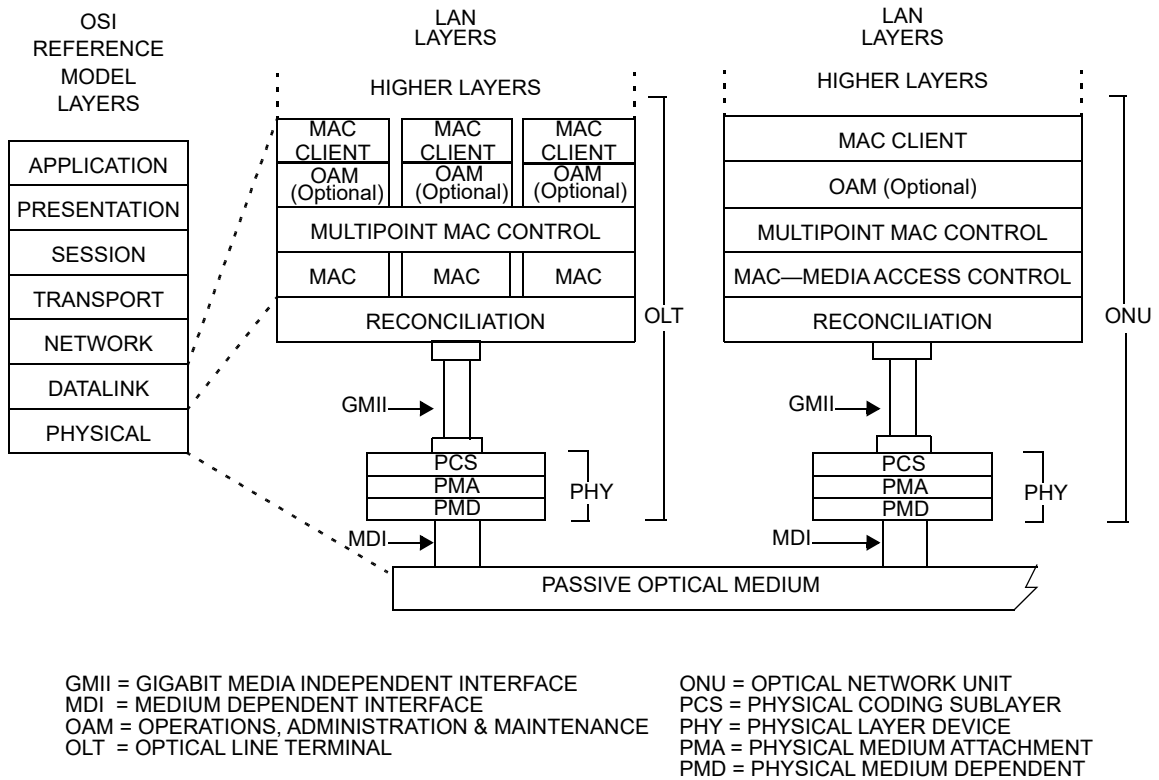


Figure 9-2—Relationship of Multipoint MAC control and the OSI protocol stack

9.1.1.3 Physical media

The physical link in EPON comprises single-mode fiber. The OLT and ONUs are connected through a passive optical network comprising sections of single-mode fiber interconnected with passive splitter/coupler devices.

The term *downstream* denotes transmission from the OLT to all connected ONUs, while the term *upstream* denotes transmission from the connected ONUs (one at the time) to the OLT. Upstream and downstream transmissions are wavelength division multiplexed (WDM) into a single strand of single-mode fiber, sharing the same physical link.

The downstream transmission channel is 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 (LLIDs) 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 toward OLT only when granted a transmission opportunity (slot).

9.1.1.4 PMD specifications

The EPON PMD specifications are based on a wavelength plan similar to that used by ITU-T G.983.1. The OLT and ONU optical parameters were derived in part from earlier 1000 Mb/s Ethernet PMD specifications, with the addition of WDM capabilities, and burst mode operation for ONU transmitters and the OLT receiver.

The upstream burst mode operation capability corresponds directly to the TDMA operation in the upstream direction, where queued data is burst from individual ONUs at full data rate for the duration of the allocated transmission period. Once completed, the ONU goes silent and another ONU starts transmitting its data.

9.1.1.5 Point-to-point emulation

The downstream link is a broadcast medium, which means that all data transmitted by the OLT is received by all connected ONUs. In order to facilitate compliance of EPON with Ethernet architecture, the P2PE function was included in the RS, creating a series of logical links between the OLT and connected ONUs. An additional broadcast link is also provided for delivery of any broadcast content. In this way, EPON becomes a collection of logical P2P connections established between the OLT and the ONUs. Therefore, the OLT can be seen as an Ethernet device with N+1 logical ports (N P2P logical interfaces and 1 broadcast interface, where N designates the number of connected ONUs).

Logical links also provide a solution for privacy of data, which otherwise would be shared by all subscribers connected to a single OLT port. In this way, each subscriber is isolated and restricted to accessing data streams addressed only to that particular subscriber.

This concept is illustrated in Figure 9-3, which shows an example of an EPON with a single OLT and three connected ONUs.

The single copy broadcast channel (addressed with a special, reserved LLID, see 65.1.3.1 of IEEE Std 802.3) was added to take advantage of the broadcast transmission capability of the underlying physical medium. In this way, it is very simple and very bandwidth efficient to deliver broadcast content to all ONUs at the same time, avoiding the need to replicate data into a series of P2P links.

1

2

3

4

5

6

7

8

9

Editor’s Note (to be removed prior to publication):

Reference to IEEE Std 802.1D was replaced with IEEE Std 802.1Q (text and Figure 9-3) per Maintenance Request 1383 (see https://www.ieee802.org/3/maint/requests/maint_1383.pdf)

10

11

12

13

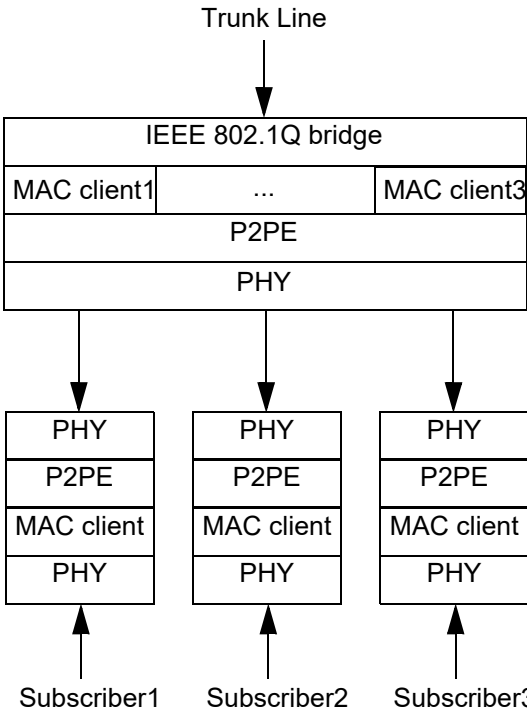
14

15

16

17

The ONUs filter all downstream data and drop all frames addressed to other devices. Only broadcast frames and frames with correct unicast logical link ID (LLID) are admitted and processed. The LLID replaces two octets of the Ethernet frame preamble, identifying a logical link established between the OLT and the given ONU during the discovery and registration process. The LLID indicates the destination port in the downstream and the source port in the upstream. The logical links are used effectively to prevent EPON from violating the IEEE 802.1Q bridging rules.



53

54

55

56

57

58

59

60

61

62

63

64

65

Figure 9-3—Example of point-to-point emulation used between an OLT and three ONUs

9.1.1.6 Principles of the MPCP

56

57

58

59

60

61

62

63

64

65

The EPON standard comprises a mechanism for media access control, referred to as the Multipoint Control Protocol (MPCP). An access network architecture is different from a typical LAN environment, primarily in terms of network provisioning. An access network is an administrated environment, with an operator providing services and subscribers consuming it depending on service provisioning contracts. The operator controls the network, manages traffic and medium access, and enforces the service level agreements (SLAs). For instance, the available bandwidth is controlled and subscribers may be billed for services. In this sense, the access network (and EPON specifically) requires a media access control protocol that provides a mechanism for station discovery and registration as well as bandwidth provisioning capabilities.

1 In the MPCP, the OLT is considered to be the master, controlling a series of connected ONUs (slave
2 devices). The OLT manages the network and controls access to network resources from individual slave
3 devices. The MPCP is also used for provisioning upstream channel access to individual slave devices via a
4 MPCPDU pair, i.e., GATE and REPORT. The MPCP is part of the MAC control layer, and MPCPDUs are
5 considered MAC control messages, carrying a specific Ethertype of 0x8808. These messages are not
6 forwarded outside of the EPON domain and are used to manage the EPON link only.
7
8

9
10 A concept of time exists in the MPCP in order to schedule the upstream transmission. A timestamp, which is
11 transmitted in the MPCPDUs downstream by the OLT and received by the connected ONUs, is used to
12 synchronize slave devices to the master device clock. This coordinates upstream transmissions from
13 individual ONUs so that the transmissions arrive at the OLT at precisely the anticipated time, and thus, data
14 from different ONUs does not overlap.
15

16
17 The MPCP plane is also used to measure the round-trip time (RTT) for each connected ONU. Each
18 MPCPDU carries a generalized timestamp field, which is filled in by the transmitting station with the
19 current value of its MPCP clock at the time when the given MPCPDU is transmitted. The RTT is measured
20 first during the discovery and registration process and then updated regularly upon each exchange of
21 MPCPDUs between the OLT and one of the ONUs. RTT is used by the OLT bandwidth scheduler to
22 schedule upstream transmission slots for individual ONUs in a non-overlapping manner. The IEEE 802.3
23 EPON standard provides support for the network diameter (distance between the OLT and the farthest ONU)
24 of nominally up to 20 km, which corresponds to the RTT of approximately 200 μ s. However, nothing in the
25 standard precludes support for larger network diameters.
26
27

28
29 The TDMA control is performed using a pair of MPDPUs, namely GATE generated by the OLT to indicate
30 a future transmission opportunity to an ONU and REPORT generated by the ONU with information on the
31 current queue status (bandwidth demand). Internal structure and possible encoding of GATE and REPORT
32 MPCPDUs are defined in Clause 64 of IEEE Std 802.3.
33

34
35 A scheduling algorithm at the OLT, which is not defined in IEEE Std 802.3, is responsible for dividing the
36 bandwidth and controlling the transmission delay of each ONU according to its SLA. The MPCP defines a
37 closed loop operation in order for this algorithm to be efficient. The MPCP allows the ONUs to report on the
38 amount of bandwidth they require for transmission using a special REPORT message. This allows allocating
39 bandwidth to an ONU only when requested, relying on the statistical burst property of the traffic, and
40 allowing different peak bandwidths for different ONUs at different times, hence, allowing oversubscription
41 of the bandwidth. The REPORT message reports the amount of data waiting in the ONU queues.
42
43

44 In addition, the MPCP defines a protocol of auto-discovery and registration of ONUs.
45

46
47 The MPCP registration process is presented in Figure 9-4, while details are described in Clause 64 of
48 IEEE Std 802.3.
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

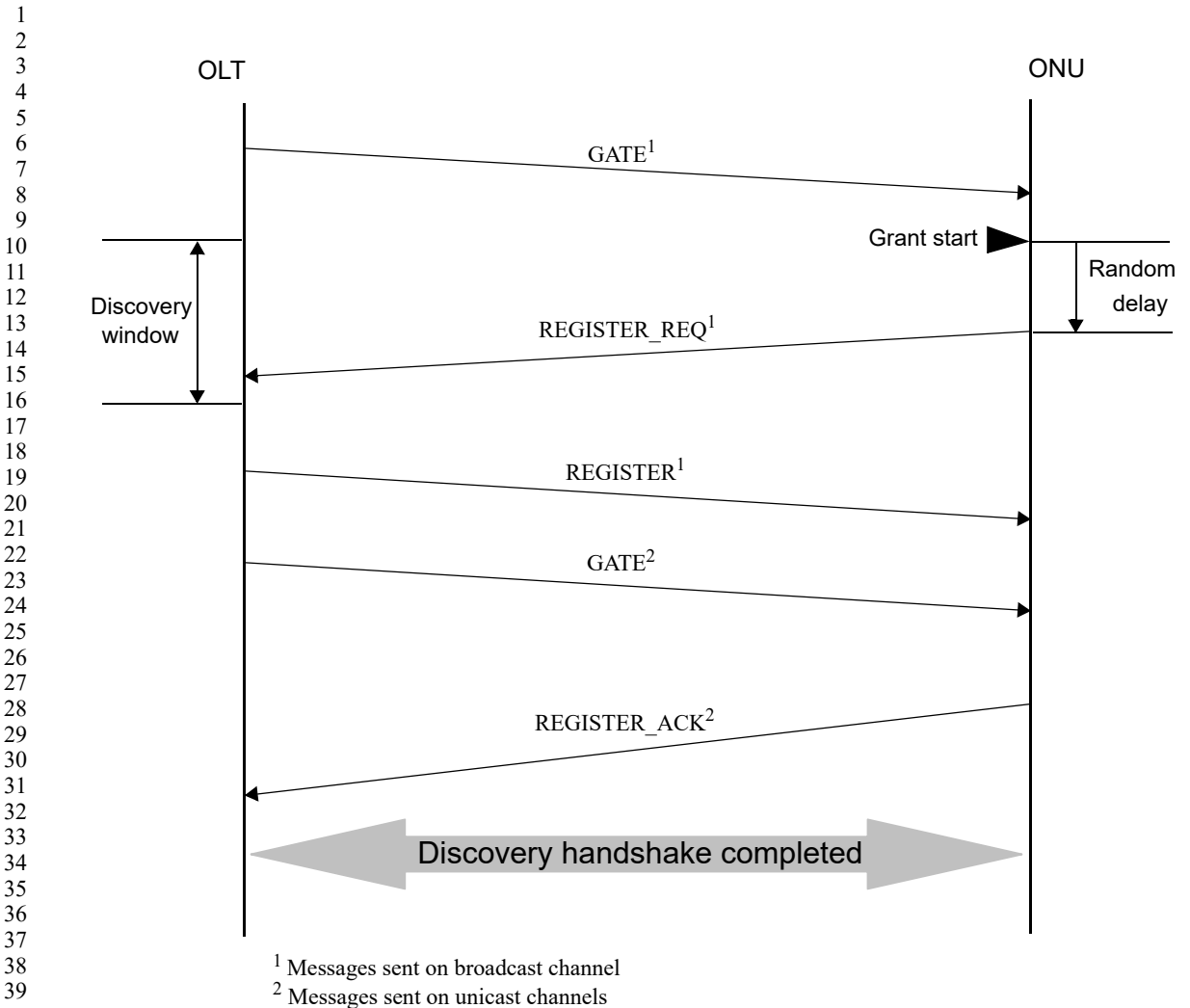


Figure 9-4—Discovery handshake message exchange

A new ONU requests to register during a special upstream window (called Discovery Window), sending the REGISTER_REQ MPCPDU. More than one ONU may attempt registration during that window, which means that their REGISTER_REQ MPCPDUs can potentially collide at the OLT receiver, since the ONU-specific RTT is not yet known and transmissions from individual ONUs cannot be scheduled in 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 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 a network or not. The registration process is completed with the ONU sending REGISTER_ACK MPCPDU to the OLT, confirming assigned parameters and registration in the network. From that point onward, the OLT can schedule transmissions from that ONU using its LLID, using the measured RTT 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.

9.1.1.7 Forward error correction (FEC)

The optional FEC mechanism is defined to enhance the EPON link budget. All the passive components of the fiber plant attenuate the optical signal; thus, the target distance (network diameter) and the number of supported splits are limited by the available link budget. The optional FEC mechanism increases the available link budget by improving the link BER from 10^{-4} to 10^{-12} (the target BER at the MAC), effectively increasing the target network diameter and/or split ratio. The target use of the increased power budget remains at the sole discretion of the network architects and is out of the scope of IEEE Std 802.3.

The optional FEC used in EPON is frame-based, meaning that parity information is added at the end of each Ethernet packet. Extra space between individual Ethernet packets is 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 9-5 presents the structure of an FEC-protected EPON frame.

The optional FEC function is added to the extended Gigabit Ethernet PCS per 65.2 in IEEE Std 802.3. The added, optional FEC function introduces a fixed delay in the receive path and in the transmit path.

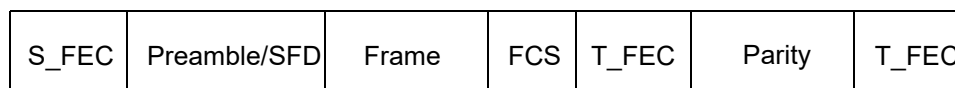


Figure 9-5—FEC-protected frame

9.1.2 Management architecture

All EPON layers are accompanied by a management interface that is controlled through mechanisms defined in Clause 30 of IEEE Std 802.3. Since IEEE Std 802.3 specifications may be used for different applications (and hence are extensible), and some of the clauses may be used separately, the management clause allocates a separate package for each independent layer. The structure of the MIB modules follows this separation.

Figure 9-6 presents the relation of the MIB module groups to the individual IEEE 802.3 layers.

Editor's Note (to be removed prior to publication):

Reference to IEEE Std 802.1D was replaced with IEEE Std 802.1Q (Figure 9-6) per Maintenance Request 1383 (see https://www.ieee802.org/3/maint/requests/maint_1383.pdf)

The association is straightforward for the ONU interface. There is one logical and one physical interface, and a single copy of each layer can be remotely queried by the OLT.

The OLT has a single physical interface and N logical interfaces, one for each logical link connected to an ONU. There is also one logical interface for the single copy broadcast link. Per layering diagram in Figure 9-6, the MAC sublayer is virtually replicated. Therefore, in this clause it was elected that management of logical interfaces is performed in the manner identical to management of any physical interfaces—an interface index is allocated for each one of the logical links, and an additional interface index is allocated for the OLT.

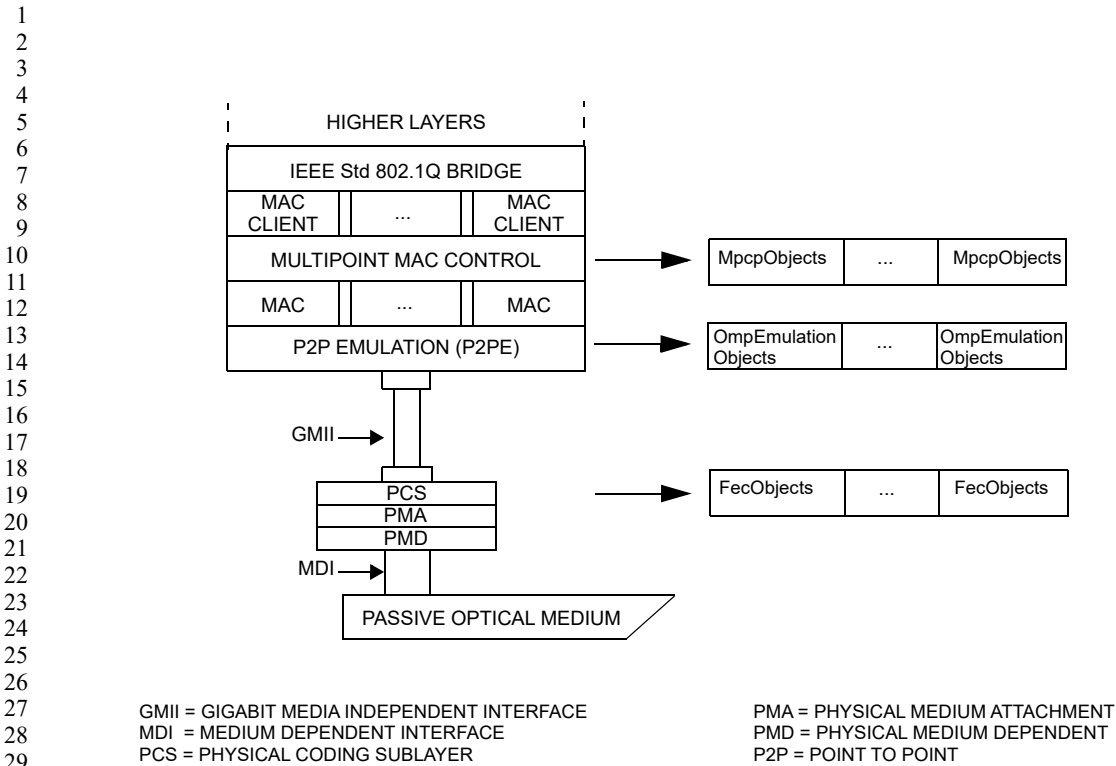


Figure 9-6—Relationship of the MIB groups to the EPON sublayers

For each physical interface, there would be an entry (ifIndex) in the tables of the interface MIB module defined in IETF RFC 2863, the MAU MIB module defined in Clause 13, and the Ethernet-like MIB module defined in Clause 10. Additionally, there would be entries (ifIndexes) for the virtual interfaces of the OLT interface. The justification for the additional allocation of indexes is that the virtual interfaces are quite well distinguished, as they connect different physical ONUs from the OLT side. For instance, there is a meaning for separate bad frames counter or bad octets counter for each virtual link, as the ONUs can be differently distanced. This is quite similar to a case of separate physical interfaces.

The same partition concept exists for the MIB module of this clause. Each row in the tables is indexed according to the ifIndex; specifically, there is a row for each virtual link. There are some control objects that are shared and are the same for the virtual interfaces (and they should have the same value for each ifIndex), but most of the objects have different values for N+1 logical interfaces at the OLT. This is done for each MIB group. It is different from the EPON layering diagram, which presents the P2MP layer as a single layer, while duplicating the MAC and MAC client layers (please see Figure 9-6). However, from a management perspective, it is more convenient to partition the management of the layers for the virtual links, as the atomic managed entity is the virtual link. It is also convenient to use the interface index of the virtual link for that purpose, as it is already used to index the rows of the virtual links at the Interface, MAU, and Ethernet-like interface MIBs.

9.2 MIB structure

This subclause defines the DOT3 EPON MIB module. The DOT3 EPON MIB module defines the objects used for management of the IEEE Std 802.3 EPON interfaces. These MIB objects are included in the following four groups:

- a) MPCP MIB objects—MIB objects related to Clause 64 of IEEE Std 802.3, Multipoint Control Protocol attributes. The following tables are presented in this group:
 - 1) The dot3MpcpControlTable defines the objects used for the configuration and status indication, which are per logical link, of MPCP compliant interfaces.
 - 2) The dot3MpcpStatTable defines the statistics objects that are per logical link, of MPCP compliant interfaces.
 - 3) The operational mode of an OLT/ONU for the tables is defined by the dot3MpcpMode object in the dot3MpcpControlTable.
- b) The OMPEmulation MIB objects—MIB objects related to Clause 65 of IEEE Std 802.3, point-to-point emulation attributes. The following tables are presented in this group:
 - 1) The dot3OmpEmulationTable defines the objects used for the configuration and status indication, which are per logical links, of OMPEmulation compliant interfaces.
 - 2) The dot3OmpEmulationStatTable defines the statistics objects that are per logical link, of OMPEmulation compliant interfaces.
 - 3) The operational mode of an OLT/ONU for the tables is defined by the dot3OmpEmulationType object in the dot3OmpEmulationTable.
- c) The FEC MIB objects—MIB objects related to Clause 60 and Clause 65 of IEEE Std 802.3, EPON FEC attributes. The following table is presented in this group:
 - 1) The dot3EponFecTable defines the objects used for the configuration and status indication, which are per logical link, of FEC EPON compliant interfaces.
- d) The EPON extended package MIB objects—MIB objects used for configuration and status indication with extended capabilities of the EPON interfaces. The following tables are presented in this group:
 - 1) The dot3ExtPkgControlTable defines the objects, which are per logical link, used for the configuration and status indication of EPON compliant interfaces.
 - 2) The dot3ExtPkgQueueTable defines the objects, which are per logical link, and per queue, used for the configuration and status indication of the ONU queues reported in the MPCP REPORT message, of EPON compliant interfaces.
 - 3) The dot3ExtPkgQueueSetsTable defines the objects, which are per logical link, per queue, and per queue_set, used for the configuration and status indication of the ONU queue_sets reported in the MPCP REPORT message, of EPON compliant interfaces.
 - 4) The dot3ExtPkgOptIfTable defines the objects, which are per logical link, used for the control and status indication of the optical interface of EPON compliant interfaces.

The interface MIB module defined in IETF RFC 2863 defines the interface index (ifIndex). Interface Index, as specified in IETF RFC 2863, is used in this MIB module as an index to the EPON MIB tables. The ifIndex is used to denote the physical interface and the virtual link interfaces at the OLT. The OLT interface and the virtual link interfaces are stacked using the ifStack table defined in IETF RFC 2863 and the ifInvStack defined in IETF RFC 2864. The OLT interface is the lower layer of all other interfaces associated with the virtual links.

As described in 9.1.2, each row in the tables is indexed according to the ifIndex; specifically, there is a row for each virtual link. There are a few control objects that are shared and have the same value for the virtual interfaces (and they should have the same value for each ifIndex), but most of the objects have different values for N+1 logical interfaces at the OLT. This is done for each MIB group. It is a bit different from the EPON layering diagram, which presents the P2MP layer as a single layer while duplicating the MAC and MAC client layers. However, from a management perspective, it is more convenient to partition the management of the layers for the virtual links, as the atomic managed entity is the virtual link. It is also convenient to use the interface index of the virtual link for that purpose, as it is already used to index the rows of the virtual links at the Interface, MAU, and Ethernet-like interface MIB modules.

The creation of the rows of the ONU interface is done at initialization. Table 9-1 presents the MPCP control table of ONU1 after initialization. A single row exists in the table.

Table 9-1—MPCP control table of ONU1 after initialization

MPCP control MIB object	Value
ifIndex	100
dot3MpcpOperStatus	true
dot3MpcpAdminState	true
dot3MpcpMode	onu
dot3MpcpSyncTime	0
dot3MpcpLinkID	0
dot3MpcpRemoteMACAddress	00:00:00:00:00:00
dot3MpcpRegistrationState	unregistered
dot3MpcpTransmitElapsed	0
dot3MpcpReceiveElapsed	0
dot3MpcpRoundTripTime	0

Table 9-2 presents the MPCP control table of ONU1 in working mode. A single row exists in the table.

Table 9-2—MPCP control table of ONU1 in working mode

MPCP control MIB object	Value
ifIndex	100
dot3MpcpOperStatus	true
dot3MpcpAdminState	true
dot3MpcpMode	onu
dot3MpcpSyncTime	25
dot3MpcpLinkID	1
dot3MpcpRemoteMACAddress	OLT_MAC_Address ^a
dot3MpcpRegistrationState	registered
dot3MpcpTransmitElapsed	10
dot3MpcpReceiveElapsed	10
dot3MpcpRoundTripTime	100

^aOLT_MAC_Address is the MAC address of the OLT EPON interface.

The creation of the rows of the OLT interface and the broadcast virtual interface is done at initialization.

The creation of rows of the virtual interfaces at the OLT is done when the link is established (ONU registers) and the deletion is done when the link is deleted (ONU deregisters).

Table 9-3 presents the MPCP control table of the OLT after initialization, before the ONUs register. A single row exists in this table associated with the virtual broadcast link.

Table 9-3—MPCP control table of the OLT after initialization

MPCP control MIB object	Value
ifIndex	165535
dot3MpcpOperStatus	true
dot3MpcpAdminState	true
dot3MpcpMode	olt
dot3MpcpSyncTime	25
dot3MpcpLinkID	65535
dot3MpcpRemoteMACAddress	BRCT_MAC_Address ^a
dot3MpcpRegistrationState	registered
dot3MpcpTransmitElapsed	10
dot3MpcpReceiveElapsed	100000
dot3MpcpRoundTripTime	0

^aBRCT_MAC_Address is the MAC address of the broadcast EPON interface, which is the OLT MAC address.

Table 9-4 presents the MPCP control table of the OLT in working mode. Three rows exist in the table associated with the virtual links.

9.3 Relationship to other MIB modules

9.3.1 Relation to the Interfaces Group MIB and Ethernet-like interface MIB

This MIB module extends the objects of the Interfaces Group MIB and the Ethernet-like interface MIB for the EPON type interface. Therefore, if this module is implemented, the Interfaces Group MIB module

Table 9-4—MPCP control table of the OLT in working mode

MPCP control MIB object	Value	Value	Value
ifIndex	100001	100002	165535
dot3MpcpOperStatus	true	true	true
dot3MpcpAdminState	true	true	true
dot3MpcpMode	olt	olt	olt
dot3MpcpSyncTime	25	25	25
dot3MpcpLinkID	1	2	65535
dot3MpcpRemote MACAddress	ONU1_MAC_Address ^a	ONU2_MAC_Address ^b	BRCT_MAC_Address ^c
dot3MpcpRegistrationState	registered	registered	registered
dot3MpcpTransmitElapsed	10	10	10
dot3MpcpReceiveElapsed	10	10	10
dot3MpcpRoundTripTime	100	60	0

^aONU1_MAC_Address is the MAC address of the ONU1 EPON interface.

^bONU2_MAC_Address is the MAC address of the ONU2 EPON interface.

^cBRCT_MAC_Address is the MAC address of the broadcast EPON interface, which is the OLT MAC address.

defined in IETF RFC 2863 and the Ethernet-like interface MIB module defined in Clause 10 shall also be implemented.

Thus, each managed EPON interface would have a corresponding entry in the mandatory tables of the Ethernet-like MIB module found in Clause 10, and likewise in the tables of the Interfaces Group MIB module found in IETF RFC 2863. Also, each managed virtual EPON interface would have a corresponding entry in the mandatory tables of the Ethernet-like MIB module found in Clause 10, and likewise in the tables of the Interfaces Group MIB module found in IETF RFC 2863 with a dedicated ifIndex for this interface.

In this clause, there is no replication of the objects from these MIBs. Therefore, for instance, the clause is defining the dot3MpcpRemoteMACAddress only while assuming that the local MAC address object is already defined in Clause 10.

This clause defines the specific EPON objects of an ONU interface and an OLT interface. Information in the tables is per LLID. The rows in the EPON MIB tables referring to the LLIDs are denoted with the corresponding ifIndexes of the virtual link interfaces.

Note that all virtual interfaces have the same physical MAC address at the OLT since the physical OLT interface used by all virtual interfaces is the same. The value of this physical MAC interface is specified in 64.1.2 of IEEE Std 802.3. The corresponding object of the Ethernet-like interface MIB is replicated for all virtual interfaces.

For example, the values of the Interfaces Group MIB objects are presented in the following tables, for an OLT with three registered ONUs.

Table 9-5 presents the objects of the Interfaces Group MIB of an ONU in working mode.

Table 9-5—Interfaces Group MIB of an ONU in working mode

Interfaces Group MIB object	Value
ifIndex	1
ifDescr	“interface description”
ifType	ethernetCsmacd (6) 1000base-Px
ifMtu	MTU size (1522)
ifSpeed	1000000000
ifPhysAddress	ONU_MAC_Address ^a
ifAdminStatus	up
ifOperStatus	Up
ifLastChange	ONUup_time
ifInOctets	ONU_octets_number
ifInUcastPkts	ONU_unicast_frame_number
ifInNUcastPkts	ONU_non_unicast_frame_number
ifInDiscards	ONU_discard_frame_number
ifInErrors	ONU_error_frame_number
ifInUnknownProtos	ONU_unknown_frame_number
ifOutOctets	ONU_octets_number
ifOutUcastPkts	ONU_unicast_frame_number
ifOutNUcastPkts	ONU_non_unicast_frame_number
ifOutDiscards	ONU_discard_frame_number
ifOutErrors	ONU_error_frame_number
ifOutQLen	ONU_queue_frame_number

^aONU_MAC_Address is the MAC address of the ONU EPON interface.

Table 9-6 presents the objects of the Interfaces Group MIB of the ONU interface.

Table 9-6—Interfaces Group MIB of the ONU interface

Interfaces Group MIB object	Value
ifIndex	100
ifDescr	“interface description”
ifType	ethernetCsmacd (6) 1000base-Px
ifMtu	MTU size (1522)
ifSpeed	1000000000
ifPhysAddress	ONU_MAC_Address ^a
ifAdminStatus	up
ifOperStatus	Up
ifLastChange	up_time
ifInOctets	ONU1_octets_number
ifInUcastPkts	ONU1_unicast_frame_number
ifInNUcastPkts	ONU1_non_unicast_frame_number
ifInDiscards	ONU1_discard_frame_number
ifInErrors	ONU1_error_frame_number
ifInUnknownProtos	ONU1_unknown_frame_number
ifOutOctets	ONU1_octets_number
ifOutUcastPkts	ONU1_unicast_frame_number
ifOutNUcastPkts	ONU1_non_unicast_frame_number
ifOutDiscards	ONU1_discard_frame_number
ifOutErrors	ONU1_error_frame_number
ifOutQLen	ONU1_queue_frame_number

^aONU_MAC_Address is the MAC address of the ONU EPON interface.

The following values will be set in the ifStack and ifInvStack tables related to this example.

ifStackTable:

- ifStackHigherLayer = 100, ifStackLowerLayer = 1 – map between the physical interface and the ONU

ifInvStackTable:

- ifStackLowerLayer = 1, ifStackHigherLayer = 100 – map between the ONU and the physical interface

Table 9-7 presents the Interfaces Group MIB objects of an OLT interface.

Table 9-7—Interfaces Group MIB objects of an OLT interface

Interfaces Group MIB object	Value
ifIndex	2
ifDescr	“interface description”
ifType	ethernetCsmacd (6) 1000base-Px
ifMtu	MTU size (1522)
ifSpeed	1000000000
ifPhysAddress	OLT_MAC_Address ^a
ifAdminStatus	up
ifOperStatus	Up
ifLastChange	OLTup_time
ifInOctets	OLT_octets_number
ifInUcastPkts	OLT_unicast_frame_number
ifInNUcastPkts	OLT_non_unicast_frame_number
ifInDiscards	OLT_discard_frame_number
ifInErrors	OLT_error_frame_number
ifInUnknownProtos	OLT_unknown_frame_number
ifOutOctets	OLT_octets_number
ifOutUcastPkts	OLT_unicast_frame_number
ifOutNUcastPkts	OLT_non_unicast_frame_number

Table 9-7—Interfaces Group MIB objects of an OLT interface (continued)

Interfaces Group MIB object	Value
ifOutDiscards	OLT_discard_frame_number
ifOutErrors	OLT_error_frame_number
ifOutQLen	OLT_queue_frame_number

^aOLT_MAC_Address is the MAC address of the OLT EPON interface.

Table 9-8 presents the Interfaces Group MIB objects of an OLT interface, associated with the virtual link interfaces.

Table 9-8—Interfaces Group MIB objects of an OLT interface, associated with the virtual link interfaces

Interface MIB object	Value	Value	Value
ifIndex	200001	200002	265535
ifDescr	“interface description”	“interface description”	“interface description”
ifType	ethernetCsmacd (6)	ethernetCsmacd (6)	ethernetCsmacd (6)
ifMtu	MTUsize(1522)	MTUsize(1522)	MTUsize(1522)
ifSpeed	1000000000	1000000000	1000000000
ifPhysAddress	OLT_MAC_Address ^a	OLT_MAC_Address	OLT_MAC_Address
ifAdminStatus	up	up	up
ifOperStatus	Up	Up	Up
ifLastChange	ONU1_up_time	ONU2_up_time	up_time
ifInOctets	ONU1_octets_number	ONU2_octets_number	BRCT_octets_number
ifInUcastPkts	ONU1_unic_frame_num	ONU2_unic_frame_num	BRCT_unic_frame_num
ifInNUcastPkts	ONU1_non_unic_frame_num	ONU2_non_unic_frame_num	BRCT_non_unic_frame_num
ifInDiscards	ONU1_disc_frame_num	ONU2_disc_frame_num	BRCT_disc_frame_numr
ifInErrors	ONU1_err_frame_num	ONU2_err_frame_num	BRCT_err_frame_num
ifInUnknownProtos	ONU1_unknw_frame_num	ONU2_unknw_frame_num	BRCT_unknw_frame_num
ifOutOctets	ONU1_octets_number	ONU2_octets_number	BRCT_octets_number
ifOutUcastPkts	ONU1_unic_frame_num	ONU2_unic_frame_num	BRCT_unic_frame_num
ifOutNUcastPkts	ONU1_non_unic_frame_num	ONU2_non_unic_frame_num	BRCT_non_unic_frame_num

Table 9-8—Interfaces Group MIB objects of an OLT interface, associated with the virtual link interfaces (*continued*)

Interface MIB object	Value	Value	Value
ifOutDiscards	ONU1_disc_frame_num	ONU2_disc_frame_num	BRCT_disc_frame_num
ifOutErrors	ONU1_err_frame_num	ONU2_err_frame_num	BRCT_err_frame_num
ifOutQLen	ONU1_queue_frame_num	ONU2_queue_frame_num	BRCT_queue_frame_num

^aOLT_MAC_Address is the MAC address of the OLT EPON interface.

The following values will be set in the ifStack and ifInvStack tables related to this example:

ifStackTable:

- ifStackHigherLayer = 265535, ifStackLowerLayer = 2 – map between the OLT physical interface and its broadcast virtual interface
- ifStackHigherLayer = 200001, ifStackLowerLayer = 2 – map between the OLT physical interface and its virtual interface of the 1st ONU
- ifStackHigherLayer = 200002, ifStackLowerLayer = 2 – map between the OLT physical interface and its virtual interface of the 2nd ONU
- ifStackHigherLayer = 200003, ifStackLowerLayer = 2 – map between the OLT physical interface and its virtual interface of the 3rd ONU

ifInvStackTable:

- ifStackLowerLayer = 2, ifStackHigherLayer = 265535 – map between the broadcast interface of the OLT and the OLT physical interface
- ifStackLowerLayer = 2, ifStackHigherLayer = 200001 – map between the OLT virtual interface of the 1st ONU and the OLT physical interface
- ifStackLowerLayer = 2, ifStackHigherLayer = 200002 – map between the OLT virtual interface of the 2nd ONU and the OLT physical interface
- ifStackLowerLayer = 2, ifStackHigherLayer = 200003 – map between the OLT virtual interface of the 3rd ONU and the OLT physical interface

The rows for the ONU interface, the OLT interface, and the OLT broadcast interface are created in initialization. The creation of a row for a virtual link is done when the virtual link is established (ONU registers), and deletion is done when the virtual link is deleted (ONU deregisters).

The EPON MIB module also extends the Interfaces Group MIB module with a set of counters, which are specific for the EPON interface. The EPON MIB module implements the same handling of the counters when the operation of the interface starts or stops. The interface MIB clause describes the possible behavior of counters when an interface is re-initialized using the ifCounterDiscontinuityTime indicator, indicating the discontinuity of the counters. See Section 3.1.5 of IETF RFC 2863 for more information. The counters of the EPON MIB should be handled in a similar manner.

9.3.2 Relation to the IEEE 802.3 MAU MIBs

The MAU types of the EPON Interface are defined in Clause 13. This clause assumes the implementation of the MAU MIB for this purpose and does not repeat the EPON MAU types. Therefore, if this module is implemented, the MAU-MIB module defined in Clause 13 shall also be implemented.

The handling of the ifMAU tables for the EPON case is similar to the handling described in the former subclause for the Interface and Ethernet-like interface MIBs. A single row exists for the ONU in the ifMauTable. A row for each virtual link (N+1 rows) exists at the OLT, with a separate value of ifMauIfIndex for each virtual link.

As specified above, the rows for the ONU interface, the OLT interface, and the OLT broadcast interface are created in initialization. The creation of a row for a virtual link is done when the virtual link is established (ONU registers), and deletion is done when the virtual link is deleted (ONU deregisters).

9.3.3 Relation to the Ethernet OAM MIB

The EPON interfaces are intended for use in optical subscriber access networks and most probably will be accompanied with the implementation of the OAM protocol defined in Clause 57 of IEEE Std 802.3. Therefore, the Ethernet OAM MIB module defined in Clause 6 may be implemented when this MIB module is implemented defining managed objects for the OAM protocol that are complementary to the EPON MIB module.

9.3.4 Relation to the bridge MIB

Editor's Note (to be removed prior to publication):

Reference to IEEE Std 802.1D was replaced with IEEE Std 802.1Q per Maintenance Request 1383 (see https://www.ieee802.org/3/maint/requests/maint_1383.pdf)

It is very probable that an EPON OLT will implement a bridging functionality above the EPON interface layer, bridging between the EPON users and the network. Bridge functionality is specified in IEEE Std 802.1Q. In this scenario, the virtual ports of the EPON are corresponding to the virtual bridge ports. There is a direct mapping between the bridge ports and the LLIDs, which are virtual EPON channels.

Therefore, the bridge MIB modules defined in IEEE Std 802.1Q [B5] may be implemented when the EPON MIB module is implemented for an EPON OLT, defining managed objects for the bridge layer.

The values of dot1dBasePortIfIndex would correspond to the ifIndex of the virtual port (1 for LLID1, 2 for LLID2, etc.).

The broadcast virtual EPON interface of the OLT has no direct mapping to a virtual bridge port as it is not port specific but used for broadcast traffic.

9.4 Mapping of IEEE 802.3 managed objects

This subclause contains the mapping between the managed objects defined in this clause and the attributes defined in Clause 30 of IEEE Std 802.3. Table 9-9 provides the mapping between the dot3EPON MIB module MPCP objects and the MPCP attributes of Clause 30 of IEEE Std 802.3.

Table 9-10 provides the mapping between the dot3EPON MIB module OMPEmulation objects and the OMPE attributes of Clause 30 of IEEE Std 802.3.

Table 9-9—oMPCP managed object class (30.3.5 of IEEE Std 802.3)

dot3EPON MIB module object	IEEE 802.3 attribute	Reference
ifIndex	aMPCPID	30.3.5.1.1
dot3MpcpOperStatus	aMPCPAdminState	30.3.5.1.2
dot3MpcpMode	aMPCPMode	30.3.5.1.3
dot3MpcpLinkID	aMPCPLinkID	30.3.5.1.4
dot3MpcpRemoteMACAddress	aMPCPRemoteMACAddress	30.3.5.1.5
dot3MpcpRegistrationState	aMPCPRegistrationState	30.3.5.1.6
dot3MpcpMACCtrlFramesTransmitted	aMPCPMACCtrlFramesTransmitted	30.3.5.1.7
dot3MpcpMACCtrlFramesReceived	aMPCPMACCtrlFramesReceived	30.3.5.1.8
dot3MpcpTxGate	aMPCPTxGate	30.3.5.1.9
dot3MpcpTxRegAck	aMPCPTxRegAck	30.3.5.1.10
dot3MpcpTxRegister	aMPCPTxRegister	30.3.5.1.11
dot3MpcpTxRegRequest	aMPCPTxRegRequest	30.3.5.1.12
dot3MpcpTxReport	aMPCPTxReport	30.3.5.1.13
dot3MpcpRxGate	aMPCPRxGate	30.3.5.1.14
dot3MpcpRxRegAck	aMPCPRxRegAck	30.3.5.1.15
dot3MpcpRxRegister	aMPCPRxRegister	30.3.5.1.16
dot3MpcpRxRegRequest	aMPCPRxRegRequest	30.3.5.1.17
dot3MpcpRxReport	aMPCPRxReport	30.3.5.1.18
dot3MpcpTransmitElapsed	aMPCPTransmitElapsed	30.3.5.1.19
dot3MpcpReceiveElapsed	aMPCPReceiveElapsed	30.3.5.1.20
dot3MpcpRoundTripTime	aMPCPRoundTripTime	30.3.5.1.21
dot3MpcpDiscoveryWindowsSent	aMPCPDiscoveryWindowsSent	30.3.5.1.22
dot3MpcpDiscoveryTimeout	aMPCPDiscoveryTimeout	30.3.5.1.23
dot3MpcpMaximumPendingGrants	aMPCPMaximumPendingGrants	30.3.5.1.24
dot3MpcpAdminState	aMPCPAdminControl	30.3.5.2.1
dot3MpcpSyncTime	SyncTime	64.3.3.2

Table 9-10—oOMPEmulation managed object class (30.3.7 of IEEE Std 802.3)

dot3EPON MIB module object	IEEE 802.3 attribute	Reference
ifIndex	aOMPEmulationID	30.3.7.1.1
dot3OmpEmulationType	aOMPEmulationType	30.3.7.1.2
dot3OmpEmulationSLDErrors	aSLDErrors	30.3.7.1.3
dot3OmpEmulationCRC8Errors	aCRC8Errors	30.3.7.1.4
dot3OmpEmulationGoodLLID	aGoodLLID	30.3.7.1.5
dot3OmpEmulationOnuPonCastLLID	aONUPONcastLLID	30.3.7.1.6
dot3OmpEmulationOltPonCastLLID	aOLTPONcastLLID	30.3.7.1.7
dot3OmpEmulationBadLLID	aBadLLID	30.3.7.1.8
dot3OmpEmulationBroadcastBitNotOnuLLid	N/A	—
dot3OmpEmulationOnuLLIDNotBroadcast	N/A	—
dot3OmpEmulationBroadcastBitPlusOnuLlid	N/A	—
dot3OmpEmulationNotBroadcastBitNotOnuLlid	N/A	—

Table 9-11 provides the mapping between the dot3EPON MIB module FEC objects and the MAU attributes of Clause 30 of IEEE Std 802.3.

Table 9-11—oMAU managed object class (30.5.1 of IEEE Std 802.3)

dot3EPON MIB module object	IEEE 802.3 attribute	Reference
dot3EponFecPCSCodingViolation	aPCSCodingViolation	30.5.1.1.14
dot3EponFecAbility	aFECAbility	30.5.1.1.15
dot3EponFecMode	aFECmode	30.5.1.1.16
dot3EponFecCorrectedBlocks	aFECCorrectedBlocks	30.5.1.1.17
dot3EponFecUncorrectableBlocks	aFECUncorrectableBlocks	30.5.1.1.18
dot3EponFecBufferHeadCodingViolation	N/A	—

9.5 Security considerations for Ethernet passive optical network (EPON) MIB module

There are number of managed objects defined in this MIB module that have a MAX-ACCESS clause of read-write or read-create. Writing to these objects can have potentially disruptive effects on network operation, including those listed in 9.5.1 to 9.5.13.

9.5.1 dot3MpcpAdminState

Changing the dot3MpcpAdminState state can lead to disabling the Multipoint Control Protocol on the respective interface, leading to the interruption of service for the users connected to the respective EPON interface.

9.5.2 dot3EponFecMode

Changing the dot3EponFecMode state can lead to disabling the Forward Error Correction on the respective interface, which can lead to a degradation of the optical link, and therefore, it may lead to an interruption of service for the users connected to the respective EPON interface.

9.5.3 dot3ExtPkgObjectReset

Changing the dot3ExtPkgObjectReset state can lead to a reset of the respective interface leading to an interruption of service for the users connected to the respective EPON interface.

9.5.4 dot3ExtPkgObjectPowerDown

Changing the dot3ExtPkgObjectPowerDown state can lead to a power down of the respective interface, leading to an interruption of service for the users connected to the respective EPON interface.

9.5.5 dot3ExtPkgObjectFecEnabled

Changing the dot3ExtPkgObjectFecEnabled state can lead to disabling the Forward Error Correction on the respective interface, which can lead to a degradation of the optical link, and therefore, it may lead to an interruption of service for the users connected to the respective EPON interface.

9.5.6 dot3ExtPkgObjectRegisterAction

Changing the dot3ExtPkgObjectRegisterAction state can lead to a change in the registration state of the respective interface, leading to a deregistration and an interruption of service for the users connected to the respective EPON interface.

9.5.7 dot3ExtPkgObjectReportNumThreshold

Changing the dot3ExtPkgObjectReportNumThreshold can lead to a change in the reporting of the ONU interface and therefore to a change in the bandwidth allocation of the respective interface. This change may lead to a degradation or an interruption of service for the users connected to the respective EPON interface.

9.5.8 dot3ExtPkgObjectReportThreshold

Changing the dot3ExtPkgObjectReportThreshold can lead to a change in the reporting of the ONU interface and therefore to a change in the bandwidth allocation of the respective interface. This change may lead to a degradation or an interruption of service for the users connected to the respective EPON interface.

9.5.9 dot3ExtPkgOptIfLowerInputPowerThreshold

Changing the dot3ExtPkgOptIfLowerInputPowerThreshold can lead to a Threshold Crossing Alert (TCA) being sent for the respective interface. This alert may be leading to an interruption of service for the users connected to the respective EPON interface, depending on the system action on such an alert.

9.5.10 dot3ExtPkgOptIfUpperInputPowerThreshold

Changing the dot3ExtPkgOptIfUpperInputPowerThreshold can lead to a Threshold Crossing Alert (TCA) being sent for the respective interface. This alert may be leading to an interruption of service for the users connected to the respective EPON interface, depending on the system action on such an alert.

9.5.11 dot3ExtPkgOptIfLowerOutputPowerThreshold

Changing the dot3ExtPkgOptIfLowerOutputPowerThreshold can lead to a Threshold Crossing Alert (TCA) being sent for the respective interface. This alert may be leading to an interruption of service for the users connected to the respective EPON interface, depending on the system action on such an alert.

9.5.12 dot3ExtPkgOptIfUpperOutputPowerThreshold

Changing the dot3ExtPkgOptIfUpperOutputPowerThreshold can lead to a Threshold Crossing Alert (TCA) being sent for the respective interface. This alert may be leading to an interruption of service for the users connected to the respective EPON interface, depending on the system action on such an alert.

9.5.13 dot3ExtPkgOptIfTransmitEnable

Changing the dot3ExtPkgOptIfTransmitEnable state can lead to a halt in the optical transmission of the respective interface, leading to an interruption of service for the users connected to the respective EPON interface.

9.6 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL¹⁶:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C9mib.txt

¹⁶Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
1 IEEE8023-DOT3-EPON-MIB DEFINITIONS ::= BEGIN
2
3     IMPORTS
4         MODULE-IDENTITY, OBJECT-TYPE, Counter32,
5         Integer32, Unsigned32, Counter64, org
6         FROM SNMPv2-SMI
7         TruthValue, MacAddress
8         FROM SNMPv2-TC
9         ifIndex
10        FROM IF-MIB
11        MODULE-COMPLIANCE, OBJECT-GROUP
12        FROM SNMPv2-CONF
13
14    ;
15
16    ieee8023dot3EponMIB MODULE-IDENTITY
17        LAST-UPDATED "201304110000Z" -- April 11, 2013
18        ORGANIZATION
19            "IEEE 802.3 working group"
20        CONTACT-INFO
21            "WG-URL: http://www.ieee802.org/3/index.html
22             WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
23
24             Contact: Howard Frazier
25             Postal: 3151 Zanker Road
26                   San Jose, CA 95134
27                   USA
28             Tel:    +1.408.922.8164
29             E-mail: hfrazier@broadcom.com"
30
31    DESCRIPTION
32        "The objects in this MIB module are used to manage the
33         Ethernet in the First Mile (EFM) Ethernet Passive Optical
34         Network (EPON) Interfaces as defined in IEEE Std 802.3
35         Clauses 60, 64, and 65.
36
37         Of particular interest are Clause 64 (MultiPoint Control
38         Protocol - MPCP), Clause 65 (Point-to-Multipoint
39         Reconciliation Sublayer - P2MP RS), Clause 60 (Ethernet
40         Passive Optical Network Physical Medium Dependent - EPON
41         PMDs), Clause 30, 'Management', and Clause 45, 'Management
42         Data Input/Output (MDIO) Interface'."
43
44    REVISION      "201304110000Z" -- April 11, 2013
45    DESCRIPTION
46        "Revision, based on an earlier version in IEEE Std 802.3.1-2011."
47
48    REVISION      "201102020000Z" -- February 2, 2011
49    DESCRIPTION
50        "Initial version, based on an earlier version published
51         as RFC 4837."
52
53    ::= { org ieee(111) standards-association-numbers-series-standards(2)
54          lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 9 }
55
56    dot3EponObjects OBJECT IDENTIFIER ::= { ieee8023dot3EponMIB 1}
57
58    dot3EponConformance OBJECT IDENTIFIER ::= { ieee8023dot3EponMIB 2}
59
60    -- MPCP MIB modules definitions (IEEE Std 802.3, Clause 30.3.5)
61
62
63
64
65
```

```

1  dot3EponMpcpObjects
2      OBJECT IDENTIFIER ::= { dot3EponObjects 1 }
3
4  dot3MpcpControlTable OBJECT-TYPE
5      SYNTAX  SEQUENCE OF Dot3MpcpControlEntry
6      MAX-ACCESS  not-accessible
7      STATUS  current
8      DESCRIPTION
9
10         "A Table of dot3 MultiPoint Control Protocol (MPCP)
11         MIB objects. The entries in the table are control and
12         status objects of the MPCP.
13         Each object has a row for every virtual link denoted by
14         the corresponding ifIndex.
15         The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
16         register (15-bit field and a broadcast bit) limiting the
17         number of virtual links to 32768. Typically the number
18         of expected virtual links in a PON is like the number of
19         ONUs, which is 32-64, plus an additional entry for
20         broadcast LLID."
21
22     ::= { dot3EponMpcpObjects 1 }
23
24  dot3MpcpControlEntry OBJECT-TYPE
25      SYNTAX  Dot3MpcpControlEntry
26      MAX-ACCESS  not-accessible
27      STATUS  current
28      DESCRIPTION
29
30         "An entry in the dot3 MPCP Control table.
31         Rows exist for an OLT interface and an ONU interface.
32         A row in the table is denoted by the ifIndex of the link
33         and it is created when the ifIndex is created.
34         The rows in the table for an ONU interface are created
35         at system initialization.
36         The row in the table corresponding to the OLT ifIndex
37         and the row corresponding to the broadcast virtual link
38         are created at system initialization.
39         A row in the table corresponding to the ifIndex of a
40         virtual links is created when a virtual link is
41         established (ONU registers) and deleted when the virtual
42         link is deleted (ONU deregisters)."
```

INDEX { ifIndex }

```

44     ::= { dot3MpcpControlTable 1}
45
46
47  Dot3MpcpControlEntry ::=
48      SEQUENCE {
49          dot3MpcpOperStatus          TruthValue,
50          dot3MpcpAdminState          TruthValue,
51          dot3MpcpMode                INTEGER,
52          dot3MpcpSyncTime             Unsigned32,
53          dot3MpcpLinkID              Unsigned32,
54          dot3MpcpRemoteMACAddress     MacAddress,
55          dot3MpcpRegistrationState    INTEGER,
56          dot3MpcpTransmitElapsed      Unsigned32,
57          dot3MpcpReceiveElapsed       Unsigned32,
58          dot3MpcpRoundTripTime        Unsigned32,
59          dot3MpcpMaximumPendingGrants Unsigned32
60      }
61
62
63  dot3MpcpOperStatus OBJECT-TYPE
64      SYNTAX  TruthValue
65
```

```
1      MAX-ACCESS read-only
2      STATUS current
3      DESCRIPTION
4          "This object reflects the operational state of the
5           MultiPoint MAC Control sublayer as defined in
6           IEEE Std 802.3, Clause 64 and Clause 77. When the value is
7           true(1), the interface will act as if the MultiPoint Control
8           Protocol is enabled. When the value is false(2), the interface
9           will act as if the MultiPoint Control Protocol is
10          disabled. The operational state can be changed using the
11          dot3MpcpAdminState object.
12          This object is applicable for an OLT, with the same
13          value for all virtual interfaces, and for an ONU."
14      REFERENCE "IEEE Std 802.3, 30.3.5.1.2."
15      ::= { dot3MpcpControlEntry 1 }
16
17  dot3MpcpAdminState OBJECT-TYPE
18      SYNTAX TruthValue
19      MAX-ACCESS read-write
20      STATUS current
21      DESCRIPTION
22          "This object is used to define the admin state of the
23           MultiPoint MAC Control sublayer, as defined in
24           IEEE Std 802.3, Clause 64, and to reflect its state.
25           When selecting the value as true(1), the MultiPoint
26           Control Protocol of the interface is enabled.
27           When selecting the value as false(2), the MultiPoint
28           Control Protocol of the interface is disabled.
29           This object reflects the administrative state of the
30           MultiPoint Control Protocol of the interface.
31           The write operation is not restricted in this document
32           and can be done at any time. Changing
33           dot3MpcpAdminState state can lead to disabling the
34           MultiPoint Control Protocol on the respective interface,
35           leading to the interruption of service for the users
36           connected to the respective EPON interface.
37           This object is applicable for an OLT, with the same
38           value for all virtual interfaces, and for an ONU."
39      REFERENCE "IEEE Std 802.3, 30.3.5.2.1."
40      DEFVAL { false }
41      ::= { dot3MpcpControlEntry 2 }
42
43  dot3MpcpMode OBJECT-TYPE
44      SYNTAX INTEGER {
45          olt(1),
46          onu(2)
47      }
48      MAX-ACCESS read-only
49      STATUS current
50      DESCRIPTION
51          "This object is used to identify the operational
52           state of the MultiPoint MAC Control sublayer as
53           defined in IEEE Std 802.3, Clause 64 and Clause 77. Reading
54           olt(1) for an OLT (server) mode and onu(2) for an ONU (client)
55           mode. This object is used to identify the operational mode
56           for the MPCP tables.
57           This object is applicable for an OLT, with the same
58           value for all virtual interfaces, and for an ONU."
59      REFERENCE "IEEE Std 802.3, 30.3.5.1.3."
```

```
1      DEFVAL { olt }
2      ::= { dot3MpcpControlEntry 3 }
3
4  dot3MpcpSyncTime OBJECT-TYPE
5      SYNTAX  Unsigned32
6      UNITS   "TQ (16 ns)"
7      MAX-ACCESS  read-only
8      STATUS   current
9      DESCRIPTION
10
11         "An object that reports the 'sync lock time' of the
12         OLT receiver in increments of Time Quanta (TQ)-16ns
13         as defined in IEEE Std 802.3, Clauses 60, 64, and 65. The
14         value returned shall be (sync lock time ns)/16, rounded up
15         to the nearest TQ. If this value exceeds (2^32-1), the
16         value (2^32-1) shall be returned. This object is applicable
17         for an OLT, with distinct values for all virtual interfaces,
18         and for an ONU."
19
20     REFERENCE  "IEEE Std 802.3, 64.3.3.2."
21     ::= { dot3MpcpControlEntry 4 }
22
23  dot3MpcpLinkID OBJECT-TYPE
24      SYNTAX  Unsigned32
25      MAX-ACCESS  read-only
26      STATUS   current
27      DESCRIPTION
28
29         "An object that identifies the Logical Link
30         Identifier (LLID) associated with the MAC of the virtual
31         link as specified in IEEE Std 802.3, 65.1.3.2.2 or 76.2.6.1.3.2,
32         as appropriate.
33         This object is applicable for an OLT and an ONU. At the
34         OLT, it has a distinct value for each virtual interface.
35         The ONU and the corresponding virtual MAC of the OLT,
36         for the same virtual link, have the same value.
37         Value is assigned when the ONU registers.
38         Value is freed when the ONU deregisters."
39
40     REFERENCE  "IEEE Std 802.3, 30.3.5.1.4."
41     ::= { dot3MpcpControlEntry 5 }
42
43  dot3MpcpRemoteMACAddress OBJECT-TYPE
44      SYNTAX  MacAddress
45      MAX-ACCESS  read-only
46      STATUS   current
47      DESCRIPTION
48
49         "An object that identifies the source_address
50         parameter of the last MPCPDUs passed to the MAC Control.
51         This value is updated on reception of a valid frame with
52         1) a destination Field equal to the reserved multicast
53         address for MAC Control as specified in IEEE Std 802.3, Annex
54         31A; 2) the lengthOrType field value equal to the reserved
55         Type for MAC Control as specified in IEEE Std 802.3, Annex
56         31A; 3) an MPCP subtype value equal to the subtype
57         reserved for MPCP as specified in IEEE Std 802.3, Annex 31A.
58         This object is applicable for an OLT and an ONU. At the
59         OLT, it has a distinct value for each virtual interface.
60         The value reflects the MAC address of the remote entity
61         and therefore the OLT holds a value for each LLID, which
62         is the MAC address of the ONU; the ONU has a single
63         value that is the OLT MAC address."
64
65     REFERENCE  "IEEE Std 802.3, 30.3.5.1.5."
```

```
1      ::= { dot3MpcpControlEntry 6 }
2
3  dot3MpcpRegistrationState OBJECT-TYPE
4      SYNTAX  INTEGER {
5          unregistered(1),
6          registering(2),
7          registered(3)
8      }
9
10     MAX-ACCESS  read-only
11     STATUS  current
12     DESCRIPTION
13         "An object that identifies the registration state
14         of the MultiPoint MAC Control sublayer as defined in
15         IEEE Std 802.3, Clause 64. When this object has the
16         enumeration unregistered(1), the interface is
17         unregistered and may be used for registering a link
18         partner. When this object has the enumeration
19         registering(2), the interface is in the process of
20         registering a link-partner. When this object has the
21         enumeration registered(3), the interface has an
22         established link-partner.
23         This object is applicable for an OLT and an ONU. At the
24         OLT, it has a distinct value for each virtual interface."
25     REFERENCE  "IEEE Std 802.3, 30.3.5.1.6."
26     ::= { dot3MpcpControlEntry 7 }
27
28
29  dot3MpcpTransmitElapsed OBJECT-TYPE
30      SYNTAX  Unsigned32
31      UNITS   "TQ (16 ns)"
32      MAX-ACCESS  read-only
33      STATUS  current
34      DESCRIPTION
35          "An object that reports the interval from the last
36          MPCP frame transmission in increments of Time Quanta
37          (TQ)-16ns. The value returned shall be (interval from
38          last MPCP frame transmission in ns)/16. If this value
39          exceeds (2^32-1), the value (2^32-1) shall be returned.
40          This object is applicable for an OLT and an ONU. At the
41          OLT, it has a distinct value for each virtual interface."
42      REFERENCE  "IEEE Std 802.3, 30.3.5.1.19."
43      ::= { dot3MpcpControlEntry 8 }
44
45
46
47  dot3MpcpReceiveElapsed OBJECT-TYPE
48      SYNTAX  Unsigned32
49      UNITS   "TQ (16 ns)"
50      MAX-ACCESS  read-only
51      STATUS  current
52      DESCRIPTION
53          "An object that reports the interval from last MPCP frame
54          reception in increments of Time Quanta (TQ)-16ns. The
55          value returned shall be (interval from last MPCP frame
56          reception in ns)/16. If this value exceeds (2^32-1), the
57          value (2^32-1) shall be returned.
58          This object is applicable for an OLT and an ONU. At the
59          OLT, it has a distinct value for each virtual interface."
60      REFERENCE  "IEEE Std 802.3, 30.3.5.1.20."
61      ::= { dot3MpcpControlEntry 9 }
62
63
64
65  dot3MpcpRoundTripTime OBJECT-TYPE
```



```
1      SYNTAX  Unsigned32 (0..'ffff'h)
2      UNITS      "TQ (16 ns)"
3      MAX-ACCESS  read-only
4      STATUS  current
5      DESCRIPTION
6          "An object that reports the MPCP round trip time in
7           increments of Time Quanta (TQ)-16ns. The value returned
8           shall be (round trip time in ns)/16. If this value
9           exceeds (2^16-1), the value (2^16-1) shall be returned.
10          This object is applicable for an OLT. At the
11          OLT, it has a distinct value for each virtual interface."
12      REFERENCE  "IEEE Std 802.3, 30.3.5.1.21."
13      ::= { dot3MpcpControlEntry 10 }
14
15
16  dot3MpcpMaximumPendingGrants OBJECT-TYPE
17      SYNTAX  Unsigned32 (0..255)
18      MAX-ACCESS  read-only
19      STATUS  current
20      DESCRIPTION
21          "An object that reports the maximum number of grants
22           that an ONU can store for handling. The maximum number
23           of grants that an ONU can store for handling has a
24           range of 0 to 255.
25           This object is applicable for an OLT and an ONU. At the
26           OLT, it has a distinct value for each virtual interface.
27           At the OLT, the value should be zero."
28      REFERENCE  "IEEE Std 802.3, 30.3.5.1.24."
29      ::= { dot3MpcpControlEntry 11 }
30
31
32
33
34
35  dot3MpcpStatTable OBJECT-TYPE
36      SYNTAX      SEQUENCE OF Dot3MpcpStatEntry
37      MAX-ACCESS  not-accessible
38      STATUS      current
39      DESCRIPTION
40          "This table defines the list of statistics counters of
41           an interface implementing the IEEE Std 802.3, Clause 64 MPCP.
42           Each object has a row for every virtual link denoted by
43           the corresponding ifIndex.
44           The LLID field, as defined in IEEE Std 802.3, is a 2-byte
45           register (15-bit field and a broadcast bit) limiting the
46           number of virtual links to 32768. Typically the number of
47           expected virtual links in a PON is like the number of
48           ONUs, which is 32-64, plus an additional entry for
49           broadcast LLID."
50      ::= { dot3EponMpcpObjects 2 }
51
52
53  dot3MpcpStatEntry OBJECT-TYPE
54      SYNTAX      Dot3MpcpStatEntry
55      MAX-ACCESS  not-accessible
56      STATUS      current
57      DESCRIPTION
58          "An entry in the table of statistics counters of the
59           IEEE Std 802.3, Clause 64, MPCP interface.
60           Rows exist for an OLT interface and an ONU interface.
61           A row in the table is denoted by the ifIndex of the link
62           and it is created when the ifIndex is created.
63           The rows in the table for an ONU interface are created
64           The rows in the table for an ONU interface are created
65           The rows in the table for an ONU interface are created
```

```

1         at system initialization.
2         The row in the table corresponding to the OLT ifIndex
3         and the row corresponding to the broadcast virtual link
4         are created at system initialization.
5         A row in the table corresponding to the ifIndex of a
6         virtual link is created when a virtual link is
7         established (ONU registers) and deleted when the virtual
8         link is deleted (ONU deregisters)."
```

INDEX { ifIndex }

::= { dot3MpcpStatTable 1 }

Dot3MpcpStatEntry ::=

SEQUENCE {

dot3MpcpMACCtrlFramesTransmitted	Counter64,
dot3MpcpMACCtrlFramesReceived	Counter64,
dot3MpcpDiscoveryWindowsSent	Counter32,
dot3MpcpDiscoveryTimeout	Counter32,
dot3MpcpTxRegRequest	Counter64,
dot3MpcpRxRegRequest	Counter64,
dot3MpcpTxRegAck	Counter64,
dot3MpcpRxRegAck	Counter64,
dot3MpcpTxReport	Counter64,
dot3MpcpRxReport	Counter64,
dot3MpcpTxGate	Counter64,
dot3MpcpRxGate	Counter64,
dot3MpcpTxRegister	Counter64,
dot3MpcpRxRegister	Counter64

}

dot3MpcpMACCtrlFramesTransmitted OBJECT-TYPE

SYNTAX Counter64

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of MPCP frames passed to the MAC sublayer for transmission. This counter is incremented when a MA_CONTROL.request service primitive is generated within the MAC control sublayer with an opcode indicating an MPCP frame. This object is applicable for an OLT and an ONU. At the OLT it has a distinct value for each virtual interface. Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime object of the Interfaces Group MIB module."

REFERENCE "IEEE Std 802.3, 30.3.5.1.7."

::= { dot3MpcpStatEntry 1 }

dot3MpcpMACCtrlFramesReceived OBJECT-TYPE

SYNTAX Counter64

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of MPCP frames passed by the MAC sublayer to the MAC Control sublayer. This counter is incremented when a ReceiveFrame function call returns a valid frame with

```
1          1) a lengthOrType field value equal to the reserved
2          Type for 802.3_MAC_Control as specified in IEEE Std 802.3
3          31.4.1.3, and
4          2) an opcode indicating an MPCP frame.
5          This object is applicable for an OLT and an ONU. At the
6          OLT, it has a distinct value for each virtual interface.
7          Discontinuities of this counter can occur at
8          re-initialization of the management system and at other
9          times, as indicated by the value of the
10         ifCounterDiscontinuityTime object of the Interfaces Group MIB
11         module."
12
13     REFERENCE    "IEEE Std 802.3, 30.3.5.1.8."
14     ::= { dot3MpcpStatEntry 2}
15
16 dot3MpcpDiscoveryWindowsSent OBJECT-TYPE
17     SYNTAX Counter32
18     MAX-ACCESS read-only
19     STATUS current
20     DESCRIPTION
21         "A count of discovery windows generated. The counter is
22         incremented by one for each generated discovery window.
23         This object is applicable for an OLT and an ONU. At the
24         OLT, it has a distinct value for each virtual interface.
25         At the ONU, the value should be zero.
26         Discontinuities of this counter can occur at
27         re-initialization of the management system and at other
28         times, as indicated by the value of the
29         ifCounterDiscontinuityTime object of the Interfaces Group MIB
30         module."
31
32     REFERENCE    "IEEE Std 802.3, 30.3.5.1.22."
33     ::= { dot3MpcpStatEntry 3}
34
35
36 dot3MpcpDiscoveryTimeout OBJECT-TYPE
37     SYNTAX Counter32
38     MAX-ACCESS read-only
39     STATUS current
40     DESCRIPTION
41         "A count of the number of times a discovery timeout
42         occurs. Increment the counter by one for each discovery
43         processing state-machine reset resulting from timeout
44         waiting for message arrival.
45         This object is applicable for an OLT and an ONU. At the
46         OLT, it has a distinct value for each virtual interface.
47         Discontinuities of this counter can occur at
48         re-initialization of the management system and at other
49         times, as indicated by the value of the
50         ifCounterDiscontinuityTime object of the Interfaces Group MIB
51         module."
52
53     REFERENCE    "IEEE Std 802.3, 30.3.5.1.23."
54     ::= { dot3MpcpStatEntry 4}
55
56
57 dot3MpcpTxRegRequest OBJECT-TYPE
58     SYNTAX Counter64
59     UNITS "frames"
60     MAX-ACCESS read-only
61     STATUS current
62     DESCRIPTION
63         "A count of the number of times a REGISTER_REQ MPCP
64         frame transmission occurs. Increment the counter by one
65
```

```
1         for each REGISTER_REQ MPCP frame transmitted as defined
2         in IEEE Std 802.3, Clause 64.
3         This object is applicable for an OLT and an ONU. At the
4         OLT, it has a distinct value for each virtual interface.
5         At the OLT, the value should be zero.
6         Discontinuities of this counter can occur at
7         re-initialization of the management system and at other
8         times, as indicated by the value of the
9         ifCounterDiscontinuityTime object of the Interfaces Group MIB
10        module."
11    REFERENCE    "IEEE Std 802.3, 30.3.5.1.12."
12    ::= { dot3MpcpStatEntry 5}
13
14    dot3MpcpRxRegRequest OBJECT-TYPE
15        SYNTAX Counter64
16        UNITS      "frames"
17        MAX-ACCESS read-only
18        STATUS current
19        DESCRIPTION
20            "A count of the number of times a REGISTER_REQ MPCP
21            frame reception occurs.
22            Increment the counter by one for each REGISTER_REQ MPCP
23            frame received as defined in IEEE Std 802.3, Clause 64.
24            This object is applicable for an OLT and an ONU. At the
25            OLT, it has a distinct value for each virtual interface.
26            At the ONU, the value should be zero.
27            Discontinuities of this counter can occur at
28            re-initialization of the management system and at other
29            times, as indicated by the value of the
30            ifCounterDiscontinuityTime object of the Interfaces Group MIB
31            module."
32    REFERENCE    "IEEE Std 802.3, 30.3.5.1.17."
33    ::= { dot3MpcpStatEntry 6}
34
35    dot3MpcpTxRegAck OBJECT-TYPE
36        SYNTAX Counter64
37        UNITS      "frames"
38        MAX-ACCESS read-only
39        STATUS current
40        DESCRIPTION
41            "A count of the number of times a REGISTER_ACK MPCP
42            frame transmission occurs. Increment the counter by one
43            for each REGISTER_ACK MPCP frame transmitted as defined
44            in IEEE Std 802.3, Clause 64.
45            This object is applicable for an OLT and an ONU. At the
46            OLT, it has a distinct value for each virtual interface.
47            At the OLT, the value should be zero.
48            Discontinuities of this counter can occur at
49            re-initialization of the management system and at other
50            times, as indicated by the value of the
51            ifCounterDiscontinuityTime object of the Interfaces Group MIB
52            module."
53    REFERENCE    "IEEE Std 802.3, 30.3.5.1.10."
54    ::= { dot3MpcpStatEntry 7}
55
56    dot3MpcpRxRegAck OBJECT-TYPE
57        SYNTAX Counter64
58        UNITS      "frames"
59        MAX-ACCESS read-only
```

```
1      STATUS current
2      DESCRIPTION
3          "A count of the number of times a REGISTER_ACK MPCP
4          frame reception occurs.
5          Increment the counter by one for each REGISTER_ACK MPCP
6          frame received as defined in IEEE Std 802.3, Clause 64.
7          This object is applicable for an OLT and an ONU. At the
8          OLT, it has a distinct value for each virtual interface.
9          At the ONU, the value should be zero.
10         Discontinuities of this counter can occur at
11         re-initialization of the management system and at other
12         times, as indicated by the value of the
13         ifCounterDiscontinuityTime object of the Interfaces Group MIB
14         module."
15
16     REFERENCE "IEEE Std 802.3, 30.3.5.1.15."
17     ::= { dot3MpcpStatEntry 8}
18
19 dot3MpcpTxReport OBJECT-TYPE
20     SYNTAX Counter64
21     UNITS "frames"
22     MAX-ACCESS read-only
23     STATUS current
24     DESCRIPTION
25         "A count of the number of times a REPORT MPCP frame
26         transmission occurs. Increment the counter by one for
27         each REPORT MPCP frame transmitted as defined in
28         IEEE Std 802.3, Clause 64.
29         This object is applicable for an OLT and an ONU. At the
30         OLT, it has a distinct value for each virtual interface.
31         At the ONU, the value should be zero.
32         Discontinuities of this counter can occur at
33         re-initialization of the management system and at other
34         times, as indicated by the value of the
35         ifCounterDiscontinuityTime object of the Interfaces Group MIB
36         module."
37     REFERENCE "IEEE Std 802.3, 30.3.5.1.13."
38     ::= { dot3MpcpStatEntry 9}
39
40 dot3MpcpRxReport OBJECT-TYPE
41     SYNTAX Counter64
42     UNITS "frames"
43     MAX-ACCESS read-only
44     STATUS current
45     DESCRIPTION
46         "A count of the number of times a REPORT MPCP frame
47         reception occurs.
48         Increment the counter by one for each REPORT MPCP frame
49         received as defined in IEEE Std 802.3, Clause 64.
50         This object is applicable for an OLT and an ONU. At the
51         OLT, it has a distinct value for each virtual interface.
52         At the ONU, the value should be zero.
53         Discontinuities of this counter can occur at
54         re-initialization of the management system and at other
55         times, as indicated by the value of the
56         ifCounterDiscontinuityTime object of the Interfaces Group MIB
57         module."
58     REFERENCE "IEEE Std 802.3, 30.3.5.1.18."
59     ::= { dot3MpcpStatEntry 10}
60
61
62
63
64
65
```

```
1  dot3MpcpTxGate OBJECT-TYPE
2      SYNTAX Counter64
3      UNITS      "frames"
4      MAX-ACCESS read-only
5      STATUS current
6      DESCRIPTION
7          "A count of the number of times a GATE MPCP frame
8              transmission occurs.
9              Increment the counter by one for each GATE MPCP frame
10             transmitted as defined in IEEE Std 802.3, Clause 64.
11             This object is applicable for an OLT and an ONU. At the
12             OLT, it has a distinct value for each virtual interface.
13             At the ONU, the value should be zero.
14             Discontinuities of this counter can occur at
15             re-initialization of the management system and at other
16             times, as indicated by the value of the
17             ifCounterDiscontinuityTime object of the Interfaces Group MIB
18             module."
19
20      REFERENCE "IEEE Std 802.3, 30.3.5.1.9."
21      ::= { dot3MpcpStatEntry 11}
22
23
24  dot3MpcpRxGate OBJECT-TYPE
25      SYNTAX Counter64
26      UNITS      "frames"
27      MAX-ACCESS read-only
28      STATUS current
29      DESCRIPTION
30          "A count of the number of times a GATE MPCP frame
31              reception occurs.
32              Increment the counter by one for each GATE MPCP frame
33              received as defined in IEEE Std 802.3, Clause 64.
34              This object is applicable for an OLT and an ONU. At the
35              OLT, it has a distinct value for each virtual interface.
36              At the OLT, the value should be zero.
37              Discontinuities of this counter can occur at
38              re-initialization of the management system and at other
39              times, as indicated by the value of the
40              ifCounterDiscontinuityTime object of the Interfaces Group MIB
41              module."
42
43      REFERENCE "IEEE Std 802.3, 30.3.5.1.14."
44      ::= { dot3MpcpStatEntry 12}
45
46
47  dot3MpcpTxRegister OBJECT-TYPE
48      SYNTAX Counter64
49      UNITS      "frames"
50      MAX-ACCESS read-only
51      STATUS current
52      DESCRIPTION
53          "A count of the number of times a REGISTER MPCP frame
54              transmission occurs.
55              Increment the counter by one for each REGISTER MPCP
56              frame transmitted as defined in IEEE Std 802.3, Clause 64.
57              This object is applicable for an OLT and an ONU. At the
58              OLT, it has a distinct value for each virtual interface.
59              At the ONU, the value should be zero.
60              Discontinuities of this counter can occur at
61              re-initialization of the management system and at other
62              times, as indicated by the value of the
63              ifCounterDiscontinuityTime object of the Interfaces Group MIB
64              module."
65
```

```
1         module."
2     REFERENCE    "IEEE Std 802.3, 30.3.5.1.11."
3     ::= { dot3MpcpStatEntry 13}
4
5     dot3MpcpRxRegister OBJECT-TYPE
6         SYNTAX    Counter64
7         UNITS      "frames"
8         MAX-ACCESS read-only
9         STATUS    current
10        DESCRIPTION
11            "A count of the number of times a REGISTER MPCP frame
12             reception occurs.
13             Increment the counter by one for each REGISTER MPCP
14             frame received as defined in IEEE Std 802.3, Clause 64.
15             This object is applicable for an OLT and an ONU. At the
16             OLT, it has a distinct value for each virtual interface.
17             At the OLT, the value should be zero.
18             Discontinuities of this counter can occur at
19             re-initialization of the management system and at other
20             times, as indicated by the value of the
21             ifCounterDiscontinuityTime object of the Interfaces Group MIB
22             module."
23        REFERENCE    "IEEE Std 802.3, 30.3.5.1.16."
24        ::= { dot3MpcpStatEntry 14}
25
26    -- Optical Multi Point Emulation (OMPEmulation)
27    -- managed object definitions
28
29    dot3OmpEmulationObjects OBJECT IDENTIFIER ::= {dot3EponObjects 2}
30
31    dot3OmpEmulationTable OBJECT-TYPE
32        SYNTAX    SEQUENCE OF Dot3OmpEmulationEntry
33        MAX-ACCESS not-accessible
34        STATUS    current
35        DESCRIPTION
36            "A table of dot3 OmpEmulation MIB objects. The table
37             contain objects for the management of the OMPEmulation
38             sublayer.
39             Each object has a row for every virtual link denoted by
40             the corresponding ifIndex.
41             The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
42             register (15-bit field and a broadcast bit) limiting the
43             number of virtual links to 32768. Typically the number
44             of expected virtual links in a PON is like the number of
45             ONUs, which is 32-64, plus an additional entry for
46             broadcast LLID."
47        ::= { dot3OmpEmulationObjects 1 }
48
49    dot3OmpEmulationEntry OBJECT-TYPE
50        SYNTAX    Dot3OmpEmulationEntry
51        MAX-ACCESS not-accessible
52        STATUS    current
53        DESCRIPTION
54            "An entry in the dot3 OmpEmulation table.
55             Rows exist for an OLT interface and an ONU interface.
56             A row in the table is denoted by the ifIndex of the link
57             and it is created when the ifIndex is created.
58             The rows in the table for an ONU interface are created
59             at system initialization.
```

```
1           The row in the table corresponding to the OLT ifIndex
2           and the row corresponding to the broadcast virtual link
3           are created at system initialization.
4           A row in the table corresponding to the ifIndex of a
5           virtual links is created when a virtual link is
6           established (ONU registers) and deleted when the virtual
7           link is deleted (ONU deregisters)."
```

INDEX { ifIndex }

::= { dot3OmpEmulationTable 1 }

Dot3OmpEmulationEntry ::=

SEQUENCE {

dot3OmpEmulationType INTEGER

}

dot3OmpEmulationType OBJECT-TYPE

SYNTAX INTEGER {

unknown(1),

olt(2),

onu(3)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An object that indicates the mode of operation of the Reconciliation Sublayer for Point-to-Point Emulation (see IEEE Std 802.3, 65.1 or 76.2 as appropriate). unknown(1) value is assigned in initialization; true state or type is not yet known. olt(2) value is assigned when the sublayer is operating in OLT mode. onu(3) value is assigned when the sublayer is operating in ONU mode. This object is applicable for an OLT, with the same value for all virtual interfaces, and for an ONU."

REFERENCE "IEEE Std 802.3, 30.3.7.1.2."

::= { dot3OmpEmulationEntry 1 }

dot3OmpEmulationStatTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot3OmpEmulationStatEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table defines the list of statistics counters of IEEE Std 802.3, Clause 65, OMPEmulation sublayer. Each object has a row for every virtual link denoted by the corresponding ifIndex. The LLID field, as defined in the IEEE Std 802.3, is a 2-byte register (15-bit field and a broadcast bit) limiting the number of virtual links to 32768. Typically the number of expected virtual links in a PON is like the number of ONUs, which is 32-64, plus an additional entry for broadcast LLID."

::= { dot3OmpEmulationObjects 2 }

dot3OmpEmulationStatEntry OBJECT-TYPE

SYNTAX Dot3OmpEmulationStatEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table of statistics counters of


```
1         IEEE Std 802.3, Clause 65, OMPEmulation sublayer.
2         Rows exist for an OLT interface and an ONU interface.
3         A row in the table is denoted by the ifIndex of the link
4         and it is created when the ifIndex is created.
5         The rows in the table for an ONU interface are created
6         at system initialization.
7         The row in the table corresponding to the OLT ifIndex
8         and the row corresponding to the broadcast virtual link
9         are created at system initialization.
10        A row in the table corresponding to the ifIndex of a
11        virtual links is created when a virtual link is
12        established (ONU registers) and deleted when the virtual
13        link is deleted (ONU deregisters)."
```

INDEX { ifIndex }

::= { dot3OmpEmulationStatTable 1 }

Dot3OmpEmulationStatEntry::=

SEQUENCE {

dot3OmpEmulationSLDErrors	Counter64,
dot3OmpEmulationCRC8Errors	Counter64,
dot3OmpEmulationBadLLID	Counter64,
dot3OmpEmulationGoodLLID	Counter64,
dot3OmpEmulationOnuPonCastLLID	Counter64,
dot3OmpEmulationOltPonCastLLID	Counter64,
dot3OmpEmulationBroadcastBitNotOnuLlid	Counter64,
dot3OmpEmulationOnuLLIDNotBroadcast	Counter64,
dot3OmpEmulationBroadcastBitPlusOnuLlid	Counter64,
dot3OmpEmulationNotBroadcastBitNotOnuLlid	Counter64

}

dot3OmpEmulationSLDErrors OBJECT-TYPE

SYNTAX Counter64

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of frames received that do not contain a valid SLD field as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as appropriate. This object is applicable for an OLT and an ONU. At the OLT, it has a distinct value for each virtual interface. Discontinuities of this counter can occur at re-initialization of the management system and at other times, as indicated by the value of the ifCounterDiscontinuityTime object of the Interfaces Group MIB module."

REFERENCE "IEEE Std 802.3, 30.3.7.1.3."

::= { dot3OmpEmulationStatEntry 1 }

dot3OmpEmulationCRC8Errors OBJECT-TYPE

SYNTAX Counter64

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of frames received that contain a valid SLD field, as defined in IEEE Std 802.3, 65.1.3.3.1 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."

```
1         This object is applicable for an OLT and an ONU. At the
2         OLT, it has a distinct value for each virtual interface.
3         Discontinuities of this counter can occur at
4         re-initialization of the management system and at other
5         times, as indicated by the value of the
6         ifCounterDiscontinuityTime object of the Interfaces Group MIB
7         module."
8
9     REFERENCE    "IEEE Std 802.3, 30.3.7.1.4."
10    ::= { dot3OmpEmulationStatEntry 2}
11
12 dot3OmpEmulationBadLLID OBJECT-TYPE
13     SYNTAX Counter64
14     UNITS      "frames"
15     MAX-ACCESS read-only
16     STATUS current
17     DESCRIPTION
18         "A count of frames received that contain a valid SLD field in an
19         OLT, and pass the CRC-8 check, but are discarded due to the
20         LLID check. The SLD is defined in IEEE Std 802.3, 65.1.3.3.1
21         or 76.2.6.1.3.1, as appropriate. The CRC-8 check is defined in
22         IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate. The
23         LLID check is defined in IEEE Std 802.3, 65.1.3.3.2 or
24         76.2.6.1.3.2, as appropriate.
25         This object is applicable for an OLT and an ONU. At the
26         OLT, it has a distinct value for each virtual interface.
27         Discontinuities of this counter can occur at
28         re-initialization of the management system and at other
29         times, as indicated by the value of the
30         ifCounterDiscontinuityTime object of the Interfaces Group MIB
31         module."
32     REFERENCE    "IEEE Std 802.3, 30.3.7.1.8."
33    ::= { dot3OmpEmulationStatEntry 3}
34
35 dot3OmpEmulationGoodLLID OBJECT-TYPE
36     SYNTAX Counter64
37     UNITS      "frames"
38     MAX-ACCESS read-only
39     STATUS current
40     DESCRIPTION
41         "A count of frames received that contain a valid SLD
42         field, as defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1,
43         as appropriate, and pass the CRC-8 check as defined in
44         IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate.
45         This object is applicable for an OLT and an ONU. At the
46         OLT, it has a distinct value for each virtual interface.
47         Discontinuities of this counter can occur at
48         re-initialization of the management system and at other
49         times, as indicated by the value of the
50         ifCounterDiscontinuityTime object of the Interfaces Group MIB
51         module."
52     REFERENCE    "IEEE Std 802.3, 30.3.7.1.5."
53    ::= { dot3OmpEmulationStatEntry 4}
54
55 dot3OmpEmulationOnuPonCastLLID OBJECT-TYPE
56     SYNTAX Counter64
57     UNITS      "frames"
58     MAX-ACCESS read-only
59     STATUS current
60     DESCRIPTION
```

```
1      "A count of frames received that: 1) contain a valid SLD field
2      in an ONU, 2) meet the rules for frame acceptance, and
3      3) pass the CRC-8 check. The SLD is defined in
4      IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as appropriate. The
5      rules for LLID acceptance are defined in IEEE Std 802.3, 65.1.3.3.2
6      or 76.2.6.1.3.2, as appropriate. The CRC-8 check is defined
7      in IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate.
8      This object is applicable for an OLT and an ONU. At the
9      OLT, it has a distinct value for each virtual interface.
10     At the OLT, the value should be zero.
11     Discontinuities of this counter can occur at
12     re-initialization of the management system and at other
13     times, as indicated by the value of the
14     ifCounterDiscontinuityTime object of the Interfaces Group MIB
15     module."
16
17     REFERENCE      "IEEE Std 802.3, 30.3.7.1.6."
18     ::= { dot3OmpEmulationStatEntry 5}
19
20
21 dot3OmpEmulationOltPonCastLLID OBJECT-TYPE
22     SYNTAX Counter64
23     UNITS      "frames"
24     MAX-ACCESS read-only
25     STATUS current
26     DESCRIPTION
27         "A count of frames received that contain a valid SLD field, as
28         defined in IEEE Std 802.3, 65.1.3.3.1 or 76.2.6.1.3.1, as
29         appropriate, pass the CRC-8 check, as defined in
30         IEEE Std 802.3, 65.1.3.3.3 or 76.2.6.1.3.3, as appropriate,
31         and meet the rules of acceptance for an OLT defined in
32         IEEE Std 802.3, 65.1.3.3.2 or 76.2.6.1.3.2, as appropriate.
33         This object is applicable for an OLT and an ONU. At the
34         OLT, it has a distinct value for each virtual interface.
35         At the ONU, the value should be zero.
36         Discontinuities of this counter can occur at
37         re-initialization of the management system and at other
38         times, as indicated by the value of the
39         ifCounterDiscontinuityTime object of the Interfaces Group MIB
40         module."
41
42     REFERENCE      "IEEE Std 802.3, 30.3.7.1.7."
43     ::= { dot3OmpEmulationStatEntry 6}
44
45
46 dot3OmpEmulationBroadcastBitNotOnuLlid OBJECT-TYPE
47     SYNTAX Counter64
48     UNITS      "frames"
49     MAX-ACCESS read-only
50     STATUS current
51     DESCRIPTION
52         "A count of frames received that contain a valid SLD
53         field, as defined in IEEE Std 802.3,
54         65.1.3.3.1, pass the CRC-8 check, as defined in
55         IEEE Std 802.3, 65.1.3.3.3, and contain the broadcast
56         bit in the LLID and not the ONU's LLID (frame accepted)
57         as defined in IEEE Std 802.3, Clause 65.
58         This object is applicable for an OLT and an ONU. At the
59         OLT, it has a distinct value for each virtual interface.
60         At the OLT, the value should be zero.
61         Discontinuities of this counter can occur at
62         re-initialization of the management system and at other
63         times, as indicated by the value of the
```

```
1             ifCounterDiscontinuityTime object of the Interfaces Group MIB
2             module."
3 ::= { dot3OmpEmulationStatEntry 7}
4
5 dot3OmpEmulationOnuLLIDNotBroadcast OBJECT-TYPE
6     SYNTAX Counter64
7     UNITS      "frames"
8     MAX-ACCESS read-only
9     STATUS current
10    DESCRIPTION
11        "A count of frames received that contain a valid SLD
12         field, as defined in IEEE Std 802.3,
13         65.1.3.3.1, pass the CRC-8 check, as defined in
14         IEEE Std 802.3, 65.1.3.3.3, and contain the ONU's LLID
15         as defined in IEEE Std 802.3, Clause 65.
16         This object is applicable for an OLT and an ONU. At the
17         OLT, it has a distinct value for each virtual interface.
18         At the OLT, the value should be zero.
19         Discontinuities of this counter can occur at
20         re-initialization of the management system and at other
21         times, as indicated by the value of the
22         ifCounterDiscontinuityTime object of the Interfaces Group MIB
23         module."
24 ::= { dot3OmpEmulationStatEntry 8}
25
26 dot3OmpEmulationBroadcastBitPlusOnuLlid OBJECT-TYPE
27     SYNTAX Counter64
28     UNITS      "frames"
29     MAX-ACCESS read-only
30     STATUS current
31     DESCRIPTION
32        "A count of frames received that contain a valid SLD
33         field, as defined in IEEE Std 802.3,
34         65.1.3.3.1, pass the CRC-8 check, as defined in
35         IEEE Std 802.3, 65.1.3.3.3, and contain the broadcast
36         bit in the LLID and match the ONU's LLID (frame
37         reflected) as defined in IEEE Std 802.3, Clause 65.
38         This object is applicable for an OLT and an ONU. At the
39         OLT, it has a distinct value for each virtual interface.
40         At the OLT, the value should be zero.
41         Discontinuities of this counter can occur at
42         re-initialization of the management system and at other
43         times, as indicated by the value of the
44         ifCounterDiscontinuityTime object of the Interfaces Group MIB
45         module."
46 ::= { dot3OmpEmulationStatEntry 9}
47
48 dot3OmpEmulationNotBroadcastBitNotOnuLlid OBJECT-TYPE
49     SYNTAX Counter64
50     UNITS      "frames"
51     MAX-ACCESS read-only
52     STATUS current
53     DESCRIPTION
54        "A count of frames received that contain a valid SLD
55         field, as defined in IEEE Std 802.3,
56         65.1.3.3.1, pass the CRC-8 check, as defined in
57         IEEE Std 802.3, 65.1.3.3.3, and do not contain
58         the ONU's LLID as defined in IEEE Std 802.3, Clause 65.
59         This object is applicable for an OLT and an ONU. At the
```

```

1         OLT, it has a distinct value for each virtual interface.
2         At the OLT, the value should be zero.
3         Discontinuities of this counter can occur at
4         re-initialization of the management system and at other
5         times, as indicated by the value of the
6         ifCounterDiscontinuityTime object of the Interfaces Group MIB
7         module."
8
9     ::= { dot3OmpEmulationStatEntry 10}
10
11 -- FEC managed object definitions (30.5.1)
12
13 dot3EponFecObjects OBJECT IDENTIFIER ::= {dot3EponObjects 3}
14
15 dot3EponFecTable OBJECT-TYPE
16     SYNTAX SEQUENCE OF Dot3EponFecEntry
17     MAX-ACCESS not-accessible
18     STATUS current
19     DESCRIPTION
20         "A table of dot3 EPON FEC management objects.
21         The entries in the table are control and status objects
22         and statistic counters for the FEC layer.
23         Each object has a row for every virtual link denoted by
24         the corresponding ifIndex.
25         The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
26         register (15-bit field and a broadcast bit) limiting the
27         number of virtual links to 32768. Typically the number
28         of expected virtual links in a PON is like the number of
29         ONUs, which is 32-64, plus an additional entry for
30         broadcast LLID."
31
32     ::= { dot3EponFecObjects 1 }
33
34
35 dot3EponFecEntry OBJECT-TYPE
36     SYNTAX Dot3EponFecEntry
37     MAX-ACCESS not-accessible
38     STATUS current
39     DESCRIPTION
40         "An entry in the dot3 EPON FEC table.
41         Rows exist for an OLT interface and an ONU interface.
42         A row in the table is denoted by the ifIndex of the link
43         and it is created when the ifIndex is created.
44         The rows in the table for an ONU interface are created
45         at system initialization.
46         The row in the table corresponding to the OLT ifIndex
47         and the row corresponding to the broadcast virtual link
48         are created at system initialization.
49         A row in the table corresponding to the ifIndex of a
50         virtual links is created when a virtual link is
51         established (ONU registers) and deleted when the virtual
52         link is deleted (ONU deregisters)."
```

55 INDEX { ifIndex}	
56 ::= { dot3EponFecTable 1 }	
57	
58 Dot3EponFecEntry ::=	
59 SEQUENCE {	
60 dot3EponFecPCSCodingViolation	Counter64,
61 dot3EponFecAbility	INTEGER,
62 dot3EponFecMode	INTEGER,
63 dot3EponFecCorrectedBlocks	Counter64,
64 dot3EponFecUncorrectableBlocks	Counter64,
65	

```
1          dot3EponFecBufferHeadCodingViolation      Counter64
2      }
3
4  dot3EponFecPCSCodingViolation OBJECT-TYPE
5      SYNTAX      Counter64
6      UNITS       "octets"
7      MAX-ACCESS  read-only
8      STATUS      current
9      DESCRIPTION
10         "For a 100 Mb/s operation, it is a count of the number of
11         times an invalid code-group is received, other than the
12         /H/ code-group. For a 1000 Mb/s operation, it is a count
13         of the number of times an invalid codegroup is received,
14         other than the /V/ code-group. /H/ denotes a special
15         4b5b codeword of the IEEE Std 802.3 Clause 24 100 Mb/s PCS layer,
16         and /V/ denotes a special 8b10b codeword of the IEEE Std 802.3
17         Clause 36 1000 Mb/s PCS layer.
18         This object is applicable for an OLT and an ONU. At the
19         OLT, it has a distinct value for each virtual interface.
20         Discontinuities of this counter can occur at
21         re-initialization of the management system and at other
22         times, as indicated by the value of the
23         ifCounterDiscontinuityTime object of the Interfaces Group MIB
24         module."
25     REFERENCE  "IEEE Std 802.3, 30.5.1.1.14."
26     ::= { dot3EponFecEntry 1}
27
28  dot3EponFecAbility OBJECT-TYPE
29      SYNTAX      INTEGER {
30          unknown(1),
31          supported(2),
32          unsupported(3)
33      }
34      MAX-ACCESS  read-only
35      STATUS      current
36      DESCRIPTION
37         "An object that indicates the support of operation of the
38         optional FEC sublayer of the 1000BASE-PX PHY specified
39         in IEEE Std 802.3, 65.2.
40         unknown(1) value is assigned in the initialization, for non
41         FEC support state or type not yet known. unsupported(3)
42         value is assigned when the sublayer is not supported.
43         supported(2) value is assigned when the sublayer is
44         supported.
45         This object is applicable for an OLT, with the same
46         value for all virtual interfaces, and for an ONU.
47         The FEC counters will have a zero value when the
48         interface is not supporting FEC.
49         The counters:
50         dot3EponFecPCSCodingViolation - not affected by FEC
51         ability.
52         dot3EponFecCorrectedBlocks - has a zero value when
53         dot3EponFecAbility is unknown(1) and unsupported(3).
54         dot3EponFecUncorrectableBlocks - has a zero value when
55         dot3EponFecAbility is unknown(1) and unsupported(3).
56         dot3EponFecBufferHeadCodingViolation - has a zero value
57         when dot3EponFecAbility is unknown(1) and
58         unsupported(3)."
```

REFERENCE "IEEE Std 802.3, 30.5.1.1.15."

```
1      ::= { dot3EponFecEntry 2}
2
3  dot3EponFecMode OBJECT-TYPE
4      SYNTAX  INTEGER {
5          unknown(1),
6          disabled(2),
7          enabled(3)
8      }
9
10     MAX-ACCESS  read-write
11     STATUS  current
12     DESCRIPTION
13         "An object that defines the mode of operation of the
14         optional FEC sublayer of the 1000BASE-PX PHY, specified
15         in IEEE Std 802.3, 65.2, and reflects its state.
16         A GET operation returns the current mode of operation
17         of the PHY. A SET operation changes the mode of
18         operation of the PHY to the indicated value.
19         unknown(1) value is assigned in the initialization for non
20         FEC support state or type not yet known.
21         disabled(2) value is assigned when the FEC sublayer is
22         operating in disabled mode.
23         enabled(3) value is assigned when the FEC sublayer is
24         operating in FEC mode.
25         The write operation is not restricted in this document
26         and can be done at any time. Changing dot3EponFecMode
27         state can lead to disabling the Forward Error Correction
28         on the respective interface, which can lead to a
29         degradation of the optical link, and therefore may lead
30         to an interruption of service for the users connected to
31         the respective EPON interface.
32         This object is applicable for an OLT and an ONU. At the
33         OLT, it has a distinct value for each virtual interface.
34         The counting of
35         the FEC counters will stop when the FEC of the interface
36         is disabled.
37         The counters:
38         dot3EponFecPCSCodingViolation - not affected by FEC
39         mode.
40         dot3EponFecCorrectedBlocks - stops counting when
41         Rx_FEC is not enabled. (unknown(1) and disabled(2)).
42         dot3EponFecUncorrectableBlocks - stops counting when
43         Rx_FEC is not enabled (unknown(1) and disabled(2)).
44         dot3EponFecBufferHeadCodingViolation - stops counting
45         when Rx_FEC is not enabled (unknown(1) and
46         disabled(2)).
47         The object:
48         dot3EponFecAbility - indicates the FEC ability and
49         is not affected by the dot3EponFecMode object."
50     REFERENCE  "IEEE Std 802.3, 30.5.1.1.16."
51     DEFVAL { unknown }
52     ::= { dot3EponFecEntry 3}
53
54  dot3EponFecCorrectedBlocks OBJECT-TYPE
55      SYNTAX  Counter64
56      MAX-ACCESS  read-only
57      STATUS  current
58      DESCRIPTION
59          "For 1000BASE-PX, 10GBASE-PR or 10/1GBASE-PRX PHYs, it is a
60          count of corrected FEC blocks. This counter will not
```

```
1         increment for other PHY Types. Increment the counter by
2         one for each received block that is corrected by the FEC
3         function in the PHY.
4         This object is applicable for an OLT and an ONU. At the
5         OLT, it has a distinct value for each virtual interface.
6         Discontinuities of this counter can occur at
7         re-initialization of the management system and at other
8         times, as indicated by the value of the
9         ifCounterDiscontinuityTime object of the Interfaces Group MIB
10        module."
11    REFERENCE    "IEEE Std 802.3, 30.5.1.1.17."
12    ::= { dot3EponFecEntry 4}
13
14    dot3EponFecUncorrectableBlocks OBJECT-TYPE
15        SYNTAX Counter64
16        MAX-ACCESS read-only
17        STATUS current
18        DESCRIPTION
19            "For 1000BASE-PX, 10GBASE-PR or 10/1GBASE-PRX PHYs, it is a
20            count of uncorrectable FEC blocks. This counter will not
21            increment for other PHY Types. Increment the counter by
22            one for each FEC block that is determined to be
23            uncorrectable by the FEC function in the PHY.
24            This object is applicable for an OLT and an ONU. At the
25            OLT, it has a distinct value for each virtual interface.
26            Discontinuities of this counter can occur at
27            re-initialization of the management system and at other
28            times, as indicated by the value of the
29            ifCounterDiscontinuityTime object of the Interfaces Group MIB
30            module."
31    REFERENCE    "IEEE Std 802.3, 30.5.1.1.18."
32    ::= { dot3EponFecEntry 5}
33
34    dot3EponFecBufferHeadCodingViolation OBJECT-TYPE
35        SYNTAX Counter64
36        UNITS      "octets"
37        MAX-ACCESS read-only
38        STATUS current
39        DESCRIPTION
40            "For a 1000 Mb/s operation, it is a count of the number of
41            invalid code-group received directly from the link. The
42            value has a meaning only in 1000 Mb/s mode and it is
43            zero otherwise.
44            This object is applicable for an OLT and an ONU. At the
45            OLT, it has a distinct value for each virtual interface.
46            Discontinuities of this counter can occur at
47            re-initialization of the management system and at other
48            times, as indicated by the value of the
49            ifCounterDiscontinuityTime object of the Interfaces Group MIB
50            module."
51    ::= { dot3EponFecEntry 6}
52
53    -- ExtendedPackage managed object definitions
54
55    dot3ExtPkgObjects OBJECT IDENTIFIER ::= {dot3EponObjects 4}
56
57    dot3ExtPkgControlObjects OBJECT IDENTIFIER ::= { dot3ExtPkgObjects 1}
58
59    dot3ExtPkgControlTable OBJECT-TYPE
```



```
1      SYNTAX SEQUENCE OF Dot3ExtPkgControlEntry
2      MAX-ACCESS not-accessible
3      STATUS current
4      DESCRIPTION
5          "A table of Extended package Control management
6           objects. Entries in the table are control and status
7           indication objects of an EPON interface, which are
8           gathered in an extended package as an addition to the
9           objects based on the IEEE Std 802.3, Clause 30, attributes.
10          Each object has a row for every virtual link denoted by
11          the corresponding ifIndex.
12          The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
13          register (15-bit field and a broadcast bit) limiting the
14          number of virtual links to 32768. Typically the number
15          of expected virtual links in a PON is like the number of
16          ONUs, which is 32-64, plus an additional entry for
17          broadcast LLID."
18      ::= { dot3ExtPkgControlObjects 1 }
19
20
21
22  dot3ExtPkgControlEntry OBJECT-TYPE
23      SYNTAX Dot3ExtPkgControlEntry
24      MAX-ACCESS not-accessible
25      STATUS current
26      DESCRIPTION
27          "An entry in the Extended package Control table.
28           Rows exist for an OLT interface and an ONU interface.
29           A row in the table is denoted by the ifIndex of the link
30           and it is created when the ifIndex is created.
31           The rows in the table for an ONU interface are created
32           at system initialization.
33           The row in the table corresponding to the OLT ifIndex
34           and the row corresponding to the broadcast virtual link
35           are created at system initialization.
36           A row in the table corresponding to the ifIndex of a
37           virtual links is created when a virtual link is
38           established (ONU registers) and deleted when the virtual
39           link is deleted (ONU deregisters)."
```

```
40      INDEX { ifIndex }
41      ::= { dot3ExtPkgControlTable 1 }
42
43
44
45  Dot3ExtPkgControlEntry ::=
46      SEQUENCE {
47          dot3ExtPkgObjectReset                INTEGER,
48          dot3ExtPkgObjectPowerDown            TruthValue,
49          dot3ExtPkgObjectNumberOfLLIDs        Unsigned32,
50          dot3ExtPkgObjectFecEnabled           INTEGER,
51          dot3ExtPkgObjectReportMaximumNumQueues Unsigned32,
52          dot3ExtPkgObjectRegisterAction       INTEGER
53      }
54
55
56  dot3ExtPkgObjectReset OBJECT-TYPE
57      SYNTAX INTEGER {
58          running(1),
59          reset(2)
60      }
61      MAX-ACCESS read-write
62      STATUS current
63      DESCRIPTION
64          "This object is used to reset the EPON interface. The
```

```
1         interface may be unavailable while the reset occurs and
2         data may be lost.
3         Setting this object to running(1) will cause the
4         interface to enter into running mode. Setting this
5         object to reset(2) will cause the interface to go into
6         reset mode. When getting running(1), the interface is in
7         running mode. When getting reset(2), the interface is in
8         reset mode.
9
10        The write operation is not restricted in this document
11        and can be done at any time. Changing
12        dot3ExtPkgObjectReset state can lead to a reset of the
13        respective interface, leading to an interruption of
14        service for the users connected to the respective EPON
15        interface.
16        This object is applicable for an OLT and an ONU. At the
17        OLT, it has a distinct value for each virtual interface.
18        A reset for a specific virtual interface resets only
19        this virtual interface and not the physical interface.
20        Thus, a virtual link that is malfunctioning can be
21        reset without affecting the operation of other virtual
22        interfaces.
23        The reset can cause Discontinuities in the values of the
24        counters of the interface, similar to re-initialization
25        of the management system. Discontinuity should be
26        indicated by the ifCounterDiscontinuityTime object of
27        the Interfaces Group MIB module."
28
29        DEFVAL { running }
30        ::= { dot3ExtPkgControlEntry 1 }
31
32
33        dot3ExtPkgObjectPowerDown OBJECT-TYPE
34            SYNTAX TruthValue
35            MAX-ACCESS read-write
36            STATUS current
37            DESCRIPTION
38                "This object is used to power down the EPON interface.
39                The interface may be unavailable while the power down
40                occurs and data may be lost.
41                Setting this object to true(1) will cause the interface
42                to enter into power down mode. Setting this object to
43                false(2) will cause the interface to go out of power
44                down mode. When getting true(1), the interface is in
45                power down mode. When getting false(2), the interface is
46                not in power down mode.
47                The write operation is not restricted in this document
48                and can be done at any time. Changing
49                dot3ExtPkgObjectPowerDown state can lead to a power down
50                of the respective interface, leading to an interruption
51                of service of the users connected to the respective EPON
52                interface.
53                This object is applicable for an OLT and an ONU. At the
54                OLT, it has a distinct value for each virtual interface.
55                A power down/up of a specific virtual interface affects
56                only the virtual interface and not the physical
57                interface. Hence a virtual link, which needs a certain
58                handling, can be powered down and then powered up without
59                disrupting the operation of other virtual interfaces.
60                The object is relevant when the admin state of the
61                interface is active as set by the dot3MpcpAdminState."
62
63        DEFVAL { false }
```

```
1      ::= { dot3ExtPkgControlEntry 2 }
2
3  dot3ExtPkgObjectNumberOfLLIDs OBJECT-TYPE
4      SYNTAX  Unsigned32
5      MAX-ACCESS  read-only
6      STATUS  current
7      DESCRIPTION
8
9          "A read only object that indicates the number of
10         registered LLIDs. The initialization value is 0.
11         This object is applicable for an OLT with the same
12         value for all virtual interfaces and for an ONU.
13         The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
14         register (15-bit field and a broadcast bit) limiting the
15         number of virtual links to 32768. Typically the number
16         of expected virtual links in a PON is like the number of
17         ONUs, which is 32-64, plus an additional entry for
18         broadcast LLID. At the ONU the
19         number of LLIDs for an interface is one."
20
21      ::= { dot3ExtPkgControlEntry 3 }
22
23  dot3ExtPkgObjectFecEnabled OBJECT-TYPE
24      SYNTAX  INTEGER {
25          noFecEnabled(1),
26          fecTxEnabled(2),
27          fecRxEnabled(3),
28          fecTxRxEnabled(4)
29      }
30      MAX-ACCESS  read-write
31      STATUS  current
32      DESCRIPTION
33
34          "An object defining the FEC mode of operation of the
35          interface, and indicating its state. The modes defined in
36          this object are extensions to the FEC modes defined in
37          the dot3EponFecMode object.
38          When noFECEnabled(1), the interface does not enable FEC
39          mode.
40          When fecTxEnabled(2), the interface enables the FEC
41          transmit mode.
42          When fecRxEnabled(3), the interface enables the FEC
43          receive mode.
44          When fecTxRxEnabled(4), the interface enables the FEC
45          transmit and receive mode.
46          This object is applicable for an OLT and an ONU. At the
47          OLT, it has a distinct value for each virtual interface.
48          The FEC counters are referring to the receive path. The
49          FEC counters will stop when the FEC receive mode of the
50          interface is disabled, as defined by fecRxEnabled(3)
51          and fecTxRxEnabled(4) values.
52          The counters:
53          dot3EponFecPCSCodingViolation - not affected by FEC
54          mode.
55          dot3EponFecCorrectedBlocks - stops counting when
56          Rx_FEC is not enabled (noFecEnabled(1) and
57          fecTxEnabled(2)).
58          dot3EponFecUncorrectableBlocks - stops counting when
59          Rx_FEC is not enabled (noFecEnabled(1) and
60          fecTxEnabled(2)).
61          dot3EponFecBufferHeadCodingViolation - stops counting
62          when Rx_FEC is not enabled (noFecEnabled(1) and
```

```
1         fecTxEnabled(2)).
2     The objects:
3         dot3EponFecAbility - indicates the FEC ability and is
4         not affected by the FEC mode.
5         dot3EponFecMode - indicates the FEC mode for combined RX
6         and TX.
7     The write operation is not restricted in this document
8     and can be done at any time. Changing
9     dot3ExtPkgObjectFecEnabled state can lead to disabling
10    the Forward Error Correction on the respective interface,
11    which can lead to a degradation of the optical link, and
12    therefore may lead to an interruption of service for the
13    users connected to the respective EPON interface."
14    DEFVAL { noFecEnabled }
15    ::= { dot3ExtPkgControlEntry 4 }
16
17    dot3ExtPkgObjectReportMaximumNumQueues OBJECT-TYPE
18        SYNTAX  Unsigned32 (0..7)
19        MAX-ACCESS  read-only
20        STATUS  current
21        DESCRIPTION
22            "An object, that defines the maximal number of queues in
23            the REPORT message as defined in IEEE Std 802.3, Clause 64. For
24            further information please see the description of the
25            queue table.
26            This object is applicable for an OLT and an ONU. At the
27            OLT, it has a distinct value for each virtual interface."
28        DEFVAL { 0 }
29        ::= { dot3ExtPkgControlEntry 5 }
30
31    dot3ExtPkgObjectRegisterAction OBJECT-TYPE
32        SYNTAX  INTEGER {
33            none(1),
34            register(2),
35            deregister(3),
36            reregister(4)
37        }
38        MAX-ACCESS  read-write
39        STATUS  current
40        DESCRIPTION
41            "An object configuring the registration state of an
42            interface, and indicating its registration state.
43            Write operation changes the registration state to its new
44            value.
45            Read operation returns the value of the state.
46            The registration state is reflected in this object and in
47            the dot3MpcpRegistrationState object.
48            none(1) indicates an unknown state,
49            register(2) indicates a registered LLID,
50            deregister(3) indicates a deregistered LLID,
51            reregister(4) indicates an LLID that is reregistering.
52            The following list describes the operation of the
53            interface, as specified in the IEEE Std 802.3, when a write
54            operation is setting a value.
55            none(1) - not doing any action.
56            register(2) - registering an LLID that has been requested
57            for registration (The LLID is in registering mode.
58            dot3MpcpRegistrationState - registering(2) ).
59            deregister(3) - deregisters an LLID that is registered
```

```

1         (dot3MpcpRegistrationState - registered(3) ).
2         reregister(4) - reregister an LLID that is registered
3         (dot3MpcpRegistrationState - registered(3) ).
4         The behavior of an ONU and OLT interfaces, at each one
5         of the detailed operation at each state, is described in
6         the registration state machine of figure 64-22,
7         IEEE Std 802.3.
8         This object is applicable for an OLT and an ONU. At the
9         OLT, it has a distinct value for each virtual interface.
10        The write operation is not restricted in this document
11        and can be done at any time. Changing
12        dot3ExtPkgObjectRegisterAction state can lead to a change
13        in the registration state of the respective interface
14        leading to a deregistration and an interruption of
15        service of the users connected to the respective EPON
16        interface."
17        DEFVAL { none }
18        ::= { dot3ExtPkgControlEntry 6 }
19
20
21
22        dot3ExtPkgQueueTable OBJECT-TYPE
23            SYNTAX SEQUENCE OF Dot3ExtPkgQueueEntry
24            MAX-ACCESS not-accessible
25            STATUS current
26            DESCRIPTION
27                "A table of the extended package objects for queue
28                management. The IEEE Std 802.3 MPCP defines a report message
29                of the occupancy of the transmit queues for the feedback
30                BW request from the ONUs. These queues serve the uplink
31                transmission of the ONU and data is gathered there until
32                the ONU is granted for transmission.
33                The management table of the queues is added here mainly
34                to control the reporting and to gather some statistics
35                of their operation. This table is not duplicating
36                existing management objects of bridging queues,
37                specified in IEEE Std 802.1D, since the existence of a
38                dedicated transmit queuing mechanism is implied in the
39                IEEE Std 802.3, and the ONU may be a device that is not a
40                bridge with embedded bridging queues.
41                The format of the REPORT message, as specified
42                in IEEE Std 802.3, is presented below:
43
44                +-----+
45                |          Destination Address          |
46                +-----+
47                |          Source Address              |
48                +-----+
49                |          Length/Type                 |
50                +-----+
51                |          OpCode                      |
52                +-----+
53                |          TimeStamp                   |
54                +-----+
55                |          Number of queue Sets        |
56                +-----+
57                |          Report bitmap                |
58                +-----+
59                |          Queue 0 report              |
60                +-----+
61                |          Queue 1 report              |
62                +-----+
63
64                /|\
65                |
66                | repeated for
67                | every
68                | queue_set

```

1		Queue 2 report		
2	+-----		+-----	
3		Queue 3 report		
4	+-----		+-----	
5		Queue 4 report		
6	+-----		+-----	
7		Queue 5 report		
8	+-----		+-----	
9		Queue 6 report		
10	+-----		+-----	
11		Queue 7 report		
12	+-----		+-----	
13		Pad/reserved		\\ /
14	+-----		+-----	
15		FCS		
16	+-----		+-----	

The 'Queue report' field reports the occupancy of each uplink transmission queue.

The number of queue sets defines the number of the reported sets, as would be explained in the description of the dot3ExtPkgQueueSetsTable table. For each set the report bitmap defines which queue is present in the report, meaning that although the MPCP REPORT message can report up to 8 queues in a REPORT message, the actual number is flexible. The Queue table has a variable size that is limited by the dot3ExtPkgObjectReportMaximumNumQueues object, as an ONU can have fewer queues to report.

The entries in the table are control and status indication objects for managing the queues of an EPON interface that are gathered in an extended package as an addition to the objects that are based on the IEEE Std 802.3 attributes.

Each object has a row for every virtual link and for every queue in the report.

The LLID field, as defined in the IEEE Std 802.3, is a 2-byte register (15-bit field and a broadcast bit) limiting the number of virtual links to 32768. Typically the number of expected virtual links in a PON is like the number of ONUs, which is 32-64, plus an additional entry for broadcast LLID.

The number of queues is between 0 and 7 and limited by dot3ExtPkgObjectReportMaximumNumQueues."

::= { dot3ExtPkgControlObjects 2 }

dot3ExtPkgQueueEntry OBJECT-TYPE

SYNTAX Dot3ExtPkgQueueEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the Extended package Queue table. At the OLT, the rows exist for each ifIndex and dot3QueueIndex. At the ONU, rows exist for the single ifIndex for each dot3QueueIndex.

Rows in the table are created when the ifIndex of the link is created. A set of rows per queue are added for each ifIndex, denoted by the dot3QueueIndex.

A set of rows per queue in the table, for an ONU

```
1         interface, are created at the system initialization.
2         A set of rows per queue in the table, corresponding to
3         the OLT ifIndex and a set of rows per queue
4         corresponding to the broadcast virtual link, are
5         created at the system initialization.
6         A set of rows per queue in the table, corresponding to
7         the ifIndex of a virtual link, are created when the
8         virtual link is established (ONU registers), and deleted
9         when the virtual link is deleted (ONU deregisters)."
```

10 INDEX { ifIndex, dot3QueueIndex }

11 ::= { dot3ExtPkgQueueTable 1 }

12

13

14 Dot3ExtPkgQueueEntry ::=

15 SEQUENCE {

16 dot3QueueIndex Unsigned32,

17 dot3ExtPkgObjectReportNumThreshold Unsigned32,

18 dot3ExtPkgObjectReportMaximumNumThreshold Unsigned32,

19 dot3ExtPkgStatTxFramesQueue Counter64,

20 dot3ExtPkgStatRxFramesQueue Counter64,

21 dot3ExtPkgStatDroppedFramesQueue Counter64

22 }

23

24

25 dot3QueueIndex OBJECT-TYPE

26 SYNTAX Unsigned32 (0..7)

27 MAX-ACCESS not-accessible

28 STATUS current

29 DESCRIPTION

30 "An object that identifies an index for the queue table

31 reflecting the queue index of the queues that are

32 reported in the MPCP REPORT message as defined in

33 IEEE Std 802.3, Clause 64.

34 The number of queues is between 0 and 7, and limited by

35 dot3ExtPkgObjectReportMaximumNumQueues."

36 ::= { dot3ExtPkgQueueEntry 1 }

37

38

39

40 dot3ExtPkgObjectReportNumThreshold OBJECT-TYPE

41 SYNTAX Unsigned32 (0..7)

42 MAX-ACCESS read-write

43 STATUS current

44 DESCRIPTION

45 "An object that defines the number of thresholds for each

46 queue in the REPORT message as defined in IEEE Std 802.3,

47 Clause 64.

48 Each queue_set reporting will provide information on the

49 queue occupancy of frames below the matching Threshold.

50 Read operation reflects the number of thresholds.

51 Write operation sets the number of thresholds for each

52 queue.

53 The write operation is not restricted in this document

54 and can be done at any time. Value cannot exceed the

55 maximal value defined by the

56 dot3ExtPkgObjectReportMaximumNumThreshold object.

57 Changing dot3ExtPkgObjectReportNumThreshold can lead to

58 a change in the reporting of the ONU interface and

59 therefore to a change in the bandwidth allocation of the

60 respective interface. This change may lead a degradation

61 or an interruption of service of the users connected to

62 the respective EPON interface.

63 This object is applicable for an OLT and an ONU. At the

64

65

```
1             OLT, it has a distinct value for each virtual interface
2             and for each queue. At the ONU, it has a distinct value
3             for each queue."
4     DEFVAL { 0 }
5     ::= { dot3ExtPkgQueueEntry 2 }
6
7     dot3ExtPkgObjectReportMaximumNumThreshold OBJECT-TYPE
8     SYNTAX  Unsigned32 (0..7)
9     MAX-ACCESS  read-only
10    STATUS  current
11    DESCRIPTION
12        "An object, that defines the maximal number of thresholds
13        for each queue in the REPORT message as defined in
14        IEEE Std 802.3, Clause 64. Each queue_set reporting will
15        provide information on the queue occupancy of frames
16        below the matching Threshold.
17        This object is applicable for an OLT and an ONU. At the
18        OLT, it has a distinct value for each virtual interface
19        and for each queue. At the ONU, it has a distinct value
20        for each queue."
21    DEFVAL { 0 }
22    ::= { dot3ExtPkgQueueEntry 3 }
23
24    dot3ExtPkgStatTxFramesQueue OBJECT-TYPE
25    SYNTAX  Counter64
26    UNITS      "frames"
27    MAX-ACCESS  read-only
28    STATUS  current
29    DESCRIPTION
30        "A count of the number of times a frame transmission
31        occurs from the corresponding 'Queue'.
32        Increment the counter by one for each frame transmitted,
33        which is an output of the 'Queue'.
34        The 'Queue' marking matches the REPORT MPCP message
35        Queue field as defined in IEEE Std 802.3, Clause 64.
36        This object is applicable for an OLT and an ONU. At the
37        OLT, it has a distinct value for each virtual interface
38        and for each queue. At the ONU, it has a distinct value
39        for each queue.
40        At the OLT the value should be zero.
41        Discontinuities of this counter can occur at
42        re-initialization of the management system and at other
43        times, as indicated by the value of the
44        ifCounterDiscontinuityTime object of the Interfaces Group MIB
45        module."
46    ::= { dot3ExtPkgQueueEntry 4}
47
48    dot3ExtPkgStatRxFramesQueue OBJECT-TYPE
49    SYNTAX  Counter64
50    UNITS      "frames"
51    MAX-ACCESS  read-only
52    STATUS  current
53    DESCRIPTION
54        "A count of the number of times a frame reception
55        occurs from the corresponding 'Queue'.
56        Increment the counter by one for each frame received,
57        which is an input to the corresponding 'Queue'.
58        The 'Queue' marking matches the REPORT MPCP message
59        Queue field as defined in IEEE Std 802.3, Clause 64.
```



```
1           This object is applicable for an OLT and an ONU. At the
2           OLT, it has a distinct value for each virtual interface
3           and for each queue. At the ONU, it has a distinct value
4           for each queue.
5           Discontinuities of this counter can occur at
6           re-initialization of the management system and at other
7           times, as indicated by the value of the
8           ifCounterDiscontinuityTime object of the Interfaces Group MIB
9           module."
10          ::= { dot3ExtPkgQueueEntry 5}
11
12
13  dot3ExtPkgStatDroppedFramesQueue OBJECT-TYPE
14      SYNTAX Counter64
15      UNITS "frames"
16      MAX-ACCESS read-only
17      STATUS current
18      DESCRIPTION
19          "A count of the number of times a frame drop
20          occurs from the corresponding 'Queue'.
21          Increment the counter by one for each frame dropped
22          from the corresponding 'Queue'.
23          The 'Queue' marking matches the REPORT MPCP message
24          Queue field as defined in IEEE Std 802.3, Clause 64.
25          This object is applicable for an OLT and an ONU. At the
26          OLT, it has a distinct value for each virtual interface
27          and for each queue. At the ONU, it has a distinct value
28          for each queue.
29          At the OLT, the value should be zero.
30          Discontinuities of this counter can occur at
31          re-initialization of the management system and at other
32          times, as indicated by the value of the
33          ifCounterDiscontinuityTime object of the Interfaces Group MIB
34          module."
35          ::= { dot3ExtPkgQueueEntry 6}
36
37
38
39  dot3ExtPkgQueueSetsTable OBJECT-TYPE
40      SYNTAX SEQUENCE OF Dot3ExtPkgQueueSetsEntry
41      MAX-ACCESS not-accessible
42      STATUS current
43      DESCRIPTION
44          "A table of Extended package objects used for the
45          management of the queue_sets. Entries are control and
46          status indication objects of an EPON interface, which
47          are gathered in an extended package as an addition to
48          the objects based on the IEEE Std 802.3 attributes. The
49          objects in this table are specific for the queue_sets,
50          which are reported in the MPCP REPORT message as defined
51          in IEEE Std 802.3, Clause 64.
52          The IEEE Std 802.3 MPCP defines a report message of the
53          occupancy of the transmit queues for the feedback BW
54          request from the ONUs. These queues serve the uplink
55          transmission of the ONU and data is gathered there until
56          the ONU is granted for transmission.
57          The management table of the queues_sets is added here
58          mainly to control the reporting and to gather some
59          statistics of their operation. This table is not
60          duplicating existing management objects of bridging
61          queues, specified in IEEE Std 802.1D, since the existence of a
62          dedicated transmit queuing mechanism is implied in the
```

IEEE Std 802.3, and the ONU may be a device that is not a bridge with embedded bridging queues. The format of the REPORT message, as specified in IEEE Std 802.3, is presented below:

Destination Address	
Source Address	
Length/Type	
OpCode	
TimeStamp	
Number of queue Sets	
Report bitmap	/ \
Queue 0 report	repeated for every queue_set
Queue 1 report	
Queue 2 report	
Queue 3 report	
Queue 4 report	
Queue 5 report	
Queue 6 report	
Queue 7 report	
Pad/reserved	\ /
FCS	

As can be seen from the message format, the ONU interface reports of the status of up to 8 queues and it can report in a single MPCP REPORT message of a few sets of queues.

The number of queue_sets defines the number of the reported sets, and it can reach a value of up to 8. It means that an ONU can hold a variable number of sets between 0 and 7.

The dot3ExtPkgQueueSetsTable table has a variable queue_set size that is limited by the dot3ExtPkgObjectReportMaximumNumThreshold object as an ONU can have fewer queue_sets to report.

The 'Queue report' field reports the occupancy of each uplink transmission queue. The queue_sets can be used to report the occupancy of the queues in a few levels as to allow granting, in an accurate manner, of only part of the data available in the queues. A Threshold is defined for each queue_set to define the level of the queue that is counted for the report of the occupancy.

```

1      The threshold is reflected in the queue_set table by the
2      dot3ExtPkgObjectReportThreshold object.
3      For each queue set, the report bitmap defines which
4      queues are present in the report, meaning that
5      although the MPCP REPORT message can report of up to 8
6      queues in a REPORT message, the actual number is
7      flexible.
8      The dot3ExtPkgQueueSetsTable table has a variable queue
9      size that is limited by the
10     dot3ExtPkgObjectReportMaximumNumQueues object as an ONU
11     can have fewer queues to report.
12     Each object has a row for every virtual link, for each
13     queue in the report and for each queue_set in the queue.
14     The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
15     register (15-bit field and a broadcast bit) limiting the
16     number of virtual links to 32768. Typically the number
17     of expected virtual links in a PON is like the number of
18     ONUs, which is 32-64, plus an additional entry for
19     broadcast LLID.
20     The number of queues is between 0 and 7 and limited by
21     dot3ExtPkgObjectReportMaximumNumQueues.
22     The number of queues_sets is between 0 and 7 and limited
23     by dot3ExtPkgObjectReportMaximumNumThreshold."
24     ::= { dot3ExtPkgControlObjects 3 }
25
26
27
28     dot3ExtPkgQueueSetsEntry OBJECT-TYPE
29     SYNTAX Dot3ExtPkgQueueSetsEntry
30     MAX-ACCESS not-accessible
31     STATUS current
32     DESCRIPTION
33         "An entry in the Extended package queue_set table. At
34         the OLT, the rows exist for each ifIndex,
35         dot3QueueSetQueueIndex and dot3QueueSetIndex. At the
36         ONU, rows exist for the single ifIndex, for each
37         dot3QueueSetQueueIndex and dot3QueueSetIndex.
38         Rows in the table are created when the ifIndex of the
39         link is created. A set of rows per queue and per
40         queue_set are added for each ifIndex, denoted by
41         dot3QueueSetIndex and dot3QueueSetQueueIndex.
42         A set of rows per queue and per queue_set in the table,
43         for an ONU interface are created at system
44         initialization.
45         A set of rows per queue and per queue_Set in the table,
46         corresponding to the OLT ifIndex and a set of rows per
47         queue and per queue_set, corresponding to the broadcast
48         virtual link, are created at system initialization.
49         A set of rows per queue and per queue_set in the table,
50         corresponding to the ifIndex of a virtual link are
51         created when the virtual link is established (ONU
52         registers) and deleted when the virtual link is deleted
53         (ONU deregisters)."
```

```

54     INDEX { ifIndex,
55             dot3QueueSetQueueIndex, dot3QueueSetIndex }
56     ::= { dot3ExtPkgQueueSetsTable 1 }
57
58
59
60
61
62
63
64
65
```

```

Dot3ExtPkgQueueSetsEntry ::=
SEQUENCE {
    dot3QueueSetQueueIndex      Unsigned32,
    dot3QueueSetIndex           Unsigned32,

```

```
1      dot3ExtPkgObjectReportThreshold          Unsigned32
2      }
3
4  dot3QueueSetQueueIndex OBJECT-TYPE
5      SYNTAX  Unsigned32 (0..7)
6      MAX-ACCESS  not-accessible
7      STATUS  current
8      DESCRIPTION
9          "An object that identifies the queue index for the
10             dot3ExtPkgQueueSetsTable table. The queues are reported
11             in the MPCP REPORT message as defined in IEEE Std 802.3,
12             Clause 64.
13             The number of queues is between 0 and 7, and limited by
14             dot3ExtPkgObjectReportMaximumNumQueues.
15             Value corresponds to the dot3QueueIndex of the queue
16             table."
17 ::= { dot3ExtPkgQueueSetsEntry 1 }
18
19 dot3QueueSetIndex OBJECT-TYPE
20     SYNTAX  Unsigned32 (0..7)
21     MAX-ACCESS  not-accessible
22     STATUS  current
23     DESCRIPTION
24         "An object that identifies the queue_set index for the
25             dot3ExtPkgQueueSetsTable table. The queues are reported
26             in the MPCP REPORT message as defined in IEEE Std 802.3,
27             Clause 64.
28             The number of queues_sets is between 0 and 7, and
29             limited by dot3ExtPkgObjectReportMaximumNumThreshold."
30 ::= { dot3ExtPkgQueueSetsEntry 2 }
31
32 dot3ExtPkgObjectReportThreshold OBJECT-TYPE
33     SYNTAX  Unsigned32
34     UNITS    "TQ (16 ns)"
35     MAX-ACCESS  read-write
36     STATUS  current
37     DESCRIPTION
38         "An object that defines the value of a threshold report
39             for each queue_set in the REPORT message as defined in
40             IEEE Std 802.3, Clause 64. The number of sets for each queue
41             is dot3ExtPkgObjectReportNumThreshold.
42             In the REPORT message, each queue_set reporting will
43             provide information on the occupancy of the queues for
44             frames below the matching Threshold.
45             The value returned shall be in Time quanta (TQ), which
46             is 16 ns or 2 octets increments.
47             Read operation provides the threshold value. Write
48             operation sets the value of the threshold.
49             The write operation is not restricted in this document
50             and can be done at any time. Changing
51             dot3ExtPkgObjectReportThreshold can lead to a change in
52             the reporting of the ONU interface and therefore to a
53             change in the bandwidth allocation of the respective
54             interface. This change may lead a degradation or an
55             interruption of service for the users connected to the
56             respective EPON interface.
57             This object is applicable for an OLT and an ONU. At the
58             OLT, it has a distinct value for each virtual interface,
59             for each queue and for each queue_set. At the ONU, it has
```

```

1          a distinct value for each queue and for each queue_set."
2      DEFVAL { 0 }
3      ::= { dot3ExtPkgQueueSetsEntry 3 }
4
5  --Optical Interface status tables
6
7  dot3ExtPkgOptIfTable OBJECT-TYPE
8      SYNTAX      SEQUENCE OF Dot3ExtPkgOptIfEntry
9      MAX-ACCESS not-accessible
10     STATUS      current
11     DESCRIPTION
12         "This table defines the control and status indication
13         objects for the optical interface of the EPON interface.
14         Each object has a row for every virtual link denoted by
15         the corresponding ifIndex.
16         The LLID field, as defined in the IEEE Std 802.3, is a 2-byte
17         register (15-bit field and a broadcast bit) limiting the
18         number of virtual links to 32768. Typically the number
19         of expected virtual links in a PON is like the number of
20         ONUs, which is 32-64, plus an additional entry for
21         broadcast LLID.
22         Although the optical interface is a physical interface,
23         there is a row in the table for each virtual interface.
24         The reason for having a separate row for each virtual
25         link is that the OLT has a separate link for each one of
26         the ONUs. For instance, ONUs could be in different
27         distances with different link budgets and different
28         receive powers, therefore having different power alarms.
29         It is quite similar to a case of different physical
30         interfaces."
31     ::= { dot3ExtPkgControlObjects 5}
32
33  dot3ExtPkgOptIfEntry OBJECT-TYPE
34      SYNTAX      Dot3ExtPkgOptIfEntry
35      MAX-ACCESS not-accessible
36      STATUS      current
37      DESCRIPTION
38          "An entry in the optical interface table of the EPON
39          interface.
40          Rows exist for an OLT interface and an ONU interface.
41          A row in the table is denoted by the ifIndex of the link
42          and it is created when the ifIndex is created.
43          The rows in the table for an ONU interface are created
44          at system initialization.
45          The row in the table corresponding to the OLT ifIndex
46          and the row corresponding to the broadcast virtual link
47          are created at system initialization.
48          A row in the table corresponding to the ifIndex of a
49          virtual links is created when a virtual link is
50          established (ONU registers) and deleted when the virtual
51          link is deleted (ONU deregisters)."
```

INDEX	{ ifIndex }
-------	-------------

```

58     ::= { dot3ExtPkgOptIfTable 1 }
59
60  Dot3ExtPkgOptIfEntry ::=
61      SEQUENCE {
62          dot3ExtPkgOptIfSuspectedFlag      TruthValue,
63          dot3ExtPkgOptIfInputPower          Integer32,
64          dot3ExtPkgOptIfLowInputPower       Integer32,
```

```

1      dot3ExtPkgOptIfHighInputPower          Integer32,
2      dot3ExtPkgOptIfLowerInputPowerThreshold Integer32,
3      dot3ExtPkgOptIfUpperInputPowerThreshold Integer32,
4      dot3ExtPkgOptIfOutputPower             Integer32,
5      dot3ExtPkgOptIfLowOutputPower          Integer32,
6      dot3ExtPkgOptIfHighOutputPower         Integer32,
7      dot3ExtPkgOptIfLowerOutputPowerThreshold Integer32,
8      dot3ExtPkgOptIfUpperOutputPowerThreshold Integer32,
9      dot3ExtPkgOptIfSignalDetect            TruthValue,
10     dot3ExtPkgOptIfTransmitAlarm           TruthValue,
11     dot3ExtPkgOptIfTransmitEnable          TruthValue
12     }
13
14
15     dot3ExtPkgOptIfSuspectedFlag OBJECT-TYPE
16         SYNTAX      TruthValue
17         MAX-ACCESS  read-only
18         STATUS      current
19         DESCRIPTION
20             "This object is a reliability indication.
21              If true, the data in this entry may be unreliable.
22              This object is applicable for an OLT and an ONU. At the
23              OLT, it has a distinct value for each virtual interface."
24             ::= { dot3ExtPkgOptIfEntry 1 }
25
26
27     dot3ExtPkgOptIfInputPower OBJECT-TYPE
28         SYNTAX      Integer32
29         UNITS       "0.1 dbm"
30         MAX-ACCESS  read-only
31         STATUS      current
32         DESCRIPTION
33             "The optical power monitored at the input.
34              This object is applicable for an OLT and an ONU. At the
35              OLT, it has a distinct value for each virtual interface."
36             ::= { dot3ExtPkgOptIfEntry 2 }
37
38
39     dot3ExtPkgOptIfLowInputPower OBJECT-TYPE
40         SYNTAX      Integer32
41         UNITS       "0.1 dbm"
42         MAX-ACCESS  read-only
43         STATUS      current
44         DESCRIPTION
45             "The lowest optical power monitored at the input during the
46              current 15-minute interval.
47              This object is applicable for an OLT and an ONU. At the
48              OLT, it has a distinct value for each virtual interface."
49             ::= { dot3ExtPkgOptIfEntry 3 }
50
51     dot3ExtPkgOptIfHighInputPower OBJECT-TYPE
52         SYNTAX      Integer32
53         UNITS       "0.1 dbm"
54         MAX-ACCESS  read-only
55         STATUS      current
56         DESCRIPTION
57             "The highest optical power monitored at the input during the
58              current 15-minute interval.
59              This object is applicable for an OLT and an ONU. At the
60              OLT, it has a distinct value for each virtual interface."
61             ::= { dot3ExtPkgOptIfEntry 4 }
62
63
64     dot3ExtPkgOptIfLowerInputPowerThreshold OBJECT-TYPE
65

```

```
1      SYNTAX  Integer32
2      UNITS   "0.1 dbm"
3      MAX-ACCESS  read-write
4      STATUS   current
5      DESCRIPTION
6          "The lower limit threshold on input power. If
7           dot3ExtPkgOptIfInputPower drops to this value or below,
8           a Threshold Crossing Alert (TCA) should be sent.
9           Reading will present the threshold value. Writing will
10          set the value of the threshold.
11          The write operation is not restricted in this document
12          and can be done at any time. Changing
13          dot3ExtPkgOptIfLowerInputPowerThreshold can lead to a Threshold
14          Crossing Alert (TCA) being sent for the respective interface.
15          This alert may be leading to an interruption of service for the
16          users connected to the respective EPON interface, depending on
17          the system action on such an alert.
18          This object is applicable for an OLT and an ONU. At the
19          OLT, it has a distinct value for each virtual interface."
20      ::= { dot3ExtPkgOptIfEntry 5 }
21
22  dot3ExtPkgOptIfUpperInputPowerThreshold OBJECT-TYPE
23      SYNTAX  Integer32
24      UNITS   "0.1 dbm"
25      MAX-ACCESS  read-write
26      STATUS   current
27      DESCRIPTION
28          "The upper limit threshold on input power. If
29           dot3ExtPkgOptIfInputPower reaches or exceeds this value,
30           a Threshold Crossing Alert (TCA) should be sent.
31           Reading will present the threshold value. Writing will
32           set the value of the threshold.
33           The write operation is not restricted in this document
34           and can be done at any time. Changing
35           dot3ExtPkgOptIfUpperInputPowerThreshold can lead to a Threshold
36           Crossing Alert (TCA) being sent for the respective interface.
37           This alert may be leading to an interruption of service for the
38           users connected to the respective EPON interface, depending on
39           the system action on such an alert.
40           This object is applicable for an OLT and an ONU. At the
41           OLT, it has a distinct value for each virtual interface."
42      ::= { dot3ExtPkgOptIfEntry 6 }
43
44  dot3ExtPkgOptIfOutputPower OBJECT-TYPE
45      SYNTAX  Integer32
46      UNITS   "0.1 dbm"
47      MAX-ACCESS  read-only
48      STATUS   current
49      DESCRIPTION
50          "The optical power monitored at the output.
51          This object is applicable for an OLT and an ONU. At the
52          OLT, it has a distinct value for each virtual interface."
53      ::= { dot3ExtPkgOptIfEntry 7 }
54
55  dot3ExtPkgOptIfLowOutputPower OBJECT-TYPE
56      SYNTAX  Integer32
57      UNITS   "0.1 dbm"
58      MAX-ACCESS  read-only
59      STATUS   current
```

```
1      DESCRIPTION
2      "The lowest optical power monitored at the output during the
3      current 15-minute interval.
4      This object is applicable for an OLT and an ONU. At the
5      OLT, it has a distinct value for each virtual interface."
6      ::= { dot3ExtPkgOptIfEntry 8 }
7
8
9      dot3ExtPkgOptIfHighOutputPower OBJECT-TYPE
10     SYNTAX  Integer32
11     UNITS   "0.1 dbm"
12     MAX-ACCESS  read-only
13     STATUS    current
14     DESCRIPTION
15     "The highest optical power monitored at the output during the
16     current 15-minute interval.
17     This object is applicable for an OLT and an ONU. At the
18     OLT, it has a distinct value for each virtual interface."
19     ::= { dot3ExtPkgOptIfEntry 9 }
20
21
22     dot3ExtPkgOptIfLowerOutputPowerThreshold OBJECT-TYPE
23     SYNTAX  Integer32
24     UNITS   "0.1 dbm"
25     MAX-ACCESS  read-write
26     STATUS    current
27     DESCRIPTION
28     "The lower limit threshold on output power. If
29     dot3ExtPkgOptIfOutputPower drops to this value or below,
30     a Threshold Crossing Alert (TCA) should be sent.
31     Reading will present the threshold value. Writing will
32     set the value of the threshold.
33     The write operation is not restricted in this document
34     and can be done at any time. Changing
35     dot3ExtPkgOptIfLowerOutputPowerThreshold can lead to a Threshold
36     Crossing Alert (TCA) being sent for the respective interface.
37     This alert may be leading to an interruption of service for the
38     users connected to the respective EPON interface, depending on
39     the system action on such an alert.
40     This object is applicable for an OLT and an ONU. At the
41     OLT, it has a distinct value for each virtual interface."
42     ::= { dot3ExtPkgOptIfEntry 10 }
43
44
45
46     dot3ExtPkgOptIfUpperOutputPowerThreshold OBJECT-TYPE
47     SYNTAX  Integer32
48     UNITS   "0.1 dbm"
49     MAX-ACCESS  read-write
50     STATUS    current
51     DESCRIPTION
52     "The upper limit threshold on output power. If
53     dot3ExtPkgOptIfOutputPower reaches or exceeds this value,
54     a Threshold Crossing Alert (TCA) should be sent.
55     Reading will present the threshold value. Writing will
56     set the value of the threshold.
57     The write operation is not restricted in this document
58     and can be done at any time. Changing
59     dot3ExtPkgOptIfUpperOutputPowerThreshold can lead to a Threshold
60     Crossing Alert (TCA) being sent for the respective interface.
61     This alert may be leading to an interruption of service of the
62     users connected to the respective EPON interface, depending on
63     the system action on such an alert.
64
65
```



```
1      This object is applicable for an OLT and an ONU. At the
2      OLT, it has a distinct value for each virtual interface."
3      ::= { dot3ExtPkgOptIfEntry 11 }
4
5      dot3ExtPkgOptIfSignalDetect OBJECT-TYPE
6          SYNTAX      TruthValue
7          MAX-ACCESS   read-only
8          STATUS      current
9          DESCRIPTION
10             "When getting true(1), there is a valid optical signal at
11             the receive that is above the optical power level for
12             signal detection. When getting false(2) the optical
13             signal at the receive is below the optical power level
14             for signal detection.
15             This object is applicable for an OLT and an ONU. At the
16             OLT, it has a distinct value for each virtual interface."
17             DEFVAL { false }
18             ::= { dot3ExtPkgOptIfEntry 12 }
19
20      dot3ExtPkgOptIfTransmitAlarm OBJECT-TYPE
21          SYNTAX      TruthValue
22          MAX-ACCESS   read-only
23          STATUS      current
24          DESCRIPTION
25             "When getting true(1) there is a non-valid optical signal
26             at the transmit of the interface, either a higher level
27             or lower level than expected. When getting false(2) the
28             optical signal at the transmit is valid and in the
29             required range.
30             This object is applicable for an OLT and an ONU. At the
31             OLT, it has a distinct value for each virtual interface."
32             DEFVAL { false }
33             ::= { dot3ExtPkgOptIfEntry 13 }
34
35      dot3ExtPkgOptIfTransmitEnable OBJECT-TYPE
36          SYNTAX      TruthValue
37          MAX-ACCESS   read-write
38          STATUS      current
39          DESCRIPTION
40             "Setting this object to true(1) will cause the optical
41             interface to start transmission (according to the
42             control protocol specified for the logical interface).
43             Setting this object to false(2) will cause the
44             interface to stop the optical transmission.
45             When getting true(1), the optical interface is in
46             transmitting mode (obeying to the logical control
47             protocol).
48             When getting false(2), the optical interface is not in
49             transmitting mode.
50             The write operation is not restricted in this document
51             and can be done at any time. Changing
52             dot3ExtPkgOptIfTransmitEnable state can lead to a halt
53             in the optical transmission of the respective interface
54             leading to an interruption of service of the users
55             connected to the respective EPON interface.
56             The object is relevant when the admin state of the
57             interface is active as set by the dot3MpcpAdminState.
58             This object is applicable for an OLT and an ONU. At the
59             OLT it, has a distinct value for each virtual interface."
```

```
1      DEFVAL { false }
2      ::= { dot3ExtPkgOptIfEntry 14 }
3
4      --
5      -- The MulticastIDs Table
6      --
7  dot3RecognizedMulticastIDsTable OBJECT-TYPE
8      SYNTAX      SEQUENCE OF Dot3RecognizedMulticastIDsEntry
9      MAX-ACCESS not-accessible
10     STATUS      current
11     DESCRIPTION
12         "A table of MulticastIDs to be recognized by this device."
13     REFERENCE   "IEEE Std 802.3, 30.3.5.1.25."
14     ::= { dot3EponObjects 5 }
15
16
17 dot3RecognizedMulticastIDsEntry OBJECT-TYPE
18     SYNTAX      Dot3RecognizedMulticastIDsEntry
19     MAX-ACCESS not-accessible
20     STATUS      current
21     DESCRIPTION
22         "An entry in the table of MulticastIDs to be recognized by this
23         device."
24     INDEX       { ifIndex, dot3RecognizedMulticastIDIndex }
25     ::= { dot3RecognizedMulticastIDsTable 1 }
26
27
28 Dot3RecognizedMulticastIDsEntry ::=
29     SEQUENCE {
30         dot3RecognizedMulticastIDIndex    Unsigned32,
31         dot3RecognizedMulticastID        Unsigned32
32     }
33
34
35 dot3RecognizedMulticastIDIndex OBJECT-TYPE
36     SYNTAX      Unsigned32 (0..127)
37     MAX-ACCESS not-accessible
38     STATUS      current
39     DESCRIPTION
40         "An index into the table of MulticastIDs to be recognized by this
41         device."
42     ::= { dot3RecognizedMulticastIDsEntry 1 }
43
44
45 dot3RecognizedMulticastID OBJECT-TYPE
46     SYNTAX      Unsigned32
47     MAX-ACCESS read-write
48     STATUS      current
49     DESCRIPTION
50         "An Unsigned32 representing a single MulticastID to be recognized
51         by this device."
52     REFERENCE   "IEEE Std 802.3, 30.3.5.1.25."
53     ::= { dot3RecognizedMulticastIDsEntry 2 }
54
55
56 -- Conformance statements
57
58 -- Conformance Groups
59
60 dot3EponGroups          OBJECT IDENTIFIER ::= { dot3EponConformance 1 }
61
62 dot3MpcpGroupBase OBJECT-GROUP
63     OBJECTS {
64         dot3MpcpOperStatus,
```

```
1         dot3MpcpAdminState,
2         dot3MpcpMode,
3         dot3MpcpSyncTime,
4         dot3MpcpLinkID,
5         dot3MpcpRemoteMACAddress,
6         dot3MpcpRegistrationState,
7         dot3MpcpMaximumPendingGrants,
8         dot3MpcpTransmitElapsed,
9         dot3MpcpReceiveElapsed,
10        dot3MpcpRoundTripTime
11    }
12    STATUS current
13    DESCRIPTION
14        "A collection of objects of dot3 Mpcp Control entity state
15         definition. Objects are per LLID."
16    ::= { dot3EponGroups 1 }
17
18    dot3MpcpGroupStat OBJECT-GROUP
19        OBJECTS {
20            dot3MpcpMACCtrlFramesTransmitted,
21            dot3MpcpMACCtrlFramesReceived,
22            dot3MpcpDiscoveryWindowsSent,
23            dot3MpcpDiscoveryTimeout,
24            dot3MpcpTxRegRequest,
25            dot3MpcpRxRegRequest,
26            dot3MpcpTxRegAck,
27            dot3MpcpRxRegAck,
28            dot3MpcpTxReport,
29            dot3MpcpRxReport,
30            dot3MpcpTxGate,
31            dot3MpcpRxGate,
32            dot3MpcpTxRegister,
33            dot3MpcpRxRegister
34        }
35    STATUS current
36    DESCRIPTION
37        "A collection of objects of dot3 Mpcp Statistics.
38         Objects are per LLID."
39    ::= { dot3EponGroups 2 }
40
41    dot3OmpGroupID OBJECT-GROUP
42        OBJECTS {
43            dot3OmpEmulationType
44        }
45    STATUS current
46    DESCRIPTION
47        "A collection of objects of dot3 OMP emulation entity
48         state definition. Objects are per LLID."
49    ::= { dot3EponGroups 3 }
50
51    dot3OmpGroupStat OBJECT-GROUP
52        OBJECTS {
53            dot3OmpEmulationSLDErrors,
54            dot3OmpEmulationCRC8Errors,
55            dot3OmpEmulationBadLLID,
56            dot3OmpEmulationGoodLLID,
57            dot3OmpEmulationOnuPonCastLLID,
58            dot3OmpEmulationOltPonCastLLID,
59            dot3OmpEmulationBroadcastBitNotOnuLlid,
```

```
1          dot3OmpEmulationOnuLLIDNotBroadcast,
2          dot3OmpEmulationBroadcastBitPlusOnuLlid,
3          dot3OmpEmulationNotBroadcastBitNotOnuLlid
4      }
5      STATUS current
6      DESCRIPTION
7          "A collection of objects of dot3 OMP emulation
8           Statistics. Objects are per LLID."
9      ::= { dot3EponGroups 4 }
10
11
12 dot3EponFecGroupAll OBJECT-GROUP
13     OBJECTS {
14         dot3EponFecPCSCodingViolation,
15         dot3EponFecAbility,
16         dot3EponFecMode,
17         dot3EponFecCorrectedBlocks,
18         dot3EponFecUncorrectableBlocks,
19         dot3EponFecBufferHeadCodingViolation
20     }
21     STATUS current
22     DESCRIPTION
23         "A collection of objects of dot3 FEC group control and
24          statistics. Objects are per LLID."
25     ::= { dot3EponGroups 5 }
26
27
28 dot3ExtPkgGroupControl OBJECT-GROUP
29     OBJECTS {
30         dot3ExtPkgObjectReset,
31         dot3ExtPkgObjectPowerDown,
32         dot3ExtPkgObjectNumberOfLLIDs,
33         dot3ExtPkgObjectFecEnabled,
34         dot3ExtPkgObjectReportMaximumNumQueues,
35         dot3ExtPkgObjectRegisterAction
36     }
37     STATUS current
38     DESCRIPTION
39         "A collection of objects of dot3ExtPkg control
40          definition. Objects are per LLID."
41     ::= { dot3EponGroups 6 }
42
43
44 dot3ExtPkgGroupQueue OBJECT-GROUP
45     OBJECTS {
46         dot3ExtPkgObjectReportNumThreshold,
47         dot3ExtPkgObjectReportMaximumNumThreshold,
48         dot3ExtPkgStatTxFramesQueue,
49         dot3ExtPkgStatRxFramesQueue,
50         dot3ExtPkgStatDroppedFramesQueue
51     }
52     STATUS current
53     DESCRIPTION
54         "A collection of objects of dot3ExtPkg Queue
55          control. Objects are per LLID, per queue."
56     ::= { dot3EponGroups 7 }
57
58
59 dot3ExtPkgGroupQueueSets OBJECT-GROUP
60     OBJECTS {
61         dot3ExtPkgObjectReportThreshold
62     }
63     STATUS current
```

```
1      DESCRIPTION
2          "A collection of objects of dot3ExtPkg queue_set
3          control. Objects are per LLID, per queue, per
4          queue_set."
5      ::= { dot3EponGroups 8 }
6
7  dot3ExtPkgGroupOptIf OBJECT-GROUP
8      OBJECTS {
9          dot3ExtPkgOptIfSuspectedFlag,
10         dot3ExtPkgOptIfInputPower,
11         dot3ExtPkgOptIfLowInputPower,
12         dot3ExtPkgOptIfHighInputPower,
13         dot3ExtPkgOptIfLowerInputPowerThreshold,
14         dot3ExtPkgOptIfUpperInputPowerThreshold,
15         dot3ExtPkgOptIfOutputPower,
16         dot3ExtPkgOptIfLowOutputPower,
17         dot3ExtPkgOptIfHighOutputPower,
18         dot3ExtPkgOptIfLowerOutputPowerThreshold,
19         dot3ExtPkgOptIfUpperOutputPowerThreshold,
20         dot3ExtPkgOptIfSignalDetect,
21         dot3ExtPkgOptIfTransmitAlarm,
22         dot3ExtPkgOptIfTransmitEnable
23     }
24     STATUS current
25     DESCRIPTION
26         "A collection of objects of control and status indication
27         of the optical interface.
28         Objects are per LLID."
29     ::= { dot3EponGroups 9 }
30
31  dot3EponGroupMulticastIDs OBJECT-GROUP
32      OBJECTS {
33         dot3RecognizedMulticastID
34     }
35     STATUS current
36     DESCRIPTION
37         "One of a set of MulticastIDs recognized by an EPON interface."
38     ::= { dot3EponGroups 10 }
39
40  -- Compliance statements
41
42  dot3EponCompliances
43      OBJECT IDENTIFIER ::= { dot3EponConformance 2 }
44
45  dot3MPCPCompliance MODULE-COMPLIANCE
46      STATUS current
47      DESCRIPTION "The compliance statement for MultiPoint
48                  Control Protocol interfaces."
49
50      MODULE -- this module
51      MANDATORY-GROUPS { dot3MpcpGroupBase }
52
53      GROUP dot3MpcpGroupStat
54      DESCRIPTION "This group is mandatory for all MPCP supporting
55                  interfaces for statistics collection."
56      ::= { dot3EponCompliances 1 }
57
58  dot3OmpeCompliance MODULE-COMPLIANCE
59      STATUS current
```

```
1      DESCRIPTION "The compliance statement for OMPEmulation
2                  interfaces."
3
4      MODULE -- this module
5      MANDATORY-GROUPS { dot3OmpeGroupID}
6
7      GROUP      dot3OmpeGroupStat
8      DESCRIPTION "This group is mandatory for all OMPemulation
9                  supporting interfaces for statistics collection."
10
11
12      ::= { dot3EponCompliances 2}
13
14      dot3EponFecCompliance MODULE-COMPLIANCE
15      STATUS      current
16      DESCRIPTION "The compliance statement for FEC EPON interfaces.
17                  This group is mandatory for all FEC supporting
18                  interfaces for control and statistics collection."
19
20
21      MODULE -- this module
22      MANDATORY-GROUPS { dot3EponFecGroupAll }
23
24      ::= { dot3EponCompliances 3}
25
26
27      dot3ExtPkgCompliance MODULE-COMPLIANCE
28      STATUS      current
29      DESCRIPTION "The compliance statement for EPON Interfaces
30                  using the extended package."
31
32      MODULE -- this module
33      MANDATORY-GROUPS { dot3ExtPkgGroupControl }
34
35      GROUP      dot3ExtPkgGroupQueue
36      DESCRIPTION " This group is mandatory for all EPON interfaces
37                  supporting REPORT queue management of the extended
38                  package."
39
40      GROUP      dot3ExtPkgGroupQueueSets
41      DESCRIPTION " This group is mandatory for all EPON interfaces
42                  supporting REPORT queue_sets management of the
43                  extended package."
44
45      GROUP      dot3ExtPkgGroupOptIf
46      DESCRIPTION "This group is mandatory for all EPON interfaces
47                  supporting optical interfaces management,
48                  of the extended package."
49
50
51      ::= { dot3EponCompliances 4}
52
53
54      dot3EponMulticastIDsCompliance MODULE-COMPLIANCE
55      STATUS      current
56      DESCRIPTION "The compliance statement for EPON Interfaces that
57                  support MulticastIDs."
58
59      MODULE -- this module
60      MANDATORY-GROUPS { dot3EponGroupMulticastIDs }
61
62      ::= { dot3EponCompliances 5 }
63
64      END
65
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

10. Ethernet-like interface MIB module

10.1 Introduction

This clause defines a portion of the MIB for use with SNMP. In particular, it defines objects for managing Ethernet-like interfaces.

10.2 Overview

Instances of these object types represent attributes of an interface to an Ethernet-like communications medium.

The definitions presented here are based on Clause 30 of IEEE Std 802.3. Implementors of these MIB objects should note that IEEE Std 802.3 explicitly describes (in the form of Pascal pseudocode) when, where, and how various MAC attributes are measured. IEEE Std 802.3 also describes the effects of MAC actions that may be invoked by manipulating instances of the MIB objects defined here.

To the extent that some of the attributes defined in IEEE Std 802.3 are represented by previously defined objects in MIB-2 from IETF RFC 1213 or in the Interfaces Group MIB defined in IETF RFC 2863, such attributes are not redundantly represented by objects defined in this clause. Among the attributes represented by objects defined in other MIB module specifications are the number of octets transmitted or received on a particular interface, the number of frames transmitted or received on a particular interface, the promiscuous status of an interface, the MAC address of an interface, and multicast information associated with an interface.

10.2.1 Relation to MIB-2

This subclause applies only when this MIB is used in conjunction with the IETF RFC 1213 interface group.

The relationship between an Ethernet-like interface and an interface in the context of MIB-2 is one-to-one. As such, the value of an ifIndex object instance can be directly used to identify corresponding instances of the objects defined herein.

10.2.2 Relation to the Interfaces Group MIB

The Interfaces Group MIB defined in IETF RFC 2863 requires that any MIB that is an adjunct of the Interfaces Group MIB clarify specific areas within the Interfaces Group MIB. These areas were intentionally left vague in the Interfaces Group MIB to avoid overconstraining the MIB, thereby precluding management of certain media-types.

Section 4 of IETF RFC 2863 enumerates several areas that a media-specific MIB must (wherein the word “must” is used in accordance with the requirements of IETF RFC 2119) clarify. Each of these areas is addressed in a following subclause. The implementor is referred to IETF RFC 2863 in order to understand the general intent of these areas.

10.2.2.1 ifRcvAddressTable

This table contains all IEEE 802.3 addresses, unicast, multicast, and broadcast, for which this interface will receive packets and forward them up to a higher layer entity for local consumption. The format of the address, contained in ifRcvAddressAddress, is the same as for ifPhysAddress.

In the event that the interface is part of a MAC bridge, this table does not include unicast addresses that are accepted for possible forwarding out some other port. This table is explicitly not intended to provide a bridge address filtering mechanism.

10.2.2.2 ifType

All Ethernet-like interfaces shall return ethernetCsmacd(6) for ifType. Information on the particular port type and operating speed is available from ifSpeed in the Interfaces Group MIB, and ifMauType in the MAU-MIB module defined in Clause 13. All Ethernet-like interfaces shall also implement the MAU-MIB module defined in Clause 13.¹⁷

10.2.2.3 ifXxxOctets

The Interfaces Group MIB octet counters, ifInOctets, ifOutOctets, ifHCInOctets, and ifHCOctets, include all octets in valid frames sent or received on the interface, including the MAC header and FCS, but not the preamble, start of frame delimiter, or extension octets. This corresponds to the definition of frameSize/8 in 4.2.7.1 of IEEE Std 802.3 (frameSize is defined in bits rather than in octets, and it is defined as $2 \times \text{addressSize} + \text{lengthOrTypeSize} + \text{dataSize} + \text{crcSize}$). They do not include the number of octets in collided or failed transmit attempts, since the MAC layer driver typically does not have visibility to count these octets. They also do not include octets in received invalid frames, since this information is normally not passed to the MAC layer, and since non-promiscuous MAC implementations cannot reliably determine whether an invalid frame was actually addressed to this station.

Note that these counters do include octets in valid MAC control frames sent or received on the interface, as well as octets in otherwise valid received MAC frames that are discarded by the MAC layer for some reason (insufficient buffer space, unknown protocol, etc.).

Note that the octet counters in IF-MIB do not exactly match the definition of the octet counters in IEEE Std 802.3. aOctetsTransmittedOK and aOctetsReceivedOK count only the octets in the clientData and Pad fields, whereas ifInOctets and ifOutOctets include the entire MAC frame, including MAC header and FCS. However, the IF-MIB counters can be derived from the IEEE 802.3 counters as follows in Equation (1) and Equation (2):

$$\text{ifInOctets} = \text{aOctetsReceivedOK} + (18 \times \text{aFramesReceivedOK}) \quad (1)$$

$$\text{ifOutOctets} = \text{aOctetsTransmittedOK} + (18 \times \text{aFramesTransmittedOK}) \quad (2)$$

Another difference to keep in mind between the IF-MIB counters and IEEE 802.3 counters is that, in IEEE Std 802.3, the frame counters and octet counters are incremented together. aOctetsTransmittedOK counts the number of octets in frames that were counted by aFramesTransmittedOK. aOctetsReceivedOK counts the number of octets in frames that were counted by aFramesReceivedOK. This is not the case with the IF-MIB counters. The IF-MIB octet counters count the number of octets sent to or received from the layer below this interface, whereas the packet counters count the number of packets sent to or received from the layer above. Therefore, received MAC Control frames, ifInDiscards, and ifInUnknownProtos are counted by ifInOctets, but not by ifInXcastPkts. Transmitted MAC Control frames are counted by ifOutOctets, but not by ifOutXcastPkts. ifOutDiscards and ifOutErrors are counted by ifOutXcastPkts, but not by ifOutOctets.

¹⁷There are three other interface types defined in IANAifType-MIB for Ethernet, namely, fastEther(62), fastEtherFX(69), and gigabitEthernet(117). Management applications should be prepared to receive these obsolete ifType values from older implementations.

10.2.2.4 ifXxxXcastPkts

The packet counters in the IF-MIB do not exactly match the definition of the frame counters in IEEE Std 802.3. aFramesTransmittedOK counts the number of frames successfully transmitted on the interface, whereas ifOutUcastPkts, ifOutMulticastPkts, and ifOutBroadcastPkts count the number of transmit requests made from a higher layer, whether or not the transmit attempt was successful. This means that packets counted by ifOutErrors or ifOutDiscards are also counted by ifOutXcastPkts, but they are not counted by aFramesTransmittedOK. This also means that, since MAC Control frames are generated by a sublayer internal to the interface layer rather than by a higher layer, they are not counted by ifOutXcastPkts, but they are counted by aFramesTransmittedOK:

$$\begin{aligned} \text{aFramesTransmittedOK} = & \text{ifOutUcastPkts} + \text{ifOutMulticastPkts} \\ & + \text{ifOutBroadcastPkts} + \text{dot3OutPauseFrames} \\ & - (\text{ifOutErrors} + \text{ifOutDiscards}) \end{aligned} \quad (3)$$

Similarly, aFramesReceivedOK counts the number of frames received successfully by the interface, whether or not they are passed to a higher layer, whereas ifInUcastPkts, ifInMulticastPkts, and ifInBroadcastPkts count only the number of packets passed to a higher layer. This means that packets counted by ifInDiscards or ifInUnknownProtos are also counted by aFramesReceivedOK, but they are not counted by ifInXcastPkts. This also means that, since MAC Control frames are consumed by a sublayer internal to the interface layer and not passed to a higher layer, they are not counted by ifInXcastPkts, but they are counted by aFramesReceivedOK:

$$\begin{aligned} \text{aFramesReceivedOK} = & \text{ifInUcastPkts} + \text{ifInMulticastPkts} \\ & + \text{ifInBroadcastPkts} + \text{dot3InPauseFrames} \\ & + \text{ifInDiscards} + \text{ifInUnknownProtos} \end{aligned} \quad (4)$$

This specification chooses to treat MAC control frames as being originated and consumed within the interface and not counted by the IF-MIB packet counters. MAC control frames are normally sent as multicast packets. In many network environments, MAC control frames can greatly outnumber multicast frames carrying actual data. If MAC control frames were included in the ifInMulticastPkts and ifOutMulticastPkts, the count of data-carrying multicast packets would tend to be drowned out by the count of MAC control frames, rendering those counters considerably less useful.

To better understand the issues surrounding the mapping of the IF-MIB packet and octet counters to an Ethernet interface, it is useful to refer to a Case diagram (Case and Partridge [B2]) for the IF-MIB counters, with modifications to show the proper interpretation for the Ethernet interface. This is depicted in Figure 10-1.

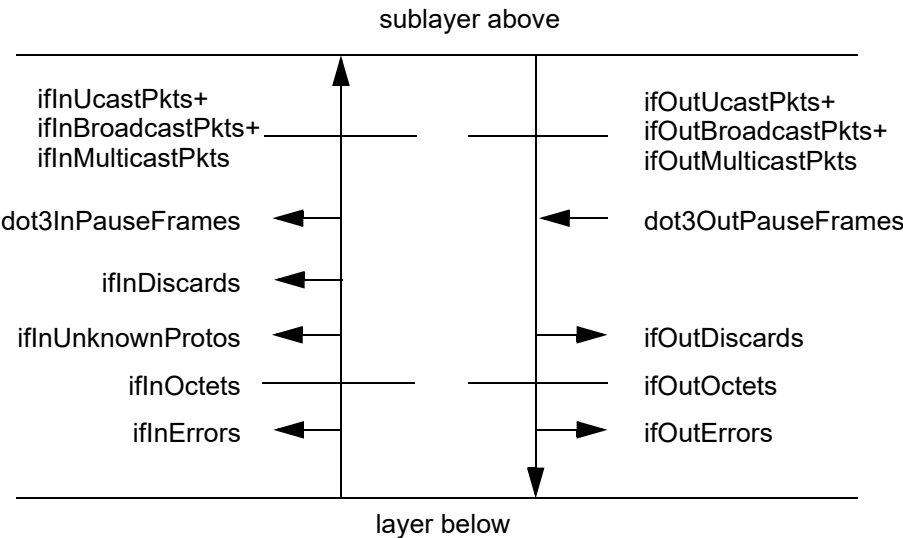


Figure 10-1—Case diagram for the IF-MIB counters

10.2.2.5 ifMtu

The defined standard MTU for Ethernet-like interfaces is 1500 octets. However, many implementations today support larger packet sizes than IEEE Std 802.3. The value of this object shall reflect the actual MTU in use on the interface, whether it matches the standard MTU or not.

This value should reflect the value seen by the MAC client interface. When a higher layer protocol, like IP, is running over Ethernet framing, this is the MTU that will be seen by that higher layer protocol. However, most Ethernet-like interfaces today run multiple protocols that use a mix of different framing types. For example, an IEEE 802.2 LLC type 1 client protocol will see an MTU of 1497 octets on an interface using the IEEE standard maximum packet size, and a protocol running over SNAP will see an MTU of 1492 octets on an interface using the IEEE standard maximum packet size. However, since the specification mandates using the MTU as seen at the MAC client interface, the value of ifMtu would be reported as 1500 octets in these cases.

10.2.2.6 ifSpeed and ifHighSpeed

For Ethernet-like interfaces operating at 1000 Megabits per second (Mb/s) or less, ifSpeed will represent the current operational speed of the interface in bits per second. For such interface types, this will be equal to 1 000 000 (1 million), 10 000 000 (10 million), 100 000 000 (100 million), or 1 000 000 000 (1 billion). ifHighSpeed will represent the current operational speed in millions of bits per second. For such Ethernet-like interfaces, this will be equal to 1, 10, 100, or 1000. If the interface implements Auto-Negotiation, Auto-Negotiation is enabled for this interface, and the interface has not yet negotiated to an operational speed, then these objects should reflect the maximum speed supported by the interface.

For Ethernet-like interfaces operating at greater than 1000 Mb/s, ifHighSpeed will represent the current operational speed of the interface in millions of bits per second. Note that for WAN implementations, this will be the payload data rate over the WAN interface sublayer. For current implementations, this will be equal to 10 000 for LAN implementations of 10 Gb/s, and 9294 for WAN implementations of the 10 Gb/s MAC over an OC-192 PHY. For these speeds, ifSpeed should report a maximum unsigned 32-bit value of 4 294 967 295 as specified in IETF RFC 2863.

1 These objects shall indicate the correct line speed regardless of the current duplex mode. They shall not
2 indicate a doubled value when operating in full-duplex mode. The duplex mode of the interface may be
3 determined by examining either the dot3StatsDuplexStatus object in this MIB module or the ifMauType
4 MAU-MIB module object defined in Clause 13.
5

6 7 **10.2.2.7 ifPhysAddress**

8
9 This object contains the IEEE 802.3 address that is placed in the source-address field of any Ethernet,
10 Starlan, or IEEE 802.3 frames that originate at this interface. Usually this will be kept in ROM on the
11 interface hardware. Some systems may set this address via software.
12

13
14 In a system where there are several such addresses, the designer has a tougher choice. The address chosen
15 should be the one most likely to be of use to network management (e.g., the address placed in ARP
16 responses for systems that are primarily IP systems).
17

18
19 If the designer truly cannot choose, use of the factory-provided ROM address is suggested.
20

21
22 If the address cannot be determined, an octet string of zero length should be returned.
23

24 The address is stored in binary in this object. The address is stored in “canonical” bit order; that is, the Group
25 Bit is positioned as the low-order bit of the first octet. Thus, the first byte of a multicast address would have
26 the bit 0x01 set.
27

28 29 **10.2.2.8 Specific Interfaces Group MIB objects**

30
31 Table 10-1 provides specific implementation guidelines for applying the Interfaces Group objects to
32 Ethernet-like interfaces.
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Table 10-1—Implementation guidelines

Object	Guidelines
ifIndex	Each Ethernet-like interface is represented by an ifEntry. The dot3StatsTable in this MIB module is indexed by dot3StatsIndex. The interface identified by a particular value of dot3StatsIndex is the same interface as identified by the same value of ifIndex.
ifDescr	Refer to IETF RFC 2863.
ifType	Refer to 10.2.2.2.
ifMtu	Refer to 10.2.2.5.
ifSpeed	Refer to 10.2.2.6.
ifPhysAddress	Refer to 10.2.2.7.
ifAdminStatus	Write access is not required. Support for “testing” is not required.
ifOperStatus	The operational state of the interface. Support for “testing” is not required. The value “dormant” has no meaning for an Ethernet-like interface.
ifLastChange	Refer to IETF RFC 2863.
ifInOctets	The number of octets in valid MAC frames received on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames received on this interface. See 10.2.2.3.
ifInUcastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer and are not passed to any higher layer protocol. See 10.2.2.4.
ifInDiscards	Refer to IETF RFC 2863.
ifInErrors	The sum for this interface of dot3StatsAlignmentErrors, dot3StatsFCSErrors, dot3StatsFrameTooLongs, and dot3StatsInternalMacReceiveErrors.
ifInUnknownProtos	Refer to IETF RFC 2863.
ifOutOctets	The number of octets transmitted in valid MAC frames on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames transmitted on this interface. See 10.2.2.3.
ifOutUcastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol. See 10.2.2.4.
ifOutDiscards	Refer to IETF RFC 2863.
ifOutErrors	The sum for this interface of: dot3StatsSQETestErrors, dot3StatsLateCollisions, dot3StatsExcessiveCollisions, dot3StatsInternalMacTransmitErrors and dot3StatsCarrierSenseErrors.

Table 10-1—Implementation guidelines (continued)

Object	Guidelines
ifName	Locally significant textual name for the interface (e.g., lan0).
ifInMulticastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer and are not passed to any higher layer protocol. See 10.2.2.4.
ifInBroadcastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer, and are not passed to any higher layer protocol. See 10.2.2.4.
ifOutMulticastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol. See 10.2.2.4.
ifOutBroadcastPkts	Refer to IETF RFC 2863. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol. See 10.2.2.4.
ifHCInOctets, ifHCOctets	64-bit versions of counters. Required for Ethernet-like interfaces that are capable of operating at 20 Mb/s or faster, even if the interface is currently operating at less than 20 Mb/s.
ifHCInUcastPkts, ifHCInMulticastPkts, ifHCInBroadcastPkts, ifHCOUcastPkts, ifHCOUmulticastPkts, ifHCOUbroadcastPkts	64-bit versions of packet counters. Required for Ethernet-like interfaces that are capable of operating at 640 Mb/s or faster, even if the interface is currently operating at less than 640 Mb/s.
ifLinkUpDownTrapEnable	Refer to IETF RFC 2863. Default is “enabled.”
ifHighSpeed	Refer to 10.2.2.6.
ifPromiscuousMode	Refer to IETF RFC 2863.
ifConnectorPresent	This will normally be “true.” It will be “false” in the case where this interface uses the WAN Interface Sublayer. See Clause 12 for details.
ifAlias	Refer to IETF RFC 2863.
ifCounterDiscontinuityTime	Refer to IETF RFC 2863. Note that a discontinuity in the Interfaces Group MIB counters may also indicate a discontinuity in some or all of the counters in this MIB that are associated with that interface.
ifStackHigherLayer, ifStackLowerLayer, ifStackStatus	Refer to 11.2.1.1.
ifRcvAddressAddress, ifRcvAddressStatus, ifRcvAddressType	Refer to 10.2.2.1.

10.2.3 Relation to the IEEE 802.3 MAU-MIB module

Support for the mauModIfCompl3 compliance statement of the MAU-MIB module defined in Clause 13 is required for Ethernet-like interfaces. This MIB module is needed in order to allow applications to determine

the current MAU type in use by the interface, and to control autonegotiation and duplex mode for the interface. Implementing this MIB module without implementing the MAU-MIB module would leave applications with no standard way to determine the media type in use, and no standard way to control the duplex mode of the interface.

10.2.4 Mapping of IEEE 802.3 managed objects

The mapping of IEEE 802.3 managed objects to SNMP objects is shown in Table 10-2.

Table 10-2—Mapping of IEEE 802.3 managed objects

IEEE 802.3 managed object		Corresponding SNMP object
oMacEntity	.aMACID	dot3StatsIndex or IF-MIB – ifIndex
	.aFramesTransmittedOK	IF-MIB – ifOutUcastPkts + ifOutMulticastPkts + ifOutBroadcastPkts ^a
	.aSingleCollisionFrames	dot3StatsSingleCollisionFrames
	.aMultipleCollisionFrames	dot3StatsMultipleCollisionFrames
	.aFramesReceivedOK	IF-MIB – ifInUcastPkts + ifInMulticastPkts + ifInBroadcastPkts ^a
	.aFrameCheckSequenceErrors	dot3StatsFCSErrors
	.aAlignmentErrors	dot3StatsAlignmentErrors
	.aOctetsTransmittedOK	IF-MIB – ifOutOctets ^a
	.aFramesWithDeferredXmissions	dot3StatsDeferredTransmissions
	.aLateCollisions	dot3StatsLateCollisions
	.aFramesAbortedDueToXSColls	dot3StatsExcessiveCollisions
	.aFramesLostDueToIntMACXmitError	dot3StatsInternalMacTransmitErrors
	.aCarrierSenseErrors	dot3StatsCarrierSenseErrors
	.aOctetsReceivedOK	IF-MIB – ifInOctets ^a
	.aFramesLostDueToIntMACRcvError	dot3StatsInternalMacReceiveErrors
	.aPromiscuousStatus	IF-MIB – ifPromiscuousMode
	.aReadMulticastAddressList	IF-MIB – ifRevAddressTable
	.aMulticastFramesXmittedOK	IF-MIB – ifOutMulticastPkts ^a
	.aBroadcastFramesXmittedOK	IF-MIB – ifOutBroadcastPkts ^a
	.aMulticastFramesReceivedOK	IF-MIB – ifInMulticastPkts ^a
	.aBroadcastFramesReceivedOK	IF-MIB – ifInBroadcastPkts ^a
	.aFrameTooLongErrors	dot3StatsFrameTooLongs
	.aReadWriteMACAddress	IF-MIB – ifPhysAddress
	.aCollisionFrames	dot3CollFrequencies
	.aDuplexStatus	dot3StatsDuplexStatus
	.aRateControlAbility	dot3StatsRateControlAbility
	.aMaxFrameLength	dot3StatsMaxFrameLength

Table 10-2—Mapping of IEEE 802.3 managed objects (*continued*)

IEEE 802.3 managed object		Corresponding SNMP object
	.aSlowProtocolFrameLimit	dot3SlowProtocolFrameLimit
	.aRateControlStatus	dot3StatsRateControlStatus
	.acAddGroupAddress	IF-MIB - ifRcvAddressTable
	.acDeleteGroupAddress	IF-MIB - ifRcvAddressTable
	.acExecuteSelfTest	dot3TestLoopBack
oPHYEntity	.aPHYID	dot3StatsIndex or IF-MIB – ifIndex
	.aSQETestErrors	dot3StatsSQETestErrors
	.aSymbolErrorDuringCarrier	dot3StatsSymbolErrors
oMACControlEntity	.aMACControlID	dot3StatsIndex or IF-MIB – ifIndex
	.aMACControlFunctionsSupported	dot3ControlFunctionsSupported and dot3ControlFunctionsEnabled
	.aUnsupportedOpcodesReceived	dot3ControlInUnknownOpcodes
oPAUSEEntity	.aPAUSEMACCtrlFramesTransmitted	dot3OutPauseFrames
	.aPAUSEMACCtrlFramesReceived	dot3InPauseFrames

^aNote that the octet counters in IF-MIB do not exactly match the definition of the octet counters in IEEE Std 802.3. See 10.2.2.3 for details. Also note that the packet counters in the IF-MIB do not exactly match the definition of the frame counters in IEEE Std 802.3. See 10.2.2.4 for details.

10.3 Security considerations for Ethernet-like interface MIB module

There is one management object defined in this MIB that has a MAX-ACCESS clause of read-write. That object, dot3PauseAdminMode, may be used to change the flow control configuration on a network interface, which may result in dropped packets, or sending flow control packets on links where the link partner will not understand them. Either action could be detrimental to network performance.

Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

Most of the objects in this MIB module contain statistical information about particular network links. In some network environments, this information may be considered sensitive. It is thus important to control GET and/or NOTIFY access to these objects and possibly to encrypt the values of these objects when sending them over the network via SNMP.

10.4 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL¹⁸:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C10mib.txt

¹⁸Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```
1      IEEE8023-EtherLike-MIB DEFINITIONS ::= BEGIN
2
3      IMPORTS
4          MODULE-IDENTITY, OBJECT-TYPE,
5          Integer32, Counter32, Counter64, org, Unsigned32
6          FROM SNMPv2-SMI
7          MODULE-COMPLIANCE, OBJECT-GROUP
8          FROM SNMPv2-CONF
9          TruthValue
10         FROM SNMPv2-TC
11         ifIndex, InterfaceIndex
12         FROM IF-MIB;
13
14
15     ieee8023etherMIB MODULE-IDENTITY
16         LAST-UPDATED "201304110000Z" -- April 11, 2013
17         ORGANIZATION
18             "IEEE 802.3 working group"
19         CONTACT-INFO
20             "WG-URL: http://www.ieee802.org/3/index.html
21             WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
22
23             Contact: Howard Frazier
24             Postal: 3151 Zanker Road
25                   San Jose, CA 95134
26                   USA
27             Tel:    +1.408.922.8164
28             E-mail: hfrazier@broadcom.com"
29
30
31         DESCRIPTION "The MIB module to describe generic objects for
32                     Ethernet-like network interfaces."
33
34
35         REVISION    "201304110000Z" -- April 11, 2013
36         DESCRIPTION
37             "Revision, based on an earlier version in IEEE Std 802.3.1-2011."
38
39
40         REVISION    "201102020000Z" -- February 2, 2011
41         DESCRIPTION
42             "Initial version, based on an earlier version published
43             in RFC 3635."
44
45         ::= { org ieee(111) standards-association-numbers-series-standards(2)
46             lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 10 }
47
48
49     ieee8023etherMIBObjects OBJECT IDENTIFIER ::= { ieee8023etherMIB 1 }
50
51     -- the Ethernet-like Statistics group
52
53
54     dot3StatsTable OBJECT-TYPE
55         SYNTAX      SEQUENCE OF Dot3StatsEntry
56         MAX-ACCESS  not-accessible
57         STATUS      current
58         DESCRIPTION "Statistics for a collection of Ethernet-like
59                     interfaces attached to a particular system.
60                     There will be one row in this table for each
61                     Ethernet-like interface in the system."
62         ::= { ieee8023etherMIBObjects 2 }
63
64
65     dot3StatsEntry OBJECT-TYPE
```

```
1      SYNTAX      Dot3StatsEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION "Statistics for a particular interface to an
5                  Ethernet-like medium."
6      INDEX       { dot3StatsIndex }
7      ::= { dot3StatsTable 1 }
8
9
10     Dot3StatsEntry ::=
11         SEQUENCE {
12
13             dot3StatsIndex          InterfaceIndex,
14             dot3StatsAlignmentErrors Counter32,
15             dot3StatsFCSErrors      Counter32,
16             dot3StatsSingleCollisionFrames Counter32,
17             dot3StatsMultipleCollisionFrames Counter32,
18             dot3StatsSQETestErrors  Counter32,
19             dot3StatsDeferredTransmissions Counter32,
20             dot3StatsLateCollisions Counter32,
21             dot3StatsExcessiveCollisions Counter32,
22             dot3StatsInternalMacTransmitErrors Counter32,
23             dot3StatsCarrierSenseErrors Counter32,
24             dot3StatsFrameTooLongs Counter32,
25             dot3StatsInternalMacReceiveErrors Counter32,
26             dot3StatsSymbolErrors Counter32,
27             dot3StatsDuplexStatus    INTEGER,
28             dot3StatsRateControlAbility TruthValue,
29             dot3StatsRateControlStatus INTEGER,
30             dot3StatsMaxFrameLength INTEGER
31         }
32
33
34
35     dot3StatsIndex OBJECT-TYPE
36         SYNTAX      InterfaceIndex
37         MAX-ACCESS  not-accessible
38         STATUS      current
39         DESCRIPTION "An index value that uniquely identifies an
40                     interface to an Ethernet-like medium. The
41                     interface identified by a particular value of
42                     this index is the same interface as identified
43                     by the same value of ifIndex."
44         REFERENCE   "IETF RFC 2863, ifIndex"
45         ::= { dot3StatsEntry 1 }
46
47
48     dot3StatsAlignmentErrors OBJECT-TYPE
49         SYNTAX      Counter32
50         MAX-ACCESS  read-only
51         STATUS      current
52         DESCRIPTION "A count of frames received on a particular
53                     interface that are not an integral number of
54                     octets in length and do not pass the FCS check.
55
56                     The count represented by an instance of this
57                     object is incremented when the alignmentError
58                     status is returned by the MAC service to the
59                     LLC (or other MAC user). Received frames for
60                     which multiple error conditions pertain are,
61                     according to the conventions of IEEE 802.3
62                     Layer Management, counted exclusively according
63                     to the error status presented to the LLC."
64
65
```

This counter does not increment for group encoding schemes greater than 4 bits per group.

For interfaces operating at 10 Gb/s, this counter can roll over in less than 5 minutes if it is incrementing at its maximum rate. Since that amount of time could be less than a management station's poll cycle time, in order to avoid a loss of information, a management station is advised to poll the dot3HCStatsAlignmentErrors object for 10 Gb/s or faster interfaces.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

REFERENCE "IEEE Std 802.3, 30.3.1.1.7,
aAlignmentErrors"
::= { dot3StatsEntry 2 }

dot3StatsFCSErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of frames received on a particular interface that are an integral number of octets in length but do not pass the FCS check. This count does not include frames received with frame-too-long or frame-too-short error.

The count represented by an instance of this object is incremented when the frameCheckError status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions pertain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

Note: Coding errors detected by the Physical Layer for speeds above 10 Mb/s will cause the frame to fail the FCS check.

For interfaces operating at 10 Gb/s, this counter can roll over in less than 5 minutes if

it is incrementing at its maximum rate. Since that amount of time could be less than a management station's poll cycle time, in order to avoid a loss of information, a management station is advised to poll the dot3HCStatsFCSErrors object for 10 Gb/s or faster interfaces.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the

```
1          value of ifCounterDiscontinuityTime."
2      REFERENCE  "IEEE Std 802.3, 30.3.1.1.6,
3                  aFrameCheckSequenceErrors."
4      ::= { dot3StatsEntry 3 }
5
6  dot3StatsSingleCollisionFrames OBJECT-TYPE
7      SYNTAX      Counter32
8      MAX-ACCESS  read-only
9      STATUS      current
10     DESCRIPTION "A count of frames that are involved in a single
11                 collision, and are subsequently transmitted
12                 successfully.
13
14                 A frame that is counted by an instance of this
15                 object is also counted by the corresponding
16                 instance of either the ifOutUcastPkts,
17                 ifOutMulticastPkts, or ifOutBroadcastPkts,
18                 and is not counted by the corresponding
19                 instance of the dot3StatsMultipleCollisionFrames
20                 object.
21
22                 This counter does not increment when the
23                 interface is operating in full-duplex mode.
24
25                 Discontinuities in the value of this counter can
26                 occur at re-initialization of the management
27                 system, and at other times as indicated by the
28                 value of ifCounterDiscontinuityTime."
29     REFERENCE  "IEEE Std 802.3, 30.3.1.1.3,
30                 aSingleCollisionFrames."
31     ::= { dot3StatsEntry 4 }
32
33  dot3StatsMultipleCollisionFrames OBJECT-TYPE
34     SYNTAX      Counter32
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION "A count of frames that are involved in more
38                 than one collision and are subsequently
39                 transmitted successfully.
40
41                 A frame that is counted by an instance of this
42                 object is also counted by the corresponding
43                 instance of either the ifOutUcastPkts,
44                 ifOutMulticastPkts, or ifOutBroadcastPkts,
45                 and is not counted by the corresponding
46                 instance of the dot3StatsSingleCollisionFrames
47                 object.
48
49                 This counter does not increment when the
50                 interface is operating in full-duplex mode.
51
52                 Discontinuities in the value of this counter can
53                 occur at re-initialization of the management
54                 system, and at other times as indicated by the
55                 value of ifCounterDiscontinuityTime."
56     REFERENCE  "IEEE Std 802.3, 30.3.1.1.4,
57                 aMultipleCollisionFrames."
58     ::= { dot3StatsEntry 5 }
```

dot3StatsSQETestErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of times that the SQE TEST ERROR is received on a particular interface. The SQE TEST ERROR is set in accordance with the rules for verification of the SQE detection mechanism in the PLS Carrier Sense Function as described in IEEE Std 802.3, 7.2.4.6.

This counter does not increment on interfaces operating at speeds greater than 10 Mb/s, or on interfaces operating in full-duplex mode.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

REFERENCE "IEEE Std 802.3, 7.2.4.6, also 30.3.2.1.4, aSQETestErrors."

::= { dot3StatsEntry 6 }

dot3StatsDeferredTransmissions OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy.

The count represented by an instance of this object does not include frames involved in collisions.

This counter does not increment when the interface is operating in full-duplex mode.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

REFERENCE "IEEE Std 802.3, 30.3.1.1.9, aFramesWithDeferredXmissions."

::= { dot3StatsEntry 7 }

dot3StatsLateCollisions OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "The number of times that a collision is detected on a particular interface later than one slotTime into the transmission of a packet.

A (late) collision included in a count represented by an instance of this object is also considered as a (generic) collision for

1 purposes of other collision-related
2 statistics.
3
4 This counter does not increment when the
5 interface is operating in full-duplex mode.
6
7 Discontinuities in the value of this counter can
8 occur at re-initialization of the management
9 system, and at other times as indicated by the
10 value of ifCounterDiscontinuityTime."
11
12 REFERENCE "IEEE Std 802.3, 30.3.1.1.10,
13 aLateCollisions."
14 ::= { dot3StatsEntry 8 }
15
16 dot3StatsExcessiveCollisions OBJECT-TYPE
17 SYNTAX Counter32
18
19 MAX-ACCESS read-only
20 STATUS current
21 DESCRIPTION "A count of frames for which transmission on a
22 particular interface fails due to excessive
23 collisions.
24
25 This counter does not increment when the
26 interface is operating in full-duplex mode.
27
28 Discontinuities in the value of this counter can
29 occur at re-initialization of the management
30 system, and at other times as indicated by the
31 value of ifCounterDiscontinuityTime."
32
33 REFERENCE "IEEE Std 802.3, 30.3.1.1.11,
34 aFramesAbortedDueToXSColls."
35 ::= { dot3StatsEntry 9 }
36
37
38 dot3StatsInternalMacTransmitErrors OBJECT-TYPE
39 SYNTAX Counter32
40
41 MAX-ACCESS read-only
42 STATUS current
43 DESCRIPTION "A count of frames for which transmission on a
44 particular interface fails due to an internal
45 MAC sublayer transmit error. A frame is only
46 counted by an instance of this object if it is
47 not counted by the corresponding instance of
48 either the dot3StatsLateCollisions object, the
49 dot3StatsExcessiveCollisions object, or the
50 dot3StatsCarrierSenseErrors object.
51
52 The precise meaning of the count represented by
53 an instance of this object is implementation-
54 specific. In particular, an instance of this
55 object may represent a count of transmission
56 errors on a particular interface that are not
57 otherwise counted.
58
59 For interfaces operating at 10 Gb/s, this
60 counter can roll over in less than 5 minutes if
61 it is incrementing at its maximum rate. Since
62 that amount of time could be less than a
63 management station's poll cycle time, in order
64
65

```
1          to avoid a loss of information, a management
2          station is advised to poll the
3          dot3HCStatsInternalMacTransmitErrors object for
4          10 Gb/s or faster interfaces.
5
6          Discontinuities in the value of this counter can
7
8          occur at re-initialization of the management
9          system, and at other times as indicated by the
10         value of ifCounterDiscontinuityTime."
11
12     REFERENCE    "IEEE Std 802.3, 30.3.1.1.12,
13                 aFramesLostDueToIntMACXmitError."
14     ::= { dot3StatsEntry 10 }
15
16     dot3StatsCarrierSenseErrors OBJECT-TYPE
17         SYNTAX      Counter32
18         MAX-ACCESS   read-only
19         STATUS       current
20         DESCRIPTION  "The number of times that the carrier sense
21                     condition was lost or never asserted when
22                     attempting to transmit a frame on a particular
23                     interface.
24
25                     The count represented by an instance of this
26                     object is incremented at most once per
27                     transmission attempt, even if the carrier sense
28                     condition fluctuates during a transmission
29                     attempt.
30
31                     This counter does not increment when the
32                     interface is operating in full-duplex mode.
33
34                     Discontinuities in the value of this counter can
35                     occur at re-initialization of the management
36                     system, and at other times as indicated by the
37                     value of ifCounterDiscontinuityTime."
38     REFERENCE    "IEEE Std 802.3, 30.3.1.1.13,
39                 aCarrierSenseErrors."
40     ::= { dot3StatsEntry 11 }
41
42     -- { dot3StatsEntry 12 } is not assigned
43
44     dot3StatsFrameTooLongs OBJECT-TYPE
45         SYNTAX      Counter32
46         MAX-ACCESS   read-only
47         STATUS       current
48         DESCRIPTION  "A count of frames received on a particular
49                     interface that exceed the maximum permitted
50                     frame size.
51
52                     The count represented by an instance of this
53                     object is incremented when the frameTooLong
54                     status is returned by the MAC service to the
55                     LLC (or other MAC user). Received frames for
56                     which multiple error conditions pertain are,
57
58                     according to the conventions of IEEE 802.3
59                     Layer Management, counted exclusively according
60                     to the error status presented to the LLC.
61
62
63
64
65
```

```
1
2       For interfaces operating at 10 Gb/s, this
3       counter can roll over in less than 80 minutes if
4       it is incrementing at its maximum rate. Since
5       that amount of time could be less than a
6       management station's poll cycle time, in order
7       to avoid a loss of information, a management
8       station is advised to poll the
9       dot3HCStatsFrameTooLongs object for 10 Gb/s
10      or faster interfaces.
11
12
13      Discontinuities in the value of this counter can
14      occur at re-initialization of the management
15      system, and at other times as indicated by the
16      value of ifCounterDiscontinuityTime."
17      REFERENCE      "IEEE Std 802.3, 30.3.1.1.25,
18                    aFrameTooLongErrors."
19      ::= { dot3StatsEntry 13 }
20
21
22      -- { dot3StatsEntry 14 } is not assigned
23
24      -- { dot3StatsEntry 15 } is not assigned
25
26      dot3StatsInternalMacReceiveErrors OBJECT-TYPE
27          SYNTAX      Counter32
28          MAX-ACCESS  read-only
29          STATUS      current
30          DESCRIPTION "A count of frames for which reception on a
31                    particular interface fails due to an internal
32                    MAC sublayer receive error. A frame is only
33                    counted by an instance of this object if it is
34                    not counted by the corresponding instance of
35                    either the dot3StatsFrameTooLongs object, the
36                    dot3StatsAlignmentErrors object, or the
37                    dot3StatsFCSErrors object.
38
39                    The precise meaning of the count represented by
40                    an instance of this object is implementation-
41                    specific. In particular, an instance of this
42                    object may represent a count of receive errors
43                    on a particular interface that are not
44                    otherwise counted.
45
46                    For interfaces operating at 10 Gb/s, this
47                    counter can roll over in less than 5 minutes if
48                    it is incrementing at its maximum rate. Since
49                    that amount of time could be less than a
50                    management station's poll cycle time, in order
51                    to avoid a loss of information, a management
52                    station is advised to poll the
53                    dot3HCStatsInternalMacReceiveErrors object for
54                    10 Gb/s or faster interfaces.
55
56                    Discontinuities in the value of this counter can
57                    occur at re-initialization of the management
58                    system, and at other times as indicated by the
59                    value of ifCounterDiscontinuityTime."
60      REFERENCE      "IEEE Std 802.3, 30.3.1.1.15,
```

```
1             aFramesLostDueToIntMACRcvError."  
2 ::= { dot3StatsEntry 16 }  
3  
4 dot3StatsSymbolErrors OBJECT-TYPE  
5     SYNTAX      Counter32  
6     MAX-ACCESS  read-only  
7     STATUS      current  
8     DESCRIPTION "For an interface operating at 100 Mb/s, the  
9                 number of times there was an invalid data symbol  
10                when a valid carrier was present.  
11  
12                For an interface operating in half-duplex mode  
13                at 1000 Mb/s, the number of times the receiving  
14                media is non-idle (a carrier event) for a period  
15                of time equal to or greater than slotTime, and  
16                during which there was at least one occurrence  
17                of an event that causes the PHY to indicate  
18                'Data reception error' or 'carrier extend error'  
19                on the GMII.  
20  
21                For an interface operating in full-duplex mode  
22                at 1000 Mb/s, the number of times the receiving  
23                media is non-idle (a carrier event) for a period  
24                of time equal to or greater than minFrameSize,  
25                and during which there was at least one  
26                occurrence of an event that causes the PHY to  
27                indicate 'Data reception error' on the GMII.  
28  
29                For an interface operating at 10 Gb/s, 40 Gb/s, and  
30                100 Gb/s, it is a count of the number of times the  
31                receiving media is non-idle (the time between the  
32                Start of Packet Delimiter and the End of Packet  
33                Delimiter) for a period of time equal to or greater  
34                than minFrameSize, and during which there was at least  
35                one occurrence of an event that causes the PHY to  
36                indicate 'Receive Error' on the XGMII, the XLGMII,  
37                or the CGMII.  
38  
39                The count represented by an instance of this  
40                object is incremented at most once per carrier  
41                event, even if multiple symbol errors occur  
42                during the carrier event. This count does  
43                not increment if a collision is present.  
44  
45                This counter does not increment when the  
46                interface is operating at 10 Mb/s.  
47  
48                For interfaces operating at 10 Gb/s, this  
49                counter can roll over in less than 5 minutes if  
50                it is incrementing at its maximum rate. Since  
51                that amount of time could be less than a  
52                management station's poll cycle time, in order  
53                to avoid a loss of information, a management  
54                station is advised to poll the  
55                dot3HCStatsSymbolErrors object for 10 Gb/s  
56                or faster interfaces.  
57  
58                Discontinuities in the value of this counter can  
59                occur at re-initialization of the management  
60  
61  
62  
63  
64  
65
```

```
1          system, and at other times as indicated by the
2          value of ifCounterDiscontinuityTime."
3      REFERENCE    "IEEE Std 802.3, 30.3.2.1.5,
4                  aSymbolErrorDuringCarrier."
5      ::= { dot3StatsEntry 17 }
6
7      dot3StatsDuplexStatus OBJECT-TYPE
8          SYNTAX      INTEGER {
9              unknown(1),
10             halfDuplex(2),
11             fullDuplex(3)
12          }
13      MAX-ACCESS    read-only
14      STATUS        current
15      DESCRIPTION   "The current mode of operation of the MAC
16                    entity. 'unknown' indicates that the current
17                    duplex mode could not be determined.
18
19                    Management control of the duplex mode is
20                    accomplished through the MAU MIB. When
21                    an interface does not support autonegotiation,
22                    or when autonegotiation is not enabled, the
23                    duplex mode is controlled using
24                    ifMauDefaultType. When autonegotiation is
25                    supported and enabled, duplex mode is controlled
26                    using ifMauAutoNegAdvertisedBits. In either
27                    case, the currently operating duplex mode is
28                    reflected both in this object and in ifMauType.
29
30                    Note that this object provides redundant
31                    information with ifMauType. Normally, redundant
32                    objects are discouraged. However, in this
33                    instance, it allows a management application to
34                    determine the duplex status of an interface
35                    without having to know every possible value of
36                    ifMauType. This was felt to be sufficiently
37                    valuable to justify the redundancy."
38      REFERENCE    "IEEE Std 802.3, 30.3.1.1.32,
39                  aDuplexStatus."
40      ::= { dot3StatsEntry 18 }
41
42      dot3StatsRateControlAbility OBJECT-TYPE
43          SYNTAX      TruthValue
44          MAX-ACCESS    read-only
45          STATUS        current
46          DESCRIPTION   "'true' for interfaces operating at speeds above
47                        1000 Mb/s that support Rate Control through
48                        lowering the average data rate of the MAC
49                        sublayer, with frame granularity, and 'false'
50                        otherwise."
51      REFERENCE    "IEEE Std 802.3, 30.3.1.1.33,
52                  aRateControlAbility."
53      ::= { dot3StatsEntry 19 }
54
55      dot3StatsRateControlStatus OBJECT-TYPE
56          SYNTAX      INTEGER {
57              rateControlOff(1),
58              rateControlOn(2),
59              unknown(3)
```

```

1          }
2      MAX-ACCESS read-only
3      STATUS current
4      DESCRIPTION "The current Rate Control mode of operation of
5      the MAC sublayer of this interface."
6      REFERENCE "IEEE Std 802.3, 30.3.1.1.34,
7      aRateControlStatus."
8      ::= { dot3StatsEntry 20 }
9
10
11 dot3StatsMaxFrameLength OBJECT-TYPE
12     SYNTAX INTEGER {
13         unknown(1),
14         baseFrame(2),
15         qTaggedFrame(3),
16         envelopeFrame(4)
17     }
18     MAX-ACCESS read-only
19     STATUS current
20     DESCRIPTION "This indicates the MAC frame length at
21     which the dot3StatsFrameTooLongs counter is
22     incremented."
23     REFERENCE "IEEE Std 802.3, 30.3.1.1.37, aMaxFrameLength."
24     ::= { dot3StatsEntry 21 }
25
26
27 -- the Ethernet-like Collision Statistics group
28
29 -- Implementation of this group is optional; it is appropriate
30 -- for all systems which have the necessary metering
31
32
33 dot3CollTable OBJECT-TYPE
34     SYNTAX SEQUENCE OF Dot3CollEntry
35     MAX-ACCESS not-accessible
36     STATUS current
37     DESCRIPTION "A collection of collision histograms for a
38     particular set of interfaces."
39     REFERENCE "IEEE Std 802.3, 30.3.1.1.30,
40     aCollisionFrames."
41     ::= { ieee8023etherMIBObjects 5 }
42
43
44 dot3CollEntry OBJECT-TYPE
45     SYNTAX Dot3CollEntry
46     MAX-ACCESS not-accessible
47     STATUS current
48     DESCRIPTION "A cell in the histogram of per-frame
49     collisions for a particular interface. An
50
51     instance of this object represents the
52     frequency of individual MAC frames for which
53     the transmission (successful or otherwise) on a
54     particular interface is accompanied by a
55     particular number of media collisions."
56     INDEX { ifIndex, dot3CollCount }
57     ::= { dot3CollTable 1 }
58
59
60 Dot3CollEntry ::=
61     SEQUENCE {
62         dot3CollCount Integer32,
63         dot3CollFrequencies Counter32
64     }
65

```

```
1
2      -- { dot3CollEntry 1 } is no longer in use
3
4      dot3CollCount OBJECT-TYPE
5          SYNTAX      Integer32 (1..16)
6          MAX-ACCESS  not-accessible
7          STATUS      current
8          DESCRIPTION "The number of per-frame media collisions for
9                      which a particular collision histogram cell
10                     represents the frequency on a particular
11                     interface."
12      ::= { dot3CollEntry 2 }
13
14
15      dot3CollFrequencies OBJECT-TYPE
16          SYNTAX      Counter32
17          MAX-ACCESS  read-only
18          STATUS      current
19          DESCRIPTION "A count of individual MAC frames for which the
20                     transmission (successful or otherwise) on a
21                     particular interface occurs after the
22                     frame has experienced exactly the number
23                     of collisions in the associated
24                     dot3CollCount object.
25
26
27                     For example, a frame which is transmitted
28                     on interface 77 after experiencing
29                     exactly 4 collisions would be indicated
30                     by incrementing only dot3CollFrequencies.77.4.
31                     No other instance of dot3CollFrequencies would
32                     be incremented in this example.
33
34
35                     This counter does not increment when the
36                     interface is operating in full-duplex mode.
37
38
39                     Discontinuities in the value of this counter can
40
41                     occur at re-initialization of the management
42                     system, and at other times as indicated by the
43                     value of ifCounterDiscontinuityTime."
44      ::= { dot3CollEntry 3 }
45
46
47      dot3ControlTable OBJECT-TYPE
48          SYNTAX      SEQUENCE OF Dot3ControlEntry
49          MAX-ACCESS  not-accessible
50          STATUS      current
51          DESCRIPTION "A table of descriptive and status information
52                     about the MAC Control sublayer on the
53                     Ethernet-like interfaces attached to a
54                     particular system. There will be one row in
55                     this table for each Ethernet-like interface in
56                     the system which implements the MAC Control
57                     sublayer. If some, but not all, of the
58                     Ethernet-like interfaces in the system implement
59                     the MAC Control sublayer, there will be fewer
60                     rows in this table than in the dot3StatsTable."
61      ::= { ieee8023etherMIBObjects 9 }
62
63
64      dot3ControlEntry OBJECT-TYPE
65          SYNTAX      Dot3ControlEntry
```

```

1      MAX-ACCESS    not-accessible
2      STATUS        current
3      DESCRIPTION   "An entry in the table, containing information
4                    about the MAC Control sublayer on a single
5                    Ethernet-like interface."
6      INDEX          { dot3StatsIndex }
7      ::= { dot3ControlTable 1 }
8
9
10     Dot3ControlEntry ::=
11     SEQUENCE {
12         dot3ControlFunctionsSupported    BITS,
13         dot3ControlInUnknownOpcodes      Counter32,
14         dot3HCControlInUnknownOpcodes     Counter64
15     }
16
17     dot3ControlFunctionsSupported OBJECT-TYPE
18     SYNTAX      BITS {
19         pause(0),    -- 802.3 pause flow control
20         mpcp(1),     -- 802.3 multi-point control protocol
21         pfc(2)       -- 802.3 priority-based flow control
22     }
23
24     MAX-ACCESS    read-only
25     STATUS        current
26     DESCRIPTION   "A list of the possible MAC Control functions
27                   implemented for this interface."
28     REFERENCE     "IEEE Std 802.3, 30.3.3.2,
29                   aMACControlFunctionsSupported."
30
31     ::= { dot3ControlEntry 1 }
32
33
34     dot3ControlInUnknownOpcodes OBJECT-TYPE
35     SYNTAX      Counter32
36     MAX-ACCESS    read-only
37     STATUS        current
38     DESCRIPTION   "A count of MAC Control frames received on this
39                   interface that contain an opcode that is not
40                   supported by this device.
41
42                   For interfaces operating at 10 Gb/s, this
43                   counter can roll over in less than 5 minutes if
44                   it is incrementing at its maximum rate. Since
45                   that amount of time could be less than a
46                   management station's poll cycle time, in order
47                   to avoid a loss of information, a management
48                   station is advised to poll the
49                   dot3HCControlInUnknownOpcodes object for 10 Gb/s
50                   or faster interfaces.
51
52                   Discontinuities in the value of this counter can
53                   occur at re-initialization of the management
54                   system, and at other times as indicated by the
55                   value of ifCounterDiscontinuityTime."
56     REFERENCE     "IEEE Std 802.3, 30.3.3.5,
57                   aUnsupportedOpcodesReceived"
58     ::= { dot3ControlEntry 2 }
59
60
61     dot3HCControlInUnknownOpcodes OBJECT-TYPE
62     SYNTAX      Counter64
63     MAX-ACCESS    read-only
64
65

```



```

1      STATUS      current
2      DESCRIPTION  "A count of MAC Control frames received on this
3                    interface that contain an opcode that is not
4                    supported by this device.
5
6                    This counter is a 64-bit version of
7                    dot3ControlInUnknownOpcodes. It should be used
8                    on interfaces operating at 10 Gb/s or faster.
9
10                   Discontinuities in the value of this counter can
11                   occur at re-initialization of the management
12                   system, and at other times as indicated by the
13                   value of ifCounterDiscontinuityTime."
14
15      REFERENCE    "IEEE Std 802.3, 30.3.3.5,
16                    aUnsupportedOpcodesReceived"
17
18      ::= { dot3ControlEntry 3 }
19
20
21  dot3PauseTable OBJECT-TYPE
22      SYNTAX      SEQUENCE OF Dot3PauseEntry
23      MAX-ACCESS  not-accessible
24      STATUS      current
25      DESCRIPTION "A table of descriptive and status information
26                  about the MAC Control PAUSE function on the
27                  Ethernet-like interfaces attached to a
28                  particular system. There will be one row in
29                  this table for each Ethernet-like interface in
30                  the system which supports the MAC Control PAUSE
31                  function (i.e., the 'pause' bit in the
32                  corresponding instance of
33                  dot3ControlFunctionsSupported is set). If some,
34                  but not all, of the Ethernet-like interfaces in
35                  the system implement the MAC Control PAUSE
36                  function (for example, if some interfaces only
37                  support half-duplex), there will be fewer rows
38                  in this table than in the dot3StatsTable."
39
40      ::= { ieee8023etherMIBObjects 10 }
41
42
43  dot3PauseEntry OBJECT-TYPE
44      SYNTAX      Dot3PauseEntry
45      MAX-ACCESS  not-accessible
46      STATUS      current
47      DESCRIPTION "An entry in the table, containing information
48                  about the MAC Control PAUSE function on a single
49                  Ethernet-like interface."
50
51      INDEX      { dot3StatsIndex }
52      ::= { dot3PauseTable 1 }
53
54  Dot3PauseEntry ::=
55
56      SEQUENCE {
57          dot3PauseAdminMode      INTEGER,
58          dot3PauseOperMode      INTEGER,
59          dot3InPauseFrames      Counter32,
60          dot3OutPauseFrames     Counter32,
61          dot3HCInPauseFrames    Counter64,
62          dot3HCOutPauseFrames   Counter64
63      }
64
65

```

```
1      dot3PauseAdminMode OBJECT-TYPE
2          SYNTAX      INTEGER {
3              disabled(1),
4              enabledXmit(2),
5              enabledRcv(3),
6              enabledXmitAndRcv(4)
7          }
8
9
10         MAX-ACCESS  read-write
11         STATUS      current
12         DESCRIPTION "This object is used to configure the default
13                     administrative PAUSE mode for this interface.
14
15                     This object represents the
16                     administratively-configured PAUSE mode for this
17                     interface. If Auto-Negotiation is not enabled
18                     or is not implemented for the active MAU
19                     attached to this interface, the value of this
20                     object determines the operational PAUSE mode
21                     of the interface whenever it is operating in
22                     full-duplex mode. In this case, a set to this
23                     object will force the interface into the
24                     specified mode.
25
26                     If Auto-Negotiation is implemented and enabled
27                     for the MAU attached to this interface, the
28                     PAUSE mode for this interface is determined by
29                     Auto-Negotiation, and the value of this object
30                     denotes the mode to which the interface will
31                     automatically revert if/when Auto-Negotiation is
32                     later disabled. Note that when Auto-Negotiation
33                     is running, administrative control of the PAUSE
34                     mode may be accomplished using the
35                     ifMauAutoNegCapAdvertisedBits object in the
36                     MAU-MIB module.
37
38                     Note that the value of this object is ignored
39                     when the interface is not operating in
40                     full-duplex mode.
41
42                     An attempt to set this object to
43                     'enabledXmit(2)' or 'enabledRcv(3)' will fail
44                     on interfaces that do not support operation
45                     at greater than 100 Mb/s."
46 ::= { dot3PauseEntry 1 }
47
48
49
50
51      dot3PauseOperMode OBJECT-TYPE
52          SYNTAX      INTEGER {
53              disabled(1),
54              enabledXmit(2),
55              enabledRcv(3),
56              enabledXmitAndRcv(4)
57          }
58
59         MAX-ACCESS  read-only
60         STATUS      current
61         DESCRIPTION "This object reflects the PAUSE mode currently
62                     in use on this interface, as determined by
63                     either (1) the result of the Auto-Negotiation
```

```
1         function or (2) if Auto-Negotiation is not
2         enabled or is not implemented for the active MAU
3         attached to this interface, by the value of
4         dot3PauseAdminMode. Interfaces operating at
5         100 Mb/s or less will never return
6         'enabledXmit(2)' or 'enabledRcv(3)'. Interfaces
7         operating in half-duplex mode will return
8         'disabled(1)'. Interfaces on which
9         Auto-Negotiation is enabled but not yet
10        completed should return the value
11        'disabled(1)'."
12    ::= { dot3PauseEntry 2 }
13
14
15    dot3InPauseFrames OBJECT-TYPE
16        SYNTAX      Counter32
17        MAX-ACCESS   read-only
18        STATUS       current
19        DESCRIPTION  "A count of MAC Control frames received on this
20        interface with an opcode indicating the PAUSE
21        operation.
22
23        This counter does not increment when the
24        interface is operating in half-duplex mode.
25
26        For interfaces operating at 10 Gb/s, this
27        counter can roll over in less than 5 minutes if
28        it is incrementing at its maximum rate. Since
29        that amount of time could be less than a
30        management station's poll cycle time, in order
31        to avoid a loss of information, a management
32        station is advised to poll the
33        dot3HCInPauseFrames object for 10 Gb/s or
34        faster interfaces.
35
36        Discontinuities in the value of this counter can
37        occur at re-initialization of the management
38        system, and at other times as indicated by the
39        value of ifCounterDiscontinuityTime."
40    REFERENCE      "IEEE Std 802.3, 30.3.4.3,
41    aPAUSEMACCtrlFramesReceived."
42    ::= { dot3PauseEntry 3 }
43
44
45    dot3OutPauseFrames OBJECT-TYPE
46        SYNTAX      Counter32
47        MAX-ACCESS   read-only
48        STATUS       current
49
50    DESCRIPTION  "A count of MAC Control frames transmitted on
51    this interface with an opcode indicating the
52    PAUSE operation.
53
54    This counter does not increment when the
55    interface is operating in half-duplex mode.
56
57    For interfaces operating at 10 Gb/s, this
58    counter can roll over in less than 5 minutes if
59    it is incrementing at its maximum rate. Since
60    that amount of time could be less than a
61    management station's poll cycle time, in order
```

1 to avoid a loss of information, a management
2 station is advised to poll the
3 dot3HCOutPauseFrames object for 10 Gb/s or
4 faster interfaces.
5
6 Discontinuities in the value of this counter can
7 occur at re-initialization of the management
8 system, and at other times as indicated by the
9 value of ifCounterDiscontinuityTime."
10 REFERENCE "IEEE Std 802.3, 30.3.4.2,
11 aPAUSEMACCtrlFramesTransmitted."
12 ::= { dot3PauseEntry 4 }
13
14
15 dot3HCInPauseFrames OBJECT-TYPE
16 SYNTAX Counter64
17 MAX-ACCESS read-only
18 STATUS current
19 DESCRIPTION "A count of MAC Control frames received on this
20 interface with an opcode indicating the PAUSE
21 operation."
22
23 This counter does not increment when the
24 interface is operating in half-duplex mode.
25
26 This counter is a 64-bit version of
27 dot3InPauseFrames. It should be used on
28 interfaces operating at 10 Gb/s or faster.
29
30 Discontinuities in the value of this counter can
31 occur at re-initialization of the management
32 system, and at other times as indicated by the
33 value of ifCounterDiscontinuityTime."
34 REFERENCE "IEEE Std 802.3, 30.3.4.3,
35 aPAUSEMACCtrlFramesReceived."
36 ::= { dot3PauseEntry 5 }
37
38
39
40
41 dot3HCOutPauseFrames OBJECT-TYPE
42 SYNTAX Counter64
43 MAX-ACCESS read-only
44 STATUS current
45 DESCRIPTION "A count of MAC Control frames transmitted on
46 this interface with an opcode indicating the
47 PAUSE operation."
48
49 This counter does not increment when the
50 interface is operating in half-duplex mode.
51
52 This counter is a 64-bit version of
53 dot3OutPauseFrames. It should be used on
54 interfaces operating at 10 Gb/s or faster.
55
56 Discontinuities in the value of this counter can
57 occur at re-initialization of the management
58 system, and at other times as indicated by the
59 value of ifCounterDiscontinuityTime."
60 REFERENCE "IEEE Std 802.3, 30.3.4.2,
61 aPAUSEMACCtrlFramesTransmitted."
62 ::= { dot3PauseEntry 6 }
63
64
65

```
1
2      dot3HCStatsTable OBJECT-TYPE
3          SYNTAX      SEQUENCE OF Dot3HCStatsEntry
4          MAX-ACCESS   not-accessible
5          STATUS       current
6          DESCRIPTION  "A table containing 64-bit versions of error
7                      counters from the dot3StatsTable. The 32-bit
8                      versions of these counters may roll over quite
9                      quickly on higher speed Ethernet interfaces.
10                     The counters that have 64-bit versions in this
11                     table are the counters that apply to full-duplex
12                     interfaces, since 10 Gb/s and faster
13                     Ethernet-like interfaces do not support
14                     half-duplex, and very few 1000 Mb/s
15                     Ethernet-like interfaces support half-duplex.
16
17                     Entries in this table are recommended for
18                     interfaces capable of operating at 1000 Mb/s or
19                     faster, and are required for interfaces capable
20                     of operating at 10 Gb/s or faster. Lower speed
21                     Ethernet-like interfaces do not need entries in
22                     this table, in which case there may be fewer
23                     entries in this table than in the
24                     dot3StatsTable. However, implementations
25                     containing interfaces with a mix of speeds may
26                     choose to implement entries in this table for
27                     all Ethernet-like interfaces."
28
29 ::= { ieee8023etherMIBObjects 11 }
30
31
32
33      dot3HCStatsEntry OBJECT-TYPE
34          SYNTAX      Dot3HCStatsEntry
35          MAX-ACCESS   not-accessible
36          STATUS       current
37          DESCRIPTION  "An entry containing 64-bit statistics for a
38                      single Ethernet-like interface."
39
40          INDEX        { dot3StatsIndex }
41
42 ::= { dot3HCStatsTable 1 }
43
44
45      Dot3HCStatsEntry ::=
46          SEQUENCE {
47              dot3HCStatsAlignmentErrors      Counter64,
48              dot3HCStatsFCSErrors             Counter64,
49              dot3HCStatsInternalMacTransmitErrors Counter64,
50              dot3HCStatsFrameTooLongs         Counter64,
51              dot3HCStatsInternalMacReceiveErrors Counter64,
52              dot3HCStatsSymbolErrors          Counter64,
53              dot3HCStatsTransmitLPIMicroseconds Counter64,
54              dot3HCStatsReceiveLPIMicroseconds Counter64,
55              dot3HCStatsTransmitLPITransitions Counter64,
56              dot3HCStatsReceiveLPITransitions Counter64
57          }
58
59      dot3HCStatsAlignmentErrors OBJECT-TYPE
60          SYNTAX      Counter64
61          MAX-ACCESS   read-only
62          STATUS       current
63          DESCRIPTION  "A count of frames received on a particular
64                      interface that are not an integral number of
65                      octets in length and do not pass the FCS check."
```

The count represented by an instance of this object is incremented when the alignmentError status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions pertain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

This counter does not increment for group encoding schemes greater than 4 bits per group.

This counter is a 64-bit version of dot3StatsAlignmentErrors. It should be used on interfaces operating at 10 Gb/s or faster.

Discontinuities in the value of this counter can occur at re-initialization of the management

system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

REFERENCE "IEEE Std 802.3, 30.3.1.1.7,
aAlignmentErrors"
::= { dot3HCStatsEntry 1 }

dot3HCStatsFCSErrors OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of frames received on a particular interface that are an integral number of octets in length but do not pass the FCS check. This count does not include frames received with frame-too-long or frame-too-short error.

The count represented by an instance of this object is incremented when the frameCheckError status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions pertain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

Note: Coding errors detected by the Physical Layer for speeds above 10 Mb/s will cause the frame to fail the FCS check.

This counter is a 64-bit version of dot3StatsFCSErrors. It should be used on interfaces operating at 10 Gb/s or faster.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

REFERENCE "IEEE Std 802.3, 30.3.1.1.6,
aFrameCheckSequenceErrors."

```
1      ::= { dot3HCStatsEntry 2 }
2
3  dot3HCStatsInternalMacTransmitErrors OBJECT-TYPE
4      SYNTAX      Counter64
5      MAX-ACCESS  read-only
6      STATUS      current
7      DESCRIPTION "A count of frames for which transmission on a
8                  particular interface fails due to an internal
9                  MAC sublayer transmit error. A frame is only
10
11                  counted by an instance of this object if it is
12                  not counted by the corresponding instance of
13                  either the dot3StatsLateCollisions object, the
14                  dot3StatsExcessiveCollisions object, or the
15                  dot3StatsCarrierSenseErrors object.
16
17                  The precise meaning of the count represented by
18                  an instance of this object is implementation-
19                  specific. In particular, an instance of this
20                  object may represent a count of transmission
21                  errors on a particular interface that are not
22                  otherwise counted.
23
24                  This counter is a 64-bit version of
25                  dot3StatsInternalMacTransmitErrors. It should
26                  be used on interfaces operating at 10 Gb/s or
27                  faster.
28
29                  Discontinuities in the value of this counter can
30                  occur at re-initialization of the management
31                  system, and at other times as indicated by the
32                  value of ifCounterDiscontinuityTime."
33      REFERENCE   "IEEE Std 802.3, 30.3.1.1.12,
34                  aFramesLostDueToIntMACXmitError."
35      ::= { dot3HCStatsEntry 3 }
36
37  dot3HCStatsFrameTooLongs OBJECT-TYPE
38      SYNTAX      Counter64
39      MAX-ACCESS  read-only
40      STATUS      current
41      DESCRIPTION "A count of frames received on a particular
42                  interface that exceed the maximum permitted
43                  frame size.
44
45                  The count represented by an instance of this
46                  object is incremented when the frameTooLong
47                  status is returned by the MAC service to the
48                  LLC (or other MAC user). Received frames for
49                  which multiple error conditions pertain are,
50                  according to the conventions of IEEE 802.3
51                  Layer Management, counted exclusively according
52                  to the error status presented to the LLC.
53
54                  This counter is a 64-bit version of
55                  dot3StatsFrameTooLongs. It should be used on
56                  interfaces operating at 10 Gb/s or faster.
57
58                  Discontinuities in the value of this counter can
```

occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."
REFERENCE "IEEE Std 802.3, 30.3.1.1.25, aFrameTooLongErrors."
 ::= { dot3HCStatsEntry 4 }

dot3HCStatsInternalMacReceiveErrors OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A count of frames for which reception on a particular interface fails due to an internal MAC sublayer receive error. A frame is only counted by an instance of this object if it is not counted by the corresponding instance of either the dot3StatsFrameTooLongs object, the dot3StatsAlignmentErrors object, or the dot3StatsFCSErrors object.

The precise meaning of the count represented by an instance of this object is implementation-specific. In particular, an instance of this object may represent a count of receive errors on a particular interface that are not otherwise counted.

This counter is a 64-bit version of dot3StatsInternalMacReceiveErrors. It should be used on interfaces operating at 10 Gb/s or faster.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

REFERENCE "IEEE Std 802.3, 30.3.1.1.15, aFramesLostDueToIntMACRcvError."
 ::= { dot3HCStatsEntry 5 }

dot3HCStatsSymbolErrors OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION "For an interface operating at 100 Mb/s, the number of times there was an invalid data symbol when a valid carrier was present.

For an interface operating in half-duplex mode at 1000 Mb/s, the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than slotTime, and during which there was at least one occurrence of an event that causes the PHY to indicate 'Data reception error' or 'carrier extend error' on the GMII.

For an interface operating in full-duplex mode

at 1000 Mb/s, the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than minFrameSize, and during which there was at least one occurrence of an event that causes the PHY to indicate 'Data reception error' on the GMII.

For an interface operating at 10 Gb/s, 40 Gb/s and 100 Gb/s, the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than minFrameSize, and during which there was at least one occurrence of an event that causes the PHY to indicate 'Receive Error' on the XGMII, the XLGMII, or the CGMII.

The count represented by an instance of this object is incremented at most once per carrier event, even if multiple symbol errors occur during the carrier event. This count does not increment if a collision is present.

This counter is a 64-bit version of dot3StatsSymbolErrors. It should be used on interfaces operating at 10 Gb/s or faster.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

REFERENCE "IEEE Std 802.3, 30.3.2.1.5,
aSymbolErrorDuringCarrier."
::= { dot3HCStatsEntry 6 }

dot3HCStatsTransmitLPIMicroseconds OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count reflecting the amount of time 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).

This counter has a maximum increment rate of 1 000 000 counts per second."

REFERENCE "IEEE Std 802.3, 30.3.2.1.8 aTransmitLPIMicroseconds."
::= { dot3HCStatsEntry 7 }

dot3HCStatsReceiveLPIMicroseconds OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count reflecting the amount of time that the LPI_INDICATION parameter has the value ASSERT. 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).

This counter has a maximum increment rate of

```

1           1 000 000 counts per second."
2 REFERENCE  "IEEE Std 802.3, 30.3.2.1.9 aReceiveLPIMicroseconds."
3 ::= { dot3HCStatsEntry 8 }
4
5
6 dot3HCStatsTransmitLPITransitions OBJECT-TYPE
7 SYNTAX      Counter64
8 MAX-ACCESS  read-only
9 STATUS      current
10 DESCRIPTION "A count of occurrences of the transition from
11              state LPI_DEASSERTED to state LPI_ASSERTED of
12              the LPI transmit state diagram is the RS.
13              The state transition corresponds to the assertion
14              of the LPI_REQUEST parameter. The request is indicated
15              to the PHY according to the requirements of the RS
16              (see IEEE Std 802.3 22.7, 35.4, 46.4.)
17
18              This counter has a maximum increment rate of 50 000
19              counts per second at 100 Mb/s; 90 000 counts per
20              second at 1000 Mb/s; and 230 000 counts per second
21              at 10 Gb/s."
22 REFERENCE  "IEEE Std 802.3, 30.3.2.1.10 aTransmitLPITransitions."
23 ::= { dot3HCStatsEntry 9 }
24
25
26 dot3HCStatsReceiveLPITransitions OBJECT-TYPE
27 SYNTAX      Counter64
28 MAX-ACCESS  read-only
29 STATUS      current
30 DESCRIPTION "A count of occurrences of the transition from DEASSERT
31              to ASSERT of the LPI_INDICATE parameter. The
32              indication reflects the state of the PHY according to
33              the requirements of the RS
34              (see IEEE Std 802.3 22.7, 35.4, and 46.4).
35
36              This counter has a maximum increment rate of 50 000
37              counts per second at 100 Mb/s; 90 000 counts per second
38              at 1000 Mb/s; and 230 000 counts per second at 10 Gb/s."
39 REFERENCE  "IEEE Std 802.3, 30.3.2.1.11 aReceiveLPITransitions."
40 ::= { dot3HCStatsEntry 10 }
41
42
43 dot3SlowProtocolFrameLimit OBJECT-TYPE
44 SYNTAX      Integer32
45 MAX-ACCESS  read-write
46 STATUS      current
47 DESCRIPTION "The maximum number of Slow Protocol frames
48              of a given subtype that can be transmitted
49              in a one second interval. The default value
50              is 10."
51 REFERENCE  "IEEE Std 802.3, 30.3.1.1.38,
52              aSlowProtocolFrameLimit."
53 DEFVAL     { 10 }
54 ::= { ieee8023etherMIBObjects 12 }
55
56
57 dot3ExtensionTable OBJECT-TYPE
58 SYNTAX      SEQUENCE OF Dot3ExtensionEntry
59 MAX-ACCESS  not-accessible
60 STATUS      current
61 DESCRIPTION "A table of status information
62              about the Extension MAC Control frames transmitted
63
64
65
```

```

1          and received on the Ethernet-like interfaces attached
2          to a particular system. There will be one row in
3          this table for each Ethernet-like interface in
4          the system which supports Extension MAC Control
5          function (i.e., the 'mpcp' bit in the
6          corresponding instance of
7          dot3ControlFunctionsSupported is set). If some,
8          but not all, of the Ethernet-like interfaces in
9          the system implement the Extension MAC Control
10         function, there will be fewer rows
11         in this table than in the dot3StatsTable."
12
13         ::= { ieee8023etherMIBObjects 13 }
14
15     dot3ExtensionEntry OBJECT-TYPE
16         SYNTAX      Dot3ExtensionEntry
17         MAX-ACCESS   not-accessible
18         STATUS       current
19         DESCRIPTION  "An entry in the table, containing information
20                     about the Extension MAC Control function on a single
21                     Ethernet-like interface."
22         INDEX        { dot3StatsIndex }
23         ::= { dot3ExtensionTable 1 }
24
25
26     Dot3ExtensionEntry ::=
27
28         SEQUENCE {
29             dot3HCInExtensionFrames      Counter64,
30             dot3HCOutExtensionFrames      Counter64,
31             dot3ExtensionMacCtrlStatus    Unsigned32
32         }
33
34
35     dot3HCInExtensionFrames OBJECT-TYPE
36         SYNTAX      Counter64
37         MAX-ACCESS   read-only
38         STATUS       current
39         DESCRIPTION  "A count of Extension MAC Control frames received on
40                     this interface.
41
42                     Discontinuities in the value of this counter can
43                     occur at re-initialization of the management
44                     system, and at other times as indicated by the
45                     value of ifCounterDiscontinuityTime."
46         REFERENCE    "IEEE Std 802.3, 30.3.8.2
47                     aEXTENSIONMACCtrlFramesReceived."
48
49         ::= { dot3ExtensionEntry 1 }
50
51
52     dot3HCOutExtensionFrames OBJECT-TYPE
53         SYNTAX      Counter64
54         MAX-ACCESS   read-only
55         STATUS       current
56         DESCRIPTION  "A count of Extension MAC Control frames transmitted on
57                     this interface.
58
59                     Discontinuities in the value of this counter can
60                     occur at re-initialization of the management
61                     system, and at other times as indicated by the
62                     value of ifCounterDiscontinuityTime."
63         REFERENCE    "IEEE Std 802.3, 30.3.8.1
64
65

```

```

1          aEXTENSIONMACCtrlFramesTransmitted."
2      ::= { dot3ExtensionEntry 2 }
3
4  dot3ExtensionMacCtrlStatus OBJECT-TYPE
5      SYNTAX      Unsigned32
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION "The current EXTENSIONMACCtrlStatus as described in
9                  IEEE Std 802.3, 30.3.8.3."
10     REFERENCE   "IEEE Std 802.3, 30.3.8.3, aEXTENSIONMACCtrlStatus."
11     ::= { dot3ExtensionEntry 3 }
12
13
14  dot3PFCTable OBJECT-TYPE
15      SYNTAX      SEQUENCE OF Dot3PFCEntree
16      MAX-ACCESS  not-accessible
17      STATUS      current
18      DESCRIPTION "A table of descriptive and status information
19                  about the MAC Control Priority-based Flow Control
20                  function on the Ethernet-like interfaces attached to
21                  a particular system. There will be one row in
22                  this table for each Ethernet-like interface in
23                  the system which supports the MAC Control PFC
24                  function (i.e., the 'pfc' bit in the
25                  corresponding instance of
26                  dot3ControlFunctionsSupported is set). If some,
27                  but not all, of the Ethernet-like interfaces in
28                  the system implement the MAC Control PFC
29                  function (for example, if some interfaces only
30                  support half-duplex), there will be fewer rows
31                  in this table than in the dot3StatsTable."
32     ::= { ieee8023etherMIBObjects 14 }
33
34
35  dot3PFCEntree OBJECT-TYPE
36      SYNTAX      Dot3PFCEntree
37      MAX-ACCESS  not-accessible
38      STATUS      current
39      DESCRIPTION "An entry in the table, containing information
40                  about the MAC Control PFC function on a single
41                  Ethernet-like interface."
42     INDEX       { dot3StatsIndex }
43     ::= { dot3PFCTable 1 }
44
45
46  Dot3PFCEntree ::=
47
48      SEQUENCE {
49          dot3PFCAdminMode          INTEGER,
50          dot3PFCOperMode          INTEGER,
51          dot3HCInPFCFrames        Counter64,
52          dot3HCOuPFCFrames        Counter64
53      }
54
55
56  dot3PFCAdminMode OBJECT-TYPE
57      SYNTAX      INTEGER {
58          disabled(1),
59          enabled(2)
60      }
61      MAX-ACCESS  read-write
62      STATUS      current
63      DESCRIPTION "This object is used to configure the default
64

```

```
1      administrative PFC mode for this interface.
2
3      This object represents the
4      administratively-configured PFC mode for this
5      interface. The value of this
6      object determines the operational PFC mode
7      of the interface. A set to this
8      object will force the interface into the
9      specified mode.
10
11      Note that the value of this object is ignored
12      when the interface is not operating in
13      full-duplex mode."
14      ::= { dot3PFCEntry 1 }
15
16
17      dot3PFCOperMode OBJECT-TYPE
18          SYNTAX      INTEGER {
19              disabled(1),
20              enabled(2)
21          }
22          MAX-ACCESS   read-only
23          STATUS        current
24          DESCRIPTION   "This object reflects the PFC mode currently
25                        in use on this interface, as determined by
26                        by the value of dot3PFCAdminMode."
27          REFERENCE     "IEEE Std 802.3, 30.3.3.6 aPFCenableStatus"
28          ::= { dot3PFCEntry 2 }
29
30
31      dot3HCInPFCFrames OBJECT-TYPE
32          SYNTAX      Counter64
33          MAX-ACCESS   read-only
34          STATUS        current
35          DESCRIPTION   "A count of MAC Control frames received on this
36                        interface with an opcode indicating the PFC
37                        operation.
38
39                        Discontinuities in the value of this counter can
40                        occur at re-initialization of the management
41                        system, and at other times as indicated by the
42                        value of ifCounterDiscontinuityTime."
43          ::= { dot3PFCEntry 3 }
44
45
46
47
48
49      dot3HCOutPFCFrames OBJECT-TYPE
50          SYNTAX      Counter64
51          MAX-ACCESS   read-only
52          STATUS        current
53          DESCRIPTION   "A count of MAC Control frames transmitted on
54                        this interface with an opcode indicating the
55                        PFC operation.
56
57                        Discontinuities in the value of this counter can
58                        occur at re-initialization of the management
59                        system, and at other times as indicated by the
60                        value of ifCounterDiscontinuityTime."
61          ::= { dot3PFCEntry 4 }
62
63
64
65
```

```
1      -- { ieee8023etherMIBObjects 6 }, the dot3ChipSets tree,
2      -- is defined in [RFC2666]
3
4      -- Conformance statements
5
6      etherConformance OBJECT IDENTIFIER ::= { ieee8023etherMIB 2 }
7
8      etherGroups      OBJECT IDENTIFIER ::= { etherConformance 1 }
9      etherCompliances OBJECT IDENTIFIER ::= { etherConformance 2 }
10
11
12      -- Compliance statements
13
14      dot3Compliance2 MODULE-COMPLIANCE
15          STATUS      current
16          DESCRIPTION "The compliance statement for managed network
17                      entities which have Ethernet-like network
18                      interfaces.
19
20                      Note that compliance with this MIB module
21                      requires compliance with the ifCompliance3
22                      MODULE-COMPLIANCE statement of the IF-MIB
23                      (IETF RFC 2863). In addition, compliance with this
24                      MIB module requires compliance with the
25                      mauModIfCompl3 MODULE-COMPLIANCE statement of
26                      the MAU-MIB module defined in Clause 13."
27
28
29      MODULE -- this module
30          MANDATORY-GROUPS { etherStatsBaseGroup2 }
31
32
33          GROUP      etherDuplexGroup
34          DESCRIPTION "This group is mandatory for all
35                      Ethernet-like network interfaces which are
36                      capable of operating in full-duplex mode.
37                      It is highly recommended for all
38                      Ethernet-like network interfaces."
39
40
41          GROUP      etherRateControlGroup
42          DESCRIPTION "This group is mandatory for all
43                      Ethernet-like network interfaces which are
44                      capable of operating at speeds faster than
45                      1000 Mb/s. It is highly recommended for all
46                      Ethernet-like network interfaces."
47
48
49          GROUP      etherStatsLowSpeedGroup
50          DESCRIPTION "This group is mandatory for all
51                      Ethernet-like network interfaces which are
52                      capable of operating at 10 Mb/s or slower in
53                      half-duplex mode."
54
55
56          GROUP      etherStatsHighSpeedGroup
57          DESCRIPTION "This group is mandatory for all
58                      Ethernet-like network interfaces which are
59                      capable of operating at 100 Mb/s or faster."
60
61
62          GROUP      etherStatsHalfDuplexGroup
63          DESCRIPTION "This group is mandatory for all
64                      Ethernet-like network interfaces which are
65                      capable of operating in half-duplex mode."
```

1 GROUP etherHCStatsGroup
2 DESCRIPTION "This group is mandatory for all
3 Ethernet-like network interfaces which are
4 capable of operating at 10 Gb/s or faster.
5 It is recommended for all Ethernet-like
6 network interfaces which are capable of
7 operating at 1000 Mb/s or faster."
8
9
10 GROUP etherControlGroup
11 DESCRIPTION "This group is mandatory for all
12 Ethernet-like network interfaces that
13 support the MAC Control sublayer."
14
15 GROUP etherHCControlGroup
16 DESCRIPTION "This group is mandatory for all
17 Ethernet-like network interfaces that
18 support the MAC Control sublayer and are
19 capable of operating at 10 Gb/s or faster."
20
21
22 GROUP etherControlPauseGroup
23 DESCRIPTION "This group is mandatory for all
24 Ethernet-like network interfaces that
25 support the MAC Control PAUSE function."
26
27
28 GROUP etherHCControlPauseGroup
29 DESCRIPTION "This group is mandatory for all
30 Ethernet-like network interfaces that
31 support the MAC Control PAUSE function and
32 are capable of operating at 10 Gb/s or
33 faster."
34
35 GROUP etherCollisionTableGroup
36 DESCRIPTION "This group is optional. It is appropriate
37 for all Ethernet-like network interfaces
38 which are capable of operating in
39 half-duplex mode and have the necessary
40 metering. Implementation in systems with
41 such interfaces is highly recommended."
42
43
44 GROUP etherHCStatsLpiGroup
45 DESCRIPTION "This group is mandatory for all
46 Ethernet-like network interfaces that
47 support the Low Power Idle function."
48
49
50 GROUP etherSlowProtocolsGroup
51 DESCRIPTION "This group is optional. It is appropriate for
52 Ethernet-like network interfaces that implement OAM
53 as defined in Clause 57 of IEEE Std 802.3."
54
55 GROUP etherExtensionMacCtrlGroup
56 DESCRIPTION "This group is mandatory for all
57 Ethernet-like network interfaces that implement
58 Extension MAC Control."
59
60
61 GROUP etherPfcGroup
62 DESCRIPTION "This group is mandatory for all
63 Ethernet-like network interfaces that implement
64 Priority Flow Control."
65

```
1      ::= { etherCompliances 1 }
2
3  -- units of conformance
4
5  etherCollisionTableGroup OBJECT-GROUP
6      OBJECTS      { dot3CollFrequencies
7                      }
8
9      STATUS        current
10     DESCRIPTION   "A collection of objects providing a histogram
11                    of packets successfully transmitted after
12                    experiencing exactly N collisions."
13     ::= { etherGroups 1 }
14
15  etherStatsLowSpeedGroup OBJECT-GROUP
16      OBJECTS      { dot3StatsSQETestErrors }
17
18      STATUS        current
19      DESCRIPTION   "A collection of objects providing information
20                    applicable to Ethernet-like network interfaces
21                    capable of operating at 10 Mb/s or slower in
22                    half-duplex mode."
23     ::= { etherGroups 2 }
24
25  etherStatsHighSpeedGroup OBJECT-GROUP
26      OBJECTS      { dot3StatsSymbolErrors }
27
28      STATUS        current
29      DESCRIPTION   "A collection of objects providing information
30                    applicable to Ethernet-like network interfaces
31                    capable of operating at 100 Mb/s or faster."
32     ::= { etherGroups 3 }
33
34  etherDuplexGroup OBJECT-GROUP
35      OBJECTS      { dot3StatsDuplexStatus }
36
37      STATUS        current
38      DESCRIPTION   "A collection of objects providing information
39                    about the duplex mode of an Ethernet-like
40                    network interface."
41     ::= { etherGroups 4 }
42
43  etherControlGroup OBJECT-GROUP
44      OBJECTS      { dot3ControlFunctionsSupported,
45                    dot3ControlInUnknownOpcodes
46                    }
47
48      STATUS        current
49      DESCRIPTION   "A collection of objects providing information
50                    about the MAC Control sublayer on Ethernet-like
51                    network interfaces."
52     ::= { etherGroups 5 }
53
54  etherControlPauseGroup OBJECT-GROUP
55      OBJECTS      { dot3PauseAdminMode,
56                    dot3PauseOperMode,
57                    dot3InPauseFrames,
58                    dot3OutPauseFrames
59                    }
60
61      STATUS        current
62      DESCRIPTION   "A collection of objects providing information
63                    about and control of the MAC Control PAUSE
64                    function on Ethernet-like network interfaces."
65     ::= { etherGroups 6 }
```



```
1
2     etherStatsBaseGroup2 OBJECT-GROUP
3         OBJECTS      { dot3StatsAlignmentErrors,
4                       dot3StatsFCSErrors,
5                       dot3StatsInternalMacTransmitErrors,
6                       dot3StatsFrameTooLongs,
7                       dot3StatsInternalMacReceiveErrors,
8                       dot3StatsMaxFrameLength
9                       }
10
11     STATUS      current
12     DESCRIPTION "A collection of objects providing information
13               applicable to all Ethernet-like network
14               interfaces."
15     ::= { etherGroups 7 }
16
17     etherStatsHalfDuplexGroup OBJECT-GROUP
18         OBJECTS      { dot3StatsSingleCollisionFrames,
19                       dot3StatsMultipleCollisionFrames,
20                       dot3StatsDeferredTransmissions,
21                       dot3StatsLateCollisions,
22                       dot3StatsExcessiveCollisions,
23                       dot3StatsCarrierSenseErrors
24                       }
25
26     STATUS      current
27     DESCRIPTION "A collection of objects providing information
28               applicable only to half-duplex Ethernet-like
29               network interfaces."
30     ::= { etherGroups 8 }
31
32     etherHCStatsGroup OBJECT-GROUP
33         OBJECTS      { dot3HCStatsAlignmentErrors,
34                       dot3HCStatsFCSErrors,
35                       dot3HCStatsInternalMacTransmitErrors,
36                       dot3HCStatsFrameTooLongs,
37                       dot3HCStatsInternalMacReceiveErrors,
38                       dot3HCStatsSymbolErrors
39                       }
40
41     STATUS      current
42     DESCRIPTION "A collection of objects providing high-capacity
43               statistics applicable to higher-speed
44               Ethernet-like network interfaces."
45     ::= { etherGroups 9 }
46
47     etherHCControlGroup OBJECT-GROUP
48         OBJECTS      { dot3HCControlInUnknownOpcodes }
49
50     STATUS      current
51     DESCRIPTION "A collection of objects providing high-capacity
52               statistics for the MAC Control sublayer on
53               higher-speed Ethernet-like network interfaces."
54     ::= { etherGroups 10 }
55
56     etherHCControlPauseGroup OBJECT-GROUP
57         OBJECTS      { dot3HCInPauseFrames,
58                       dot3HCOutPauseFrames
59                       }
60
61     STATUS      current
62     DESCRIPTION "A collection of objects providing high-capacity
63               statistics for the MAC Control PAUSE function on
64               "
65
```

```
1             higher-speed Ethernet-like network interfaces."
2 ::= { etherGroups 11 }
3
4 etherRateControlGroup OBJECT-GROUP
5     OBJECTS      { dot3StatsRateControlAbility,
6                   dot3StatsRateControlStatus
7                   }
8     STATUS       current
9     DESCRIPTION  "A collection of objects providing information
10                  about the Rate Control function on Ethernet-like
11                  interfaces."
12 ::= { etherGroups 12 }
13
14
15 etherHCStatsLpiGroup OBJECT-GROUP
16     OBJECTS      { dot3HCStatsTransmitLPIMicroseconds,
17                   dot3HCStatsReceiveLPIMicroseconds,
18                   dot3HCStatsTransmitLPITransitions,
19                   dot3HCStatsReceiveLPITransitions
20                   }
21     STATUS       current
22     DESCRIPTION  "A collection of objects providing information
23                  about the Low Power Idle function on Ethernet-like
24                  interfaces."
25 ::= { etherGroups 13 }
26
27
28
29
30 etherSlowProtocolsGroup OBJECT-GROUP
31     OBJECTS      { dot3SlowProtocolFrameLimit }
32     STATUS       current
33     DESCRIPTION  "An object providing control and information
34                  about the frame transmission rate limit for
35                  Slow Protocols on Ethernet-like interfaces."
36 ::= { etherGroups 14 }
37
38
39 etherExtensionMacCtrlGroup OBJECT-GROUP
40     OBJECTS      { dot3HCInExtensionFrames,
41                   dot3HCOutExtensionFrames,
42                   dot3ExtensionMacCtrlStatus
43                   }
44     STATUS       current
45     DESCRIPTION  "A collection of objects providing information
46                  about the Extension MAC Control function on
47                  Ethernet-like interfaces."
48 ::= { etherGroups 15 }
49
50
51 etherPfcGroup OBJECT-GROUP
52     OBJECTS      { dot3PFCAdminMode,
53                   dot3PFCOperMode,
54                   dot3HCInPFCFrames,
55                   dot3HCOutPFCFrames
56                   }
57     STATUS       current
58     DESCRIPTION  "A collection of objects providing information
59                  about the Priority Flow Control function on
60                  Ethernet-like interfaces."
61 ::= { etherGroups 16 }
62
63
64
65 END
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

11. Ethernet in the First Mile copper (EFMCu) interfaces MIB module

11.1 Introduction

Ethernet-like interfaces have been defined in IEEE Std 802.3 known as Ethernet in the First Mile (EFM). In particular, 2BASE-TL and 10PASS-TS physical interfaces (PHYs), defined over voice-grade copper pairs, have been specified for the long and short reach, respectively. These interfaces, collectively called EFM Copper (EFMCu), are based on single-pair high-speed digital subscriber line (SHDSL; see ITU-T G.991.2) and very high-speed digital subscriber line (VDSL; see ITU-T G.993.1) technology, supporting optional physical medium entity (PME) aggregation (a.k.a. multi-pair bonding) with variable rates.

The 2BASE-TL PHY is capable of providing at least 2 Mb/s over a 2700 m long single copper pair with a mean bit error ratio (BER) of 10^{-7} (using 5 dB target noise margin).

The 10PASS-TS PHY is capable of providing at least 10 Mb/s over a 750 m long single copper pair with a mean BER of 10^{-7} (using 6 dB target noise margin). This clause defines a MIB module for use with SNMP to manage EFMCu interfaces. In addition, a MIB module is defined describing the cross-connect capability of a stacked interface.

11.2 Relation to other MIB modules

This subclause outlines the relationship of the MIB modules defined in this clause with other MIB modules described in other clauses of this standard, or the relevant RFCs. Specifically, the Interfaces Group MIB (IF-MIB), Ethernet-Like (IEEE8023-EtherLike-MIB), MAU (MAU-MIB), SHDSL (HDSL2-SHDSL-LINE-MIB), and VDSL (VDSL-LINE-EXT-MCM-MIB) modules are discussed.

11.2.1 Relation to Interfaces Group MIB module

2BASE-TL and 10PASS-TS PHYs specified in the EFM-CU-MIB module are stacked (a.k.a. aggregated or bonded) Ethernet interfaces and as such are managed using generic interface management objects defined in the IF-MIB defined in IETF RFC 2863.

The stack management (i.e., actual connection of the sublayers to the top-layer interface) is done via the ifStackTable, as defined in the IF-MIB defined in IETF RFC 2863, and its inverse ifInvStackTable, as defined in the IF-INVERTED-STACK-MIB defined in IETF RFC 2864.

The table ifCapStackTable and its inverse ifInvCapStackTable are defined in the IF-CAP-STACK-MIB module. These tables extend the stack management with an ability to describe possible connections or cross-connect capability, when a flexible cross-connect matrix is present between the interface layers. The IF-CAP-STACK-MIB module definition (Beili [B1]) can be found in:

<https://datatracker.ietf.org/doc/draft-ietf-opsawg-rfc5066bis/>

11.2.1.1 Layering model

An EFMCu interface can aggregate up to 32 physical medium entity (PME) sublayer devices (modems), using the so-called PME aggregation function (PAF).

A generic EFMCu device can have a number of physical coding sublayer (PCS) ports, each connected to a media access controller (MAC) via a media independent interface (MII) at the upper layer, and cross-connected to a number of underlying PMEs, with a single PCS per PME relationship. See 61.1 of IEEE Std 802.3 for more details.

Each PME in the aggregated EFMCu port is represented in the Interface table (ifTable) as a separate interface with ifType of shdsl(169) for 2BASE-TL or vdsl(97) for 10PASS-TS. The ifType values are defined in [IANAifType-MIB].

The ifSpeed for each PME shall return the actual data bitrate of the active PME (e.g., for 2BaseTL PMEs, it is a multiple of 64 kb/s). A zero value shall be returned when the PME is Initializing or Down.

The ifSpeed of the PCS is the sum of the current operating data rates of all PMEs in the aggregation group, without the 64/65-octet encapsulation overhead and PAF overhead, but accounting for the inter-frame gaps (IFGs).

When using the stated definition of ifSpeed for the PCS, there would be no frame loss in the configuration shown in Figure 11-1 (the test-sets are configured to generate 100% of back-to-back traffic, i.e., minimal IFG, at 10 Mb/s or 100 Mb/s, with min and max frame sizes; the EFM interfaces are aggregated to achieve the shown speed).

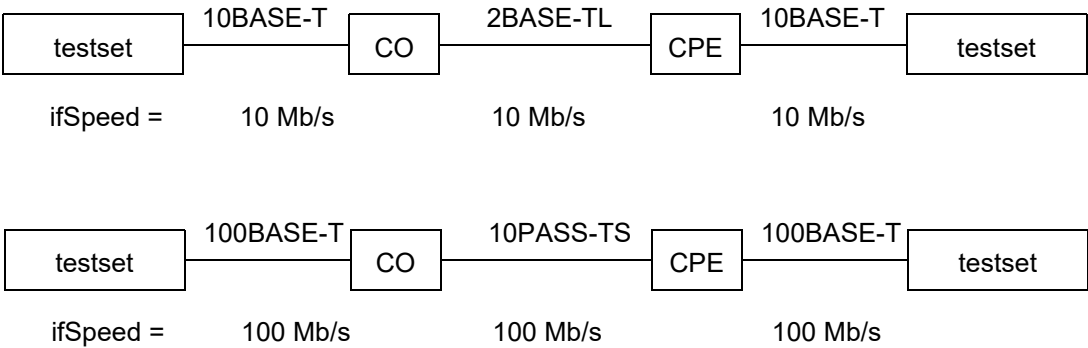
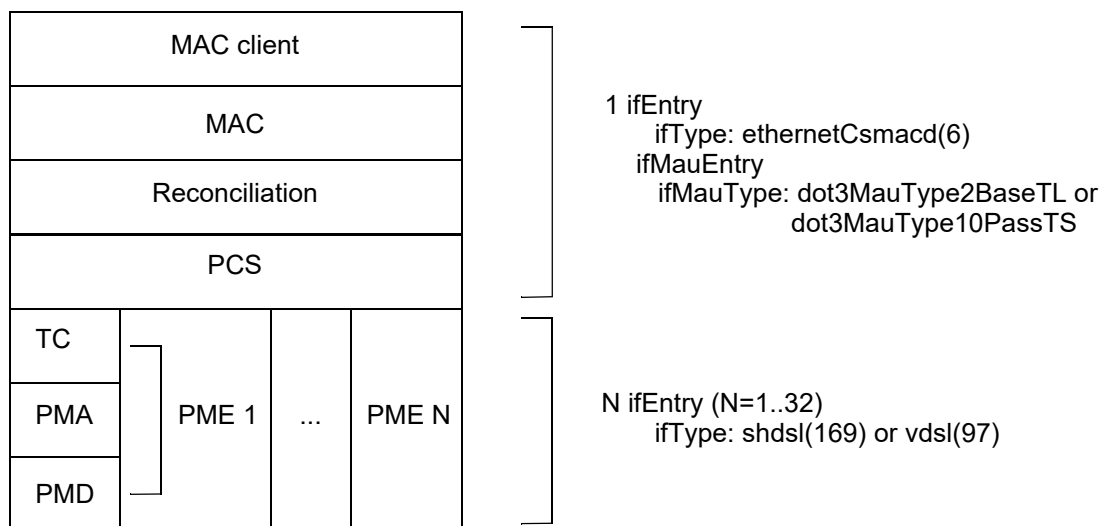


Figure 11-1—Example configuration with no frame loss

Figure 11-2 shows the IEEE 802.3 layering diagram and corresponding use of ifTable and ifMauTable.



MAC = Media Access Control
PCS = Physical Coding Sublayer
TC = Transmission Convergence

PMA = Physical Medium Attachment
PMD = Physical Medium Dependent
PME = Physical Medium Entity

Figure 11-2—Use of ifTable and ifMauTable for EFMcu ports

The ifStackTable is indexed by the ifIndex values of the aggregated EFMcu port (PCS) and the PMEs connected to it. The ifStackTable allows a Network Management application to determine which PMEs are connected to a particular PCS and change connections (if supported by the application). The ifInvStackTable, being an inverted version of the ifStackTable, provides an efficient means for a Network Management application to read a subset of the ifStackTable and thereby determine which PCS runs on top of a particular PME.

The ifCapStackTable, defined in the IF-CAP-STACK-MIB module, specifies for each higher layer interface (e.g., PCS port) a list of lower layer interfaces (e.g., PMEs), which can possibly be cross-connected to that higher layer interface, determined by the cross-connect capability of the device. This table, modeled after the ifStackTable, is read-only, reflecting the current cross-connect capability of the stacked interface, which can be dynamic in some implementations (e.g., if PMEs are located on a pluggable module and the module is pulled out). Note that PME availability per PCS, described by the ifCapStackTable, can be constrained by other parameters, for example, by the aggregation capacity of a PCS or by the PME in question being already connected to another PCS. So that a particular PME can be connected to the PCS, all respective parameters (e.g., ifCapStackTable, ifStackTable, and efmCuPAFCapacity) shall be inspected.

The ifInvCapStackTable, also defined in the IF-CAP-STACK-MIB module, describes which higher layer interfaces (e.g., PCS ports) can possibly be connected to a particular lower layer interface (e.g., PME), providing an inverted mapping of the ifCapStackTable. While it contains no additional information beyond that already contained in the ifCapStackTable, the ifInvCapStackTable has the ifIndex values in its INDEX clause in the reverse order, i.e., the lower layer interface first, and the higher layer interface second, providing an efficient means for a Network Management application to read a subset of the ifCapStackTable and thereby determine which interfaces can be connected to run on top of a particular interface.

11.2.1.2 PME aggregation function (PAF)

The PME aggregation function (PAF) allows a number of PMEs to be aggregated onto a PCS port, by fragmenting the Ethernet frames, transmitting the fragments over multiple PMEs, and assembling the original frames at the remote port. PAF is optional, meaning that a device with a single PME may perform fragmentation and reassembly if this function is supported by the device. Note that the agent is required to report on the PAF capability for all EFM-Cu ports (2BASE-TL and 10PASS-TS).

The EFM-CU-MIB module allows a network management application to query the PAF capability and enable/disable it if supported. Note that enabling PAF effectively turns on fragmentation and reassembly, even on a single-PME port.

11.2.1.3 Discovery operation

The EFM-Cu ports may optionally support discovery operation, whereby PMEs, during initialization, exchange information about their respective aggregation groups (PCS). This information can then be used to detect copper misconnections or for an automatic assignment of the local PMEs into aggregation groups instead of a fixed pre-configuration.

The MIB modules defined in this clause allow a network management application to control the EFM discovery mechanism and query its results. Note that the discovery mechanism can work only if PAF is supported and enabled.

Two tables are used by the EFM discovery mechanism: ifStackTable and ifCapStackTable. The following pseudo-code gives an example of the discovery and automatic PME assignment for a generic PAF-enabled multi-PCS EFM-Cu device, located at central office (CO), using objects defined in these MIB modules and in the IF-MIB. (Note that automatic PME assignment is only shown here for the purposes of the example. Fixed PME pre-assignment, manual assignment, or auto-assignment using an alternative internal algorithm may be chosen by a particular implementation.)

```
// Go over all PCS ports in the CO device
FOREACH pcs[i] IN CO_device
{ // Perform discovery and auto-assignment only on PAF enabled ports
  // with room for more PMEs
  IF ( pcs[i].PAFSupported AND pcs[i].NumPMEs < pcs[i].PAFCapacity )
  { // Assign a unique 6-octet local discovery code to the PCS
    // e.g., MAC address
    dc = pcs[i].DiscoveryCode = MAC[i];
    // Go over all disconnected PMEs, which can
    // potentially be connected to the PCS
    FOREACH pme[j] IN ifCapStackTable[pcs[i]] AND
      NOT IN ifStackTable[pcs[i]] // not connected
    { // Try to grab the remote RT_device, by writing the value
      // of the local 6-octet discovery code to the remote
      // discovery code register (via handshake mechanism).
      // This operation is atomic Set-if-Clear action, i.e., it
      // would succeed only if the remote discovery register was
      // zero. Read the remote discovery code register via Get
      // operation to see if the RT_device, attached via the PME
      // is indeed marked as being the CO_device peer.
      pme[j].RemoteDiscoveryCode = dc; // Set-if-Clear
      r = pme[j].RemoteDiscoveryCode; // Get
      IF ( r == dc AND pcs[i].NumPMEs < pcs[i].PAFCapacity )
      { // Remote RT_device connected via PME[j] is/was a peer
```

```

1      // for PCS[i] and there is room for another PME in the
2      // PCS[i] aggregation group (max. PAF capacity is not
3      // reached yet).
4      // Connect this PME to the PCS (via ifStackTable,
5      // ifInvStackTable being inverse of ifStackTable is
6      // updated automatically, i.e., pcs[i] is auto-added
7      // to ifInvStackTable[pme[j]])
8      ADD pme[j] TO ifStackTable[pcs[i]];
9      pcs[i].NumPMEs = pcs[i].NumPMEs + 1;
10     // Discover all other disconnected PMEs,
11     // attached to the same RT_device and connect them to
12     // the PCS provided there is enough room for more PMEs.
13     FOREACH pme[k] IN ifCapStackTable[pcs[i]] AND
14         NOT IN ifStackTable[pcs[i]]
15     { // Get Remote Discovery Code from the PME to see if
16         // it belongs to a connected RT_device "grabbed" by
17         // the CO_device.
18         r = pme[k].RemoteDiscoveryCode;
19         IF ( r == dc AND pcs[i].NumPMEs < pcs[i].PAFCapacity)
20         { // Physically connect the PME to the PCS
21             // (pcs[i] is auto-added TO ifInvStackTable[pme[k]])
22             ADD pme[k] TO ifStackTable[pcs[i]];
23             pcs[i].NumPMEs = pcs[i].NumPMEs + 1;
24         }
25     }
26     }
27     }
28     }
29     }
30     // At this point we have discovered all local PMEs which
31     // are physically connected to the same remote RT_device
32     // and connected them to PCS[i]. Go to the next PCS.
33     BREAK;
34 }
35 }
36 }
37 }
38 }
39 }
40 }

```

An SNMP Agent for an EFMcu device builds the ifCapStackTable and its inverse ifInvCapStackTable according to the information contained in the Clause 45 PME_Available_register (see 61.1.5.3 and 45.2.3.27 of IEEE Std 802.3).

Adding a PME to the ifStackTable row for a specific PCS involves actual connection of the PME to the PCS, which can be done by modifying the Clause 45 PME_Aggregate_register (see 61.1.5.3 and 45.2.3.28 of IEEE Std 802.3).

Note that the PCS port does not have to be operationally “down” for the connection to succeed. In fact, a dynamic PME addition (and removal) may be implemented with an available PME being initialized first (by setting its ifAdminStatus to “up”) and then added to an operationally “up” PCS port, by modifying a respective ifStackTable (and respective ifInvStackTable) entry.

It is recommended that a removal of the last operationally “up” PME from an operationally “up” PCS would be rejected by the implementation, as this action would completely drop the link.

11.2.1.4 EFMcu ports initialization

EFMCu ports being built on top of xDSL technology require a lengthy initialization or “training” process, before any data can pass. During this initialization, both ends of a link (peers) work cooperatively to achieve the required data rate on a particular copper pair. Sometimes, when the copper line is too long or the noise on

the line is too high, that “training” process may fail to achieve a specific target rate with required characteristics.

The ifAdminStatus object from the IF-MIB controls the desired state of a PCS with all the PMEs connected to it or of an individual PME port. Setting this object to “up” instructs a particular PCS or PME to start the initialization process, which may take tens of seconds for EFMcu ports, especially if PAF is involved. The ifOperStatus object shows the operational state of an interface (extended by the ifMauMediaAvailable object from the MAU-MIB module for PCS and efmCuPmeOperStatus defined in the EFM-CU-MIB module for PME interfaces).

A disconnected PME may be initialized by changing the ifAdminState from “down” to “up.” Changing the ifAdminState to “up” on the PCS initializes all PMEs connected to that particular PCS. Note that in case of PAF, some interfaces may fail to initialize while others succeed. The PCS is considered operationally “up” if at least one PME aggregated by its PAF is operationally “up.” When all PMEs connected to the PCS are “down,” the PCS shall be considered operationally “lowerLayerDown.” The PCS shall be considered operationally “notPresent” if it is not connected to any PME. The PCS/PME interface shall remain operationally “down” during initialization.

The efmCuPmeOperStatus defined in the EFM-CU-MIB module expands PME’s ifOperStatus value of “down” to “downReady,” “downNotReady,” and “init” values, indicating various EFMcu PME-specific states.

11.2.1.5 Usage of ifTable

Both the PME and PCS interfaces of the EFMcu PHY are managed using interface-specific management objects defined in the EFM-CU-MIB module and generic interface objects from the ifTable of IF-MIB, with all management table entries referenced by the interface index ifIndex.

Table 11-1 summarizes EFMcu-specific interpretations for some of the ifTable objects specified in the mandatory ifGeneralInformationGroup.

Table 11-1—EFMCu interpretation of IF-MIB objects

IF-MIB object	EFMCu interpretation
ifIndex	Interface index. Each PME and each PCS in the EFMcu PHY shall have a unique index, as there are some PCS- and PME-specific attributes accessible only on the PCS or the PME level.
ifType	ethernetCsmacd(6) for PCS, shdsl(169) for 2BASE-TL PME, vdsl(97) for 10PASS-TS PME. Operating data rate for the PME. For the PCS, it is the sum of the current operating data rates of all PMEs in the aggregation group, without the 64/65-octet encapsulation overhead and PAF overhead, but accounting for the Inter-Frame Gaps (IFGs).
ifSpeed	Setting this object to “up” instructs a particular PCS (with all PMEs connected to it) or PME to start the initialization process.
ifAdminStatus	Setting this object to “up” instructs a particular PCS (with all PMEs connected to it) or PME to start the initialization process.
ifOperStatus	efmCuPmeOperStatus supplements the “down” value of ifOperStatus for PMEs.

11.2.2 Relation to SHDSL MIB module

G.SHDSL.bis modems, similar to PMEs comprising a 2BASE-TL port, are described in the HDSL2-SHDSL-LINE-MIB module defined in IETF RFC 4319 [B34]. Note that not all attributes of G.SHDSL modems reflected in the HDSL2-SHDSL-LINE-MIB module have adequate management objects (Clause 30 attributes and Clause 45 registers) in IEEE Std 802.3.

Because of these differences and for the purposes of simplicity, unification of attributes common to both 2BASE-TL and 10PASS-TS PMEs, and name consistency (e.g., prefixing the 2BASE-TL PME related objects with “efmCuPme2B” instead of “hdl2shdsl”), it was decided not to reference HDSL2-SHDSL-LINE-MIB objects but to define all the relevant objects in the EFM-CU-MIB module.

However, if some functionality not available in the EFM-CU-MIB module is required and supported by the PME, e.g., performance monitoring, relevant HDSL2-SHDSL-LINE-MIB groups may be included and applied for PMEs of 2BASE-TL subtype.

11.2.3 Relation to VDSL MIB module

VDSL modems, similar to the PME(s) comprising a 10PASS-TS port, are described in the VDSL-LINE-EXT-MCM-MIB module defined in IETF RFC 4070 [B31]. Note that not all attributes of VDSL modems reflected in the VDSL-LINE-EXT-MCM-MIB module have adequate management objects (Clause 30 attributes and Clause 45 registers) in IEEE Std 802.3.

Because of these differences and for the purposes of simplicity, unification of attributes common to both 2BASE-TL and 10PASS-TS PMEs, and name consistency, it was decided not to reference VDSL-LINE-EXT-MCM-MIB objects but to define all the relevant objects in the EFM-CU-MIB module.

However, if some functionality not available in the EFM-CU-MIB module is required and supported by the PME, relevant VDSL-LINE-EXT-MCM-MIB groups may be included and applied for PMEs of 10PASS-TS subtype.

11.2.4 Relation to Ethernet-Like and MAU MIB modules

An agent implementing the objects defined in this clause shall also implement the objects required by the Ethernet-like interface MIB module defined in Clause 10 and the objects required by the MAU MIB module defined in Clause 13.

Two new values of ifMauType (OBJECT-IDENTITIES of dot3MauType) and corresponding bit definitions of ifMauTypeListBits (IANAifMauTypeListBits) have been defined in the IANA-MAU-MIB module for EFM-CU MAUs:

- dot3MauType2BaseTL and b2BaseTL, for 2BASE-TL MAU
- dot3MauType10PassTS and b10PassTS, for 10PASS-TS MAU

Additionally, the IANA-MAU-MIB module defines two new values of ifMauMediaAvailable, specifically for EFM-CU ports: availableReduced and ready (in textual convention IANAifMauMediaAvailable). Due to the PME aggregation, the EFM-CU interpretation of some possible ifMauMediaAvailable values differs from other MAUs as follows:

- unknown: the EFM-CU interface (PCS with connected PMEs) is Initializing
- ready: the interface is Down, at least one PME in the aggregation group (all PMEs connected to the PCS) is ready for handshake
- available: the interface is Up, all PMEs in the aggregation group are up

- notAvailable: the interface is Down, all PME's in the aggregation group are Down, no handshake tones are detected by any PME
- availableReduced: the interface is Up, a link fault is detected at the receive direction by one or more PME's in the aggregation group, but at least one PME is Up
- pmdLinkFault: a link fault is detected at the receive direction by all PME's in the aggregation group

As an Ethernet-like interface, every EFMcu port [an ifEntry representing a consolidation of LLC, MAC, and PCS (sub)layers] shall return an ifType of ethernetCsmacd(6). While most of the MAU characteristics are not applicable to the EFMcu ports (no Auto-Negotiation, false carriers, or jabber), they shall return an appropriate ifMauType (dot3MauType2BaseTL or dot3mauType10PassTS) in order to direct the management software to look in the EFM-CU-MIB module for the desired information. For example, the information on the particular EFMcu flavor that an EFMcu port is running is available from efmCuOperSubType, defined in the EFM-CU-MIB module.

Since EFMcu PME's are not Ethernet-like interfaces, they cannot be instantiated as MAU interface objects.

11.3 MIB structure

11.3.1 EFM copper MIB overview

The main management objects defined in the EFM-CU-MIB module are split into two groups:

- efmCuPort—containing objects for configuration, capabilities, status, and notifications, common to all EFMcu PHYs.
- efmCuPme—containing objects for configuration, capabilities, status, and notifications of EFMcu PME's.

The efmCuPme group in turn contains efmCuPme2B and efmCuPme10P groups, which define PME profiles specific to 2BASE-TL and 10PASS-TS PME's, respectively, as well as PME-specific status information.

11.3.2 PME profiles

Since a managed node can have a large number of EFMcu PHYs, provisioning every parameter on every EFMcu PHY may become burdensome. Moreover, most PME's are provisioned identically with the same set of parameters. To simplify the provisioning process, the EFM-CU-MIB module makes use of configuration profiles, similar to the HDSL2-SHDSL-LINE-MIB and VDSL-LINE-EXT-MCM-MIB modules. A profile is a set of parameters, used for either configuration or representation of a PME. The same profile can be shared by multiple PME ports using the same configuration.

The PME profiles are defined in the efmCuPme2BProfileTable and efmCuPme10PProfileTable for 2BASE-TL and 10PASS-TS PME's, respectively. There are 12 predefined standard profiles for 2BASE-TL and 22 standard profiles for 10PASS-TS, defined in IEEE Std 802.3 and dedicated for rapid provisioning of EFMcu PHYs in most scenarios. In addition, the EFM-CU-MIB defines two additional predefined profiles for “best-effort” provisioning of 2BASE-TL PME's. An ability to define new configuration profiles is also provided to allow for EFMcu deployment tailored to specific copper environments and spectral regulations.

A specific configuration or administrative profile is assigned to a specific PME via the efmCuPmeAdminProfile object. If efmCuPmeAdminProfile is zero, then the efmCuAdminProfile object of the PCS port connected to the PME determines the configuration profile (or a list of possible profiles) for that PME. This mechanism allows specifying a common profile for all PME's connected to the PCS port, with an ability to change individual PME profiles by setting the efmCuPmeAdminProfile object, which overwrites the profile set by efmCuAdminProfile.

A current operating PME profile is pointed to by the efmCuPmeOperProfile object. Note that this profile entry can be created automatically to reflect achieved parameters in adaptive (not fixed) initialization.

11.3.3 Mapping of IEEE 802.3 managed objects

This subclause contains the mapping between relevant managed objects (attributes) defined in Clause 30 of IEEE Std 802.3, and managed objects defined in this clause and in associated MIB modules, i.e., the IF-MIB defined in IETF RFC 2863. Note that the majority of the objects defined in the EFM-CU-MIB module do not have direct counterparts in Clause 30 and instead refer to Clause 45 registers.

Table 11-2—Mapping of IEEE 802.3 managed objects

IEEE 802.3 managed object		Corresponding SNMP object
oMAU - Basic Package (Mandatory)	aMAUType	ifMauType (MAU-MIB)
	aMAUTypeList	ifMauTypeListBits (MAU-MIB)
	aMediaAvailable	ifMediaAvailable (MAU-MIB)
oPAF - Basic Package (Mandatory)	aPAFID	ifIndex (IF-MIB)
	aPhyEnd	efmCuPhySide
	aPHYCurrentStatus	efmCuStatus
	aPAFSupported	efmCuPAFSupported
oPAF - PME Aggregation Package (Optional)	aPAFAdminState	efmCuPAFAdminState
	aLocalPAFCapacity	efmCuPAFCapacity
	aLocalPMEAvailable	ifCapStackTable (IF-CAP-STACK-MIB)
	aLocalPMEAggregate	ifStackTable (IF-MIB)
	aRemotePAFSupported	efmCuRemotePAFSupported
	aRemotePAFCapacity	efmCuRemotePAFCapacity
	aRemotePMEAggregate	
oPME - 10P/2B Package (Mandatory)	aPMEID	ifIndex (IF-MIB)
	aPMEAdminState aPMEStatus	ifAdminState (IF-MIB) efmCuPmeStatus
	aPMESNRMgn	efmCuPmeSnrMgn
	aTCCodingViolations	efmCuPmeTCCodingErrors
	aTCCRCErrors	efmCuPmeTCCreErrors
	aProfileSelect	efmCuAdminProfile, efmCuPmeAdminProfile
	aOperatingProfile	efmCuPmeOperProfile
	aPMEFECCorrectedBlocks	efmCuPme10PFECCorrectedBlocks
	aPMEFECCUncorrectableBlocks	efmCuPme10PFECUncorrectedBlocks

11.4 Security considerations for Ethernet in the First Mile copper interfaces MIB module

There are a number of managed objects defined in the EFM-CU-MIB module that have a MAX-ACCESS clause of read-write or read-create. Most objects are writeable only when the link is Down. Writing to these objects can have potentially disruptive effects on network operation, for example:

- Changing of `efmCuPmeAdminSubType` may lead to a potential locking of the link, as peer PMEs of the same subtype cannot exchange handshake messages.
- Changing of `efmCuPAFAdminState` to enabled may lead to a potential locking of the link, if the peer PHY does not support PAF.
- Changing of `efmCuPAFDiscoveryCode`, before the discovery operation, may lead to a wrongful discovery, for example, when two -O ports are connected to the same multi-PME -R port and both -O ports have the same Discovery register value.
- Changing PCS or PME configuration parameters (e.g., profile of a PCS or PME via `efmCuAdminProfile` or `efmCuPmeAdminProfile`) may lead to anything from link quality and rate degradation to a complete link initialization failure, as the ability of an EFMcu port to support a particular configuration depends on the copper environment.
- Activation of a PME can cause a severe degradation of service for another EFMcu PHY, whose PME(s) may be affected by the crosstalk from the newly activated PME.
- Removal of a PME from an operationally “up” EFMcu port, aggregating several PMEs, may cause the port’s rate degradation.

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

The readable objects in the EFM-CU-MIB module (i.e., those with MAX-ACCESS other than not-accessible) may be considered sensitive in some environments since, collectively, they provide information about the performance of network interfaces and can reveal some aspects of their configuration. In particular, since EFMcu can be carried over Unshielded Twisted Pair (UTP) voice-grade copper in a bundle with other pairs belonging to another operator/customer, it is theoretically possible to eavesdrop to an EFMcu transmission simply by “listening” to a crosstalk from the EFMcu pairs, especially if the parameters of the EFMcu link in question are known.

In such environments, it is important to control also GET and NOTIFY access to these objects and possibly to encrypt their values when sending them over the network via SNMP.

11.5 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL¹⁹:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C11mib.txt

¹⁹Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
1 IEEE8023-EFM-CU-MIB DEFINITIONS ::= BEGIN
2
3 IMPORTS
4   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, Integer32,
5   Unsigned32, Counter32, org
6     FROM SNMPv2-SMI -- [RFC2578]
7   TEXTUAL-CONVENTION, TruthValue, RowStatus, PhysAddress
8     FROM SNMPv2-TC -- [RFC2579]
9   MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
10     FROM SNMPv2-CONF -- [RFC2580]
11   SnmpAdminString
12     FROM SNMP-FRAMEWORK-MIB -- [RFC3411]
13   ifIndex, ifSpeed
14     FROM IF-MIB -- [RFC2863]
15 ;
16
17
18 ieee8023efmCuMIB MODULE-IDENTITY
19   LAST-UPDATED "201304110000Z" -- April 11, 2013
20   ORGANIZATION
21     "IEEE 802.3 working group"
22   CONTACT-INFO
23     "WG-URL: http://www.ieee802.org/3/index.html
24     WG-EMail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
25
26     Contact: Howard Frazier
27     Postal: 3151 Zanker Road
28             San Jose, CA 95134
29             USA
30     Tel: +1.408.922.8164
31     E-mail: hfrazier@broadcom.com"
32
33
34 DESCRIPTION
35   "The objects in this MIB module are used to manage
36   the Ethernet in the First Mile (EFM) Copper (EFMCu) Interfaces
37   2BASE-TL and 10PASS-TS, defined in IEEE Std 802.3.
38
39   Of particular interest are Clause 61, 'Physical Coding
40   Sublayer (PCS) and common specifications, type 10PASS-TS and
41   type 2BASE-TL', Clause 30, 'Management', Clause 45,
42   'Management Data Input/Output (MDIO) Interface', Annex 62A,
43   'PMD profiles for 10PASS-TS' and Annex 63A, 'PMD profiles for
44   2BASE-TL'."
45
46 REVISION "201304110000Z" -- April 11, 2013
47 DESCRIPTION
48   "Revision, based on an earlier version in IEEE Std 802.3.1-2011."
49
50 REVISION "201102020000Z" -- February 2, 2011
51 DESCRIPTION
52   "Initial version, based on an earlier version published
53   as RFC 5066."
54
55 ::= { org ieee(111) standards-association-numbers-series-standards(2)
56       lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1)
57       ieee8023efmcu(11) 2 }
58
59 -- Sections of the module
60
61
62
63
64
65
```

```
1      efmCuObjects      OBJECT IDENTIFIER ::= { ieee8023efmCuMIB 1 }
2
3      efmCuConformance OBJECT IDENTIFIER ::= { ieee8023efmCuMIB 2 }
4
5      -- Groups in the module
6
7      efmCuPort          OBJECT IDENTIFIER ::= { efmCuObjects 1 }
8
9      efmCuPme           OBJECT IDENTIFIER ::= { efmCuObjects 2 }
10
11
12     -- Textual Conventions
13
14     EfmProfileIndex ::= TEXTUAL-CONVENTION
15         DISPLAY-HINT "d"
16         STATUS      current
17         DESCRIPTION
18             "A unique value, greater than zero, for each PME configuration
19             profile in the managed EFMCu port. Values should be assigned
20             contiguously starting from 1. The value for each profile shall
21             remain constant at least from one re-initialization of the
22             entity's network management system to the next re-initialization."
23         SYNTAX      Unsigned32 (1..255)
24
25
26     EfmProfileIndexOrZero ::= TEXTUAL-CONVENTION
27         DISPLAY-HINT "d"
28         STATUS      current
29         DESCRIPTION
30             "This textual convention is an extension of the
31             EfmProfileIndex convention. The latter defines a greater than
32             zero value used to identify a PME profile in the managed EFMCu
33             port. This extension permits the additional value of zero.
34             The value of zero is object-specific and shall therefore be
35             defined as part of the description of any object that uses
36             this syntax.
37             Examples of the usage of zero value might include situations
38             where the current operational profile is unknown."
39         SYNTAX      Unsigned32 (0..255)
40
41
42     EfmProfileIndexList ::= TEXTUAL-CONVENTION
43         DISPLAY-HINT "1d:"
44
45         STATUS      current
46         DESCRIPTION
47             "This textual convention represents a list of up to 6
48             EfmProfileIndex values, any of which can be chosen for
49             configuration of a PME in a managed EFMCu port.
50             The EfmProfileIndex textual convention defines a greater than
51             zero value used to identify a PME profile.
52             The value of this object is a concatenation of zero or
53             more (up to 6) octets, where each octet contains an 8-bit
54             EfmProfileIndex value.
55             A zero-length octet string is object-specific and shall
56             therefore be defined as part of the description of any object
57             that uses this syntax. Examples of the usage of a zero-length
58             value might include situations where an object using this
59             textual convention is irrelevant for a specific EFMCu port
60             type."
61         SYNTAX      OCTET STRING (SIZE(0..6))
62
63
64
65
```

```
1      EfmTruthValueOrUnknown ::= TEXTUAL-CONVENTION
2          STATUS          current
3          DESCRIPTION
4              "This textual convention is an extension of the TruthValue
5              convention. The latter defines a Boolean value with possible
6              values of true(1) and false(2). This extension permits the
7              additional value of unknown(0), which can be returned as the
8              result of a GET operation when an exact true or false value
9              of the object cannot be determined."
10         SYNTAX          INTEGER { unknown(0), true(1), false(2) }
11
12
13 -- Port Notifications Group
14
15 efmCuPortNotifications OBJECT IDENTIFIER ::= { efmCuPort 0 }
16
17 efmCuLowRateCrossing NOTIFICATION-TYPE
18     OBJECTS {
19         ifSpeed,
20         efmCuThreshLowRate
21     }
22     STATUS          current
23     DESCRIPTION
24         "This notification indicates that the EFMcu port's data rate
25         has reached/dropped below or exceeded the low rate threshold,
26         specified by efmCuThreshLowRate.
27
28         This notification may be sent for the -O subtype ports
29         (2BaseTL-O/10PassTS-O) while the port is Up, on the crossing
30         event in both directions: from normal (rate is above the
31         threshold) to low (rate equals the threshold or below it) and
32         from low to normal. This notification is not applicable to
33         the -R subtypes.
34
35         A small debouncing period of 2.5 sec, between the detection
36         of the condition and the notification, should be implemented to
37         prevent simultaneous LinkUp/LinkDown and efmCuLowRateCrossing
38         notifications to be sent.
39
40         The adaptive nature of the EFMcu technology allows the port to
41         adapt itself to the changes in the copper environment, e.g.,
42         an impulse noise, alien crosstalk, or a micro-interruption may
43         temporarily drop one or more PMEs in the aggregation group,
44         causing a rate degradation of the aggregated EFMcu link.
45         The dropped PMEs would then try to re-initialize, possibly at
46         a lower rate than before, adjusting the rate to provide
47         required target SNR margin.
48
49         Generation of this notification is controlled by the
50         efmCuLowRateCrossingEnable object."
51     ::= { efmCuPortNotifications 1 }
52
53 -- PCS Port group
54
55 efmCuPortConfTable OBJECT-TYPE
56     SYNTAX          SEQUENCE OF EfmCuPortConfEntry
57     MAX-ACCESS      not-accessible
58     STATUS          current
59     DESCRIPTION
60         "Table for Configuration of EFMcu 2BASE-TL/10PASS-TS (PCS)
```



```
1      Ports. Entries in this table shall be maintained in a
2      persistent manner."
3      ::= { efmCuPort 1 }
4
5  efmCuPortConfEntry OBJECT-TYPE
6      SYNTAX      EfmCuPortConfEntry
7      MAX-ACCESS  not-accessible
8      STATUS      current
9      DESCRIPTION
10         "An entry in the EFMcu Port Configuration table.
11         Each entry represents an EFMcu port indexed by the ifIndex.
12         Note that an EFMcu PCS port runs on top of a single
13         or multiple PME port(s), which are also indexed by ifIndex."
14     INDEX { ifIndex }
15     ::= { efmCuPortConfTable 1 }
16
17
18  EfmCuPortConfEntry ::=
19     SEQUENCE {
20         efmCuPAFAdminState          INTEGER,
21         efmCuPAFDiscoveryCode       PhysAddress,
22         efmCuAdminProfile            EfmProfileIndexList,
23         efmCuTargetDataRate          Unsigned32,
24         efmCuTargetSnrMgn            Unsigned32,
25         efmCuAdaptiveSpectra         TruthValue,
26         efmCuThreshLowRate           Unsigned32,
27         efmCuLowRateCrossingEnable   TruthValue
28     }
29
30
31  efmCuPAFAdminState OBJECT-TYPE
32      SYNTAX      INTEGER {
33          enabled(1),
34          disabled(2)
35      }
36      MAX-ACCESS  read-write
37      STATUS      current
38      DESCRIPTION
39         "Administrative (desired) state of the PAF of the EFMcu port
40         (PCS).
41         When 'disabled', PME aggregation will not be performed by the
42         PCS. No more than a single PME can be assigned to this PCS in
43         this case.
44         When 'enabled', PAF will be performed by the PCS when the link
45         is Up, even on a single attached PME, if PAF is supported.
46
47         PCS ports incapable of supporting PAF shall return a value of
48         'disabled'. Attempts to 'enable' such ports shall be
49         rejected.
50
51         A PAF 'enabled' port with multiple PMEs assigned cannot be
52         'disabled'. Attempts to 'disable' such port shall be
53         rejected, until at most one PME is left assigned.
54
55         Changing PAFAdminState is a traffic-disruptive operation and
56         as such shall be done when the link is Down. Attempts to
57         change this object shall be rejected if the link is Up or
58         Initializing.
59
60         This object maps to the Clause 30 attribute aPAFAdminState.
61
62
63
64
65
```

1 If a Clause 45 MDIO Interface to the PCS is present, then this
2 object maps to the PAF enable bit in the 10P/2B PCS control
3 register.
4
5 This object shall be maintained in a persistent manner."
6
7 REFERENCE
8 "IEEE Std 802.3, 61.2.2, 45.2.3.26.3"
9 ::= { efmCuPortConfEntry 1 }
10 efmCuPAFDiscoveryCode OBJECT-TYPE
11 SYNTAX PhysAddress (SIZE(0|6))
12 MAX-ACCESS read-write
13 STATUS current
14 DESCRIPTION
15 "PAF Discovery Code of the EFMCu port (PCS).
16 A unique 6-octet code used by the Discovery function,
17 when PAF is supported.
18 PCS ports incapable of supporting PAF shall return a
19 zero-length octet string on an attempt to read this object.
20 An attempt to write to this object shall be rejected for such
21 ports.
22 This object shall be instantiated for the -O subtype PCS before
23 writing operations on the efmCuPAFRemoteDiscoveryCode
24 (Set_if_Clear and Clear_if_Same) are performed by PMEs
25 associated with the PCS.
26 The initial value of this object for -R subtype ports after
27 reset is all zeros. For -R subtype ports, the value of this
28 object cannot be changed directly. This value may be changed
29 as a result of writing operation on the
30 efmCuPAFRemoteDiscoveryCode object of remote PME of -O
31 subtype, connected to one of the local PMEs associated with
32 the PCS.
33
34 Discovery shall be performed when the link is Down.
35 Attempts to change this object shall be rejected (in case of
36 SNMP with the error inconsistentValue), if the link is Up or
37 Initializing.
38
39 The PAF Discovery Code maps to the local Discovery code
40 variable in PAF (note that it does not have a corresponding
41 Clause 45 register)."
42 REFERENCE
43 "IEEE Std 802.3, 61.2.2.8.3, 61.2.2.8.4, 45.2.6.6.1, 45.2.6.8,
44 61A.2"
45 ::= { efmCuPortConfEntry 2 }
46
47 efmCuAdminProfile OBJECT-TYPE
48 SYNTAX EfmProfileIndexList
49 MAX-ACCESS read-write
50 STATUS current
51 DESCRIPTION
52 "Desired configuration profile(s), common for all PMEs in the
53 EFMCu port. This object is a list of pointers to entries in
54 either efmCuPme2BProfileTable or
55 efmCuPme10PProfileTable, depending on the current
56 operating SubType of the EFMCu port as indicated by
57 efmCuPortSide.
58 The value of this object is a list of up to 6 indices of
59 profiles. If this list consists of a single profile index,
60 then all PMEs assigned to this EFMCu port shall be configured
61
62
63
64
65

according to the profile referenced by that index, unless it is overwritten by a corresponding non-zero efmCuPmeAdminProfile instance, which takes precedence over efmCuAdminProfile.

A list consisting of more than one index allows each PME in the port to be configured according to any profile specified in the list.

By default, this object has a value of 0x01, referencing the 1st entry in efmCuPme2BProfileTable or efmCuPme10PProfileTable.

This object is writeable and readable for the -O subtype (2BaseTL-O or 10PassTS-O) EFMCu ports. It is irrelevant for the -R subtype (2BaseTL-R or 10PassTS-R) ports -- a zero-length octet string shall be returned on an attempt to read this object and an attempt to change this object shall be rejected in this case.

Note that the current operational profile value is available via the efmCuPmeOperProfile object.

Any modification of this object shall be performed when the link is Down. Attempts to change this object shall be rejected, if the link is Up or Initializing. Attempts to set this object to a list with a member value that is not the value of the index for an active entry in the corresponding profile table shall be rejected.

This object maps to the Clause 30 attribute aProfileSelect.

This object shall be maintained in a persistent manner."

REFERENCE

"IEEE Std 802.3, 30.11.2.1.6"

DEFVAL { '01'H }

::= { efmCuPortConfEntry 3 }

efmCuTargetDataRate OBJECT-TYPE

SYNTAX Unsigned32(1..100000|999999)

UNITS "Kbps"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Desired EFMCu port 'net' (as seen across MII) Data Rate in kb/s, to be achieved during initialization, under spectral restrictions placed on each PME via efmCuAdminProfile or efmCuPmeAdminProfile, with the desired SNR margin specified by efmCuTargetSnrMgn.

In case of PAF, this object represents a sum of individual PME data rates, modified to compensate for fragmentation and 64/65-octet encapsulation overhead (e.g., target data rate of 10 Mb/s shall allow lossless transmission of a full-duplex 10 Mb/s Ethernet frame stream with minimal inter-frame gap).

The value is limited above by 100 Mb/s as this is the max burst rate across MII for EFMCu ports.

The value between 1 and 100000 indicates that the total data rate (ifSpeed) of the EFMCu port after initialization shall be equal to the target data rate or less, if the target data rate

cannot be achieved under spectral restrictions specified by efmCuAdminProfile/efmCuPmeAdminProfile and with the desired SNR margin. In case the copper environment allows a higher total data rate to be achieved than that specified by the target, the excess capability shall be either converted to additional SNR margin or reclaimed by minimizing transmit power as controlled by efmCuAdaptiveSpectra.

The value of 999999 means that the target data rate is not fixed and shall be set to the maximum attainable rate during initialization (Best Effort), under specified spectral restrictions and with the desired SNR margin.

This object is read-write for the -O subtype EFMcu ports (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.

Changing of the Target Data Rate shall be performed when the link is Down. Attempts to change this object shall be rejected (in case of SNMP with the error inconsistentValue), if the link is Up or Initializing.

Note that the current Data Rate of the EFMcu port is represented by the ifSpeed object of IF-MIB.

This object shall be maintained in a persistent manner."
::= { efmCuPortConfEntry 4 }

efmCuTargetSnrMgn OBJECT-TYPE
SYNTAX Unsigned32(0..21)
UNITS "dB"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Desired EFMcu port SNR margin to be achieved on all PMEs assigned to the port, during initialization. (The SNR margin is the difference between the desired SNR and the actual SNR.)

Note that IEEE Std 802.3 recommends using a default target SNR margin of 5 dB for 2BASE-TL ports and 6 dB for 10PASS-TS ports in order to achieve a mean bit error ratio (BER) of 10^{-7} at the PMA service interface.

This object is read-write for the -O subtype EFMcu ports (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.

Changing of the target SNR margin shall be performed when the link is Down. Attempts to change this object shall be rejected (in case of SNMP with the error inconsistentValue), if the link is Up or Initializing.

Note that the current SNR margin of the PMEs comprising the EFMcu port is represented by efmCuPmeSnrMgn.

This object shall be maintained in a persistent manner."
REFERENCE
"IEEE Std 802.3, 61.1.2"
::= { efmCuPortConfEntry 5 }

efmCuAdaptiveSpectra OBJECT-TYPE

```
1      SYNTAX      TruthValue
2      MAX-ACCESS  read-write
3      STATUS      current
4      DESCRIPTION
5          "Indicates how to utilize excess capacity when the copper
6          environment allows a higher total data rate to be achieved
7          than that specified by the efmCuTargetDataRate.
8
9
10         A value of true(1) indicates that the excess capability shall
11         be reclaimed by minimizing transmit power, e.g., using higher
12         constellations and Power Back-Off, in order to reduce
13         interference to other copper pairs in the binder and the
14         adverse impact to link/system performance.
15
16         A value of false(2) indicates that the excess capability shall
17         be converted to additional SNR margin and spread evenly across
18         all active PMEs assigned to the (PCS) port, to increase link
19         robustness.
20
21         This object is read-write for the -O subtype EFMcu ports
22         (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.
23
24         Changing of this object shall be performed when the link is
25         Down. Attempts to change this object shall be rejected (in
26         case of SNMP with the error inconsistentValue), if the link
27         is Up or Initializing.
28
29         This object shall be maintained in a persistent manner."
30
31 ::= { efmCuPortConfEntry 6 }
32
33
34 efmCuThreshLowRate OBJECT-TYPE
35     SYNTAX      Unsigned32(1..100000)
36     UNITS        "Kbps"
37     MAX-ACCESS  read-write
38     STATUS      current
39     DESCRIPTION
40         "This object configures the EFMcu port low-rate crossing alarm
41         threshold. When the current value of ifSpeed for this port
42         reaches/drops below or exceeds this threshold, an
43         efmCuLowRateCrossing notification may be generated if enabled
44         by efmCuLowRateCrossingEnable.
45
46         This object is read-write for the -O subtype EFMcu ports
47         (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.
48
49         This object shall be maintained in a persistent manner."
50
51 ::= { efmCuPortConfEntry 7 }
52
53
54 efmCuLowRateCrossingEnable OBJECT-TYPE
55     SYNTAX      TruthValue
56     MAX-ACCESS  read-write
57     STATUS      current
58     DESCRIPTION
59         "Indicates whether efmCuLowRateCrossing notifications should
60         be generated for this interface.
61
62         A value of true(1) indicates that efmCuLowRateCrossing
63         notification is enabled. A value of false(2) indicates that
64         the notification is disabled.
```

```
1
2      This object is read-write for the -O subtype EFMcu ports
3      (2BaseTL-O/10PassTS-O) and not available for the -R subtypes.
4
5      This object shall be maintained in a persistent manner."
6      ::= { efmCuPortConfEntry 8 }
7
8
9
10     efmCuPortCapabilityTable OBJECT-TYPE
11     SYNTAX      SEQUENCE OF EfmCuPortCapabilityEntry
12     MAX-ACCESS  not-accessible
13     STATUS      current
14     DESCRIPTION
15         "Table for Capabilities of EFMcu 2BASE-TL/10PASS-TS (PCS)
16         Ports. Entries in this table shall be maintained in a
17         persistent manner"
18     ::= { efmCuPort 2 }
19
20
21     efmCuPortCapabilityEntry OBJECT-TYPE
22     SYNTAX      EfmCuPortCapabilityEntry
23     MAX-ACCESS  not-accessible
24     STATUS      current
25     DESCRIPTION
26         "An entry in the EFMcu Port Capability table.
27         Each entry represents an EFMcu port indexed by the ifIndex.
28         Note that an EFMcu PCS port runs on top of a single
29         or multiple PME port(s), which are also indexed by ifIndex."
30     INDEX { ifIndex }
31     ::= { efmCuPortCapabilityTable 1 }
32
33
34     EfmCuPortCapabilityEntry ::=
35     SEQUENCE {
36         efmCuPAFSupported          TruthValue,
37         efmCuPeerPAFSupported      EfmTruthValueOrUnknown,
38         efmCuPAFCapacity           Unsigned32,
39         efmCuPeerPAFCapacity       Unsigned32
40     }
41
42
43     efmCuPAFSupported OBJECT-TYPE
44     SYNTAX      TruthValue
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         "PME Aggregation Function (PAF) capability of the EFMcu port
49         (PCS).
50         This object has a value of true(1) when the PCS can perform
51         PME aggregation on the available PMEs.
52         Ports incapable of PAF shall return a value of false(2).
53
54         This object maps to the Clause 30 attribute aPAFSupported.
55
56         If a Clause 45 MDIO Interface to the PCS is present,
57         then this object maps to the PAF available bit in the
58         10P/2B capability register."
59     REFERENCE
60         "IEEE Std 802.3, 61.2.2, 30.11.1.1.4, 45.2.3.25.1"
61     ::= { efmCuPortCapabilityEntry 1 }
62
63
64     efmCuPeerPAFSupported OBJECT-TYPE
```

```
1      SYNTAX      EfmTruthValueOrUnknown
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "PME Aggregation Function (PAF) capability of the EFMcu port
6          (PCS) link partner.
7          This object has a value of true(1) when the remote PCS can
8          perform PME aggregation on its available PMEs.
9          Ports whose peers are incapable of PAF shall return a value
10         of false(2).
11         Ports whose peers cannot be reached because of the link
12         state shall return a value of unknown(0).
13
14         This object maps to the Clause 30 attribute
15         aRemotePAFSupported.
16
17         If a Clause 45 MDIO Interface to the PCS is present, then
18         this object maps to the Remote PAF supported bit in the
19         10P/2B capability register."
20     REFERENCE
21         "IEEE Std 802.3, 61.2.2, 30.11.1.1.9, 45.2.3.25.2"
22     ::= { efmCuPortCapabilityEntry 2 }
23
24 efmCuPAFCapacity OBJECT-TYPE
25     SYNTAX      Unsigned32 (1..32)
26     MAX-ACCESS  read-only
27     STATUS      current
28     DESCRIPTION
29         "Number of PMEs that can be aggregated by the local PAF.
30         The number of PMEs currently assigned to a particular
31         EFMcu port (efmCuNumPMEs) is never greater than
32         efmCuPAFCapacity.
33
34         This object maps to the Clause 30 attribute
35         aLocalPAFCapacity."
36     REFERENCE
37         "IEEE Std 802.3, 61.2.2, 30.11.1.1.6"
38     ::= { efmCuPortCapabilityEntry 3 }
39
40 efmCuPeerPAFCapacity OBJECT-TYPE
41     SYNTAX      Unsigned32 (0|1..32)
42     MAX-ACCESS  read-only
43     STATUS      current
44     DESCRIPTION
45         "Number of PMEs that can be aggregated by the PAF of the peer
46         PHY (PCS port).
47         A value of 0 is returned when peer PAF capacity is unknown
48         (peer cannot be reached).
49         This object maps to the Clause 30 attribute
50         aRemotePAFCapacity."
51     REFERENCE
52         "IEEE Std 802.3, 61.2.2, 30.11.1.1.10"
53     ::= { efmCuPortCapabilityEntry 4 }
54
55 efmCuPortStatusTable OBJECT-TYPE
56     SYNTAX      SEQUENCE OF EfmCuPortStatusEntry
57     MAX-ACCESS  not-accessible
58     STATUS      current
59     DESCRIPTION
```

```

1      "This table provides overall status information of EFMcu
2      2BASE-TL/10PASS-TS ports, complementing the generic status
3      information from the ifTable of IF-MIB and ifMauTable of the
4      MAU-MIB module. Additional status information about connected PMEs
5      is available from the efmCuPmeStatusTable.
6
7      This table contains live data from the equipment. As such,
8      it is not persistent."
9      ::= { efmCuPort 3 }
10
11  efmCuPortStatusEntry OBJECT-TYPE
12      SYNTAX      EfmCuPortStatusEntry
13      MAX-ACCESS  not-accessible
14      STATUS      current
15      DESCRIPTION
16          "An entry in the EFMcu Port Status table.
17          Each entry represents an EFMcu port indexed by the ifIndex.
18          Note that an EFMcu PCS port runs on top of a single
19          or multiple PME port(s), which are also indexed by ifIndex."
20      INDEX { ifIndex }
21      ::= { efmCuPortStatusTable 1 }
22
23  EfmCuPortStatusEntry ::=
24      SEQUENCE {
25          efmCuFltStatus          BITS,
26          efmCuPortSide          INTEGER,
27          efmCuNumPMEs           Unsigned32,
28          efmCuPAFInErrors       Counter32,
29          efmCuPAFInSmallFragments Counter32,
30          efmCuPAFInLargeFragments Counter32,
31          efmCuPAFInBadFragments Counter32,
32          efmCuPAFInLostFragments Counter32,
33          efmCuPAFInLostStarts   Counter32,
34          efmCuPAFInLostEnds     Counter32,
35          efmCuPAFInOverflows    Counter32
36      }
37
38  efmCuFltStatus OBJECT-TYPE
39      SYNTAX      BITS {
40          noPeer(0),
41          peerPowerLoss(1),
42          pmeSubTypeMismatch(2),
43          lowRate(3)
44      }
45      MAX-ACCESS  read-only
46      STATUS      current
47      DESCRIPTION
48          "EFMCu (PCS) port Fault Status. This is a bitmap of possible
49          conditions. The various bit positions are:
50              noPeer          - the peer PHY cannot be reached (e.g.,
51                              no PMEs attached, all PMEs are Down,
52                              etc.). More info is available in
53                              efmCuPmeFltStatus.
54              peerPowerLoss   - the peer PHY has indicated impending
55                              unit failure due to loss of local
56                              power ('Dying Gasp').
57              pmeSubTypeMismatch - local PMEs in the aggregation group
58                              are not of the same subtype, e.g.,
59                              some PMEs in the local device are -0
60
61
62
63
64
65

```



```
1           while others are -R subtype.
2           lowRate           - ifSpeed of the port reached or dropped
3                               below efmCuThreshLowRate.
4
5           This object is intended to supplement the ifOperStatus object
6           in IF-MIB and ifMauMediaAvailable in the MAU-MIB module.
7
8           Additional information is available via the efmCuPmeFltStatus
9           object for each PME in the aggregation group (single PME if
10          PAF is disabled)."
```

REFERENCE

```
13          "IF-MIB, ifOperStatus; MAU-MIB, ifMauMediaAvailable;
14            efmCuPmeFltStatus"
15          ::= { efmCuPortStatusEntry 1 }
```

efmCuPortSide OBJECT-TYPE

```
18          SYNTAX      INTEGER {
19              subscriber(1),
20              office(2),
21              unknown(3)
22          }
23          MAX-ACCESS    read-only
24          STATUS        current
25          DESCRIPTION
26              "EFM port mode of operation (subtype).
27              The value of 'subscriber' indicates that the port is
28              designated as '-R' subtype (all PMEs assigned to this port are
29              of subtype '-R').
30              The value of the 'office' indicates that the port is
31              designated as '-O' subtype (all PMEs assigned to this port are
32              of subtype '-O').
33              The value of 'unknown' indicates that the port has no assigned
34              PMEs yet or that the assigned PMEs are not of the same side
35              (subTypePMEMismatch).
36
37              This object partially maps to the Clause 30 attribute
38              aPhyEnd."
```

REFERENCE

```
42          "IEEE Std 802.3, 61.1, 30.11.1.1.2"
43          ::= { efmCuPortStatusEntry 2 }
```

efmCuNumPMEs OBJECT-TYPE

```
46          SYNTAX      Unsigned32 (0..32)
47          MAX-ACCESS    read-only
48          STATUS        current
49          DESCRIPTION
50              "The number of PMEs that is currently aggregated by the local
51              PAF (assigned to the EFMcu port using the ifStackTable).
52              This number is never greater than efmCuPAFCapacity.
53
54              This object shall be automatically incremented or decremented
55              when a PME is added or deleted to/from the EFMcu port using
56              the ifStackTable."
```

REFERENCE

```
59          "IEEE Std 802.3, 61.2.2, 30.11.1.1.6"
60          ::= { efmCuPortStatusEntry 3 }
```

efmCuPAFInErrors OBJECT-TYPE

```
63          SYNTAX      Counter32
```

```
1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "The number of fragments that have been received across the
5          gamma interface with RxErr asserted and discarded.
6          This read-only counter is inactive (not incremented) when the
7          PAF is unsupported or disabled. Upon disabling the PAF, the
8          counter retains its previous value.
9
10
11          If a Clause 45 MDIO Interface to the PCS is present, then
12          this object maps to the 10P/2B PAF RX error register.
13
14          Discontinuities in the value of this counter can occur at
15          re-initialization of the management system, and at other times
16          as indicated by the value of ifCounterDiscontinuityTime,
17          defined in IF-MIB."
18      REFERENCE
19          "IEEE Std 802.3, 45.2.3.29"
20      ::= { efmCuPortStatusEntry 4 }
21
22
23  efmCuPAFInSmallFragments OBJECT-TYPE
24      SYNTAX      Counter32
25      MAX-ACCESS  read-only
26      STATUS      current
27      DESCRIPTION
28          "The number of fragments smaller than minFragmentSize
29          (64 bytes) that have been received across the gamma interface
30          and discarded.
31          This read-only counter is inactive when the PAF is
32          unsupported or disabled. Upon disabling the PAF, the counter
33          retains its previous value.
34
35
36          If a Clause 45 MDIO Interface to the PCS is present, then
37          this object maps to the 10P/2B PAF small fragments register.
38
39          Discontinuities in the value of this counter can occur at
40          re-initialization of the management system, and at other times
41          as indicated by the value of ifCounterDiscontinuityTime,
42          defined in IF-MIB."
43      REFERENCE
44          "IEEE Std 802.3, 45.2.3.30"
45      ::= { efmCuPortStatusEntry 5 }
46
47
48  efmCuPAFInLargeFragments OBJECT-TYPE
49      SYNTAX      Counter32
50      MAX-ACCESS  read-only
51      STATUS      current
52      DESCRIPTION
53          "The number of fragments larger than maxFragmentSize
54          (512 bytes) that have been received across the gamma interface
55          and discarded.
56          This read-only counter is inactive when the PAF is
57          unsupported or disabled. Upon disabling the PAF, the counter
58          retains its previous value.
59
60
61          If a Clause 45 MDIO Interface to the PCS is present, then
62          this object maps to the 10P/2B PAF large fragments register.
63
64          Discontinuities in the value of this counter can occur at
65
```

```
1         re-initialization of the management system, and at other times
2         as indicated by the value of ifCounterDiscontinuityTime,
3         defined in IF-MIB."
4     REFERENCE
5         "IEEE Std 802.3, 45.2.3.31"
6     ::= { efmCuPortStatusEntry 6 }
7
8
9     efmCuPAFInBadFragments OBJECT-TYPE
10        SYNTAX      Counter32
11        MAX-ACCESS   read-only
12        STATUS       current
13        DESCRIPTION
14            "The number of fragments that do not fit into the sequence
15            expected by the frame assembly function and that have been
16            received across the gamma interface and discarded (the
17            frame buffer is flushed to the next valid frame start).
18            This read-only counter is inactive when the PAF is
19            unsupported or disabled. Upon disabling the PAF, the counter
20            retains its previous value.
21
22            If a Clause 45 MDIO Interface to the PCS is present, then
23            this object maps to the 10P/2B PAF bad fragments register.
24
25            Discontinuities in the value of this counter can occur at
26            re-initialization of the management system, and at other times
27            as indicated by the value of ifCounterDiscontinuityTime,
28            defined in IF-MIB."
29        REFERENCE
30            "IEEE Std 802.3, 45.2.3.33"
31        ::= { efmCuPortStatusEntry 7 }
32
33
34     efmCuPAFInLostFragments OBJECT-TYPE
35        SYNTAX      Counter32
36        MAX-ACCESS   read-only
37        STATUS       current
38        DESCRIPTION
39            "The number of gaps in the sequence of fragments that have
40            been received across the gamma interface (the frame buffer is
41            flushed to the next valid frame start, when fragment/fragments
42            expected by the frame assembly function is/are not received).
43            This read-only counter is inactive when the PAF is
44            unsupported or disabled. Upon disabling the PAF, the counter
45            retains its previous value.
46
47            If a Clause 45 MDIO Interface to the PCS is present, then
48            this object maps to the 10P/2B PAF lost fragment register.
49
50            Discontinuities in the value of this counter can occur at
51            re-initialization of the management system, and at other times
52            as indicated by the value of ifCounterDiscontinuityTime,
53            defined in IF-MIB."
54        REFERENCE
55            "IEEE Std 802.3, 45.2.3.34"
56        ::= { efmCuPortStatusEntry 8 }
57
58
59     efmCuPAFInLostStarts OBJECT-TYPE
60        SYNTAX      Counter32
61        MAX-ACCESS   read-only
62        STATUS       current
```

1 DESCRIPTION
2 "The number of missing StartOfPacket indicators expected by
3 the frame assembly function.
4 This read-only counter is inactive when the PAF is
5 unsupported or disabled. Upon disabling the PAF, the counter
6 retains its previous value.
7
8
9 If a Clause 45 MDIO Interface to the PCS is present, then
10 this object maps to the 10P/2B PAF lost start of fragment
11 register.
12
13 Discontinuities in the value of this counter can occur at
14 re-initialization of the management system, and at other times
15 as indicated by the value of ifCounterDiscontinuityTime,
16 defined in IF-MIB."
17
18 REFERENCE
19 "IEEE Std 802.3, 45.2.3.35"
20 ::= { efmCuPortStatusEntry 9 }
21
22 efmCuPAFInLostEnds OBJECT-TYPE
23 SYNTAX Counter32
24 MAX-ACCESS read-only
25 STATUS current
26 DESCRIPTION
27 "The number of missing EndOfPacket indicators expected by the
28 frame assembly function.
29 This read-only counter is inactive when the PAF is
30 unsupported or disabled. Upon disabling the PAF, the counter
31 retains its previous value.
32
33
34 If a Clause 45 MDIO Interface to the PCS is present, then
35 this object maps to the 10P/2B PAF lost ends of fragments
36 register.
37
38 Discontinuities in the value of this counter can occur at
39 re-initialization of the management system, and at other times
40 as indicated by the value of ifCounterDiscontinuityTime,
41 defined in IF-MIB."
42
43 REFERENCE
44 "IEEE Std 802.3, 45.2.3.36"
45 ::= { efmCuPortStatusEntry 10 }
46 efmCuPAFInOverflows OBJECT-TYPE
47 SYNTAX Counter32
48 MAX-ACCESS read-only
49 STATUS current
50 DESCRIPTION
51 "The number of fragments, received across the gamma interface
52 and discarded, which would have caused the frame assembly
53 buffer to overflow.
54 This read-only counter is inactive when the PAF is
55 unsupported or disabled. Upon disabling the PAF, the counter
56 retains its previous value.
57
58
59 If a Clause 45 MDIO Interface to the PCS is present, then
60 this object maps to the 10P/2B PAF overflow register.
61
62
63 Discontinuities in the value of this counter can occur at
64 re-initialization of the management system, and at other times
65 as indicated by the value of ifCounterDiscontinuityTime,

```
1         defined in IF-MIB."
2     REFERENCE
3         "IEEE Std 802.3, 45.2.3.32"
4         ::= { efmCuPortStatusEntry 11 }
5
6     -- PME Notifications Group
7
8     efmCuPmeNotifications OBJECT IDENTIFIER ::= { efmCuPme 0 }
9
10    efmCuPmeLineAtnCROSSING NOTIFICATION-TYPE
11    OBJECTS {
12        efmCuPmeLineAtn,
13        efmCuPmeThreshLineAtn
14    }
15    STATUS          current
16    DESCRIPTION
17        "This notification indicates that the loop attenuation
18        threshold (as per the efmCuPmeThreshLineAtn
19        value) has been reached/exceeded for the 2BASE-TL/10PASS-TS
20        PME. This notification may be sent on the crossing event in
21        both directions: from normal to exceeded and from exceeded
22        to normal.
23
24        A small debouncing period of 2.5 sec, between the detection
25        of the condition and the notification, should be implemented
26        to prevent intermittent notifications from being sent.
27
28        Generation of this notification is controlled by the
29        efmCuPmeLineAtnCROSSINGEnable object."
30    ::= { efmCuPmeNotifications 1 }
31
32    efmCuPmeSnrMgnCROSSING NOTIFICATION-TYPE
33    OBJECTS {
34        efmCuPmeSnrMgn,
35        efmCuPmeThreshSnrMgn
36    }
37    STATUS          current
38    DESCRIPTION
39        "This notification indicates that the SNR margin threshold
40        (as per the efmCuPmeThreshSnrMgn value) has been
41        reached/exceeded for the 2BASE-TL/10PASS-TS PME.
42        This notification may be sent on the crossing event in
43        both directions: from normal to exceeded and from exceeded
44        to normal.
45
46        A small debouncing period of 2.5 sec, between the detection
47        of the condition and the notification, should be implemented
48        to prevent intermittent notifications from being sent.
49
50        Generation of this notification is controlled by the
51        efmCuPmeSnrMgnCROSSINGEnable object."
52    ::= { efmCuPmeNotifications 2 }
53
54    efmCuPmeDeviceFault NOTIFICATION-TYPE
55    OBJECTS {
56        efmCuPmeFltStatus
57    }
58    STATUS          current
59    DESCRIPTION
```

```
1      "This notification indicates that a fault in the PME has been
2      detected by a vendor-specific diagnostic or a self-test.
3
4      Generation of this notification is controlled by the
5      efmCuPmeDeviceFaultEnable object."
6      ::= { efmCuPmeNotifications 3 }
7
8
9      efmCuPmeConfigInitFailure NOTIFICATION-TYPE
10     OBJECTS {
11         efmCuPmeFltStatus,
12         efmCuAdminProfile,
13         efmCuPmeAdminProfile
14     }
15     STATUS      current
16     DESCRIPTION
17         "This notification indicates that PME initialization has
18         failed, due to inability of the PME link to achieve the
19         requested configuration profile.
20
21         Generation of this notification is controlled by the
22         efmCuPmeConfigInitFailEnable object."
23         ::= { efmCuPmeNotifications 4 }
24
25
26     efmCuPmeProtocolInitFailure NOTIFICATION-TYPE
27     OBJECTS {
28         efmCuPmeFltStatus,
29         efmCuPmeOperSubType
30     }
31     STATUS      current
32     DESCRIPTION
33         "This notification indicates that the peer PME was using
34         an incompatible protocol during initialization.
35
36         Generation of this notification is controlled by the
37         efmCuPmeProtocolInitFailEnable object."
38         ::= { efmCuPmeNotifications 5 }
39
40
41 -- The PME group
42
43
44     efmCuPmeConfTable OBJECT-TYPE
45     SYNTAX      SEQUENCE OF EfmCuPmeConfEntry
46     MAX-ACCESS  not-accessible
47     STATUS      current
48     DESCRIPTION
49         "Table for Configuration of common aspects for EFMcu
50         2BASE-TL/10PASS-TS PME ports (modems). Configuration of
51         aspects specific to 2BASE-TL or 10PASS-TS PME types is
52         represented in efmCuPme2BConfTable and efmCuPme10PConfTable,
53         respectively.
54
55         Entries in this table shall be maintained in a persistent
56         manner."
57         ::= { efmCuPme 1 }
58
59
60     efmCuPmeConfEntry OBJECT-TYPE
61     SYNTAX      EfmCuPmeConfEntry
62     MAX-ACCESS  not-accessible
63     STATUS      current
64     DESCRIPTION
65
```

```

1      "An entry in the EFMcu PME Configuration table.
2      Each entry represents common aspects of an EFMcu PME port
3      indexed by the ifIndex. Note that an EFMcu PME port can be
4      stacked below a single PCS port, also indexed by ifIndex,
5      possibly together with other PME ports if PAF is enabled."
6      INDEX { ifIndex }
7      ::= { efmCuPmeConfTable 1 }
8
9
10     EfmCuPmeConfEntry ::=
11     SEQUENCE {
12         efmCuPmeAdminSubType      INTEGER,
13         efmCuPmeAdminProfile      EfmProfileIndexOrZero,
14         efmCuPAFRemoteDiscoveryCode PhysAddress,
15         efmCuPmeThreshLineAtn     Integer32,
16         efmCuPmeThreshSnrMgn      Integer32,
17         efmCuPmeLineAtnCrossingEnable TruthValue,
18         efmCuPmeSnrMgnCrossingEnable TruthValue,
19         efmCuPmeDeviceFaultEnable TruthValue,
20         efmCuPmeConfigInitFailEnable TruthValue,
21         efmCuPmeProtocolInitFailEnable TruthValue
22     }
23
24
25     efmCuPmeAdminSubType OBJECT-TYPE
26     SYNTAX      INTEGER {
27         ieee2BaseTLO(1),
28         ieee2BaseTLR(2),
29         ieee10PassTSO(3),
30         ieee10PassTSR(4),
31         ieee2BaseTLor10PassTSR(5),
32         ieee2BaseTLor10PassTSO(6),
33         ieee10PassTSor2BaseTLO(7)
34     }
35
36     MAX-ACCESS      read-write
37     STATUS           current
38     DESCRIPTION
39         "Administrative (desired) subtype of the PME.
40         Possible values are:
41
42         ieee2BaseTLO      - PME shall operate as 2BaseTL-O
43         ieee2BaseTLR      - PME shall operate as 2BaseTL-R
44         ieee10PassTSO      - PME shall operate as 10PassTS-O
45         ieee10PassTSR      - PME shall operate as 10PassTS-R
46         ieee2BaseTLor10PassTSR - PME shall operate as 2BaseTL-R or
47                                10PassTS-R. The actual value will
48                                be set by the -O link partner
49                                during initialization (handshake).
50         ieee2BaseTLor10PassTSO - PME shall operate as 2BaseTL-O
51                                (preferred) or 10PassTS-O. The
52                                actual value will be set during
53                                initialization depending on the -R
54                                link partner capability (i.e., if
55                                -R is incapable of the preferred
56                                2BaseTL mode, 10PassTS will be
57                                used).
58         ieee10PassTSor2BaseTLO - PME shall operate as 10PassTS-O
59                                (preferred) or 2BaseTL-O. The
60                                actual value will be set during
61                                initialization depending on the -R
62                                link partner capability (i.e., if
63                                -R is incapable of the preferred
64                                10PassTS mode, 2BaseTL will be
65                                used).

```

10PassTS mode, 2BaseTL will be used).

Changing efmCuPmeAdminSubType is a traffic-disruptive operation and as such shall be done when the link is Down. Attempts to change this object shall be rejected if the link is Up or Initializing.

Attempts to change this object to an unsupported subtype (see efmCuPmeSubTypesSupported) shall be rejected.

The current operational subtype is indicated by the efmCuPmeOperSubType variable.

If a Clause 45 MDIO Interface to the PMA/PMD is present, then this object combines values of the Port subtype select bits and the PMA/PMD type selection bits in the 10P/2B PMA/PMD control register."

REFERENCE

"IEEE Std 802.3, 61.1, 45.2.1.14.4, 45.2.1.14.7"

::= { efmCuPmeConfEntry 1 }

efmCuPmeAdminProfile OBJECT-TYPE

SYNTAX EfmProfileIndexOrZero

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Desired PME configuration profile. This object is a pointer to an entry in either the efmCuPme2BProfileTable or the efmCuPme10PProfileTable, depending on the current operating SubType of the PME. The value of this object is the index of the referenced profile.

The value of zero (default) indicates that the PME is configured via the efmCuAdminProfile object for the PCS port to which this PME is assigned. That is, the profile referenced by efmCuPmeAdminProfile takes precedence over the profile(s) referenced by efmCuAdminProfile.

This object is writeable and readable for the CO subtype PMEs (2BaseTL-O or 10PassTS-O). It is irrelevant for the CPE subtype (2BaseTL-R or 10PassTS-R) -- a zero value shall be returned on an attempt to read this object and any attempt to change this object shall be rejected in this case.

Note that the current operational profile value is available via efmCuPmeOperProfile object.

Any modification of this object shall be performed when the link is Down. Attempts to change this object shall be rejected, if the link is Up or Initializing.

Attempts to set this object to a value that is not the value of the index for an active entry in the corresponding profile table shall be rejected.

This object maps to the Clause 30 attribute aProfileSelect.

This object shall be maintained in a persistent manner."

REFERENCE

"IEEE Std 802.3, 30.11.2.1.6"

DEFVAL { 0 }


```
1      ::= { efmCuPmeConfEntry 2 }
2
3  efmCuPAFRemoteDiscoveryCode OBJECT-TYPE
4      SYNTAX      PhysAddress (SIZE(0|6))
5      MAX-ACCESS  read-write
6      STATUS      current
7      DESCRIPTION
8
9          "PAF Remote Discovery Code of the PME port at the CO.
10         The 6-octet Discovery Code of the peer PCS connected via
11         the PME.
12         Reading this object results in a Discovery Get operation.
13         Setting this object to all zeros results in a Discovery
14         Clear_if_Same operation (the value of efmCuPAFDiscoveryCode
15         at the peer PCS shall be the same as efmCuPAFDiscoveryCode of
16         the local PCS associated with the PME for the operation to
17         succeed).
18         Writing a non-zero value to this object results in a
19         Discovery Set_if_Clear operation.
20         A zero-length octet string shall be returned on an attempt to
21         read this object when PAF aggregation is not enabled.
22
23         This object is irrelevant in CPE port (-R) subtypes: in this
24         case, a zero-length octet string shall be returned on an
25         attempt to read this object; writing to this object shall
26         be rejected.
27
28         Discovery shall be performed when the link is Down.
29         Attempts to change this object shall be rejected (in case of
30         SNMP with the error inconsistentValue), if the link is Up or
31         Initializing.
32
33         If a Clause 45 MDIO Interface to the PMA/PMD is present, then
34         this object is a function of 10P/2B aggregation discovery
35         control register, Discovery operation result bits in 10P/2B
36         aggregation and discovery status register and
37         10P/2B aggregation discovery code register."
38
39  REFERENCE
40      "IEEE Std 802.3, 61.2.2.8.4, 45.2.6.6 to 45.2.6.8"
41
42  ::= { efmCuPmeConfEntry 3 }
43
44
45  efmCuPmeThreshLineAtn OBJECT-TYPE
46      SYNTAX      Integer32(-127..128)
47      UNITS       "dB"
48      MAX-ACCESS  read-write
49      STATUS      current
50      DESCRIPTION
51
52          "Desired Line Attenuation threshold for the 2B/10P PME.
53          This object configures the line attenuation alarm threshold.
54          When the current value of Line Attenuation reaches or
55          exceeds this threshold, an efmCuPmeLineAtnCrossing
56          notification may be generated, if enabled by
57          efmCuPmeLineAtnCrossingEnable.
58
59          This object is writeable for the CO subtype PMEs (-O).
60          It is read-only for the CPE subtype (-R).
61
62          Changing of the Line Attenuation threshold shall be performed
63          when the link is Down. Attempts to change this object shall be
64          rejected (in case of SNMP with the error inconsistentValue),
65
```

```
1         if the link is Up or Initializing.
2
3         If a Clause 45 MDIO Interface to the PME is present, then this
4         object maps to the loop attenuation threshold bits in
5         the 2B PMD line quality thresholds register."
6
7     REFERENCE
8         "IEEE Std 802.3, 45.2.1.23"
9         ::= { efmCuPmeConfEntry 4 }
10
11     efmCuPmeThreshSnrMgn OBJECT-TYPE
12         SYNTAX      Integer32(-127..128)
13         UNITS        "dB"
14         MAX-ACCESS   read-write
15         STATUS       current
16         DESCRIPTION
17             "Desired SNR margin threshold for the 2B/10P PME.
18             This object configures the SNR margin alarm threshold.
19             When the current value of SNR margin reaches or exceeds this
20             threshold, an efmCuPmeSnrMgnCrossing notification may be
21             generated, if enabled by efmCuPmeSnrMgnCrossingEnable.
22             This object is writeable for the CO subtype PMEs
23             (2BaseTL-O/10PassTS-O). It is read-only for the CPE subtype
24             (2BaseTL-R/10PassTS-R)."
25
26
27             Changing of the SNR margin threshold shall be performed when
28             the link is Down. Attempts to change this object shall be
29             rejected (in case of SNMP with the error inconsistentValue),
30             if the link is Up or Initializing.
31
32
33             If a Clause 45 MDIO Interface to the PME is present, then this
34             object maps to the SNR margin threshold bits in the 2B PMD
35             line quality thresholds register."
36
37     REFERENCE
38         "IEEE Std 802.3, 45.2.1.23"
39         ::= { efmCuPmeConfEntry 5 }
40
41     efmCuPmeLineAtnCrossingEnable OBJECT-TYPE
42         SYNTAX      TruthValue
43         MAX-ACCESS   read-write
44         STATUS       current
45         DESCRIPTION
46             "Indicates whether efmCuPmeLineAtnCrossing notifications
47             should be generated for this interface.
48
49             A value of true(1) indicates that efmCuPmeLineAtnCrossing
50             notification is enabled. A value of false(2) indicates that
51             the notification is disabled."
52             ::= { efmCuPmeConfEntry 6 }
53
54
55     efmCuPmeSnrMgnCrossingEnable OBJECT-TYPE
56         SYNTAX      TruthValue
57         MAX-ACCESS   read-write
58         STATUS       current
59         DESCRIPTION
60             "Indicates whether efmCuPmeSnrMgnCrossing notifications
61             should be generated for this interface.
62
63             A value of true(1) indicates that efmCuPmeSnrMgnCrossing
64             notification is enabled. A value of false(2) indicates that
```

```
1         the notification is disabled."
2         ::= { efmCuPmeConfEntry 7 }
3
4     efmCuPmeDeviceFaultEnable OBJECT-TYPE
5         SYNTAX      TruthValue
6         MAX-ACCESS   read-write
7         STATUS       current
8         DESCRIPTION
9             "Indicates whether efmCuPmeDeviceFault notifications
10            should be generated for this interface.
11
12            A value of true(1) indicates that efmCuPmeDeviceFault
13            notification is enabled. A value of false(2) indicates that
14            the notification is disabled."
15            ::= { efmCuPmeConfEntry 8 }
16
17
18     efmCuPmeConfigInitFailEnable OBJECT-TYPE
19         SYNTAX      TruthValue
20         MAX-ACCESS   read-write
21         STATUS       current
22         DESCRIPTION
23             "Indicates whether efmCuPmeConfigInitFailure notifications
24            should be generated for this interface.
25
26            A value of true(1) indicates that efmCuPmeConfigInitFailure
27            notification is enabled. A value of false(2) indicates that
28            the notification is disabled."
29            ::= { efmCuPmeConfEntry 9 }
30
31
32     efmCuPmeProtocolInitFailEnable OBJECT-TYPE
33         SYNTAX      TruthValue
34         MAX-ACCESS   read-write
35         STATUS       current
36         DESCRIPTION
37             "Indicates whether efmCuPmeProtocolInitFailure notifications
38            should be generated for this interface.
39
40            A value of true(1) indicates that efmCuPmeProtocolInitFailure
41            notification is enabled. A value of false(2) indicates that
42            the notification is disabled."
43            ::= { efmCuPmeConfEntry 10 }
44
45
46
47
48     efmCuPmeCapabilityTable OBJECT-TYPE
49         SYNTAX      SEQUENCE OF EfmCuPmeCapabilityEntry
50         MAX-ACCESS   not-accessible
51         STATUS       current
52         DESCRIPTION
53             "Table for the configuration of common aspects for EFMcu
54            2BASE-TL/10PASS-TS PME ports (modems). The configuration of
55            aspects specific to 2BASE-TL or 10PASS-TS PME types is
56            represented in the efmCuPme2BConfTable and the
57            efmCuPme10PConfTable, respectively.
58
59            Entries in this table shall be maintained in a persistent
60            manner."
61            ::= { efmCuPme 2 }
62
63     efmCuPmeCapabilityEntry OBJECT-TYPE
64         SYNTAX      EfmCuPmeCapabilityEntry
65
```

```

1      MAX-ACCESS    not-accessible
2      STATUS        current
3      DESCRIPTION
4          "An entry in the EFMcu PME Capability table.
5          Each entry represents common aspects of an EFMcu PME port
6          indexed by the ifIndex. Note that an EFMcu PME port can be
7          stacked below a single PCS port, also indexed by ifIndex,
8          possibly together with other PME ports if PAF is enabled."
9      INDEX { ifIndex }
10     ::= { efmCuPmeCapabilityTable 1 }
11
12
13     EfmCuPmeCapabilityEntry ::=
14         SEQUENCE {
15             efmCuPmeSubTypesSupported      BITS
16         }
17
18     efmCuPmeSubTypesSupported OBJECT-TYPE
19         SYNTAX      BITS {
20             ieee2BaseTLO(0),
21             ieee2BaseTLR(1),
22             ieee10PassTSO(2),
23             ieee10PassTSR(3)
24         }
25
26     MAX-ACCESS    read-only
27     STATUS        current
28     DESCRIPTION
29         "PME supported subtypes. This is a bitmap of possible
30         subtypes. The various bit positions are:
31             ieee2BaseTLO    - PME is capable of operating as 2BaseTL-O
32             ieee2BaseTLR    - PME is capable of operating as 2BaseTL-R
33             ieee10PassTSO    - PME is capable of operating as 10PassTS-O
34             ieee10PassTSR    - PME is capable of operating as 10PassTS-R
35
36
37         The desired mode of operation is determined by
38         efmCuPmeAdminSubType, while efmCuPmeOperSubType reflects the
39         current operating mode.
40
41
42         If a Clause 45 MDIO Interface to the PCS is present, then this
43         object combines the 10PASS-TS capable and 2BASE-TL capable
44         bits in the 10P/2B PMA/PMD speed ability register and the
45         CO supported and CPE supported bits in the 10P/2B PMA/PMD
46         status register."
47     REFERENCE
48         "IEEE Std 802.3, 61.1, 45.2.1.4.7, 45.2.1.4.8, 45.2.1.15.2,
49         45.2.1.15.3"
50     ::= { efmCuPmeCapabilityEntry 1 }
51
52     efmCuPmeStatusTable OBJECT-TYPE
53         SYNTAX      SEQUENCE OF EfmCuPmeStatusEntry
54         MAX-ACCESS    not-accessible
55         STATUS        current
56         DESCRIPTION
57             "This table provides common status information of EFMcu
58             2BASE-TL/10PASS-TS PME ports. Status information specific
59             to 10PASS-TS PME is represented in efmCuPme10PStatusTable.
60
61             This table contains live data from the equipment. As such,
62             it is not persistent."
63     ::= { efmCuPme 3 }
64
65

```

```

1      efmCuPmeStatusEntry OBJECT-TYPE
2          SYNTAX      EfmCuPmeStatusEntry
3          MAX-ACCESS  not-accessible
4          STATUS      current
5          DESCRIPTION
6              "An entry in the EFMCu PME Status table.
7              Each entry represents common aspects of an EFMCu PME port
8              indexed by the ifIndex. Note that an EFMCu PME port can be
9              stacked below a single PCS port, also indexed by ifIndex,
10             possibly together with other PME ports if PAF is enabled."
11          INDEX      { ifIndex }
12          ::= { efmCuPmeStatusTable 1 }
13
14
15      EfmCuPmeStatusEntry ::=
16          SEQUENCE {
17              efmCuPmeOperStatus      INTEGER,
18              efmCuPmeFltStatus        BITS,
19              efmCuPmeOperSubType      INTEGER,
20              efmCuPmeOperProfile      EfmProfileIndexOrZero,
21              efmCuPmeSnrMgn           Integer32,
22              efmCuPmePeerSnrMgn       Integer32,
23              efmCuPmeLineAtn          Integer32,
24              efmCuPmePeerLineAtn      Integer32,
25              efmCuPmeEquivalentLength Unsigned32,
26              efmCuPmeTCCodingErrors   Counter32,
27              efmCuPmeTCCrcErrors      Counter32
28          }
29
30
31      efmCuPmeOperStatus OBJECT-TYPE
32          SYNTAX      INTEGER {
33              up(1),
34              downNotReady(2),
35              downReady(3),
36              init(4)
37          }
38
39
40      MAX-ACCESS  read-only
41      STATUS      current
42      DESCRIPTION
43          "Current PME link Operational Status. Possible values are:
44              up(1)          - The link is Up and ready to pass
45                             64/65-octet encoded frames or fragments.
46              downNotReady(2) - The link is Down and the PME does not
47                             detect Handshake tones from its peer.
48                             This value may indicate a possible
49                             problem with the peer PME.
50              downReady(3)   - The link is Down and the PME detects
51                             Handshake tones from its peer.
52              init(4)        - The link is Initializing, as a result of
53                             ifAdminStatus being set to 'up' for a
54                             particular PME or a PCS to which the PME
55                             is connected.
56
57          This object is intended to supplement the Down(2) state of
58          ifOperStatus.
59
60          This object partially maps to the Clause 30 attribute
61          aPMEStatus.
62
63
64
65

```

```
1      If a Clause 45 MDIO Interface to the PME is present, then this
2      object partially maps to PMA/PMD link status bits in 10P/2B
3      PMA/PMD status register."
4  REFERENCE
5      "IEEE Std 802.3, 30.11.2.1.3, 45.2.1.15.4"
6      ::= { efmCuPmeStatusEntry 1 }
7
8
9  efmCuPmeFltStatus OBJECT-TYPE
10     SYNTAX      BITS {
11         lossOfFraming(0),
12         snrMgnDefect(1),
13         lineAtnDefect(2),
14         deviceFault(3),
15         configInitFailure(4),
16         protocolInitFailure(5)
17     }
18     MAX-ACCESS   read-only
19     STATUS       current
20     DESCRIPTION
21         "Current/Last PME link Fault Status. This is a bitmap of
22         possible conditions. The various bit positions are:
23
24
25         lossOfFraming      - Loss of Framing for 10P or
26                             Loss of Sync word for 2B PMD or
27                             Loss of 64/65-octet framing.
28
29         snrMgnDefect       - SNR margin dropped below the
30                             threshold.
31
32         lineAtnDefect      - Line Attenuation exceeds the
33                             threshold.
34
35         deviceFault        - Indicates a vendor-dependent
36                             diagnostic or self-test fault
37                             has been detected.
38
39         configInitFailure  - Configuration initialization failure,
40                             due to inability of the PME link to
41                             support the configuration profile,
42                             requested during initialization.
43
44         protocolInitFailure - Protocol initialization failure, due
45                             to an incompatible protocol used by
46                             the peer PME during init (that could
47                             happen if a peer PMD is a regular
48                             G.SDHSL/VDSL modem instead of a
49                             2BASE-TL/10PASS-TS PME).
50
51         This object is intended to supplement ifOperStatus in IF-MIB.
52
53         This object holds information about the last fault.
54         efmCuPmeFltStatus is cleared by the device restart.
55         In addition, lossOfFraming, configInitFailure, and
56         protocolInitFailure are cleared by PME init;
57         deviceFault is cleared by successful diagnostics/test;
58         snrMgnDefect and lineAtnDefect are cleared by SNR margin
59         and Line attenuation, respectively, returning to norm and by
60         PME init.
61
62         This object partially maps to the Clause 30 attribute
63         aPMEStatus.
64
65         If a Clause 45 MDIO Interface to the PME is present, then this
66         object consolidates information from various PMA/PMD
```

```
1         registers, namely: Fault bit in PMA/PMD status 1 register,
2         10P/2B PMA/PMD link loss register,
3         10P outgoing indicator bits status register,
4         10P incoming indicator bits status register,
5         2B state defects register."
6     REFERENCE
7         "IEEE Std 802.3, 30.11.2.1.3, 45.2.1.2.1, 45.2.1.41,
8         45.2.1.42, 45.2.1.57"
9     ::= { efmCuPmeStatusEntry 2 }
10
11
12 efmCuPmeOperSubType OBJECT-TYPE
13     SYNTAX      INTEGER {
14         ieee2BaseTLO(1),
15         ieee2BaseTLR(2),
16         ieee10PassTSO(3),
17         ieee10PassTSR(4)
18     }
19     MAX-ACCESS   read-only
20     STATUS       current
21     DESCRIPTION
22         "Current operational subtype of the PME.
23         Possible values are:
24             ieee2BaseTLO          - PME operates as 2BaseTL-O
25             ieee2BaseTLR          - PME operates as 2BaseTL-R
26             ieee10PassTSO         - PME operates as 10PassTS-O
27             ieee10PassTSR         - PME operates as 10PassTS-R
28
29         The desired operational subtype of the PME can be configured
30         via the efmCuPmeAdminSubType variable.
31
32         If a Clause 45 MDIO Interface to the PMA/PMD is present, then
33         this object combines values of the Port subtype select
34         bits, the PMA/PMD type selection bits in the 10P/2B
35         PMA/PMD control register, and the PMA/PMD link status bits in
36         the 10P/2B PMA/PMD status register."
37     REFERENCE
38         "IEEE Std 802.3, 61.1, 45.2.1.14.4, 45.2.1.14.7, 45.2.1.15.4"
39     ::= { efmCuPmeStatusEntry 3 }
40
41
42 efmCuPmeOperProfile OBJECT-TYPE
43     SYNTAX      EfmProfileIndexOrZero
44     MAX-ACCESS   read-only
45     STATUS       current
46     DESCRIPTION
47         "PME current operating profile. This object is a pointer to
48         an entry in either the efmCuPme2BProfileTable or the
49         efmCuPme10PProfileTable, depending on the current operating
50         SubType of the PME as indicated by efmCuPmeOperSubType.
51         Note that a profile entry to which efmCuPmeOperProfile is
52         pointing can be created automatically to reflect achieved
53         parameters in adaptive (not fixed) initialization,
54         i.e., values of efmCuPmeOperProfile and efmCuAdminProfile or
55         efmCuPmeAdminProfile may differ.
56         The value of zero indicates that the PME is Down or
57         Initializing.
58
59         This object partially maps to the aOperatingProfile attribute
60         in Clause 30."
61     REFERENCE
```

```
1      "IEEE Std 802.3, 30.11.2.1.7"
2      ::= { efmCuPmeStatusEntry 4 }
3  efmCuPmeSnrMgn OBJECT-TYPE
4      SYNTAX      Integer32(-127..128|65535)
5      UNITS       "dB"
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "The current signal-to-noise ratio (SNR) margin with respect
10         to the received signal as perceived by the local PME.
11         The value of 65535 is returned when the PME is Down or
12         Initializing.
13
14         This object maps to the aPMESNRMgn attribute in Clause 30.
15
16         If a Clause 45 MDIO Interface is present, then this
17         object maps to the 10P/2B RX SNR margin register."
18  REFERENCE
19      "IEEE Std 802.3, 30.11.2.1.4, 45.2.1.19"
20      ::= { efmCuPmeStatusEntry 5 }
21
22  efmCuPmePeerSnrMgn OBJECT-TYPE
23      SYNTAX      Integer32(-127..128|65535)
24      UNITS       "dB"
25      MAX-ACCESS  read-only
26      STATUS      current
27      DESCRIPTION
28          "The current SNR margin in dB with respect to the received
29         signal, as perceived by the remote (link partner) PME.
30         The value of 65535 is returned when the PME is Down or
31         Initializing.
32
33         This object is irrelevant for the -R PME subtypes. The value
34         of 65535 shall be returned in this case.
35
36         If a Clause 45 MDIO Interface is present, then this
37         object maps to the 10P/2B link partner RX SNR margin
38         register."
39  REFERENCE
40      "IEEE Std 802.3, 45.2.1.20"
41      ::= { efmCuPmeStatusEntry 6}
42
43  efmCuPmeLineAtn OBJECT-TYPE
44      SYNTAX      Integer32(-127..128|65535)
45      UNITS       "dB"
46      MAX-ACCESS  read-only
47      STATUS      current
48      DESCRIPTION
49          "The current Line Attenuation in dB as perceived by the local
50         PME.
51         The value of 65535 is returned when the PME is Down or
52         Initializing.
53
54         If a Clause 45 MDIO Interface is present, then this
55         object maps to the Line Attenuation register."
56  REFERENCE
57      "IEEE Std 802.3, 45.2.1.21"
58      ::= { efmCuPmeStatusEntry 7 }
59
60
61
62
63
64
65
```



```
1      efmCuPmePeerLineAtn OBJECT-TYPE
2          SYNTAX      Integer32(-127..128|65535)
3          UNITS        "dB"
4          MAX-ACCESS   read-only
5          STATUS       current
6          DESCRIPTION
7              "The current Line Attenuation in dB as perceived by the remote
8              (link partner) PME.
9              The value of 65535 is returned when the PME is Down or
10             Initializing.
11
12             This object is irrelevant for the -R PME subtypes. The value
13             of 65535 shall be returned in this case.
14
15             If a Clause 45 MDIO Interface is present, then this
16             object maps to the 20P/2B link partner Line Attenuation
17             register."
18
19      REFERENCE
20          "IEEE Std 802.3, 45.2.1.22"
21          ::= { efmCuPmeStatusEntry 8 }
22
23
24      efmCuPmeEquivalentLength OBJECT-TYPE
25          SYNTAX      Unsigned32(0..8192|65535)
26          UNITS        "m"
27          MAX-ACCESS   read-only
28          STATUS       current
29          DESCRIPTION
30              "An estimate of the equivalent loop's physical length in
31              meters, as perceived by the PME after the link is established.
32              An equivalent loop is a hypothetical 26AWG (0.4mm) loop with a
33              perfect square root attenuation characteristic, without any
34              bridged taps.
35              The value of 65535 is returned if the link is Down or
36              Initializing or the PME is unable to estimate the equivalent
37              length.
38
39              For a 10BASE-TL PME, if a Clause 45 MDIO Interface to the PME
40              is present, then this object maps to the 10P Electrical Length
41              register."
42
43      REFERENCE
44          "IEEE Std 802.3, 45.2.1.29"
45          ::= { efmCuPmeStatusEntry 9 }
46
47
48      efmCuPmeTCCodingErrors OBJECT-TYPE
49          SYNTAX      Counter32
50          MAX-ACCESS   read-only
51          STATUS       current
52          DESCRIPTION
53              "The number of 64/65-octet encapsulation errors. This counter
54              is incremented for each 64/65-octet encapsulation error
55              detected by the 64/65-octet receive function.
56
57              This object maps to aTCCodingViolations attribute in
58              Clause 30.
59
60              If a Clause 45 MDIO Interface to the PME TC is present, then
61              this object maps to the TC coding violations register
62              (see IEEE Std 802.3 45.2.6.12).
63
64
65
```

```

1      Discontinuities in the value of this counter can occur at
2      re-initialization of the management system, and at other times
3      as indicated by the value of ifCounterDiscontinuityTime,
4      defined in IF-MIB."
5  REFERENCE
6      "IEEE Std 802.3, 61.3.3.1, 30.11.2.1.5, 45.2.6.12"
7      ::= { efmCuPmeStatusEntry 10 }
8
9
10     efmCuPmeTCCrcErrors OBJECT-TYPE
11         SYNTAX      Counter32
12         MAX-ACCESS   read-only
13         STATUS       current
14         DESCRIPTION
15             "The number of TC-CRC errors. This counter is incremented for
16             each TC-CRC error detected by the 64/65-octet receive function
17             (see IEEE Std 802.3 61.3.3.3 and IEEE Std 802.3 Figure 61-19).
18
19             This object maps to aTCCRCErrors attribute in
20             Clause 30.
21
22             If a Clause 45 MDIO Interface to the PME TC is present, then
23             this object maps to the TC CRC error register
24             (see IEEE Std 802.3 45.2.6.11).
25
26             Discontinuities in the value of this counter can occur at
27             re-initialization of the management system, and at other times
28             as indicated by the value of ifCounterDiscontinuityTime,
29             defined in IF-MIB."
30     REFERENCE
31         "IEEE Std 802.3, 61.3.3.3, 30.11.2.1.10, 45.2.6.11"
32         ::= { efmCuPmeStatusEntry 11 }
33
34
35 -- 2BASE-TL specific PME group
36
37
38     efmCuPme2B          OBJECT IDENTIFIER ::= { efmCuPme 5 }
39
40
41     efmCuPme2BProfileTable OBJECT-TYPE
42         SYNTAX      SEQUENCE OF EfmCuPme2BProfileEntry
43         MAX-ACCESS   not-accessible
44         STATUS       current
45         DESCRIPTION
46             "This table supports definitions of administrative and
47             operating profiles for 2BASE-TL PMEs.
48             The first 14 entries in this table shall be defined as
49             follows (see IEEE Std 802.3 Annex 63A):
50             -----+-----+-----+-----+-----+-----+-----+-----
51             Profile MinRate MaxRate Power Region Constellation Comment
52             index  (kb/s)  (kb/s)  (dBm)
53             -----+-----+-----+-----+-----+-----+-----+-----
54
55                 1      5696      5696      13.5      1      32-TCPAM      default
56                 2      3072      3072      13.5      1      32-TCPAM
57                 3      2048      2048      13.5      1      16-TCPAM
58                 4      1024      1024      13.5      1      16-TCPAM
59                 5        704        704      13.5      1      16-TCPAM
60                 6        512        512      13.5      1      16-TCPAM
61                 7      5696      5696      14.5      2      32-TCPAM
62                 8      3072      3072      14.5      2      32-TCPAM
63                 9      2048      2048      14.5      2      16-TCPAM
64                 10     1024      1024      13.5      2      16-TCPAM
65

```

1	11	704	704	13.5	2	16-TCPAM	
2	12	512	512	13.5	2	16-TCPAM	
3	13	192	5696	0	1	0	best effort
4	14	192	5696	0	2	0	best effort
5	-----+-----+-----+-----+-----+-----+-----						

These default entries shall be created during agent initialization and shall not be deleted.

Entries following the first 14 can be dynamically created and deleted to provide custom administrative (configuration) profiles and automatic operating profiles.

This table shall be maintained in a persistent manner."

REFERENCE

"IEEE Std 802.3, Annex 63A, 30.11.2.1.6"

::= { efmCuPme2B 2 }

efmCuPme2BProfileEntry OBJECT-TYPE

SYNTAX EfmCuPme2BProfileEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry corresponds to a single 2BASE-TL PME profile.

Each profile contains a set of parameters, used either for configuration or representation of a 2BASE-TL PME.

In case a particular profile is referenced via the efmCuPmeAdminProfile object (or efmCuAdminProfile if efmCuPmeAdminProfile is zero), it represents the desired parameters for the 2BaseTL-O PME initialization. If a profile is referenced via an efmCuPmeOperProfile object, it represents the current operating parameters of an operational PME.

Profiles may be created/deleted using the row creation/deletion mechanism via efmCuPme2BProfileRowStatus. If an active entry is referenced, the entry shall remain 'active' until all references are removed.

Default entries shall not be removed."

INDEX { efmCuPme2BProfileIndex }

::= { efmCuPme2BProfileTable 1 }

EfmCuPme2BProfileEntry ::=

SEQUENCE {

efmCuPme2BProfileIndex	EfmProfileIndex,
efmCuPme2BProfileDescr	SnmpAdminString,
efmCuPme2BRegion	INTEGER,
efmCuPme2BsMode	EfmProfileIndexOrZero,
efmCuPme2BMinDataRate	Unsigned32,
efmCuPme2BMaxDataRate	Unsigned32,
efmCuPme2BPower	Unsigned32,
efmCuPme2BConstellation	INTEGER,
efmCuPme2BProfileRowStatus	RowStatus

}

efmCuPme2BProfileIndex OBJECT-TYPE

SYNTAX EfmProfileIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

```
1         "2BASE-TL PME profile index.
2         This object is the unique index associated with this profile.
3         Entries in this table are referenced via efmCuAdminProfile or
4         efmCuPmeAdminProfile objects."
5         ::= { efmCuPme2BProfileEntry 1 }
6 efmCuPme2BProfileDescr OBJECT-TYPE
7     SYNTAX      SnmpAdminString
8     MAX-ACCESS  read-create
9     STATUS      current
10    DESCRIPTION
11        "A textual string containing information about a 2BASE-TL PME
12        profile. The string may include information about the data
13        rate and spectral limitations of this particular profile."
14        ::= { efmCuPme2BProfileEntry 2 }
15
16
17 efmCuPme2BRegion OBJECT-TYPE
18     SYNTAX      INTEGER {
19         region1(1),
20         region2(2)
21     }
22     MAX-ACCESS  read-create
23     STATUS      current
24     DESCRIPTION
25         "Regional settings for a 2BASE-TL PME, as specified in the
26         relevant Regional Annex of ITU-T Recommendation G.991.2.
27         Regional settings specify the Power Spectral Density (PSD)
28         mask and the Power Back-Off (PBO) values, and place
29         limitations on the max allowed data rate, power, and
30         constellation.
31
32         Possible values for this object are:
33             region1      - Annexes A and F (e.g., North America)
34             region2      - Annexes B and G (e.g., Europe)
35
36         Annex A/B specify regional settings for data rates from
37         192 kb/s to 2304 kb/s using 16-TCPAM encoding.
38         Annex F/G specify regional settings for rates from
39         2320 kb/s to 3840 kb/s using 16-TCPAM encoding and from
40         768 kb/s to 5696 kb/s using 32-TCPAM encoding.
41
42         If a Clause 45 MDIO Interface to the PME is present, then this
43         object partially maps to the Region bits in the 2B general
44         parameter register."
45     REFERENCE
46         "IEEE Std 802.3, 45.2.1.45; ITU-T Recommendation G.991.2,
47         Annexes A, B, F and G"
48     ::= { efmCuPme2BProfileEntry 3 }
49
50
51 efmCuPme2BsMode OBJECT-TYPE
52     SYNTAX      EfmProfileIndexOrZero
53     MAX-ACCESS  read-create
54     STATUS      current
55     DESCRIPTION
56         "Desired custom Spectral Mode for a 2BASE-TL PME. This object
57         is a pointer to an entry in efmCuPme2BsModeTable and a block
58         of entries in efmCuPme2BRateReachTable, which together define
59         (country-specific) reach-dependent rate limitations in
60         addition to those defined by efmCuPme2BRegion.
61
62
63
64
65
```

1 The value of this object is the index of the referenced
2 spectral mode.
3 The value of zero (default) indicates that no specific
4 spectral mode is applicable.
5
6 Attempts to set this object to a value that is not the value
7 of the index for an active entry in the corresponding spectral
8 mode table shall be rejected."
9
10 REFERENCE
11 "efmCuPme2BsModeTable, efmCuPme2BRateReachTable"
12 DEFVAL { 0 }
13 ::= { efmCuPme2BProfileEntry 4 }
14
15 efmCuPme2BMinDataRate OBJECT-TYPE
16 SYNTAX Unsigned32(192..5696)
17 UNITS "Kbps"
18 MAX-ACCESS read-create
19 STATUS current
20 DESCRIPTION
21 "Minimum Data Rate for the 2BASE-TL PME.
22 This object can take values of (n x 64)kb/s,
23 where n=3..60 for 16-TCPAM and n=12..89 for 32-TCPAM encoding.
24
25 The data rate of the 2BASE-TL PME is considered 'fixed' when
26 the value of this object equals that of efmCuPme2BMaxDataRate.
27 If efmCuPme2BMinDataRate is less than efmCuPme2BMaxDataRate in
28 the administrative profile, the data rate is considered
29 'adaptive', and shall be set to the maximum attainable rate
30 not exceeding efmCuPme2BMaxDataRate, under the spectral
31 limitations placed by the efmCuPme2BRegion and
32 efmCuPme2BsMode.
33
34 Note that the current operational data rate of the PME is
35 represented by the ifSpeed object of IF-MIB.
36
37 If a Clause 45 MDIO Interface to the PME is present, then this
38 object maps to the Min Data Rate1 bits in the 2B PMD
39 parameters register.
40
41 This object shall be maintained in a persistent manner."
42
43 REFERENCE
44 "IEEE Std 802.3, 45.2.1.46"
45 ::= { efmCuPme2BProfileEntry 5 }
46 efmCuPme2BMaxDataRate OBJECT-TYPE
47 SYNTAX Unsigned32(192..5696)
48 UNITS "Kbps"
49 MAX-ACCESS read-create
50 STATUS current
51 DESCRIPTION
52 "Maximum Data Rate for the 2BASE-TL PME.
53 This object can take values of (n x 64)kb/s,
54 where n=3..60 for 16-TCPAM and n=12..89 for 32-TCPAM encoding.
55
56 The data rate of the 2BASE-TL PME is considered 'fixed' when
57 the value of this object equals that of efmCuPme2BMinDataRate.
58 If efmCuPme2BMinDataRate is less than efmCuPme2BMaxDataRate in
59 the administrative profile, the data rate is considered
60 'adaptive', and shall be set to the maximum attainable rate
61 not exceeding efmCuPme2BMaxDataRate, under the spectral
62 limitations placed by the efmCuPme2BRegion and
63 efmCuPme2BsMode.
64
65 Note that the current operational data rate of the PME is
66 represented by the ifSpeed object of IF-MIB.
67 If a Clause 45 MDIO Interface to the PME is present, then this
68 object maps to the Max Data Rate1 bits in the 2B PMD
69 parameters register.
70 This object shall be maintained in a persistent manner."

```
1      limitations placed by the efmCuPme2BRegion and
2      efmCuPme2BsMode.
3
4      Note that the current operational data rate of the PME is
5      represented by the ifSpeed object of IF-MIB.
6
7      If a Clause 45 MDIO Interface to the PME is present, then this
8      object maps to the Max Data Rate1 bits in the 2B PMD
9      parameters register.
10
11     This object shall be maintained in a persistent manner."
12
13     REFERENCE
14         "IEEE Std 802.3, 45.2.1.46"
15     ::= { efmCuPme2BProfileEntry 6 }
16
17     efmCuPme2BPower OBJECT-TYPE
18     SYNTAX      Unsigned32(0|10..42)
19     UNITS       "0.5 dBm"
20     MAX-ACCESS  read-create
21     STATUS      current
22     DESCRIPTION
23         "Signal Transmit Power. Multiple of 0.5 dBm.
24         The value of 0 in the administrative profile means that the
25         signal transmit power is not fixed and shall be set to
26         maximize the attainable rate, under the spectral limitations
27         placed by the efmCuPme2BRegion and efmCuPme2BsMode.
28
29         If a Clause 45 MDIO Interface to the PME is present, then this
30         object maps to the Power1 bits in the 2B PMD parameters
31         register."
32     REFERENCE
33         "IEEE Std 802.3, 45.2.1.46"
34     ::= { efmCuPme2BProfileEntry 7 }
35
36     efmCuPme2BConstellation OBJECT-TYPE
37     SYNTAX      INTEGER {
38         adaptive(0),
39         tcpam16(1),
40         tcpam32(2)
41     }
42     MAX-ACCESS  read-create
43     STATUS      current
44     DESCRIPTION
45         "TCPAM Constellation of the 2BASE-TL PME.
46         The possible values are:
47             adaptive(0)    - either 16- or 32-TCPAM
48             tcpam16(1)     - 16-TCPAM
49             tcpam32(2)     - 32-TCPAM
50
51         The value of adaptive(0) in the administrative profile means
52         that the constellation is not fixed and shall be set to
53         maximize the attainable rate, under the spectral limitations
54         placed by the efmCuPme2BRegion and efmCuPme2BsMode.
55
56         If a Clause 45 MDIO Interface to the PME is present, then this
57         object maps to the Constellation1 bits in the 2B general
58         parameter register."
59     REFERENCE
60         "IEEE Std 802.3, 45.2.1.46"
```

```
1      ::= { efmCuPme2BProfileEntry 8 }
2
3  efmCuPme2BProfileRowStatus OBJECT-TYPE
4      SYNTAX      RowStatus
5      MAX-ACCESS  read-create
6      STATUS      current
7      DESCRIPTION
8
9          "This object controls the creation, modification, or deletion
10         of the associated entry in the efmCuPme2BProfileTable per the
11         semantics of RowStatus.
12
13         If an 'active' entry is referenced via efmCuAdminProfile or
14         efmCuPmeAdminProfile instance(s), the entry shall remain
15         'active'.
16
17         An 'active' entry shall not be modified. In order to modify
18         an existing entry, it shall be taken out of service (by setting
19         this object to 'notInService'), modified, and set 'active'
20         again."
21
22      ::= { efmCuPme2BProfileEntry 9 }
23  efmCuPme2BsModeTable OBJECT-TYPE
24      SYNTAX      SEQUENCE OF EfmCuPme2BsModeEntry
25      MAX-ACCESS  not-accessible
26      STATUS      current
27      DESCRIPTION
28
29          "This table, together with efmCu2BReachRateTable, supports
30          definition of administrative custom spectral modes for
31          2BASE-TL PMEs, describing spectral limitations in addition to
32          those specified by efmCuPme2BRegion.
33
34          In some countries, spectral regulations (e.g., UK ANFP) limit
35          the length of the loops for certain data rates. This table
36          allows these country-specific limitations to be specified.
37
38          Entries in this table referenced by the efmCuPme2BsMode
39          shall not be deleted until all the active references are
40          removed.
41
42          This table shall be maintained in a persistent manner."
43      REFERENCE
44          "efmCu2BReachRateTable"
45
46      ::= { efmCuPme2B 3 }
47
48  efmCuPme2BsModeEntry OBJECT-TYPE
49      SYNTAX      EfmCuPme2BsModeEntry
50      MAX-ACCESS  not-accessible
51      STATUS      current
52      DESCRIPTION
53
54          "Each entry specifies a spectral mode description and its
55          index, which is used to reference corresponding entries in the
56          efmCu2BReachRateTable.
57
58          Entries may be created/deleted using the row creation/
59          deletion mechanism via efmCuPme2BsModeRowStatus."
60      INDEX { efmCuPme2BsModeIndex }
61      ::= { efmCuPme2BsModeTable 1 }
62
63  EfmCuPme2BsModeEntry ::=
64      SEQUENCE {
65
```

```
1          efmCuPme2BsModeIndex          EfmProfileIndex,
2          efmCuPme2BsModeDescr          SnmpAdminString,
3          efmCuPme2BsModeRowStatus      RowStatus
4      }
5
6  efmCuPme2BsModeIndex OBJECT-TYPE
7      SYNTAX      EfmProfileIndex
8      MAX-ACCESS  not-accessible
9      STATUS      current
10     DESCRIPTION
11         "2BASE-TL PME Spectral Mode index.
12         This object is the unique index associated with this spectral
13         mode.
14         Entries in this table are referenced via the efmCuPme2BsMode
15         object."
16     ::= { efmCuPme2BsModeEntry 1 }
17
18  efmCuPme2BsModeDescr OBJECT-TYPE
19     SYNTAX      SnmpAdminString
20     MAX-ACCESS  read-create
21     STATUS      current
22     DESCRIPTION
23         "A textual string containing information about a 2BASE-TL PME
24         spectral mode. The string may include information about
25         corresponding (country-specific) spectral regulations
26         and rate/reach limitations of this particular spectral mode."
27     ::= { efmCuPme2BsModeEntry 2 }
28
29  efmCuPme2BsModeRowStatus OBJECT-TYPE
30     SYNTAX      RowStatus
31     MAX-ACCESS  read-create
32     STATUS      current
33     DESCRIPTION
34         "This object controls creation, modification, or deletion of
35         the associated entry in efmCuPme2BsModeTable per the semantics
36         of RowStatus.
37
38         If an 'active' entry is referenced via efmCuPme2BsMode
39         instance(s), the entry shall remain 'active'.
40
41         An 'active' entry shall not be modified. In order to modify
42         an existing entry, it shall be taken out of service (by setting
43         this object to 'notInService'), modified, and set 'active'
44         again."
45     ::= { efmCuPme2BsModeEntry 3 }
46
47  efmCuPme2BReachRateTable OBJECT-TYPE
48     SYNTAX      SEQUENCE OF EfmCuPme2BReachRateEntry
49     MAX-ACCESS  not-accessible
50     STATUS      current
51     DESCRIPTION
52         "This table supports the definition of administrative custom
53         spectral modes for 2BASE-TL PMEs, providing spectral
54         limitations in addition to those specified by
55         efmCuPme2BRegion.
56         The spectral regulations in some countries (e.g., UK ANFP)
57         limit the length of the loops for certain data rates.
58         This table allows these country-specific limitations to be
```


specified.

Below is an example of this table for NICC Document ND1602:2005/08:

Equivalent Length (m)	MaxRate PAM16 (kb/s)	MaxRate PAM32 (kb/s)
975	2304	5696
1125	2304	5504
1275	2304	5120
1350	2304	4864
1425	2304	4544
1500	2304	4288
1575	2304	3968
1650	2304	3776
1725	2304	3520
1800	2304	3264
1875	2304	3072
1950	2048	2688
2100	1792	2368
2250	1536	0
2400	1408	0
2550	1280	0
2775	1152	0
2925	1152	0
3150	1088	0
3375	1024	0

Entries in this table referenced by an efmCuPme2BsMode instance shall not be deleted.

This table shall be maintained in a persistent manner."

REFERENCE

"NICC Document ND1602:2005/08"

::= { efmCuPme2B 4 }

efmCuPme2BReachRateEntry OBJECT-TYPE

SYNTAX EfmCuPme2BReachRateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry specifies maximum 2BASE-TL PME data rates allowed for a certain equivalent loop length, when using 16-TCPAM or 32-TCPAM encoding.

When a 2BASE-TL PME is initialized, its data rate shall not exceed the following limitations:

- the value of efmCuPme2BMaxDataRate
- maximum data rate allowed by efmCuPme2BRegion and efmCuPme2BPower
- maximum data rate for a given encoding specified in the efmCuPme2BsModeEntry, corresponding to the equivalent loop length, estimated by the PME

efmCuPme2BEquivalentLength values should be assigned in increasing order, starting from the minimum value.

```

1      Entries may be created/deleted using the row creation/
2      deletion mechanism via efmCuPme2ReachRateRowStatus."
3      INDEX { efmCuPme2BsModeIndex, efmCuPme2BReachRateIndex }
4      ::= { efmCuPme2BReachRateTable 1 }
5
6      EfmCuPme2BReachRateEntry ::=
7      SEQUENCE {
8          efmCuPme2BReachRateIndex      EfmProfileIndex,
9          efmCuPme2BEquivalentLength    Unsigned32,
10         efmCuPme2BMaxDataRatePam16    Unsigned32,
11         efmCuPme2BMaxDataRatePam32    Unsigned32,
12         efmCuPme2BReachRateRowStatus  RowStatus
13     }
14
15
16     efmCuPme2BReachRateIndex OBJECT-TYPE
17         SYNTAX      EfmProfileIndex
18         MAX-ACCESS  not-accessible
19         STATUS      current
20         DESCRIPTION
21             "2BASE-TL custom spectral mode Reach-Rate table index.
22             This object is the unique index associated with each entry."
23         ::= { efmCuPme2BReachRateEntry 1 }
24
25
26     efmCuPme2BEquivalentLength OBJECT-TYPE
27         SYNTAX      Unsigned32(0..8192)
28         UNITS       "m"
29         MAX-ACCESS  read-create
30         STATUS      current
31         DESCRIPTION
32             "Maximum allowed equivalent loop's physical length in meters
33             for the specified data rates.
34             An equivalent loop is a hypothetical 26AWG (0.4mm) loop with a
35             perfect square root attenuation characteristic, without any
36             bridged taps."
37         ::= { efmCuPme2BReachRateEntry 2 }
38
39
40     efmCuPme2BMaxDataRatePam16 OBJECT-TYPE
41         SYNTAX      Unsigned32(0|192..5696)
42         UNITS       "Kbps"
43         MAX-ACCESS  read-create
44         STATUS      current
45         DESCRIPTION
46             "Maximum data rate for a 2BASE-TL PME at the specified
47             equivalent loop's length using TC-PAM16 encoding.
48             The value of zero means that TC-PAM16 encoding should not be
49             used at this distance."
50         ::= { efmCuPme2BReachRateEntry 3 }
51
52
53     efmCuPme2BMaxDataRatePam32 OBJECT-TYPE
54         SYNTAX      Unsigned32(0|192..5696)
55         UNITS       "Kbps"
56         MAX-ACCESS  read-create
57         STATUS      current
58         DESCRIPTION
59             "Maximum data rate for a 2BASE-TL PME at the specified
60             equivalent loop's length using TC-PAM32 encoding.
61             The value of zero means that TC-PAM32 encoding should not be
62             used at this distance."
63         ::= { efmCuPme2BReachRateEntry 4 }
64
65

```

```

1
2     efmCuPme2BReachRateRowStatus OBJECT-TYPE
3         SYNTAX      RowStatus
4         MAX-ACCESS  read-create
5         STATUS      current
6         DESCRIPTION
7             "This object controls the creation, modification, or deletion
8             of the associated entry in the efmCuPme2BReachRateTable per
9             the semantics of RowStatus.
10
11             If an 'active' entry is referenced via efmCuPme2BsMode
12             instance(s), the entry shall remain 'active'.
13
14             An 'active' entry shall not be modified. In order to modify
15             an existing entry, it shall be taken out of service (by setting
16             this object to 'notInService'), modified, and set 'active'
17             again."
18         ::= { efmCuPme2BReachRateEntry 5 }
19
20
21
22
23 -- 10PASS-TS specific PME group
24 efmCuPme10P          OBJECT IDENTIFIER ::= { efmCuPme 6 }
25
26 efmCuPme10PProfileTable OBJECT-TYPE
27     SYNTAX      SEQUENCE OF EfmCuPme10PProfileEntry
28     MAX-ACCESS  not-accessible
29     STATUS      current
30     DESCRIPTION
31         "This table supports definitions of configuration profiles for
32         10PASS-TS PMEs.
33         The first 22 entries in this table shall be defined as
34         follows (see IEEE Std 802.3 Annex 62B.3, Table 62B-1):
35         -----+-----+-----+-----+-----+-----+-----
36         Profile Bandplan UPBO BandNotch DRate URate Comment
37         Index  PSDMask#  p#    p#          p#    p#
38         -----+-----+-----+-----+-----+-----+-----
39
40         1      1        3     2,6,10,11    20     20 default profile
41         2      13        5      0          20     20
42         3       1        1      0          20     20
43         4      16        0      0          100    100
44         5      16        0      0           70     50
45         6       6        0      0           50     10
46         7      17        0      0           30     30
47         8       8        0      0           30      5
48         9       4        0      0           25     25
49        10       4        0      0           15     15
50        11      23        0      0           10     10
51        12      23        0      0            5      5
52        13      16        0     2,5,9,11    100    100
53        14      16        0     2,5,9,11     70     50
54        15       6        0     2,6,10,11    50     10
55        16      17        0     2,5,9,11    30     30
56        17       8        0     2,6,10,11    30      5
57        18       4        0     2,6,10,11    25     25
58        19       4        0     2,6,10,11    15     15
59        20      23        0     2,5,9,11    10     10
60        21      23        0     2,5,9,11     5      5
61        22      30        0      0          200    50
62         -----+-----+-----+-----+-----+-----+-----
63
64
65

```

```

1
2     These default entries shall be created during agent
3     initialization and shall not be deleted.
4
5     Entries following the first 22 can be dynamically created and
6     deleted to provide custom administrative (configuration)
7     profiles and automatic operating profiles.
8
9
10    This table shall be maintained in a persistent manner."
11    REFERENCE
12        "IEEE Std 802.3, Annex 62B.3, 30.11.2.1.6"
13    ::= { efmCuPme10P 1 }
14
15    efmCuPme10PProfileEntry OBJECT-TYPE
16        SYNTAX      EfmCuPme10PProfileEntry
17        MAX-ACCESS   not-accessible
18        STATUS       current
19        DESCRIPTION
20            "Each entry corresponds to a single 10PASS-TS PME profile.
21
22            Each profile contains a set of parameters, used either for
23            configuration or representation of a 10PASS-TS PME.
24            In case a particular profile is referenced via the
25            efmCuPmeAdminProfile object (or efmCuAdminProfile if
26            efmCuPmeAdminProfile is zero), it represents the desired
27            parameters for the 10PassTS-O PME initialization.
28            If a profile is referenced via an efmCuPmeOperProfile object,
29            it represents the current operating parameters of the PME.
30
31            Profiles may be created/deleted using the row creation/
32            deletion mechanism via efmCuPme10PProfileRowStatus. If an
33            'active' entry is referenced, the entry shall remain 'active'
34            until all references are removed.
35            Default entries shall not be removed."
36        INDEX { efmCuPme10PProfileIndex }
37        ::= { efmCuPme10PProfileTable 1 }
38
39    EfmCuPme10PProfileEntry ::=
40    SEQUENCE {
41        efmCuPme10PProfileIndex          EfmProfileIndex,
42        efmCuPme10PProfileDescr          SnmpAdminString,
43        efmCuPme10PBandplanPSDMskProfile INTEGER,
44        efmCuPme10PUPBORReferenceProfile INTEGER,
45        efmCuPme10PBandNotchProfiles    BITS,
46        efmCuPme10PPayloadDRateProfile  INTEGER,
47        efmCuPme10PPayloadURateProfile  INTEGER,
48        efmCuPme10PProfileRowStatus     RowStatus
49    }
50
51    efmCuPme10PProfileIndex OBJECT-TYPE
52        SYNTAX      EfmProfileIndex
53        MAX-ACCESS   not-accessible
54        STATUS       current
55        DESCRIPTION
56            "10PASS-TS PME profile index.
57            This object is the unique index associated with this profile.
58            Entries in this table are referenced via efmCuAdminProfile or
59            efmCuPmeAdminProfile."
60        ::= { efmCuPme10PProfileEntry 1 }

```

Copyright © 2023 IEEE. All rights reserved.
This is an unapproved IEEE Standards draft, subject to change.

```

1      profile7(7)      ANSI T1.424 FTTCab.M2      D/D/U/D/U      A
2      profile8(8)      ANSI T1.424 FTTEEx.M2      D/D/U/D/U      A
3      profile9(9)      ANSI T1.424 FTTCab.M1      U/D/U/D/x      A
4      profile10(10)    ANSI T1.424 FTTEEx.M1      U/D/U/D/x      A
5      profile11(11)    ANSI T1.424 FTTCab.M2      U/D/U/D/x      A
6      profile12(12)    ANSI T1.424 FTTEEx.M2      U/D/U/D/x      A
7      profile13(13)    ETSI TS 101 270-1 Pcab.M1.A    x/D/U/D/U      B
8      profile14(14)    ETSI TS 101 270-1 Pcab.M1.B    x/D/U/D/U      B
9      profile15(15)    ETSI TS 101 270-1 Pex.P1.M1    x/D/U/D/U      B
10     profile16(16)    ETSI TS 101 270-1 Pex.P2.M1    x/D/U/D/U      B
11     profile17(17)    ETSI TS 101 270-1 Pcab.M2      x/D/U/D/U      B
12     profile18(18)    ETSI TS 101 270-1 Pex.P1.M2    x/D/U/D/U      B
13     profile19(19)    ETSI TS 101 270-1 Pex.P2.M2    x/D/U/D/U      B
14     profile20(20)    ETSI TS 101 270-1 Pcab.M1.A    U/D/U/D/x      B
15     profile21(21)    ETSI TS 101 270-1 Pcab.M1.B    U/D/U/D/x      B
16     profile22(22)    ETSI TS 101 270-1 Pex.P1.M1    U/D/U/D/x      B
17     profile23(23)    ETSI TS 101 270-1 Pex.P2.M1    U/D/U/D/x      B
18     profile24(24)    ETSI TS 101 270-1 Pcab.M2      U/D/U/D/x      B
19     profile25(25)    ETSI TS 101 270-1 Pex.P1.M2    U/D/U/D/x      B
20     profile26(26)    ETSI TS 101 270-1 Pex.P2.M2    U/D/U/D/x      B
21     profile27(27)    ITU-T G.993.1 F.1.2.1          x/D/U/D/U      Annex F
22     profile28(28)    ITU-T G.993.1 F.1.2.2          x/D/U/D/U      Annex F
23     profile29(29)    ITU-T G.993.1 F.1.2.3          x/D/U/D/U      Annex F
24     profile30(30)    ANSI T1.424 FTTCab.M1 (ext.)    x/D/U/D/U/D     Annex A
25     -----+-----+-----+-----+-----
26     "
27
28     REFERENCE
29
30     "IEEE Std 802.3, Annex 62A"
31
32     ::= { efmCuPme10PPProfileEntry 3 }
33
34     efmCuPme10PUPBReferenceProfile OBJECT-TYPE
35     SYNTAX INTEGER {
36         profile0(0),
37         profile1(1),
38         profile2(2),
39         profile3(3),
40         profile4(4),
41         profile5(5),
42         profile6(6),
43         profile7(7),
44         profile8(8),
45         profile9(9)
46     }
47
48     MAX-ACCESS read-create
49     STATUS current
50     DESCRIPTION
51         "The 10PASS-TS PME Upstream Power Back-Off (UPBO) Reference
52         PSD Profile, as specified in 802.3 Annex 62A, table 62A-3.
53         Possible values are:
54         -----+-----+-----+-----+-----
55         Profile Name      Reference      PSD
56         -----+-----+-----+-----+-----
57         profile0(0)      no profile
58         profile1(1)      ANSI T1.424      Noise A      M1
59         profile2(2)      ANSI T1.424      Noise A      M2
60         profile3(3)      ANSI T1.424      Noise F      M1
61         profile4(4)      ANSI T1.424      Noise F      M2
62         profile5(5)      ETSI TS 101 270-1 Noise A&B
63         profile6(6)      ETSI TS 101 270-1 Noise C

```

```

1      profile7(7)  ETSI TS 101 270-1  Noise D
2      profile8(8)  ETSI TS 101 270-1  Noise E
3      profile9(9)  ETSI TS 101 270-1  Noise F
4      -----+-----
5      "
6
7      REFERENCE
8      "IEEE Std 802.3, Annex 62A.3.5"
9      ::= { efmCuPme10PProfileEntry 4 }
10
11  efmCuPme10PBandNotchProfiles  OBJECT-TYPE
12      SYNTAX  BITS {
13          profile0(0),
14          profile1(1),
15          profile2(2),
16          profile3(3),
17          profile4(4),
18          profile5(5),
19          profile6(6),
20          profile7(7),
21          profile8(8),
22          profile9(9),
23          profile10(10),
24          profile11(11)
25      }
26
27  MAX-ACCESS  read-create
28  STATUS      current
29
30  DESCRIPTION
31      "The 10PASS-TS PME Egress Control Band Notch Profile bitmap,
32      as specified in IEEE Std 802.3 Annex 62A, table 62A-4. Possible
33      values are:
34      -----+-----+-----+-----+-----+-----
35      Profile Name  G.991.3  T1.424 TS 101 270-1  StartF  EndF
36                      table    table  table          (MHz)  (MHz)
37      -----+-----+-----+-----+-----+-----
38      profile0(0)   no profile
39      profile1(1)   F-5 #01  -      -              1.810  1.825
40      profile2(2)   6-2      15-1  17              1.810  2.000
41      profile3(3)   F-5 #02  -      -              1.907  1.912
42      profile4(4)   F-5 #03  -      -              3.500  3.575
43      profile5(5)   6-2      -      17              3.500  3.800
44      profile6(6)   -        15-1  -              3.500  4.000
45      profile7(7)   F-5 #04  -      -              3.747  3.754
46      profile8(8)   F-5 #05  -      -              3.791  3.805
47      profile9(9)   6-2      -      17              7.000  7.100
48      profile10(10) F-5 #06  15-1  -              7.000  7.300
49      profile11(11) 6-2      15-1  1              10.100 10.150
50      -----+-----+-----+-----+-----+-----
51
52
53      Any combination of profiles can be specified by ORing
54      individual profiles, for example, a value of 0x2230 selects
55      profiles 2, 6, 10, and 11."
56
57  REFERENCE
58      "IEEE Std 802.3, Annex 62A.3.5"
59      ::= { efmCuPme10PProfileEntry 5 }
60
61  efmCuPme10PPayloadDRateProfile  OBJECT-TYPE
62      SYNTAX  INTEGER {
63          profile5(5),
64          profile10(10),
65

```

```
1      profile15(15),
2      profile20(20),
3      profile25(25),
4      profile30(30),
5      profile50(50),
6      profile70(70),
7      profile100(100),
8      profile140(140),
9      profile200(200)
10     }
11
12     MAX-ACCESS    read-create
13     STATUS        current
14     DESCRIPTION
15         "The 10PASS-TS PME Downstream Payload Rate Profile, as
16         specified in IEEE Std 802.3 Annex 62A. Possible values are:
17         profile5(5)      - 2.5 Mb/s
18         profile10(10)    - 5 Mb/s
19         profile15(15)    - 7.5 Mb/s
20         profile20(20)    - 10 Mb/s
21         profile25(25)    - 12.5 Mb/s
22         profile30(30)    - 15 Mb/s
23         profile50(50)    - 25 Mb/s
24         profile70(70)    - 35 Mb/s
25         profile100(100)  - 50 Mb/s
26         profile140(140)  - 70 Mb/s
27         profile200(200)  - 100 Mb/s
28
29         Each value represents a target for the PME's Downstream
30         Payload Bitrate as seen at the MII. If the payload rate of
31         the selected profile cannot be achieved based on the loop
32         environment, bandplan, and PSD mask, the PME initialization
33         shall fail."
34
35     REFERENCE
36         "IEEE Std 802.3, Annex 62A.3.6"
37     ::= { efmCuPme10PPProfileEntry 6 }
38
39
40
41     efmCuPme10PPayloadURateProfile OBJECT-TYPE
42     SYNTAX          INTEGER {
43         profile5(5),
44         profile10(10),
45         profile15(15),
46         profile20(20),
47         profile25(25),
48         profile30(30),
49         profile50(50),
50         profile70(70),
51         profile100(100)
52     }
53
54     MAX-ACCESS    read-create
55     STATUS        current
56     DESCRIPTION
57         "The 10PASS-TS PME Upstream Payload Rate Profile, as specified
58         in 802.3 Annex 62A. Possible values are:
59         profile5(5)      - 2.5 Mb/s
60         profile10(10)    - 5 Mb/s
61         profile15(15)    - 7.5 Mb/s
62         profile20(20)    - 10 Mb/s
63         profile25(25)    - 12.5 Mb/s
64         profile30(30)    - 15 Mb/s
65
```



```
1         profile50(50)      - 25 Mb/s
2         profile70(70)      - 35 Mb/s
3         profile100(100)    - 50 Mb/s
4     Each value represents a target for the PME's Upstream Payload
5     Bitrate as seen at the MII. If the payload rate of the
6     selected profile cannot be achieved based on the loop
7     environment, bandplan, and PSD mask, the PME initialization
8     shall fail."
9
10    REFERENCE
11        "IEEE Std 802.3, Annex 62A.3.6"
12    ::= { efmCuPme10PProfileEntry 7 }
13
14    efmCuPme10PProfileRowStatus OBJECT-TYPE
15        SYNTAX      RowStatus
16        MAX-ACCESS   read-create
17        STATUS       current
18        DESCRIPTION
19            "This object controls creation, modification, or deletion of
20            the associated entry in efmCuPme10PProfileTable per the
21            semantics of RowStatus.
22
23            If an active entry is referenced via efmCuAdminProfile or
24            efmCuPmeAdminProfile, the entry shall remain 'active' until
25            all references are removed.
26
27            An 'active' entry shall not be modified. In order to modify
28            an existing entry, it shall be taken out of service (by setting
29            this object to 'notInService'), modified, and set 'active'
30            again."
31    ::= { efmCuPme10PProfileEntry 8 }
32
33
34
35
36    efmCuPme10PStatusTable OBJECT-TYPE
37        SYNTAX      SEQUENCE OF EfmCuPme10PStatusEntry
38        MAX-ACCESS   not-accessible
39        STATUS       current
40        DESCRIPTION
41            "This table provides status information of EFMcu 10PASS-TS
42            PMEs (modems).
43
44            This table contains live data from the equipment. As such,
45            it is not persistent."
46    ::= { efmCuPme10P 2 }
47
48
49    efmCuPme10PStatusEntry OBJECT-TYPE
50        SYNTAX      EfmCuPme10PStatusEntry
51        MAX-ACCESS   not-accessible
52        STATUS       current
53        DESCRIPTION
54            "An entry in the EFMcu 10PASS-TS PME Status table."
55        INDEX { ifIndex }
56    ::= { efmCuPme10PStatusTable 1 }
57
58
59    EfmCuPme10PStatusEntry ::=
60        SEQUENCE {
61            efmCuPme10PFECCorrectedBlocks    Counter32,
62            efmCuPme10PFECUncorrectedBlocks  Counter32
63        }
64
65
```

```
1      efmCuPme10PFECCorrectedBlocks OBJECT-TYPE
2          SYNTAX          Counter32
3          MAX-ACCESS      read-only
4          STATUS          current
5          DESCRIPTION
6              "The number of received and corrected Forward Error Correction
7              (FEC) codewords in this 10PASS-TS PME.
8
9
10             This object maps to the aPMEFECCorrectedBlocks attribute in
11             Clause 30.
12
13             If a Clause 45 MDIO Interface to the PMA/PMD is present,
14             then this object maps to the 10P FEC correctable errors
15             register.
16
17             Discontinuities in the value of this counter can occur at
18             re-initialization of the management system, and at other times
19             as indicated by the value of ifCounterDiscontinuityTime,
20             defined in IF-MIB."
21
22         REFERENCE
23             "IEEE Std 802.3, 45.2.1.25, 30.11.2.1.8"
24             ::= { efmCuPme10PStatusEntry 1 }
25
26     efmCuPme10PFECUncorrectedBlocks OBJECT-TYPE
27         SYNTAX          Counter32
28         MAX-ACCESS      read-only
29         STATUS          current
30         DESCRIPTION
31             "The number of received uncorrectable FEC codewords in this
32             10PASS-TS PME.
33
34
35             This object maps to the aPMEFECUncorrectableBlocks attribute
36             in Clause 30.
37
38             If a Clause 45 MDIO Interface to the PMA/PMD is present,
39             then this object maps to the 10P FEC uncorrectable errors
40             register.
41
42             Discontinuities in the value of this counter can occur at
43             re-initialization of the management system, and at other times
44             as indicated by the value of ifCounterDiscontinuityTime,
45             defined in IF-MIB."
46
47         REFERENCE
48             "IEEE Std 802.3, 45.2.1.26, 30.11.2.1.9"
49             ::= { efmCuPme10PStatusEntry 2 }
50
51
52     --
53     -- Conformance statements
54     --
55
56     efmCuGroups          OBJECT IDENTIFIER ::= { efmCuConformance 1 }
57
58     efmCuCompliances OBJECT IDENTIFIER ::= { efmCuConformance 2 }
59
60     -- Object Groups
61
62     efmCuBasicGroup OBJECT-GROUP
63         OBJECTS {
64             efmCuPAFSupported,
```

```
1         efmCuAdminProfile,
2         efmCuTargetDataRate,
3         efmCuTargetSnrMgn,
4         efmCuAdaptiveSpectra,
5         efmCuPortSide,
6         efmCuFltStatus
7     }
8
9     STATUS          current
10    DESCRIPTION
11        "A collection of objects representing management information
12        common for all types of EFMcu ports."
13    ::= { efmCuGroups 1 }
14
15    efmCuPAFGroup OBJECT-GROUP
16    OBJECTS {
17        efmCuPeerPAFSupported,
18        efmCuPAFCapacity,
19        efmCuPeerPAFCapacity,
20        efmCuPAFAdminState,
21        efmCuPAFDiscoveryCode,
22        efmCuPAFRemoteDiscoveryCode,
23        efmCuNumPMEs
24    }
25
26    STATUS          current
27    DESCRIPTION
28        "A collection of objects supporting optional PME
29        Aggregation Function (PAF) and PAF discovery in EFMcu ports."
30    ::= { efmCuGroups 2 }
31
32    efmCuPAFErrorsGroup OBJECT-GROUP
33    OBJECTS {
34        efmCuPAFInErrors,
35        efmCuPAFInSmallFragments,
36        efmCuPAFInLargeFragments,
37        efmCuPAFInBadFragments,
38        efmCuPAFInLostFragments,
39        efmCuPAFInLostStarts,
40        efmCuPAFInLostEnds,
41        efmCuPAFInOverflows
42    }
43
44    STATUS          current
45    DESCRIPTION
46        "A collection of objects supporting optional error counters
47        of PAF on EFMcu ports."
48    ::= { efmCuGroups 3 }
49
50    efmCuPmeGroup OBJECT-GROUP
51    OBJECTS {
52        efmCuPmeAdminProfile,
53        efmCuPmeOperStatus,
54        efmCuPmeFltStatus,
55        efmCuPmeSubTypesSupported,
56        efmCuPmeAdminSubType,
57        efmCuPmeOperSubType,
58        efmCuPAFRemoteDiscoveryCode,
59        efmCuPmeOperProfile,
60        efmCuPmeSnrMgn,
61        efmCuPmePeerSnrMgn,
62        efmCuPmeLineAtn,
63        efmCuPmePeerLineAtn,
```

```
1         efmCuPmeEquivalentLength,
2         efmCuPmeTCCodingErrors,
3         efmCuPmeTCCrcErrors,
4         efmCuPmeThreshLineAtn,
5         efmCuPmeThreshSnrMgn
6     }
7     STATUS          current
8     DESCRIPTION
9         "A collection of objects providing information about
10        a 2BASE-TL/10PASS-TS PME."
11    ::= { efmCuGroups 4 }
12
13    efmCuAlarmConfGroup OBJECT-GROUP
14    OBJECTS {
15        efmCuThreshLowRate,
16        efmCuLowRateCrossingEnable,
17        efmCuPmeThreshLineAtn,
18        efmCuPmeLineAtnCrossingEnable,
19        efmCuPmeThreshSnrMgn,
20        efmCuPmeSnrMgnCrossingEnable,
21        efmCuPmeDeviceFaultEnable,
22        efmCuPmeConfigInitFailEnable,
23        efmCuPmeProtocolInitFailEnable
24    }
25    STATUS          current
26    DESCRIPTION
27        "A collection of objects supporting configuration of alarm
28        thresholds and notifications in EFMcu ports."
29    ::= { efmCuGroups 5 }
30
31    efmCuNotificationGroup NOTIFICATION-GROUP
32    NOTIFICATIONS {
33        efmCuLowRateCrossing,
34        efmCuPmeLineAtnCrossing,
35        efmCuPmeSnrMgnCrossing,
36        efmCuPmeDeviceFault,
37        efmCuPmeConfigInitFailure,
38        efmCuPmeProtocolInitFailure
39    }
40    STATUS          current
41    DESCRIPTION
42        "This group supports notifications of significant conditions
43        associated with EFMcu ports."
44    ::= { efmCuGroups 6 }
45
46    efmCuPme2BProfileGroup OBJECT-GROUP
47    OBJECTS {
48        efmCuPme2BProfileDescr,
49        efmCuPme2BRegion,
50        efmCuPme2BsMode,
51        efmCuPme2BMinDataRate,
52        efmCuPme2BMaxDataRate,
53        efmCuPme2BPower,
54        efmCuPme2BConstellation,
55        efmCuPme2BProfileRowStatus,
56        efmCuPme2BsModeDescr,
57        efmCuPme2BsModeRowStatus,
58        efmCuPme2BEquivalentLength,
59        efmCuPme2BMaxDataRatePam16,
```

```

1         efmCuPme2BMaxDataRatePam32,
2         efmCuPme2BReachRateRowStatus
3     }
4     STATUS          current
5     DESCRIPTION
6         "A collection of objects that constitute a configuration
7         profile for configuration of 2BASE-TL ports."
8         ::= { efmCuGroups 7}
9
10
11     efmCuPme10PProfileGroup OBJECT-GROUP
12     OBJECTS {
13         efmCuPme10PProfileDescr,
14         efmCuPme10PBandplanPSDMskProfile,
15         efmCuPme10PUPBReferenceProfile,
16         efmCuPme10PBandNotchProfiles,
17         efmCuPme10PPayloadDRateProfile,
18         efmCuPme10PPayloadURateProfile,
19         efmCuPme10PProfileRowStatus
20     }
21     STATUS          current
22     DESCRIPTION
23         "A collection of objects that constitute a configuration
24         profile for configuration of 10PASS-TS ports."
25         ::= { efmCuGroups 8 }
26
27
28     efmCuPme10PStatusGroup OBJECT-GROUP
29     OBJECTS {
30         efmCuPme10PFECCorrectedBlocks,
31         efmCuPme10PFECUncorrectedBlocks
32     }
33     STATUS          current
34     DESCRIPTION
35         "A collection of objects providing status information
36         specific to 10PASS-TS PMEs."
37         ::= { efmCuGroups 9 }
38
39
40
41 -- Compliance statements
42
43     efmCuCompliance MODULE-COMPLIANCE
44     STATUS          current
45     DESCRIPTION
46         "The compliance statement for 2BASE-TL/10PASS-TS interfaces.
47         Compliance with the following external compliance statements
48         is required:
49
50         MIB module          Compliance Statement
51         -----
52         IF-MIB              ifCompliance3
53         IEEE8023-EtherLike-MIB dot3Compliance2
54         MAU-MIB             mauModIfComp13
55
56
57         Compliance with the following external compliance statements
58         is optional for implementations supporting PME Aggregation
59         Function (PAF) with flexible cross-connect between the PCS
60         and PME ports:
61
62         MIB module          Compliance Statement
63         -----
64         IF-INVERTED-STACK-MIB ifInvCompliance
65

```

```

1          IF-CAP-STACK-MIB          ifCapStackCompliance"
2
3      MODULE -- this module
4          MANDATORY-GROUPS {
5              efmCuBasicGroup,
6              efmCuPmeGroup,
7              efmCuAlarmConfGroup,
8              efmCuNotificationGroup
9          }
10
11
12      GROUP          efmCuPme2BProfileGroup
13      DESCRIPTION
14          "Support for this group is only required for implementations
15          supporting 2BASE-TL PHY."
16
17
18      GROUP          efmCuPme10PProfileGroup
19      DESCRIPTION
20          "Support for this group is only required for implementations
21          supporting 10PASS-TS PHY."
22
23
24      GROUP          efmCuPAFGroup
25      DESCRIPTION
26          "Support for this group is only required for
27          implementations supporting PME Aggregation Function (PAF)."
28
29
30      GROUP          efmCuPAFErrorsGroup
31      DESCRIPTION
32          "Support for this group is optional for implementations
33          supporting PME Aggregation Function (PAF)."
34
35
36      GROUP          efmCuPme10PStatusGroup
37      DESCRIPTION
38          "Support for this group is optional for implementations
39          supporting 10PASS-TS PHY."
40
41
42      OBJECT          efmCuPmeSubTypesSupported
43      SYNTAX          BITS {
44          ieee2BaseTLO(0),
45          ieee2BaseTLR(1),
46          ieee10PassTSO(2),
47          ieee10PassTSR(3)
48      }
49      DESCRIPTION
50          "Support for all subtypes is not required. However, at
51          least one value shall be supported."
52
53
54      OBJECT          efmCuPmeAdminSubType
55      MIN-ACCESS      read-only
56      DESCRIPTION
57          "Write access is not required (needed only for PMEs
58          supporting more than a single subtype, e.g.,
59          ieee2BaseTLO and ieee2BaseTLR or ieee10PassTSO and
60          ieee10PassTSR)."
61
62
63      OBJECT          efmCuTargetSnrMgn
64      MIN-ACCESS      read-only
65      DESCRIPTION
66          "Write access is optional. For PHYs without write access,
67          the target SNR margin shall be fixed at 5dB for 2BASE-TL

```

```
1           and 6dB for 10PASS-TS."
2
3       OBJECT      efmCuAdaptiveSpectra
4       MIN-ACCESS  read-only
5       DESCRIPTION
6           "Write access is optional. For PHYs without write access,
7           the default value should be false."
8
9
10      ::= { efmCuCompliances 1 }
11  END
```

12. Ethernet wide area network (WAN) interface sublayer (WIS) MIB module

This clause defines a portion of the MIB for use with SNMP. In particular, it defines objects for managing IEEE 802.3 WAN interface sublayers.

12.1 Overview

The objects defined in this clause are used in conjunction with objects defined in the Interfaces Group MIB in IETF RFC 2863, the SONET/SDH Interface MIB in IETF RFC 3592, and the IEEE 802.3 MAU MIB defined in Clause 13 of this document to manage the Ethernet WAN interface sublayer (WIS) defined in IEEE Std 802.3. The WIS contains functions to perform OC-192c/VC-4-64c framing and scrambling. It resides between the Physical Coding Sublayer (PCS) and the Physical Medium Attachment (PMA) sublayer within a 10GBASE-W 10 Gb/s WAN-compatible Physical Layer device (PHY) and may be used in conjunction with any of the PCS, PMA, and physical medium dependent (PMD) sublayers defined in IEEE Std 802.3 for 10GBASE-W PHYs. Three types of 10GBASE-W PHYs are defined, distinguished by the type of optics employed: 10GBASE-SW, 10GBASE-LW, and 10GBASE-EW. The objects defined in this clause may be used to manage an Ethernet interface employing any type of 10GBASE-W PHY. They do not apply to any other kind of interface. In particular, they do not apply to so-called Ethernet line terminating equipment (ELTE) residing within a SONET network element that uses the 10GBASE-W PMA/PMD sublayers but otherwise acts as SONET line terminating equipment (LTE).

The objects presented here—along with those incorporated by reference from the Interfaces Group MIB, the SONET/SDH Interface MIB, and the IEEE 802.3 MAU MIB—are intended to provide exact representations of the mandatory attributes in the oWIS managed object class (i.e., the members of the pWISBasic package) defined in Clause 30 of IEEE Std 802.3. They are also intended to provide approximate representations of the optional attributes (i.e., the members of the pWISOptional package). Some objects with no analogs in oWIS are defined to support WIS testing features required by Clause 50 of IEEE Std 802.3.

12.1.1 Relationship to the SONET/SDH interface MIB

Since the Ethernet WAN interface sublayer was designed to be SONET-compatible, information similar to that provided by most of the members of the oWIS managed object class is available from objects defined in the SONET-MIB in IETF RFC 3592. Thus, the MIB module defined in this clause is a sparse augmentation of the SONET-MIB—in other words, every table defined here is an extension of some table in the SONET-MIB—and its compliance statement **REQUIRES** that an agent implementing the objects defined in this clause also implement the relevant SONET-MIB objects. That includes all objects required by sonetCompliance2 as well as some that it leaves optional.

It should be noted that some of the objects incorporated by reference from the SONET-MIB—specifically, the threshold objects and interval counter objects—provide only approximate representations of the corresponding oWIS attributes, as detailed in 12.1.6. An alternative approach would have been to define new objects to exactly match the oWIS definitions. That approach was rejected because the SONET-MIB objects are already used in deployed systems to manage the SONET sublayers of ATM over SONET and PPP over SONET interfaces, and it was deemed undesirable to use a different scheme to manage the SONET sublayers of 10 Gb/s WAN-compatible Ethernet interfaces. Note that the approach adopted by this clause requires no hardware support beyond that mandated by 50.3.11 of IEEE Std 802.3.

12.1.2 Relationship to the Ethernet-like interface MIB

An interface that includes the Ethernet WIS is, by definition, an Ethernet-like interface, and an agent implementing the objects defined in this clause shall also implement the objects required by the Ethernet-like interface MIB module defined in Clause 10.

1 **12.1.3 Relationship to the IEEE 802.3 MAU MIB**

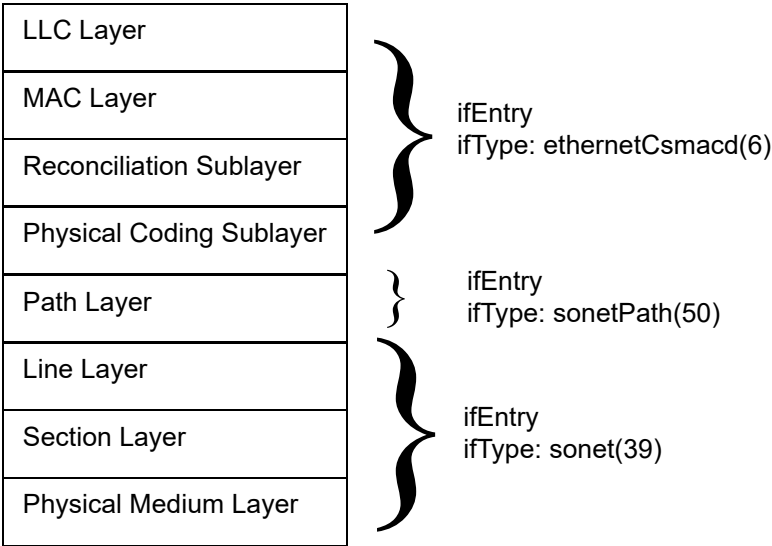
2
3 Support for the mauModIfCompl3 compliance statement of the MAU-MIB module defined in Clause 13 is
4 required for all Ethernet-like interfaces. The MAU-MIB module is needed in order to allow applications to
5 control and/or determine the media type in use; this is important for devices that can support both the
6 10GBASE-R 10 Gb/s LAN format (which does not include the WIS) and the 10GBASE-W 10 Gb/s WAN
7 format (which does include the WIS). The MAU-MIB module also provides the means to put a device in
8 standby mode or to reset it; the latter may be used to re-initialize the WIS.
9

10
11
12 **12.1.4 Use of the ifTable**

13
14 This subclause specifies how the ifTable, as defined in IETF RFC 2863, is used for the Ethernet WIS
15 application.
16

17
18 **12.1.4.1 Layering model**

19
20 Ethernet interfaces that employ the WIS are layered as defined in IEEE Std 802.3. The corresponding use of
21 the ifTable defined in IETF RFC 2863 is shown in Figure 12-1.
22



46 **Figure 12-1—Use of ifTable for an Ethernet WIS port**

47
48
49
50 The exact configuration and multiplexing of the layers is maintained in the ifStackTable in IETF RFC 2863
51 and in the ifInvStackTable in IETF RFC 2864.
52

53
54 **12.1.4.2 Use of ifTable for LLC layer/MAC sublayer/reconciliation sublayer/physical coding**
55 **sublayer**

56
57 The ifTable shall be used as specified in Clause 10 and Clause 13 for the LLC Layer/MAC sublayer/
58 reconciliation sublayer/physical coding sublayer.
59

60
61 **12.1.4.3 Use of ifTable for SONET/SDH path layer**

62
63 The ifTable shall be used as specified in IETF RFC 3592 for the SONET/SDH path layer. The value of
64 ifHighSpeed is set to 9585. ifSpeed reports a value of 4294967295.
65

12.1.4.4 Use of ifTable for SONET/SDH medium/section/line layer

The ifTable shall be used as specified in IETF RFC 3592 for the SONET/SDH Medium/Section/Line Layer. The value of ifHighSpeed is set to 9953. ifSpeed reports a value of 4294967295.

12.1.5 SONET/SDH terminology

The SONET/SDH terminology used in IEEE Std 802.3 is mostly the same as in IETF RFC 3592, but there are a few differences. In those cases, the definitions in Clause 3 take precedence.

12.1.6 Mapping of IEEE 802.3 managed objects

Table 12-1 contains the mapping between oWIS managed objects in the pWIS Basic package defined in IEEE Std 802.3 and managed objects defined in this clause and in associated MIB modules, i.e., the IF-MIB in IETF RFC 2863, the SONET-MIB in IETF RFC 3592, and the IEEE 802.3 MAU-MIB module defined in Clause 13 of this document.

Table 12-1—Mapping of IEEE 802.3 managed objects (pWIS Basic package)

IEEE 802.3 managed object		Corresponding SNMP object
oWIS - pWISBasic package	aWISID	IF-MIB - ifIndex
	aSectionStatus	SONET-MIB - sonetSectionCurrentStatus
	aLineStatus	SONET-MIB - sonetLineCurrentStatus
	aPathStatus	etherWisPathCurrentStatus
	aFarEndPathStatus	etherWisFarEndPathCurrentStatus

The Unequipped defect is not defined by IEEE Std 802.3.

Table 12-2 contains the same mapping information for the pWIS optional package.

The threshold and counter objects imported from the SONET-MIB are not completely equivalent to the corresponding IEEE 802.3 objects. The specific differences are presented in Table 12-3. Despite the semantic differences between the threshold objects and counter objects imported from the SONET-MIB and the corresponding IEEE 802.3 objects, the hardware support mandated by 50.3.11 of IEEE Std 802.3 suffices for both. See Annex 12A for details.

Table 12-2—Mapping of IEEE 802.3 managed objects (pWIS optional package)

IEEE 802.3 managed object		Corresponding SNMP object
oWIS – pWISOptional package	aSectionSESThreshold	SONET-MIB – sonetSESthresholdSe
	aSectionSESs	SONET-MIB – sonetSectionCurrentSESs + sonetSectionIntervalSESs
	aSectionESs	SONET-MIB – sonetSectionCurrentESs + sonetSectionIntervalESs
	aSectionSEFSs	SONET-MIB – sonetSectionCurrentSEFSs + sonetSectionIntervalSEFSs
	aSectionCVs	SONET-MIB – sonetSectionCurrentCVs + sonetSectionIntervalCVs
	aJ0ValueTX	etherWisSectionCurrentJ0Transmitted
	aJ0ValueRX	etherWisSectionCurrentJ0Received
	aLineSESThreshold	SONET-MIB – sonetSESthresholdSet
	aLineSESs	SONET-MIB – sonetLineCurrentSESs + sonetLineIntervalSESs
	aLineESs	SONET-MIB – sonetLineCurrentESs + sonetLineIntervalESs
	aLineCVs	SONET-MIB – sonetLineCurrentCVs + sonetLineIntervalCVs
	aFarEndLineSESs	SONET-MIB – sonetFarEndLineCurrentSESs + sonetFarEndLineIntervalSESs
	aFarEndLineESs	SONET-MIB – sonetFarEndLineCurrentESs + sonetFarEndLineIntervalESs
	aFarEndLineCVs	SONET-MIB – sonetFarEndLineCurrentCVs + sonetFarEndLineIntervalCVs
	aPathSESThreshold	SONET-MIB – sonetSESthresholdSet
	aPathSESs	SONET-MIB – sonetPathCurrentSESs + sonetPathIntervalSESs
	aPathESs	SONET-MIB – sonetPathCurrentESs + sonetPathIntervalESs
	aPathCVs	SONET-MIB – sonetPathCurrentCVs + sonetPathIntervalCVs
	aJ1ValueTX	etherWisPathCurrentJ1Transmitted

Table 12-2—Mapping of IEEE 802.3 managed objects (pWIS optional package) (continued)

IEEE 802.3 managed object		Corresponding SNMP object
oWIS - pWISOptional package (continued)	aJ1ValueRX	etherWisPathCurrentJ1Received
	aFarEndPathSESSs	SONET-MIB – sonetFarEndPathCurrentSESSs + sonetFarEndPathIntervalSESSs
	aFarEndPathESs	SONET-MIB – sonetFarEndPathCurrentESs + sonetFarEndPathIntervalESs
	aFarEndPathCVs	SONET-MIB – sonetFarEndPathCurrentCVs + sonetFarEndPathIntervalCVs

Table 12-3—IEEE 802.3 managed object and SNMP object differences

IEEE 802.3 managed object	How corresponding SNMP object differs
aSectionSESThreshold	This object is defined in IEEE Std 802.3 as an integer with one instance per interface. sonetSESThresholdSet is an enumerated value that has one instance per network element; it controls the thresholds for all layers simultaneously and allows only certain discrete values to be selected.
aSectionSESSs	This object is defined in IEEE Std 802.3 as a generalized nonresettable counter. The objects sonetSectionCurrentSESSs and sonetSectionIntervalSESSs are 15-minute interval counters.
aSectionESs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3. The objects sonetSectionCurrentESs and sonetSectionIntervalESs are 15-minute interval counters.
aSectionSEFSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3. The objects sonetSectionCurrentSEFSs and sonetSectionIntervalSEFSs are 15-minute interval counters.
aSectionCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetSectionCurrentCVs and sonetSectionIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as severely errored seconds.
aLineSESThreshold	This object is defined in IEEE Std 802.3 as an integer with one instance per interface. sonetSESThresholdSet is an enumerated value that has one instance per network element; it controls the thresholds for all layers simultaneously and allows only certain discrete values to be selected.
aLineSESSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetLineCurrentSESSs and sonetLineIntervalSESSs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as unavailable seconds.
aLineESs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetLineCurrentESs and sonetLineIntervalESs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as unavailable seconds.

Table 12-3—IEEE 802.3 managed object and SNMP object differences (continued)

IEEE 802.3 managed object	How corresponding SNMP object differs
aLineCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetLineCurrentCVs and sonetLineIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify either as severely errored seconds or as unavailable seconds.
aFarEndLineSESSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetFarEndLineCurrentSESSs and sonetFarEndLineIntervalSESSs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as unavailable seconds.
aFarEndLineESSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetFarEndLineCurrentESSs and sonetFarEndLineIntervalESSs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as unavailable seconds.
aFarEndLineCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetFarEndLineCurrentCVs and sonetFarEndLineIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify either as severely errored seconds or as unavailable seconds.
aPathSESThreshold	This object is defined in IEEE Std 802.3 as an integer with one instance per interface. sonetSESThresholdSet is an enumerated value that has one instance per network element; it controls the thresholds for all layers simultaneously and allows only certain discrete values to be selected.
aPathSESSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetPathCurrentSESSs and sonetPathIntervalSESSs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as unavailable seconds. In addition, IEEE Std 802.3 includes PLM-P and LCD-P defects in the criteria for declaring path layer severely errored seconds, while IETF RFC 3592 does not.
aPathESSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetPathCurrentESSs and sonetPathIntervalESSs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as unavailable seconds. In addition, IEEE Std 802.3 includes PLM-P and LCD-P defects in the criteria for declaring path layer errored seconds, while IETF RFC 3592 does not.
aPathCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetPathCurrentCVs and sonetPathIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify either as severely errored seconds or as unavailable seconds.

Table 12-3—IEEE 802.3 managed object and SNMP object differences (continued)

IEEE 802.3 managed object	How corresponding SNMP object differs
aFarEndPathSESSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetFarEndPathCurrentSESSs and sonetFarEndPathIntervalSESSs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as unavailable seconds. In addition, IEEE Std 802.3 includes far-end PLM-P and LCD-P defects in the criteria for declaring far-end path layer severely errored seconds, while IETF RFC 3592 does not.
aFarEndPathESSs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetFarEndPathCurrentESSs and sonetFarEndPathIntervalESSs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify as unavailable seconds. In addition, IEEE Std 802.3 includes far-end PLM-P and LCD-P defects in the criteria for declaring far-end path layer errored seconds, while IETF RFC 3592 does not.
aFarEndPathCVs	This object is defined as a generalized nonresettable counter in IEEE Std 802.3, and it is not subject to inhibiting. The objects sonetFarEndPathCurrentCVs and sonetFarEndPathIntervalCVs are 15-minute interval counters, and they are inhibited (not incremented) during 1-second intervals that qualify either as severely errored seconds or as unavailable seconds.

12.1.7 Mapping of SNMP objects to WIS station management registers

Some of the objects defined in this clause or incorporated by reference from the SONET-MIB IETF RFC 3592, or the MAU-MIB module defined in Clause 13 require WIS-specific hardware support. Subclause 50.3.11 of IEEE Std 802.3 specifies WIS management interface requirements, including a required subset of the WIS Management Data Input/Output (MDIO) registers defined in 45.2.2 of IEEE Std 802.3. Table 12-4 provides a cross-reference between those managed objects and the WIS MDIO registers from the subset in 50.3.11 of IEEE Std 802.3 required to support them. Note that the MDIO interface is optional; however, if it is not implemented, then the capabilities of the required register subset shall be provided by other means.

Table 12-4—Cross-reference between SNMP objects and WIS MDIO registers

SNMP object	WIS MDIO register(s)
ETHER-WIS - etherWisDeviceTxTestPatternMode	10G WIS control 2
ETHER-WIS - etherWisDeviceRxTestPatternMode	10G WIS control 2
ETHER-WIS - etherWisDeviceRxTestPatternErrors	10G WIS test pattern error counter
SONET-MIB - sonetMediumType	None required
SONET-MIB - sonetMediumTimeElapsed	None required
SONET-MIB - sonetMediumValidIntervals	None required
SONET-MIB - sonetMediumLineCoding	None required
SONET-MIB - sonetMediumLineType	None required

Table 12-4—Cross-reference between SNMP objects and WIS MDIO registers (continued)

SNMP object	WIS MDIO register(s)
SONET-MIB - sonetMediumCircuitIdentifier	None required
SONET-MIB - sonetMediumInvalidIntervals	None required
SONET-MIB - sonetMediumLoopbackConfig	None required
SONET-MIB - sonetSESthresholdSet	None required
ETHER-WIS - etherWisSectionCurrentJ0Transmitted	10G WIS J0 transmit
ETHER-WIS - etherWisSectionCurrentJ0Received	10G WIS J0 receive
SONET-MIB - sonetSectionCurrentStatus	10G WIS status 3
SONET-MIB - sonetSectionCurrentESs	10G WIS status 3 + 10G WIS section BIP error count
SONET-MIB - sonetSectionCurrentSESSs	
SONET-MIB - sonetSectionCurrentSEFSs	
SONET-MIB - sonetSectionCurrentCVs	
SONET-MIB - sonetSectionIntervalESs	
SONET-MIB - sonetSectionIntervalSESSs	
SONET-MIB - sonetSectionIntervalSEFSs	
SONET-MIB - sonetSectionIntervalCVs	
SONET-MIB - sonetSectionIntervalValidData	None required
SONET-MIB - sonetLineCurrentStatus	10G WIS status 3
SONET-MIB - sonetLineCurrentESs	10G WIS status 3 + 10G WIS line BIP errors
SONET-MIB - sonetLineCurrentSESSs	
SONET-MIB - sonetLineCurrentCVs	
SONET-MIB - sonetLineCurrentUASs	
SONET-MIB - sonetLineIntervalESs	
SONET-MIB - sonetLineIntervalSESSs	
SONET-MIB - sonetLineIntervalCVs	
SONET-MIB - sonetLineIntervalUASs	
SONET-MIB - sonetLineIntervalValidData	None required

Table 12-4—Cross-reference between SNMP objects and WIS MDIO registers (continued)

SNMP object	WIS MDIO register(s)
SONET-MIB - sonetFarEndLineCurrentESs	10G WIS status 3 + 10G WIS far end line BIP errors
SONET-MIB - sonetFarEndLineCurrentSEsSs	
SONET-MIB - sonetFarEndLineCurrentCVs	
SONET-MIB - sonetFarEndLineCurrentUASs	
SONET-MIB - sonetFarEndLineIntervalESs	
SONET-MIB - sonetFarEndLineIntervalSEsSs	
SONET-MIB - sonetFarEndLineIntervalCVs	
SONET-MIB - sonetFarEndLineIntervalUASs	10G WIS status 3
SONET-MIB - sonetFarEndLineIntervalValidData	
ETHER-WIS - etherWisPathCurrentStatus	10G WIS status 3
ETHER-WIS - etherWisPathCurrentJ1Transmitted	10G WIS J1 transmit
ETHER-WIS - etherWisPathCurrentJ1Received	10G WIS J1 receive
SONET-MIB - sonetPathCurrentWidth	None required
SONET-MIB - sonetPathCurrentStatus	10G WIS status 3
SONET-MIB - sonetPathCurrentESs	10G WIS status 3 + 10G WIS path block error count
SONET-MIB - sonetPathCurrentSEsSs	
SONET-MIB - sonetPathCurrentCVs	
SONET-MIB - sonetPathCurrentUASs	
SONET-MIB - sonetPathIntervalESs	
SONET-MIB - sonetPathIntervalCVs	
SONET-MIB - sonetPathIntervalUASs	None required
SONET-MIB - sonetPathIntervalValidData	
ETHER-WIS - etherWisFarEndPathCurrentStatus	10G WIS status 3

Table 12-4—Cross-reference between SNMP objects and WIS MDIO registers (continued)

SNMP object	WIS MDIO register(s)
SONET-MIB - sonetFarEndPathCurrentESs	10G WIS status 3 + 10G WIS far end path block error count
SONET-MIB - sonetFarEndPathCurrentSESs	
SONET-MIB - sonetFarEndPathCurrentCVs	
SONET-MIB - sonetFarEndPathCurrentUASs	
SONET-MIB - sonetFarEndPathIntervalESs	
SONET-MIB - sonetFarEndPathIntervalSESs	
SONET-MIB - sonetFarEndPathIntervalCVs	
SONET-MIB - sonetFarEndPathIntervalUASs	
SONET-MIB - sonetFarEndPathIntervalValidData	
MAU-MIB - ifMauIfIndex	None required
MAU-MIB - ifMauIndex	None required
MAU-MIB - ifMauType	10G WIS control 2
MAU-MIB - ifMauStatus	WIS control 1
MAU-MIB - ifMauMediaAvailable	WIS status 1 + 10G WIS status 3
MAU-MIB - ifMauMediaAvailableStateExits	
MAU-MIB - ifMauJabberState	None required
MAU-MIB - ifMauJabberingStateEnters	None required
MAU-MIB - ifMauFalseCarriers	None required
MAU-MIB - ifMauDefaultType	10G WIS control 2
MAU-MIB - ifMauAutoNegSupported	none required
MAU-MIB - ifMauTypeListBits	10G WIS status 2

12.1.8 Structure of the MIB module

Four tables are defined in this MIB module.

12.1.8.1 etherWisDeviceTable

The purpose of this table is to define managed objects to control the WIS test pattern mode. These objects are required to support mandatory and optional WIS test features specified in 50.3.8 of IEEE Std 802.3.

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

12.1.8.2 etherWisSectionCurrentTable

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

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

12.1.8.3 etherWisPathCurrentTable

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

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

12.1.8.4 etherWisFarEndPathCurrentTable

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

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

12.2 Security considerations for Ethernet wide area network (WAN) interface sublayer (WIS) MIB module

There are five managed objects defined in this MIB module that have a MAX-ACCESS clause of read-write: (1) etherWisDeviceTxTestPatternMode, (2) etherWisDeviceRxTestPatternMode, (3) etherWisDeviceRxTestPatternErrors, (4) etherWisSectionCurrentJ0Transmitted, and (5) etherWisPathCurrentJ1Transmitted. Writing to these objects can have the following potentially disruptive effects on network operation:

- Changing the transmit or receive test pattern mode or modifying the accumulated error count from a PRBS31 pattern test on an administratively disabled 10GBASE-W interface, which can interfere with an in-progress pattern test.
- Modifying the transmitted section trace and/or path trace message on an operational 10GBASE-W interface, which can cause connectivity alarms to be raised at the remote of the link.

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

The readable objects in this MIB module (i.e., those with MAX-ACCESS other than not-accessible) may be considered sensitive in some environments since, collectively, they provide information about the performance of network interfaces and can reveal some aspects of their configuration. In such environments

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

12.3 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL²⁰:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C12mib.txt

²⁰Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
1  IEEE8023-ETHER-WIS-MIB DEFINITIONS ::= BEGIN
2
3      IMPORTS
4          MODULE-IDENTITY, OBJECT-TYPE,
5          Gauge32, org
6          FROM SNMPv2-SMI
7          ifIndex
8          FROM IF-MIB
9      MODULE-COMPLIANCE, OBJECT-GROUP
10         FROM SNMPv2-CONF
11         sonetMediumStuff2, sonetSectionStuff2,
12         sonetLineStuff2, sonetFarEndLineStuff2,
13         sonetPathStuff2, sonetFarEndPathStuff2,
14         sonetMediumType, sonetMediumLineCoding,
15         sonetMediumLineType, sonetMediumCircuitIdentifier,
16         sonetMediumLoopbackConfig, sonetSESthresholdSet,
17         sonetPathCurrentWidth
18         FROM SONET-MIB;
19
20  ieee8023etherWisMIB MODULE-IDENTITY
21      LAST-UPDATED "201304110000Z" -- April 11, 2013
22      ORGANIZATION
23          "IEEE 802.3 working group"
24      CONTACT-INFO
25          "WG-URL: http://www.ieee802.org/3/index.html
26          WG-E-Mail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
27
28          Contact: Howard Frazier
29          Postal: 3151 Zanker Road
30                  San Jose, CA 95134
31                  USA
32          Tel:      +1.408.922.8164
33          E-mail:    hfrazier@broadcom.com"
34
35  DESCRIPTION
36      "The objects in this MIB module are used in conjunction
37      with objects in the SONET-MIB module and the MAU-MIB module to manage
38      the Ethernet WAN Interface Sublayer (WIS) defined in
39      IEEE Std 802.3.
40
41      Of particular interest are Clause 50, 'WAN Interface
42      Sublayer (WIS), type 10GBASE-W', Clause 30, '10 Mb/s,
43      100 Mb/s, 1000 Mb/s, and 10 Gb/s Management, and Link
44      Aggregation Management', and Clause 45, 'Management
45      Data Input/Output (MDIO) Interface'."
46
47  REVISION "201304110000Z" -- April 11, 2013
48  DESCRIPTION
49      "Revision, based on an earlier version in IEEE Std 802.3.1-2011."
50
51  REVISION "201102020000Z" -- February 2, 2011
52  DESCRIPTION
53      "Initial version, based on an earlier version published
54      as RFC 3637."
55
56      ::= { org ieee(111) standards-association-numbers-series-standards(2)
57            lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 12 }
58
59  -- The main sections of the module
```

```

1
2     etherWisObjects      OBJECT IDENTIFIER ::= { ieee8023etherWisMIB 1 }
3
4     etherWisObjectsPath OBJECT IDENTIFIER ::= { ieee8023etherWisMIB 2 }
5
6     etherWisConformance OBJECT IDENTIFIER ::= { ieee8023etherWisMIB 3 }
7
8     -- groups in the Ethernet WIS MIB module
9
10
11     etherWisDevice      OBJECT IDENTIFIER ::= { etherWisObjects 1 }
12
13     etherWisSection     OBJECT IDENTIFIER ::= { etherWisObjects 2 }
14
15     etherWisPath        OBJECT IDENTIFIER ::= { etherWisObjectsPath 1 }
16
17     etherWisFarEndPath  OBJECT IDENTIFIER ::= { etherWisObjectsPath 2 }
18
19     -- The Device group
20
21     -- These objects provide WIS extensions to
22     -- the SONET-MIB Medium Group.
23
24
25     etherWisDeviceTable OBJECT-TYPE
26         SYNTAX SEQUENCE OF EtherWisDeviceEntry
27         MAX-ACCESS not-accessible
28         STATUS current
29         DESCRIPTION
30             "The table for Ethernet WIS devices"
31             ::= { etherWisDevice 1 }
32
33
34     etherWisDeviceEntry OBJECT-TYPE
35         SYNTAX EtherWisDeviceEntry
36         MAX-ACCESS not-accessible
37         STATUS current
38         DESCRIPTION
39             "An entry in the Ethernet WIS device table. For each
40             instance of this object there shall be a corresponding
41             instance of sonetMediumEntry."
42         INDEX { ifIndex }
43         ::= { etherWisDeviceTable 1 }
44
45
46     EtherWisDeviceEntry ::=
47         SEQUENCE {
48             etherWisDeviceTxTestPatternMode    INTEGER,
49             etherWisDeviceRxTestPatternMode    INTEGER,
50             etherWisDeviceRxTestPatternErrors  Gauge32
51         }
52
53
54     etherWisDeviceTxTestPatternMode OBJECT-TYPE
55         SYNTAX INTEGER {
56             none(1),
57             squareWave(2),
58             prbs31(3),
59             mixedFrequency(4)
60         }
61         MAX-ACCESS read-write
62         STATUS current
63         DESCRIPTION
64             "This variable controls the transmit test pattern mode.
65
```

```
1      The value none(1) puts the the WIS transmit path into
2      the normal operating mode. The value squareWave(2) puts
3      the WIS transmit path into the square wave test pattern
4      mode described in IEEE Std 802.3, 50.3.8.1.
5      The value prbs31(3) puts the WIS transmit path into the
6      PRBS31 test pattern mode described in IEEE Std 802.3
7      50.3.8.2. The value mixedFrequency(4) puts the
8      WIS transmit path into the mixed frequency test pattern
9      mode described in IEEE Std 802.3, 50.3.8.3.
10     Any attempt to set this object to a value other than
11     none(1) when the corresponding instance of ifAdminStatus
12     has the value up(1) shall be rejected with the error
13     inconsistentValue, and any attempt to set the corresponding
14     instance of ifAdminStatus to the value up(1) when an
15     instance of this object has a value other than none(1)
16     shall be rejected with the error inconsistentValue."
17
18     REFERENCE
19         "IEEE Std 802.3, 50.3.8, WIS test pattern generator and
20         checker, 45.2.2.6, 10G WIS control 2 register (2.7), and
21         45.2.2.7.2, PRBS31 pattern testing ability (2.8.1)."
```

```
22     ::= { etherWisDeviceEntry 1 }
23
24
25     etherWisDeviceRxTestPatternMode OBJECT-TYPE
26         SYNTAX  INTEGER {
27             none(1),
28             prbs31(3),
29             mixedFrequency(4)
30         }
31         MAX-ACCESS  read-write
32         STATUS  current
33         DESCRIPTION
34             "This variable controls the receive test pattern mode.
35             The value none(1) puts the the WIS receive path into the
36             normal operating mode. The value prbs31(3) puts the WIS
37             receive path into the PRBS31 test pattern mode described
38             in IEEE Std 802.3, 50.3.8.2. The value
39             mixedFrequency(4) puts the WIS receive path into the mixed
40             frequency test pattern mode described in IEEE Std 802.3,
41             50.3.8.3. Any attempt to set this object to a
42             value other than none(1) when the corresponding instance
43             of ifAdminStatus has the value up(1) shall be rejected with
44             the error inconsistentValue, and any attempt to set the
45             corresponding instance of ifAdminStatus to the value up(1)
46             when an instance of this object has a value other than
47             none(1) shall be rejected with the error inconsistentValue."
48
49     REFERENCE
50         "IEEE Std 802.3, 50.3.8, WIS test pattern generator and
51         checker, 45.2.2.6, 10G WIS control 2 register (2.7), and
52         45.2.2.7.2, PRBS31 pattern testing ability (2.8.1)."
```

```
53     ::= { etherWisDeviceEntry 2 }
54
55
56     etherWisDeviceRxTestPatternErrors OBJECT-TYPE
57         SYNTAX  Gauge32 ( 0..65535 )
58         MAX-ACCESS  read-write
59         STATUS  current
60         DESCRIPTION
61             "This object counts the number of errors detected when the
62             WIS receive path is operating in the PRBS31 test pattern
63             mode. It is reset to zero when the WIS receive path
64
65
```

```
1         initially enters that mode, and it increments each time
2         the PRBS pattern checker detects an error as described in
3         IEEE Std 802.3, 50.3.8.2 unless its value is
4         65535, in which case it remains unchanged. This object is
5         writeable so that it may be reset upon explicit request
6         of a command generator application while the WIS receive
7         path continues to operate in PRBS31 test pattern mode."
8
9     REFERENCE
10        "IEEE Std 802.3, 50.3.8, WIS test pattern generator and
11        checker, 45.2.2.7.2, PRBS31 pattern testing ability
12        (2.8.1), and 45.2.2.8, 10G WIS test pattern error counter
13        register (2.9)."
```

```
14    ::= { etherWisDeviceEntry 3 }
15
16  -- The Section group
17
18  -- These objects provide WIS extensions to
19  -- the SONET-MIB Section Group.
20
21  etherWisSectionCurrentTable OBJECT-TYPE
22      SYNTAX  SEQUENCE OF EtherWisSectionCurrentEntry
23      MAX-ACCESS not-accessible
24      STATUS  current
25      DESCRIPTION
26          "The table for the current state of Ethernet WIS sections."
27      ::= { etherWisSection 1 }
28
29  etherWisSectionCurrentEntry OBJECT-TYPE
30      SYNTAX  EtherWisSectionCurrentEntry
31      MAX-ACCESS not-accessible
32      STATUS  current
33      DESCRIPTION
34          "An entry in the etherWisSectionCurrentTable. For each
35          instance of this object there shall be a corresponding
36          instance of sonetSectionCurrentEntry."
37      INDEX   { ifIndex }
38      ::= { etherWisSectionCurrentTable 1 }
39
40
41  EtherWisSectionCurrentEntry ::=
42      SEQUENCE {
43          etherWisSectionCurrentJ0Transmitted OCTET STRING,
44          etherWisSectionCurrentJ0Received   OCTET STRING
45      }
46
47
48  etherWisSectionCurrentJ0Transmitted OBJECT-TYPE
49      SYNTAX  OCTET STRING (SIZE (16))
50      MAX-ACCESS read-write
51      STATUS  current
52      DESCRIPTION
53          "This is the 16-octet section trace message that
54          is transmitted in the J0 byte. The value should
55          be '89'h followed by fifteen octets of '00'h
56          (or some cyclic shift thereof) when the section
57          trace function is not used, and the implementation
58          should use that value (or a cyclic shift thereof)
59          as a default if no other value has been set."
60
61  REFERENCE
62      "IEEE Std 802.3, 30.8.1.1.8, aJ0ValueTX."
63  ::= { etherWisSectionCurrentEntry 1 }
64
65
```

```
1      etherWisSectionCurrentJ0Received OBJECT-TYPE
2          SYNTAX OCTET STRING (SIZE (16))
3          MAX-ACCESS read-only
4          STATUS current
5          DESCRIPTION
6              "This is the 16-octet section trace message that
7              was most recently received in the J0 byte."
8          REFERENCE
9              "IEEE Std 802.3, 30.8.1.1.9, aJ0ValueRX."
10             ::= { etherWisSectionCurrentEntry 2 }
11
12
13      -- The Path group
14
15      -- These objects provide WIS extensions to
16      -- the SONET-MIB Path Group.
17
18      etherWisPathCurrentTable OBJECT-TYPE
19          SYNTAX SEQUENCE OF EtherWisPathCurrentEntry
20          MAX-ACCESS not-accessible
21          STATUS current
22          DESCRIPTION
23              "The table for the current state of Ethernet WIS paths."
24             ::= { etherWisPath 1 }
25
26
27      etherWisPathCurrentEntry OBJECT-TYPE
28          SYNTAX EtherWisPathCurrentEntry
29          MAX-ACCESS not-accessible
30          STATUS current
31          DESCRIPTION
32              "An entry in the etherWisPathCurrentTable. For each
33              instance of this object there shall be a corresponding
34              instance of sonetPathCurrentEntry."
35          INDEX { ifIndex }
36          ::= { etherWisPathCurrentTable 1 }
37
38
39      EtherWisPathCurrentEntry ::=
40          SEQUENCE {
41              etherWisPathCurrentStatus          BITS,
42              etherWisPathCurrentJ1Transmitted    OCTET STRING,
43              etherWisPathCurrentJ1Received       OCTET STRING
44          }
45
46      etherWisPathCurrentStatus OBJECT-TYPE
47          SYNTAX BITS {
48              etherWisPathLOP(0),
49              etherWisPathAIS(1),
50              etherWisPathPLM(2),
51              etherWisPathLCD(3)
52          }
53
54          MAX-ACCESS read-only
55          STATUS current
56          DESCRIPTION
57              "This variable indicates the current status of the
58              path payload with a bit map that can indicate multiple
59              defects at once. The bit positions are assigned as
60              follows:
61
62              etherWisPathLOP(0)
63                  This bit is set to indicate that an
64                  LOP-P (Loss of Pointer - Path) defect
```



```
1         is being experienced. When this
2         bit is set, sonetPathSTSLOP shall be set
3         in the corresponding instance of
4         sonetPathCurrentStatus.
5
6     etherWisPathAIS(1)
7         This bit is set to indicate that an
8         AIS-P (Alarm Indication Signal - Path)
9         defect is being experienced. When
10        this bit is set, sonetPathSTS AIS shall be
11        set in the corresponding instance of
12        sonetPathCurrentStatus.
13
14
15    etherWisPathPLM(1)
16        This bit is set to indicate that a
17        PLM-P (Payload Label Mismatch - Path)
18        defect is being experienced. When
19        this bit is set, sonetPathSignalLabelMismatch
20        shall be set in the corresponding instance of
21        sonetPathCurrentStatus.
22
23    etherWisPathLCD(3)
24        This bit is set to indicate that an
25        LCD-P (Loss of Codegroup Delination - Path)
26        defect is being experienced. Since this
27        defect is detected by the PCS and not by
28        the path layer itself, there is no
29        corresponding bit in sonetPathCurrentStatus."
30
31    REFERENCE
32        "IEEE Std 802.3, 30.8.1.1.18, aPathStatus."
33        ::= { etherWisPathCurrentEntry 1 }
34
35    etherWisPathCurrentJ1Transmitted OBJECT-TYPE
36        SYNTAX OCTET STRING (SIZE (16))
37        MAX-ACCESS read-write
38        STATUS current
39        DESCRIPTION
40            "This is the 16-octet path trace message that
41            is transmitted in the J1 byte. The value should
42            be '89'h followed by fifteen octets of '00'h
43            (or some cyclic shift thereof) when the path
44            trace function is not used, and the implementation
45            should use that value (or a cyclic shift thereof)
46            as a default if no other value has been set."
47
48    REFERENCE
49        "IEEE Std 802.3, 30.8.1.1.23, aJ1ValueTX."
50        ::= { etherWisPathCurrentEntry 2 }
51
52
53    etherWisPathCurrentJ1Received OBJECT-TYPE
54        SYNTAX OCTET STRING (SIZE (16))
55        MAX-ACCESS read-only
56        STATUS current
57        DESCRIPTION
58            "This is the 16-octet path trace message that
59            was most recently received in the J1 byte."
60
61    REFERENCE
62        "IEEE Std 802.3, 30.8.1.1.24, aJ1ValueRX."
63        ::= { etherWisPathCurrentEntry 3 }
64
65    -- The Far End Path group
```

```
1      -- These objects provide WIS extensions to
2      -- the SONET-MIB Far End Path Group.
3
4      etherWisFarEndPathCurrentTable OBJECT-TYPE
5          SYNTAX SEQUENCE OF EtherWisFarEndPathCurrentEntry
6          MAX-ACCESS not-accessible
7          STATUS current
8          DESCRIPTION
9              "The table for the current far-end state of Ethernet WIS
10             paths."
11             ::= { etherWisFarEndPath 1 }
12
13
14      etherWisFarEndPathCurrentEntry OBJECT-TYPE
15          SYNTAX EtherWisFarEndPathCurrentEntry
16          MAX-ACCESS not-accessible
17          STATUS current
18          DESCRIPTION
19              "An entry in the etherWisFarEndPathCurrentTable. For each
20              instance of this object there shall be a corresponding
21              instance of sonetFarEndPathCurrentEntry."
22          INDEX { ifIndex }
23          ::= { etherWisFarEndPathCurrentTable 1 }
24
25
26      EtherWisFarEndPathCurrentEntry ::=
27          SEQUENCE {
28              etherWisFarEndPathCurrentStatus      BITS
29          }
30
31
32      etherWisFarEndPathCurrentStatus OBJECT-TYPE
33          SYNTAX BITS {
34              etherWisFarEndPayloadDefect(0),
35              etherWisFarEndServerDefect(1)
36          }
37          MAX-ACCESS read-only
38          STATUS current
39          DESCRIPTION
40              "This variable indicates the current status at the
41              far end of the path using a bit map that can indicate
42              multiple defects at once. The bit positions are
43              assigned as follows:
44
45              etherWisFarEndPayloadDefect(0)
46                  A far end payload defect (i.e., far end
47                  PLM-P or LCD-P) is currently being signaled
48                  in G1 bits 5-7.
49
50              etherWisFarEndServerDefect(1)
51                  A far end server defect (i.e., far end
52                  LOP-P or AIS-P) is currently being signaled
53                  in G1 bits 5-7. When this bit is set,
54                  sonetPathSTSRDI shall be set in the corresponding
55                  instance of sonetPathCurrentStatus."
56          REFERENCE
57              "IEEE Std 802.3, 30.8.1.1.25, aFarEndPathStatus."
58          ::= { etherWisFarEndPathCurrentEntry 1 }
59
60      --
61      -- Conformance Statements
62      --
```

```
1
2 etherWisGroups      OBJECT IDENTIFIER ::= { etherWisConformance 1 }
3
4 etherWisCompliances OBJECT IDENTIFIER ::= { etherWisConformance 2 }
5
6 --      Object Groups
7
8 etherWisDeviceGroupBasic OBJECT-GROUP
9     OBJECTS {
10         etherWisDeviceTxTestPatternMode,
11         etherWisDeviceRxTestPatternMode
12     }
13     STATUS current
14     DESCRIPTION
15         "A collection of objects that support test
16         features required of all WIS devices."
17     ::= { etherWisGroups 1 }
18
19
20 etherWisDeviceGroupExtra OBJECT-GROUP
21     OBJECTS {
22         etherWisDeviceRxTestPatternErrors
23     }
24     STATUS current
25     DESCRIPTION
26         "A collection of objects that support
27         optional WIS device test features."
28     ::= { etherWisGroups 2 }
29
30 etherWisSectionGroup OBJECT-GROUP
31     OBJECTS {
32         etherWisSectionCurrentJ0Transmitted,
33         etherWisSectionCurrentJ0Received
34     }
35     STATUS current
36     DESCRIPTION
37         "A collection of objects that provide
38         required information about a WIS section."
39     ::= { etherWisGroups 3 }
40
41
42 etherWisPathGroup OBJECT-GROUP
43     OBJECTS {
44         etherWisPathCurrentStatus,
45         etherWisPathCurrentJ1Transmitted,
46         etherWisPathCurrentJ1Received
47     }
48     STATUS current
49     DESCRIPTION
50         "A collection of objects that provide
51         required information about a WIS path."
52     ::= { etherWisGroups 4 }
53
54
55 etherWisFarEndPathGroup OBJECT-GROUP
56     OBJECTS {
57         etherWisFarEndPathCurrentStatus
58     }
59     STATUS current
60     DESCRIPTION
61         "A collection of objects that provide required
62         information about the far end of a WIS path."
63     ::= { etherWisGroups 5 }
64
65
```

```

1
2      --      Compliance Statements
3
4  etherWisCompliance MODULE-COMPLIANCE
5      STATUS      current
6      DESCRIPTION
7          "The compliance statement for interfaces that include
8          the Ethernet WIS. Compliance with the following
9          external compliance statements is prerequisite:
10
11
12      MIB module      Compliance Statement
13      -----
14      IF-MIB          ifCompliance3
15      IF-INVERTED-STACK-MIB  ifInvCompliance
16      IEEE8023-EtherLike-MIB  dot3Compliance2
17      MAU-MIB          mauModIfCompl3"
18  MODULE -- this module
19      MANDATORY-GROUPS {
20          etherWisDeviceGroupBasic,
21          etherWisSectionGroup,
22          etherWisPathGroup,
23          etherWisFarEndPathGroup
24      }
25
26
27      OBJECT          etherWisDeviceTxTestPatternMode
28      SYNTAX          INTEGER {
29          none(1),
30          squareWave(2),
31          mixedFrequency(4)
32      }
33
34      DESCRIPTION
35          "Support for values other than none(1),
36          squareWave(2), and mixedFrequency(4)
37          is not required."
38
39
40      OBJECT          etherWisDeviceRxTestPatternMode
41      SYNTAX          INTEGER {
42          none(1),
43          mixedFrequency(4)
44      }
45
46      DESCRIPTION
47          "Support for values other than none(1)
48          and mixedFrequency(4) is not required."
49
50      GROUP          etherWisDeviceGroupExtra
51      DESCRIPTION
52          "Implementation of this group, along with support for
53          the value prbs31(3) for etherWisDeviceTxTestPatternMode
54          and etherWisDeviceRxTestPatternMode, is necessary if the
55          optional PRBS31 test pattern mode is to be supported."
56
57      OBJECT          etherWisDeviceRxTestPatternErrors
58      WRITE-SYNTAX Gauge32 ( 0 )
59      DESCRIPTION
60          "An implementation is not required to
61          allow values other than zero to be
62          written to this object."
63
64  MODULE SONET-MIB
65      MANDATORY-GROUPS {

```

```
1          sonetMediumStuff2,  
2          sonetSectionStuff2,  
3          sonetLineStuff2,  
4          sonetFarEndLineStuff2,  
5          sonetPathStuff2,  
6          sonetFarEndPathStuff2  
7      }  
8  
9  
10     OBJECT      sonetMediumType  
11     SYNTAX      INTEGER {  
12         sonet(1)  
13     }  
14     MIN-ACCESS   read-only  
15     DESCRIPTION  
16         "Write access is not required, nor is support  
17         for any value other than sonet(1)."  
18  
19  
20     OBJECT      sonetMediumLineCoding  
21     SYNTAX      INTEGER {  
22         sonetMediumNRZ(4)  
23     }  
24     MIN-ACCESS   read-only  
25     DESCRIPTION  
26         "Write access is not required, nor is support  
27         for any value other than sonetMediumNRZ(4)."  
28  
29  
30     OBJECT      sonetMediumLineType  
31     MIN-ACCESS   read-only  
32     DESCRIPTION  
33         "Write access is not required."  
34  
35  
36     OBJECT      sonetMediumCircuitIdentifier  
37     MIN-ACCESS   read-only  
38     DESCRIPTION  
39         "Write access is not required."  
40  
41  
42     OBJECT      sonetMediumLoopbackConfig  
43     SYNTAX      BITS {  
44         sonetNoLoop(0),  
45         sonetFacilityLoop(1)  
46     }  
47     MIN-ACCESS   read-only  
48     DESCRIPTION  
49         "Write access is not required, nor is support for values  
50         other than sonetNoLoop(0) and sonetFacilityLoop(1)."  
51  
52  
53     OBJECT      sonetSESthresholdSet  
54     MIN-ACCESS   read-only  
55     DESCRIPTION  
56         "Write access is not required, and only one  
57         of the enumerated values need be supported."  
58  
59  
60     OBJECT      sonetPathCurrentWidth  
61     SYNTAX      INTEGER {  
62         sts192cSTM64(6)  
63     }  
64     MIN-ACCESS   read-only  
65     DESCRIPTION  
        "Write access is not required, nor is support
```

```
1           for any value other than sts192cSTM64(6)."  
2  
3       ::= { etherWisCompliances 1 }  
4  
5   END  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

13. Ethernet medium attachment units (MAUs) MIB module

13.1 Introduction

This clause defines a portion of the MIB for use with SNMP. In particular, it defines objects for managing IEEE 802.3 medium attachment units (MAUs).

A previous version of this clause, IETF RFC 3636 [B28], defined a single MIB module. IETF RFC 4836 [B35] split the original MIB module into two, putting frequently updated object identities and textual conventions into a separate, IANA-maintained MIB module, in order to decrease the need of updating the basic MAU-MIB module. The MIB module defined in this clause incorporates the IANA-MAU-MIB module by reference.

13.2 Overview

Instances of these object types represent attributes of an IEEE 802.3 MAU. Several types of MAUs are defined in IEEE Std 802.3. These MAUs may be connected to IEEE 802.3 repeaters or to IEEE 802.3 (Ethernet-like) interfaces. For convenience, this clause refers to these devices as “repeater MAUs” and “interface MAUs.”

The definitions presented here are based on 30.5 and 30.6 of IEEE Std 802.3. This specification is intended to provide for management of all types of Ethernet/IEEE 802.3 MAUs.

13.2.1 Relationship to IETF RFC 3636 and IETF RFC 4836

The management definitions provided in this clause are intended to be a superset of those defined by IETF RFC 3636 [B28] and IETF RFC 4836 [B35].

In order to decrease the need of updating the basic MAU MIB module due to the new MAU type, Media Available state, Auto Negotiation capability, and/or Jack type introduction, all relevant object identities and textual conventions have been moved to a separate, IANA-maintained MIB module, IANA-MAU-MIB. Thus, when a new MAU type, Media Available state, Auto Negotiation capability, and/or Jack type is defined by the IEEE 802.3 working group, only the IANA-maintained module needs to be revised, leaving the basic MAU-MIB module defined in this clause unchanged.

The changes made in this revision are not entirely backward-compatible with MIB modules that currently import MAU-type object identity descriptors from the MAU-MIB; such modules need to be revised to import those DESCRIPTORS from the IANA-MAU-MIB. Similarly, any management applications that process the object identity definitions (e.g., to present the DESCRIPTION text to a user) need to get those definitions from the IANA-MAU-MIB instead of the MAU-MIB. While it is true that changes that require such adjustments are not strictly compliant with the SMIV2 rules governing MIB module revisions (see Section 10 of IETF STD 58, RFC 2578), in this case continued high maintenance costs that would result from not making these changes make the deviation from the rules justified.

13.2.2 Relationship to other MIBs

It is assumed that an agent implementing the MAU-MIB module will also implement (at least) the “system” group defined in the SNMPv2 MIB of IETF RFC 3418 [B26]. The following subclauses identify other MIBs that such an agent should implement.

13.2.2.1 Relationship to the Interfaces Group MIB

The subclauses of this clause that define interface MAU-related objects specify an extension to the Interfaces Group MIB of IETF RFC 2863. An agent implementing these interface-MAU related objects shall also implement the relevant groups of the ifCompliance3 MODULE-COMPLIANCE statement of the Interfaces Group MIB. The value of the object ifMauIfIndex is the same as the value of “ifIndex” used to instantiate the interface to which the given MAU is connected.

An agent implementing the interface-MAU related objects in the MAU-MIB module shall also fully comply with the dot3Compliance2 MODULE-COMPLIANCE statement of the Ethernet-like interface MIB defined in Clause 10. Furthermore, when the interface-MAU related objects are used to manage a 10GBASE-W PHY—i.e., when ifMauType is equal to dot3MauType10GigBaseW or any other 10GBASE-W variant—then the agent shall also support the Ethernet WAN Interface Sublayer (WIS) MIB module defined in Clause 12, and shall follow the interface layering model specified therein. In that case, the value of the object ifMauIfIndex is the same as the value of “ifIndex” for the layer at the top of the stack, i.e., for the ifTable entry that has “ifType” equal to ethernetCsmacd(6). If the interface-MAU related objects are used to manage a PHY that allows the MAU type to be changed dynamically, then the agent shall create ifTable, ifStackTable, and ifInvStackTable entries that pertain to the WIS when ifMauDefaultType is changed to a 10GBASEW variant (i.e., one of dot3MauType10GigBaseW, dot3MauType10GigBaseEW, dot3MauType10GigBaseLW, or dot3MauType10GigBaseSW) from any other type, and shall destroy the WIS-related entries when ifMauDefaultType is changed to a non-10GBASE-W type. The agent shall also change the values of “ifConnectorPresent” and “ifHighSpeed” in the ifTable entry indexed by ifMauIfIndex as specified in Clause 10 and Clause 12 when ifMauDefaultType is manipulated in this way, but shall NOT otherwise alter that entry.

NOTE—Repeater ports are not represented as interfaces in the Interfaces Group MIB.

13.2.2.2 Relationship to the IEEE 802.3 repeater MIB module

The subclause of this clause that defines repeater MAU-related objects specifies an extension to the IEEE 802.3 repeater MIB module defined in Clause 7. An agent implementing these repeater-MAU related objects shall also comply with the snmpRpTrModCompl compliance statement of the IEEE 802.3 repeater MIB module.

The values of “rpMauGroupIndex” and “rpMauPortIndex” used to instantiate a repeater MAU variable shall be the same as the values of “rpTrPortGroupIndex” and “rpTrPortIndex” used to instantiate the port to which the given MAU is connected.

13.2.3 Management of internal MAUs

In some situations, a MAU can be “internal”; i.e., its functionality is implemented entirely within a device. For example, a managed repeater may contain an internal repeater-MAU and/or an internal interface-MAU through which management communications originating on one of the repeater’s external ports pass, in order to reach the management agent associated with the repeater. Such internal MAUs may or may not be managed. If they are managed, objects describing their attributes should appear in the appropriate MIB subtree: dot3RpMauBasicGroup for internal repeater-MAUs and dot3IfMauBasicGroup for internal interface-MAUs.

13.2.4 Mapping of IEEE 802.3 managed objects

Table 13-1 depicts the mapping between relevant managed objects (attributes) defined in Clause 30 of IEEE Std 802.3 and managed objects defined in this clause.

Table 13-1—Mapping of IEEE 802.3 managed objects

IEEE 802.3 managed object		Corresponding SNMP object
oMAU	.aMAUID	rpMauIndex or ifMauIndex or broadMauIndex
	.aMAUType	rpMauType or ifMauType
	.aMAUTypeList	ifMauTypeListBits
	.aMediaAvailable	rpMauMediaAvailable or ifMauMediaAvailable
	.aLoseMediaCounter	rpMauMediaAvailableStateExits or ifMauMediaAvailableStateExits
	.aJabber	rpMauJabberState and rpMauJabberingStateEnters or ifMauJabberState and ifMauJabberingStateEnters
	.aMAUAdminState	rpMauStatus or ifMauStatus
	.aFalseCarriers	rpMauFalseCarriers or ifMauFalseCarriers
	.acResetMAU	rpMauStatus or ifMauStatus
	.acMAUAdminControl	rpMauStatus or ifMauStatus
	.nJabber	rpMauJabberTrap or ifMauJabberTrap

Table 13-1—Mapping of IEEE 802.3 managed objects (continued)

IEEE 802.3 managed object		Corresponding SNMP object
oAutoNegotiation	.aAutoNegID	ifMauIndex
	.aAutoNegAdminState	ifMauAutoNegAdminStatus
	.aAutoNegRemoteSignalling	ifMauAutoNegRemoteSignalling
	.aAutoNegAutoConfig	ifMauAutoNegConfig
	.aAutoNegLocalTechnologyAbility	ifMauAutoNegCapabilityBits
	.aAutoNegAdvertisedTechnologyAbility	ifMauAutoNegAdvertisedBits and ifMauAutoNegRemoteFaultAdvertised
	.aAutoNegReceivedTechnologyAbility	ifMauAutoNegReceivedBits and ifMauAutoNegRemoteFaultReceived
	.acAutoNegRestartAutoConfig	ifMauAutoNegRestart
	.acAutoNegAdminControl	ifMauAutoNegAdminStatus
oTimeSync	.aTimeSyncCapabilityTX	ifMauTimeSyncCapabilityTX
	.aTimeSyncCapabilityRX	ifMauTimeSyncCapabilityRX
	.aTimeSyncDelayTXmax	ifMauTimeSyncDelayTXmax
	.aTimeSyncDelayTXmin	ifMauTimeSyncDelayTXmin
	.aTimeSyncDelayRXmax	ifMauTimeSyncDelayRXmax
	.aTimeSyncDelayRXmin	ifMauTimeSyncDelayRXmin

Table 13-2 depicts the IEEE 802.3 managed objects that have not been included in the MAU-MIB module, and the reason for the exclusion.

Table 13-2—Unmapped IEEE 802.3 managed objects

IEEE 802.3 managed object		Reason for exclusion
oMAU	.aIdleErrorCount	Only useful for 100BaseT2, which is not widely implemented
oAutoNegotiation	.aAutoNegLocalSelectorAbility	Only needed for support of isoethernet (IEEE Std 802.9a-1995), which is not supported by MAU- MIB
	.aAutoNegAdvertisedSelectorAbility	
	.aAutoNegReceivedSelectorAbility	

13.2.5 Addition of new MAU types

13.2.5.1 dot3MauType OBJECT-IDENTITIES

The dot3MauType OBJECT IDENTIFIER and its OBJECT-IDENTITY definitions have been moved from the MAU-MIB module into the IANA-maintained IANA-MAU-MIB module.

When a new IEEE 802.3 MAU is defined, IANA can reissue a version of the IANA-MAU-MIB module with the new dot3MauType OBJECT-IDENTITY and its matching IANAifMauTypeListBits textual convention value and, possibly, with new IANAifMauMediaAvailable, IANAifMauAutoNegCapBits, and/or IANAifJackType values.

An Expert Review, as defined in IETF RFC 2434, is required for the addition of the new MAU, Media Available states, Auto Negotiation capabilities, and/or Jack types.

In some cases, new MAU types may require additional managed objects or may have side effects on the behavior of existing managed objects. In such cases, a standards-track specification (which may be a new document or a revision of this document) is also required. Any such document is required to note any special properties of the MAU types that it defines—for example, side effects on the ifStackTable as noted in this document for 10GBASE-W MAUs.

13.2.5.2 IANAifMauTypeListBits TEXTUAL-CONVENTION

The syntax of ifMauTypeListBits is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauTypeListBits, which can be respecified (with additional values, when defined by IEEE Std 802.3) in the IANA-maintained MIB module without issuing a new version of this document.

13.2.5.3 IANAifMauMediaAvailable TEXTUAL-CONVENTION

The syntax of ifMauMediaAvailable and rpMauMediaAvailable is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauMediaAvailable, which can be respecified (with additional values, when defined by IEEE Std 802.3) in the IANA-maintained MIB module without issuing a new version of this document.

13.2.5.4 IANAifMauAutoNegCapBits TEXTUAL-CONVENTION

The syntax of ifMauAutoNegCapabilityBits, ifMauAutoNegCapAdvertisedBits, and ifMauAutoNegCapReceivedBits objects is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauAutoNegCapBits, which can be respecified (with additional values, when defined by IEEE Std 802.3) in the IANA-maintained MIB module without issuing a new version of this document.

13.2.5.5 JackType TEXTUAL-CONVENTION

The JackType Textual Convention has been deprecated in favor of the IANAifJackType defined in the IANA-maintained MIB module, so the new Jack types can be added (when defined by IEEE Std 802.3) without issuing a new version of this document.

13.3 Security considerations for Ethernet medium attachment units (MAUs) MIB module

The IANA-MAU-MIB module does not define any management objects. Instead, it defines a set of textual conventions that are used by the MAU-MIB module and may be used by other MIB modules to define management objects. Meaningful security considerations can only be written for MIB modules that define management objects.

There are a number of management objects defined in the MAU-MIB module that have a MAX-ACCESS clause of read-write. Setting these objects can have a serious effect on the operation of the network, including:

- Enabling or disabling a MAU
- Changing a MAU's default type
- Enabling, disabling, or restarting autonegotiation
- Modifying the capabilities that a MAU advertises during autonegotiation.

Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

Some of the readable objects in the MAU-MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. In some environments, it may be undesirable to allow unauthorized parties to access statistics or status information about individual links in a network. It is thus important to control GET and/or NOTIFY access to these objects and possibly to encrypt the values of these objects when sending them over the network via SNMP.

13.4 IANA considerations

It is intended that each new MAU type, Media Available state, Auto Negotiation capability, and/or Jack type defined by the IEEE 802.3 working group and approved for publication in a revision of IEEE Std 802.3 will be added to the IANA-maintained MIB module, provided that it is suitable for being managed by the base objects in the MAU-MIB module.

For each new MAU type added, a short description of the MAU technology and, wherever possible, a reference to a publicly available specification should be specified. An Expert Review, as defined in IETF RFC 2434, is required, for each modification.

13.5 MIB module definition

An ASCII text version of the MIB definition can be found at the following URL²¹:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1C13mib.txt

The IANA-MAU-MIB module can be found at the following URL:

<http://www.iana.org/assignments/ianamau-mib>

²¹Copyright release for MIB modules: Users of this standard may freely reproduce the MIB module contained in this subclause so that it can be used for its intended purpose.

```
1  IEEE8023-MAU-MIB DEFINITIONS ::= BEGIN
2
3      IMPORTS
4          Counter32, Integer32, Counter64, Unsigned32,
5          OBJECT-TYPE, MODULE-IDENTITY, NOTIFICATION-TYPE, org
6          FROM SNMPv2-SMI          -- RFC 2578
7          TruthValue, AutonomousType
8          FROM SNMPv2-TC          -- RFC 2579
9          OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
10         FROM SNMPv2-CONF        -- RFC 2580
11         InterfaceIndex
12         FROM IF-MIB             -- RFC 2863
13         IANAifMauTypeListBits, IANAifMauMediaAvailable,
14         IANAifMauAutoNegCapBits, IANAifJackType
15         FROM IANA-MAU-MIB
16         -- http://www.iana.org/assignments/ianamau-mib
17     ;
18
19     ieee8023mauMIB MODULE-IDENTITY
20         LAST-UPDATED "201304110000Z" -- April 11, 2013
21         ORGANIZATION
22             "IEEE 802.3 working group"
23         CONTACT-INFO
24             "WG-URL: http://www.ieee802.org/3/index.html
25             WG-E-Mail: STDS-802-3-MIB@LISTSERV.IEEE.ORG
26
27             Contact: Howard Frazier
28             Postal: 3151 Zanker Road
29                 San Jose, CA 95134
30                 USA
31             Tel: +1.408.922.8164
32             E-mail: hfrazier@broadcom.com"
33
34     DESCRIPTION
35         "Management information for 802.3 MAUs."
36
37     REVISION      "201304110000Z" -- April 11, 2013
38     DESCRIPTION
39         "Revision, based on an earlier version in IEEE Std 802.3.1-2011."
40
41     REVISION      "201102020000Z" -- February 2, 2011
42     DESCRIPTION
43         "Initial version, based on an earlier version published
44         as RFC 4836."
45
46     ::= { org ieee(111) standards-association-numbers-series-standards(2)
47         lan-man-stds(802) ieee802dot3(3) ieee802dot3dot1mibs(1) 13 }
48
49     ieee8023snmpDot3MauMgt OBJECT IDENTIFIER ::= { ieee8023mauMIB 1 }
50
51     dot3RpMauBasicGroup
52         OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 1 }
53     dot3IfMauBasicGroup
54         OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 2 }
55     -- The following object is a placeholder
56     -- to preserve the arc assignments that follow it.
57     dot3PlaceholderGroup
58         OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 3 }
```

```

1      -- OIDs under the following branch are reserved for
2      -- the IANA-MAU-MIB to assign as MAU type values:
3      --                                     { ieee8023snmpDot3MauMgt 4 }
4
5      dot3IfMauAutoNegGroup
6          OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 5 }
7
8      --
9
10     -- The Basic Repeater MAU Table
11     --
12
13     rpMauTable OBJECT-TYPE
14         SYNTAX      SEQUENCE OF RpMauEntry
15         MAX-ACCESS   not-accessible
16         STATUS       current
17         DESCRIPTION  "Table of descriptive and status information
18                     about the MAU(s) attached to the ports of a
19                     repeater."
20         ::= { dot3RpMauBasicGroup 1 }
21
22
23     rpMauEntry OBJECT-TYPE
24         SYNTAX      RpMauEntry
25         MAX-ACCESS   not-accessible
26         STATUS       current
27         DESCRIPTION  "An entry in the table, containing information
28                     about a single MAU."
29         INDEX        { rpMauGroupIndex,
30                     rpMauPortIndex,
31                     rpMauIndex
32                     }
33         ::= { rpMauTable 1 }
34
35
36     RpMauEntry ::=
37         SEQUENCE {
38             rpMauGroupIndex      Integer32,
39             rpMauPortIndex       Integer32,
40             rpMauIndex           Integer32,
41             rpMauType            AutonomousType,
42             rpMauStatus          INTEGER,
43             rpMauMediaAvailable  IANAIfMauMediaAvailable,
44             rpMauMediaAvailableStateExits Counter32,
45             rpMauJabberState     INTEGER,
46             rpMauJabberingStateEnters Counter32,
47             rpMauFalseCarriers   Counter32
48         }
49
50
51
52     rpMauGroupIndex OBJECT-TYPE
53         SYNTAX      Integer32 (1..2147483647)
54         MAX-ACCESS   not-accessible
55         STATUS       current
56         DESCRIPTION  "This variable uniquely identifies the group
57                     containing the port to which the MAU described
58                     by this entry is connected.
59
60                     Note: In practice, a group will generally be
61                     a field-replaceable unit (i.e., module, card,
62                     or board) that can fit in the physical system
63                     enclosure, and the group number will correspond
64                     to a number marked on the physical enclosure.
65

```

```
1
2         The group denoted by a particular value of this
3         object is the same as the group denoted by the
4         same value of rpPtrGroupIndex."
5     REFERENCE    "RFC 2108, rpPtrGroupIndex."
6     ::= { rpMauEntry 1 }
7
8
9     rpMauPortIndex OBJECT-TYPE
10        SYNTAX      Integer32 (1..2147483647)
11        MAX-ACCESS   not-accessible
12        STATUS       current
13        DESCRIPTION  "This variable uniquely identifies the repeater
14                     port within group rpMauGroupIndex to which the
15                     MAU described by this entry is connected."
16        REFERENCE    "RFC 2108, rpPtrPortIndex."
17        ::= { rpMauEntry 2 }
18
19
20     rpMauIndex OBJECT-TYPE
21        SYNTAX      Integer32 (1..2147483647)
22        MAX-ACCESS   not-accessible
23        STATUS       current
24        DESCRIPTION  "This variable uniquely identifies the MAU
25                     described by this entry from among other
26                     MAUs connected to the same port
27                     (rpMauPortIndex)."
28        REFERENCE    "IEEE Std 802.3, 30.5.1.1.1, aMAUID."
29        ::= { rpMauEntry 3 }
30
31
32     rpMauType OBJECT-TYPE
33        SYNTAX      AutonomousType
34        MAX-ACCESS   read-only
35        STATUS       current
36        DESCRIPTION  "This object identifies the MAU type. Values for
37                     standard IEEE 802.3 MAU types are defined in the
38                     IANA maintained IANA-MAU-MIB module, as
39                     OBJECT-IDENTITIES of dot3MauType.
40                     If the MAU type is unknown, the object identifier
41                     zeroDotZero is returned."
42        REFERENCE    "IEEE Std 802.3, 30.5.1.1.2, aMAUType."
43        ::= { rpMauEntry 4 }
44
45
46     rpMauStatus OBJECT-TYPE
47        SYNTAX      INTEGER {
48                     other(1),
49                     unknown(2),
50                     operational(3),
51                     standby(4),
52                     shutdown(5),
53                     reset(6)
54                     }
55        MAX-ACCESS   read-write
56        STATUS       current
57        DESCRIPTION  "The current state of the MAU. This object may
58                     be implemented as a read-only object by those
59                     agents and MAUs that do not implement software
60                     control of the MAU state. Some agents may not
61                     support setting the value of this object to some
62                     of the enumerated values."
63
64
65
```


1 The value other(1) is returned if the MAU is in
2 a state other than one of the states 2 through
3 6.
4 The value unknown(2) is returned when the MAU's
5 true state is unknown; for example, when it is
6 being initialized.
7
8
9 A MAU in the operational(3) state is fully
10 functional; it operates, and passes signals to its
11 attached DTE or repeater port in accordance to
12 its specification.
13
14 A MAU in standby(4) state forces DI and CI to
15 idle, and the media transmitter to idle or fault,
16 if supported. Standby(4) mode only applies to
17 link type MAUs. The state of
18 rpMauMediaAvailable is unaffected.
19
20
21 A MAU in shutdown(5) state assumes the same
22 condition on DI, CI, and the media transmitter,
23 as though it were powered down or not connected.
24 The MAU may return other(1) value for the
25 rpMauJabberState and rpMauMediaAvailable objects
26 when it is in this state. For an AUI, this
27 state will remove power from the AUI.
28
29
30 Setting this variable to the value reset(6)
31 resets the MAU in the same manner as a
32 power-off, power-on cycle of at least one-half
33 second would. The agent is not required to
34 return the value reset(6).
35
36 Setting this variable to the value
37 operational(3), standby(4), or shutdown(5)
38 causes the MAU to assume the respective state,
39 except that setting a mixing-type MAU or an AUI
40 to standby(4) will cause the MAU to enter the
41 shutdown state."
42 REFERENCE "IEEE Std 802.3, 30.5.1.1.7, aMAUAdminState,
43 30.5.1.2.2, acMAUAdminControl, and 30.5.1.2.1,
44 acResetMAU."
45 ::= { rpMauEntry 5 }
46
47
48 rpMauMediaAvailable OBJECT-TYPE
49 SYNTAX IANAifMauMediaAvailable
50 MAX-ACCESS read-only
51 STATUS current
52 DESCRIPTION "This object identifies Media Available state of
53 the MAU, complementary to the rpMauStatus. Values
54 for the standard IEEE 802.3 Media Available states
55 are defined in the IANA maintained IANA-MAU-MIB
56 module, as IANAifMauMediaAvailable TC."
57 REFERENCE "IEEE Std 802.3, 30.5.1.1.4, aMediaAvailable."
58 ::= { rpMauEntry 6 }
59
60
61 rpMauMediaAvailableStateExits OBJECT-TYPE
62 SYNTAX Counter32
63 MAX-ACCESS read-only
64 STATUS current
65

```
1      DESCRIPTION "A count of the number of times that
2                  rpMauMediaAvailable for this MAU instance leaves
3                  the state available(3).
4
5                  Discontinuities in the value of this counter can
6                  occur at re-initialization of the management
7                  system and at other times, as indicated by the
8                  value of rpPtrMonitorPortLastChange."
9
10     REFERENCE    "IEEE Std 802.3, 30.5.1.1.5, aLoseMediaCounter.
11                  RFC 2108, rpPtrMonitorPortLastChange"
12     ::= { rpMauEntry 7 }
13
14     rpMauJabberState OBJECT-TYPE
15     SYNTAX        INTEGER {
16                     other(1),
17                     unknown(2),
18                     noJabber(3),
19                     jabbering(4)
20                     }
21
22     MAX-ACCESS    read-only
23     STATUS        current
24     DESCRIPTION   "The value other(1) is returned if the jabber
25                   state is not 2, 3, or 4. The agent shall
26                   return other(1) for MAU type dot3MauTypeAUI.
27
28                   The value unknown(2) is returned when the MAU's
29                   true state is unknown; for example, when it is
30                   being initialized.
31
32                   If the MAU is not jabbering the agent returns
33                   noJabber(3). This is the 'normal' state.
34
35                   If the MAU is in jabber state the agent returns
36                   the jabbering(4) value."
37
38     REFERENCE    "IEEE Std 802.3, 30.5.1.1.6, aJabber.jabberFlag."
39     ::= { rpMauEntry 8 }
40
41     rpMauJabberingStateEnters OBJECT-TYPE
42     SYNTAX        Counter32
43     MAX-ACCESS    read-only
44     STATUS        current
45     DESCRIPTION   "A count of the number of times that
46                   mauJabberState for this MAU instance enters the
47                   state jabbering(4). For MAUs of type
48                   dot3MauTypeAUI, dot3MauType100BaseT4,
49                   dot3MauType100BaseTX, dot3MauType100BaseFX, and
50                   all 1000 Mb/s types, this counter will
51                   indicate zero.
52
53                   Discontinuities in the value of this counter can
54                   occur at re-initialization of the management
55                   system and at other times, as indicated by the
56                   value of rpPtrMonitorPortLastChange."
57
58     REFERENCE    "IEEE Std 802.3, 30.5.1.1.6, aJabber.jabberCounter.
59                  RFC 2108, rpPtrMonitorPortLastChange"
60     ::= { rpMauEntry 9 }
61
62     rpMauFalseCarriers OBJECT-TYPE
63     SYNTAX        Counter32
```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION "A count of the number of false carrier events
4                  during IDLE in 100BASE-X links. This counter
5                  does not increment at the symbol rate. It can
6                  increment after a valid carrier completion at a
7                  maximum rate of once per 100 ms until the next
8                  carrier event.
9
10
11                 This counter increments only for MAUs of type
12                 dot3MauType100BaseT4, dot3MauType100BaseTX,
13                 dot3MauType100BaseFX, and all 1000 Mb/s types.
14
15                 For all other MAU types, this counter will
16                 indicate zero.
17
18                 The approximate minimum time for rollover of
19                 this counter is 7.4 hours.
20
21                 Discontinuities in the value of this counter can
22                 occur at re-initialization of the management
23                 system and at other times, as indicated by the
24                 value of rpPtrMonitorPortLastChange."
25
26      REFERENCE   "IEEE Std 802.3, 30.5.1.1.10, aFalseCarriers.
27                  RFC 2108, rpPtrMonitorPortLastChange"
28
29      ::= { rpMauEntry 10 }
30
31      -- The rpJackTable applies to MAUs attached to repeaters
32      -- which have one or more external jacks (connectors).
33      rpJackTable OBJECT-TYPE
34          SYNTAX      SEQUENCE OF RpJackEntry
35          MAX-ACCESS  not-accessible
36          STATUS      current
37          DESCRIPTION "Information about the external jacks attached
38                      to MAUs attached to the ports of a repeater."
39          ::= { dot3RpMauBasicGroup 2 }
40
41      rpJackEntry OBJECT-TYPE
42          SYNTAX      RpJackEntry
43          MAX-ACCESS  not-accessible
44          STATUS      current
45          DESCRIPTION "An entry in the table, containing information
46                      about a particular jack."
47          INDEX       { rpMauGroupIndex,
48                      rpMauPortIndex,
49                      rpMauIndex,
50                      rpJackIndex
51                      }
52          ::= { rpJackTable 1 }
53
54      RpJackEntry ::=
55          SEQUENCE {
56              rpJackIndex      Integer32,
57              rpJackType       IANAIfJackType
58          }
59
60      rpJackIndex OBJECT-TYPE
61          SYNTAX      Integer32 (1..2147483647)
62          MAX-ACCESS  not-accessible

```

```

1      STATUS      current
2      DESCRIPTION "This variable uniquely identifies the jack
3                  described by this entry from among other jacks
4                  attached to the same MAU (rpMauIndex)."
```

::= { rpJackEntry 1 }

```

7      rpJackType OBJECT-TYPE
8          SYNTAX      IANAifJackType
9          MAX-ACCESS  read-only
10         STATUS      current
11         DESCRIPTION "The jack connector type, as it appears on the
12                     outside of the system."
13         ::= { rpJackEntry 2 }
```

--

-- The Basic Interface MAU Table

--

```

19     ifMauTable OBJECT-TYPE
20         SYNTAX      SEQUENCE OF IfMauEntry
21         MAX-ACCESS  not-accessible
22         STATUS      current
23         DESCRIPTION "Table of descriptive and status information
24                     about MAU(s) attached to an interface."
25         ::= { dot3IfMauBasicGroup 1 }
```

```

28     ifMauEntry OBJECT-TYPE
29         SYNTAX      IfMauEntry
30         MAX-ACCESS  not-accessible
31         STATUS      current
32         DESCRIPTION "An entry in the table, containing information
33                     about a single MAU."
34         INDEX       { ifMauIfIndex,
35                     ifMauIndex
36                     }
37         ::= { ifMauTable 1 }
```

```

40     IfMauEntry ::=
41         SEQUENCE {
42             ifMauIfIndex      InterfaceIndex,
43             ifMauIndex        Integer32,
44             ifMauType         AutonomousType,
45             ifMauStatus       INTEGER,
46             ifMauMediaAvailable IANAifMauMediaAvailable,
47             ifMauMediaAvailableStateExits Counter32,
48             ifMauJabberState  INTEGER,
49             ifMauJabberingStateEnters Counter32,
50             ifMauFalseCarriers Counter32,
51             ifMauDefaultType  AutonomousType,
52             ifMauAutoNegSupported TruthValue,
53             ifMauTypeListBits IANAifMauTypeListBits,
54             ifMauHCFALSECarriers Counter64,
55             ifMauPCSCodingViolations Counter64,
56             ifMauFECAbility    INTEGER,
57             ifMauFECMode       INTEGER,
58             ifMauFECCorrectedBlocks Counter64,
59             ifMauFECUnCorrectableBlocks Counter64,
60             ifMauSNROpMarginChnlA Integer32,
61             ifMauSNROpMarginChnlB Integer32,
62             ifMauSNROpMarginChnlC Integer32,
```

```
1         ifMauSNROpMarginChn1D           Integer32,
2         ifMauEEESupportList              IANAifMauTypeListBits,
3         ifMauEEELDFastRetrainCount        Counter32,
4         ifMauEEELPFastRetrainCount        Counter32,
5         ifMauTimeSyncCapabilityTX          TruthValue,
6         ifMauTimeSyncCapabilityRX          TruthValue,
7         ifMauTimeSyncDelayTXmax            Integer32,
8         ifMauTimeSyncDelayTXmin            Integer32,
9         ifMauTimeSyncDelayRXmax            Integer32,
10        ifMauTimeSyncDelayRXmin            Integer32,
11    }
12
13
14    ifMauIfIndex OBJECT-TYPE
15        SYNTAX      InterfaceIndex
16        MAX-ACCESS  not-accessible
17        STATUS      current
18        DESCRIPTION "This variable uniquely identifies the interface
19                    to which the MAU described by this entry is
20                    connected."
21        REFERENCE   "RFC 2863, ifIndex"
22        ::= { ifMauEntry 1 }
23
24
25    ifMauIndex OBJECT-TYPE
26        SYNTAX      Integer32 (1..2147483647)
27        MAX-ACCESS  not-accessible
28        STATUS      current
29        DESCRIPTION "This variable uniquely identifies the MAU
30                    described by this entry from among other MAUs
31                    connected to the same interface (ifMauIfIndex)."
32        REFERENCE   "IEEE Std 802.3, 30.5.1.1.1, aMAUID."
33        ::= { ifMauEntry 2 }
34
35
36    ifMauType OBJECT-TYPE
37        SYNTAX      AutonomousType
38        MAX-ACCESS  read-only
39        STATUS      current
40        DESCRIPTION "This object identifies the MAU type. Values for
41                    standard IEEE 802.3 MAU types are defined in the
42                    IANA maintained IANA-MAU-MIB module, as
43                    OBJECT-IDENTITIES of dot3MauType.
44                    If the MAU type is unknown, the object identifier
45                    zeroDotZero is returned.
46
47                    This object represents the operational type of
48                    the MAU, as determined by either 1) the result
49                    of the Auto-Negotiation function or 2) if
50                    Auto-Negotiation is not enabled or is not
51                    implemented for this MAU, by the value of the
52                    object ifMauDefaultType. In case 2), a set to
53                    the object ifMauDefaultType will force the MAU
54                    into the new operating mode."
55        REFERENCE   "IEEE Std 802.3, 30.5.1.1.2, aMAUType."
56        ::= { ifMauEntry 3 }
57
58
59
60    ifMauStatus OBJECT-TYPE
61        SYNTAX      INTEGER {
62                    other(1),
63                    unknown(2),
64                    operational(3),
65
```

```
1             standby(4),
2             shutdown(5),
3             reset(6)
4         }
5     MAX-ACCESS read-write
6     STATUS current
7     DESCRIPTION "The current state of the MAU. This object may
8                 be implemented as a read-only object by those
9                 agents and MAUs that do not implement software
10                control of the MAU state. Some agents may not
11                support setting the value of this object to some
12                of the enumerated values.
13
14                The value other(1) is returned if the MAU is in
15                a state other than one of the states 2 through
16                6.
17
18                The value unknown(2) is returned when the MAU's
19                true state is unknown; for example, when it is
20                being initialized.
21
22                A MAU in the operational(3) state is fully
23                functional; it operates, and passes signals to its
24                attached DTE or repeater port in accordance to
25                its specification.
26
27                A MAU in standby(4) state forces DI and CI to
28                idle and the media transmitter to idle or fault,
29                if supported. Standby(4) mode only applies to
30                link type MAUs. The state of
31                ifMauMediaAvailable is unaffected.
32
33                A MAU in shutdown(5) state assumes the same
34                condition on DI, CI, and the media transmitter,
35                as though it were powered down or not connected.
36                The MAU may return other(1) value for the
37                ifMauJabberState and ifMauMediaAvailable objects
38                when it is in this state. For an AUI, this
39                state will remove power from the AUI.
40
41                Setting this variable to the value reset(6)
42                resets the MAU in the same manner as a
43                power-off, power-on cycle of at least one-half
44                second would. The agent is not required to
45                return the value reset(6).
46
47                Setting this variable to the value
48                operational(3), standby(4), or shutdown(5)
49                causes the MAU to assume the respective state,
50                except that setting a mixing-type MAU or an AUI
51                to standby(4) will cause the MAU to enter the
52                shutdown state."
```

REFERENCE "IEEE Std 802.3, 30.5.1.1.7, aMAUAdminState,
30.5.1.2.2, acMAUAdminControl, and 30.5.1.2.1,
acResetMAU."

```
 ::= { ifMauEntry 4 }

ifMauMediaAvailable OBJECT-TYPE
SYNTAX      IANAifMauMediaAvailable
```

```
1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION "This object identifies Media Available state of
4                  the MAU, complementary to the ifMauStatus. Values
5                  for the standard IEEE 802.3 Media Available states
6                  are defined in the IANA maintained IANA-MAU-MIB
7                  module, as IANAifMauMediaAvailable TC."
8      REFERENCE   "IEEE Std 802.3, 30.5.1.1.4, aMediaAvailable."
9      ::= { ifMauEntry 5 }
10
11
12  ifMauMediaAvailableStateExits OBJECT-TYPE
13      SYNTAX      Counter32
14      MAX-ACCESS  read-only
15      STATUS      current
16      DESCRIPTION "A count of the number of times that
17                  ifMauMediaAvailable for this MAU instance leaves
18                  the state available(3).
19
20                  Discontinuities in the value of this counter can
21                  occur at re-initialization of the management
22                  system and at other times, as indicated by the
23                  value of ifCounterDiscontinuityTime."
24      REFERENCE   "IEEE Std 802.3, 30.5.1.1.5, aLoseMediaCounter.
25                  RFC 2863, ifCounterDiscontinuityTime."
26      ::= { ifMauEntry 6 }
27
28
29  ifMauJabberState OBJECT-TYPE
30      SYNTAX      INTEGER {
31                  other(1),
32                  unknown(2),
33                  noJabber(3),
34                  jabbering(4)
35              }
36      MAX-ACCESS  read-only
37      STATUS      current
38      DESCRIPTION "The value other(1) is returned if the jabber
39                  state is not 2, 3, or 4. The agent shall
40                  return other(1) for MAU type dot3MauTypeAUI.
41
42                  The value unknown(2) is returned when the MAU's
43                  true state is unknown; for example, when it is
44                  being initialized.
45
46                  If the MAU is not jabbering the agent returns
47                  noJabber(3). This is the 'normal' state.
48
49                  If the MAU is in jabber state the agent returns
50                  the jabbering(4) value."
51      REFERENCE   "IEEE Std 802.3, 30.5.1.1.6, aJabber.jabberFlag."
52      ::= { ifMauEntry 7 }
53
54
55  ifMauJabberingStateEnters OBJECT-TYPE
56      SYNTAX      Counter32
57      MAX-ACCESS  read-only
58      STATUS      current
59      DESCRIPTION "A count of the number of times that
60                  mauJabberState for this MAU instance enters the
61                  state jabbering(4). This counter will
62                  indicate zero for MAUs of type dot3MauTypeAUI
63
64
65
```

```
1          and those of speeds above 10 Mb/s.
2
3          Discontinuities in the value of this counter can
4          occur at re-initialization of the management
5          system and at other times, as indicated by the
6          value of ifCounterDiscontinuityTime."
7      REFERENCE  "IEEE Std 802.3, 30.5.1.1.6, aJabber.jabberCounter.
8                  RFC 2863, ifCounterDiscontinuityTime."
9      ::= { ifMauEntry 8 }
10
11  ifMauFalseCarriers OBJECT-TYPE
12      SYNTAX      Counter32
13      MAX-ACCESS  read-only
14      STATUS      current
15      DESCRIPTION "A count of the number of false carrier events
16                  during IDLE in 100BASE-X and 1000BASE-X links.
17
18                  For all other MAU types, this counter will
19                  indicate zero. This counter does not
20                  increment at the symbol rate.
21
22                  It can increment after a valid carrier
23                  completion at a maximum rate of once per 100 ms
24                  for 100BASE-X and once per 10us for 1000BASE-X
25                  until the next CarrierEvent.
26
27                  This counter can roll over very quickly. A
28                  management station is advised to poll the
29                  ifMauHCFALSECarriers instead of this counter in
30                  order to avoid loss of information.
31
32                  Discontinuities in the value of this counter can
33                  occur at re-initialization of the management
34                  system and at other times, as indicated by the
35                  value of ifCounterDiscontinuityTime."
36      REFERENCE  "IEEE Std 802.3, 30.5.1.1.10, aFalseCarriers.
37                  RFC 2863, ifCounterDiscontinuityTime."
38      ::= { ifMauEntry 9 }
39
40  ifMauDefaultType OBJECT-TYPE
41      SYNTAX      AutonomousType
42      MAX-ACCESS  read-write
43      STATUS      current
44      DESCRIPTION "This object identifies the default
45                  administrative baseband MAU type to be used in
46                  conjunction with the operational MAU type
47                  denoted by ifMauType.
48
49                  The set of possible values for this object is
50                  the same as the set defined for the ifMauType
51                  object.
52
53                  This object represents the
54                  administratively-configured type of the MAU. If
55                  Auto-Negotiation is not enabled or is not
56                  implemented for this MAU, the value of this
57                  object determines the operational type of the
58                  MAU. In this case, a set to this object will
59                  force the MAU into the specified operating mode.
```


1
2 If Auto-Negotiation is implemented and enabled
3 for this MAU, the operational type of the MAU
4 is determined by Auto-Negotiation, and the value
5 of this object denotes the type to which the MAU
6 will automatically revert if/when
7 Auto-Negotiation is later disabled.
8
9
10 It may be necessary to provide for underlying hardware
11 implementations which do not follow the exact behavior
12 specified above.
13 In particular, when ifMauAutoNegAdminStatus transitions
14 from enabled to disabled, the agent implementation shall
15 verify that the operational type of the MAU
16 (as reported by ifMauType) correctly transitions to
17 the value specified by this object, rather than
18 continuing to operate at the value earlier
19 determined by the Auto-Negotiation function."
20
21 REFERENCE "IEEE Std 802.3, 30.5.1.1.1, aMAUID, and 22.2.4.1.4."
22 ::= { ifMauEntry 10 }
23
24 ifMauAutoNegSupported OBJECT-TYPE
25 SYNTAX TruthValue
26 MAX-ACCESS read-only
27 STATUS current
28 DESCRIPTION "This object indicates whether or not
29 Auto-Negotiation is supported on this MAU."
30 ::= { ifMauEntry 11 }
31
32
33 ifMauTypeListBits OBJECT-TYPE
34 SYNTAX IANAifMauTypeListBits
35 MAX-ACCESS read-only
36 STATUS current
37 DESCRIPTION "A value that uniquely identifies the set of
38 possible IEEE 802.3 types that the MAU could be.
39 If Auto-Negotiation is present on this MAU, this
40 object will map to ifMauAutoNegCapabilityBits.
41
42 Note that this MAU may be capable of operating
43 as a MAU type that is beyond the scope of this
44 MIB. This is indicated by returning the
45 bit value bOther in addition to any bit values
46 for standard capabilities that are listed in the
47 IANAifMauTypeListBits TC."
48 ::= { ifMauEntry 12 }
49
50
51 ifMauHCFALSECarriers OBJECT-TYPE
52 SYNTAX Counter64
53 MAX-ACCESS read-only
54 STATUS current
55 DESCRIPTION "A count of the number of false carrier events
56 during IDLE in 100BASE-X and 1000BASE-X links.
57
58 For all other MAU types, this counter will
59 indicate zero. This counter does not
60 increment at the symbol rate.
61
62 This counter is a 64-bit version of
63 ifMauFalseCarriers. Since the 32-bit version of
64
65

```
1         this counter can roll over very quickly,
2         management stations are advised to poll the
3         64-bit version instead, in order to avoid loss
4         of information.
5
6         Discontinuities in the value of this counter can
7         occur at re-initialization of the management
8         system and at other times, as indicated by the
9         value of ifCounterDiscontinuityTime."
10
11     REFERENCE "IEEE Std 802.3, 30.5.1.1.10, aFalseCarriers.
12               RFC 2863, ifCounterDiscontinuityTime."
13     ::= { ifMauEntry 13 }
14
15     ifMauPCSCodingViolations OBJECT-TYPE
16         SYNTAX      Counter64
17         MAX-ACCESS   read-only
18         STATUS       current
19         DESCRIPTION  "Generalized nonresettable counter. This counter
20                     has a maximum increment rate of 25 000 000
21                     counts per second for 100 Mb/s implementations and
22                     125 000 000 counts per second for 1000 Mb/s
23                     implementations.
24
25                     For 100 Mb/s operation it is a count of the number
26                     of events that cause the PHY to indicate 'Data
27                     reception with errors' on the MII (see IEEE Std 802.3
28                     Table 22-2).
29
30                     For 1000 Mb/s operation it is a count of the
31                     number of events that cause the PHY to indicate 'Data
32                     reception error' or 'Carrier Extend Error' on the GMII
33                     (see IEEE Std 802.3, Table 35-2). The contents of this
34                     attribute is undefined when FEC is operating."
35
36                     REFERENCE "IEEE Std 802.3, 30.5.1.1.14 aPCSCodingViolations."
37     ::= {ifMauEntry 14}
38
39
40     ifMauFECAbility OBJECT-TYPE
41         SYNTAX      INTEGER {
42                     unknown(1),
43                     supported(2),
44                     notsupported(3)
45                     }
46         MAX-ACCESS   read-only
47         STATUS       current
48         DESCRIPTION  "A read-only value that indicates if the
49                     PHY supports an optional FEC sublayer for
50                     forward error correction (see IEEE Std 802.3, 65.2
51                     and IEEE Std 802.3, Clause 74).
52
53                     If an IEEE Std 802.3 Clause 45 MDIO Interface to the
54                     PCS is present, then this attribute will map to the
55                     FEC capability register (see IEEE Std 802.3, 45.2.8.2)."
56
57                     REFERENCE "IEEE Std 802.3, 30.5.1.1.15 aFECAbility."
58     ::= {ifMauEntry 15}
59
60
61     ifMauFECMode OBJECT-TYPE
62         SYNTAX      INTEGER {
63                     unknown(1),
64                     disabled(2),
65
```

```
1          enabled(3)
2      }
3      MAX-ACCESS read-write
4      STATUS current
5      DESCRIPTION "A read-write value that indicates the mode of
6                  operation of the optional FEC sublayer for forward
7                  error correction (see IEEE Std 802.3, 65.2 and
8                  IEEE Std 802.3, Clause 74).
9
10                 A GET operation returns the current mode of operation
11                 of the PHY. A SET operation changes the mode of
12                 operation of the PHY to the indicated value. When
13                 IEEE Std 802.3 Clause 73 Auto-Negotiation is enabled
14                 a SET operation is not allowed and a GET operation maps
15                 to the variable FEC enabled in Clause 74.
16
17                 If an IEEE Std 802.3 Clause 45 MDIO Interface to the
18                 PCS is present, then this object will map to the FEC
19                 control register (see IEEE Std 802.3 45.2.8.3) for
20                 1000BASE-PX or FEC enable bit in the BASE-R FEC control
21                 register (see IEEE Std 802.3 45.2.1.90)."
```

24 REFERENCE "IEEE Std 802.3. 30.5.1.1.16 aFECMode."
25 ::= {ifMauEntry 16}

27 ifMauFECCorrectedBlocks OBJECT-TYPE
28 SYNTAX Counter64
29 MAX-ACCESS read-only
30 STATUS deprecated
31 DESCRIPTION
32 "***** THIS OBJECT IS DEPRECATED *****
33
34
35 Generalized nonresettable counter. This counter
36 has a maximum increment rate of 1 200 000
37 counts per second for 1000 Mb/s implementations,
38 and 5 000 000 counts per second for 10 Gb/s
39 implementations.
40
41 For 1000BASE-PX PHYs or 10GBASE-R PHYs, a count
42 of corrected FEC blocks. This counter will not
43 increment for other PHY types.
44 Increment the counter by one for each received block
45 that is corrected by the FEC function in the PHY.
46 If a Clause 45 MDIO Interface to the PCS is present,
47 then this object will map to the FEC corrected blocks
48 counter (see IEEE Std 802.3, 45.2.8.5 and 45.2.1.91)"
50 REFERENCE "IEEE Std 802.3. 30.5.1.1.17 aFECCorrectedBlocks."
51 ::= {ifMauEntry 17}

53 ifMauFECUnCorrectableBlocks OBJECT-TYPE
54 SYNTAX Counter64
55 MAX-ACCESS read-only
56 STATUS deprecated
57 DESCRIPTION
58 "***** THIS OBJECT IS DEPRECATED *****
59
60
61 Generalized nonresettable counter. This counter
62 has a maximum increment rate of 1 200 000
63 counts per second for 1000 Mb/s implementations,
64 and 5 000 000 counts per second for 10 Gb/s
65

1 implementations.

2

3 For 1000BASE-PX PHYs or 10GBASE-R PHYs, a count

4 of uncorrectable FEC blocks. This counter will not

5 increment for other PHY types.

6 Increment the counter by one for each received block

7 that is determined to be uncorrectable by the FEC

8 function in the PHY.

9

10

11 If a Clause 45 MDIO Interface to the PCS is present,

12 then this object will map to the FEC uncorrectable

13 blocks counter (see IEEE Std 802.3 45.2.8.6 and

14 45.2.1.92)"

15 REFERENCE "IEEE Std 802.3. 30.5.1.1.18 aFECUnCorrectableBlocks."

16 ::= {ifMauEntry 18}

17

18 ifMauSNROpMarginChnlA OBJECT-TYPE

19 SYNTAX Integer32 (-127..127)

20 MAX-ACCESS read-only

21 STATUS current

22 DESCRIPTION "The current SNR operating margin measured at the

23 slicer input for channel A for the 10GBASE-T PMA.

24 It is reported in units of 0.1 dB to an accuracy of

25 0.5 dB within the range of -12.7 dB to 12.7 dB.

26 If an IEEE Std 802.3 Clause 45 MDIO Interface to the

27 PMA/PMD is present, then this attribute maps to the SNR

28 operating margin channel A register

29 (see IEEE Std 802.3, 45.2.1.65)."

30 REFERENCE "IEEE Std 802.3, 30.5.1.1.19 aSNROpMarginChnlA."

31 ::= {ifMauEntry 19}

32

33

34

35 ifMauSNROpMarginChnlB OBJECT-TYPE

36 SYNTAX Integer32 (-127..127)

37 MAX-ACCESS read-only

38 STATUS current

39 DESCRIPTION "The current SNR operating margin measured at the

40 slicer input for channel B for the 10GBASE-T PMA.

41 It is reported in units of 0.1 dB to an accuracy of

42 0.5 dB within the range of -12.7 dB to 12.7 dB.

43 If an IEEE Std 802.3 Clause 45 MDIO Interface to the

44 PMA/PMD is present, then this attribute maps to the SNR

45 operating margin channel B register

46 (see IEEE Std 802.3, 45.2.1.66)."

47 REFERENCE "IEEE Std 802.3, 30.5.1.1.20 aSNROpMarginChnlB."

48 ::= {ifMauEntry 20}

49

50

51

52 ifMauSNROpMarginChnlC OBJECT-TYPE

53 SYNTAX Integer32 (-127..127)

54 MAX-ACCESS read-only

55 STATUS current

56 DESCRIPTION "The current SNR operating margin measured at the

57 slicer input for channel C for the 10GBASE-T PMA.

58 It is reported in units of 0.1 dB to an accuracy of

59 0.5 dB within the range of -12.7 dB to 12.7 dB.

60 If an IEEE Std 802.3 Clause 45 MDIO Interface to the

61 PMA/PMD is present, then this attribute maps to the SNR

62 operating margin channel C register

63 (see IEEE Std 802.3, 45.2.1.67)."

64 REFERENCE "IEEE Std 802.3, 30.5.1.1.21 aSNROpMarginChnlC."

65

```
1      ::= {ifMauEntry 21}
2
3  ifMauSNROpMarginChnlD OBJECT-TYPE
4      SYNTAX      Integer32 (-127..127)
5      MAX-ACCESS  read-only
6      STATUS      current
7      DESCRIPTION "The current SNR operating margin measured at the
8                  slicer input for channel D for the 10GBASE-T PMA.
9                  It is reported in units of 0.1 dB to an accuracy of
10                 0.5 dB within the range of -12.7 dB to 12.7 dB.
11                 If an IEEE Std 802.3 Clause 45 MDIO Interface to the
12                 PMA/PMD is present, then this attribute maps to the SNR
13                 operating margin channel D register
14                 (see IEEE Std 802.3, 45.2.1.68)."
```

15

```
16      REFERENCE  "IEEE Std 802.3, 30.5.1.1.22 aSNROpMarginChnlD."
17      ::= {ifMauEntry 22}
18
19  ifMauEEESupportList OBJECT-TYPE
20      SYNTAX      IANAifMauTypeListBits
21      MAX-ACCESS  read-only
22      STATUS      current
23      DESCRIPTION "A read-only list of the possible PHY types for which
24                  the underlying system supports Energy-Efficient Ethernet
25                  (EEE) as defined in IEEE Std 802.3 Clause 78."
26      REFERENCE  "IEEE Std 802.3, 30.5.1.1.23 aEEESupportList."
27      ::= { ifMauEntry 23 }
28
29  ifMauEEELDFastRetrainCount OBJECT-TYPE
30      SYNTAX      Counter32
31      MAX-ACCESS  read-only
32      STATUS      current
33      DESCRIPTION "A count of the number of 10GBASE-T fast retrains
34                  initiated by the local device. The indication reflects
35                  the state of the PHY event counter (see IEEE Std 802.3,
36                  45.2.1.78.2 and 55.4.5.1.)"
37      REFERENCE  "IEEE Std 802.3, 30.5.1.1.24 aLDFastRetrainCount."
38      ::= { ifMauEntry 24 }
39
40  ifMauEEELPFastRetrainCount OBJECT-TYPE
41      SYNTAX      Counter32
42      MAX-ACCESS  read-only
43      STATUS      current
44      DESCRIPTION "A count of the number of 10GBASE-T fast retrains
45                  initiated by the link partner. The indication reflects
46                  the state of the PHY event counter (see IEEE Std 802.3,
47                  45.2.1.78.1 and 55.4.5.1.)"
48      REFERENCE  "IEEE Std 802.3, 30.5.1.1.25 aLPFastRetrainCount."
49      ::= { ifMauEntry 25 }
50
51  ifMauTimeSyncCapabilityTX OBJECT-TYPE
52      SYNTAX      TruthValue
53      MAX-ACCESS  read-only
54      STATUS      current
55      DESCRIPTION "This object indicates whether or not transmit
56                  Time Sync is supported on this MAU."
57      REFERENCE  "IEEE Std 802.3, 30.13.1.1 aTimeSyncCapabilityTX."
58      ::= { ifMauEntry 26 }
59
60  ifMauTimeSyncCapabilityRX OBJECT-TYPE
```

```
1      SYNTAX      TruthValue
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION "This object indicates whether or not receive
5                  Time Sync is supported on this MAU."
6      REFERENCE   "IEEE Std 802.3, 30.13.1.2 aTimeSyncCapabilityRX."
7      ::= { ifMauEntry 27 }
8
9
10     ifMauTimeSyncDelayTXmax OBJECT-TYPE
11         SYNTAX      Integer32
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION "The maximum data delay as specified in IEEE Std 802.3
15                     90.7, expressed in units of ns.
16
17                     If an IEEE Std 802.3 Clause 45 MDIO Interface to
18                     PMA/PMD, WIS, PCS, PHY XS, DTE XS and/or TC is
19                     present, then the value stored in this attribute
20                     represents the maximum transmit path data delay
21                     values, consisting of the sum of the values of the
22                     registers in the instantiated sublayers (for each MMD,
23                     in case of multiple instances)"
24         REFERENCE   "IEEE Std 802.3, 30.13.1.3 aTimeSyncDelayTXmax."
25         ::= { ifMauEntry 28 }
26
27
28     ifMauTimeSyncDelayTXmin OBJECT-TYPE
29         SYNTAX      Integer32
30         MAX-ACCESS  read-only
31         STATUS      current
32         DESCRIPTION "The minimum data delay as specified in IEEE Std 802.3
33                     90.7, expressed in units of ns.
34
35                     If an IEEE Std 802.3 Clause 45 MDIO Interface to
36                     PMA/PMD, WIS, PCS, PHY XS, DTE XS and/or TC is
37                     present, then the value stored in this attribute
38                     represents the minimum transmit path data delay
39                     values, consisting of the sum of the values of the
40                     registers in the instantiated sublayers (for each MMD,
41                     in case of multiple instances)"
42         REFERENCE   "IEEE Std 802.3, 30.13.1.4 aTimeSyncDelayTXmin."
43         ::= { ifMauEntry 29 }
44
45
46
47     ifMauTimeSyncDelayRXmax OBJECT-TYPE
48         SYNTAX      Integer32
49         MAX-ACCESS  read-only
50         STATUS      current
51         DESCRIPTION "The maximum data delay as specified in IEEE Std 802.3
52                     90.7, expressed in units of ns.
53
54                     If an IEEE Std 802.3 Clause 45 MDIO Interface to
55                     PMA/PMD, WIS, PCS, PHY XS, DTE XS and/or TC is
56                     present, then the value stored in this attribute
57                     represents the maximum receive path data delay
58                     values, consisting of the sum of the values of the
59                     registers in the instantiated sublayers (for each MMD,
60                     in case of multiple instances)"
61         REFERENCE   "IEEE Std 802.3, 30.13.1.5 aTimeSyncDelayRXmax."
62         ::= { ifMauEntry 30 }
63
64
65
```

```
1      ifMauTimeSyncDelayRXmin OBJECT-TYPE
2          SYNTAX      Integer32
3          MAX-ACCESS  read-only
4          STATUS      current
5          DESCRIPTION "The minimum data delay as specified in IEEE Std 802.3
6                      90.7, expressed in units of ns.
7
8                      If an IEEE Std 802.3 Clause 45 MDIO Interface to
9                      PMA/PMD, WIS, PCS, PHY XS, DTE XS and/or TC is
10                     present, then the value stored in this attribute
11                     represents the minimum receive path data delay
12                     values, consisting of the sum of the values of the
13                     registers in the instantiated sublayers (for each MMD,
14                     in case of multiple instances)"
15
16          REFERENCE   "IEEE Std 802.3, 30.13.1.6 aTimeSyncDelayRXmin."
17          ::= { ifMauEntry 31 }
18
19
20
21      -- The ifJackTable applies to MAUs attached to interfaces
22      -- which have one or more external jacks (connectors).
23
24      ifJackTable OBJECT-TYPE
25          SYNTAX      SEQUENCE OF IfJackEntry
26          MAX-ACCESS  not-accessible
27          STATUS      current
28          DESCRIPTION "Information about the external jacks attached
29                      to MAUs attached to an interface."
30          ::= { dot3IfMauBasicGroup 2 }
31
32
33      ifJackEntry OBJECT-TYPE
34          SYNTAX      IfJackEntry
35          MAX-ACCESS  not-accessible
36          STATUS      current
37          DESCRIPTION "An entry in the table, containing information
38                      about a particular jack."
39          INDEX       { ifMauIfIndex,
40                      ifMauIndex,
41                      ifJackIndex
42                      }
43          ::= { ifJackTable 1 }
44
45
46      IfJackEntry ::=
47          SEQUENCE {
48              ifJackIndex      Integer32,
49              ifJackType       IANAifJackType
50          }
51
52
53      ifJackIndex OBJECT-TYPE
54          SYNTAX      Integer32 (1..2147483647)
55          MAX-ACCESS  not-accessible
56          STATUS      current
57          DESCRIPTION "This variable uniquely identifies the jack
58                      described by this entry from among other jacks
59                      attached to the same MAU."
60          ::= { ifJackEntry 1 }
61
62
63      ifJackType OBJECT-TYPE
64          SYNTAX      IANAifJackType
65          MAX-ACCESS  read-only
```

```

1      STATUS      current
2      DESCRIPTION "The jack connector type, as it appears on the
3                  outside of the system."
4      ::= { ifJackEntry 2 }
5
6      --
7      -- The MAU Per-PCS Lane Statistics Table
8      --
9
10
11     ifMauPerPCSLaneStatsTable OBJECT-TYPE
12         SYNTAX      SEQUENCE OF IfMauPerPCSLaneStatsEntry
13         MAX-ACCESS   not-accessible
14         STATUS      current
15         DESCRIPTION "Table of Per-PCS lane status information
16                     about MAUs attached to an interface."
17         ::= { dot3IfMauBasicGroup 3 }
18
19
20     ifMauPerPCSLaneStatsEntry OBJECT-TYPE
21         SYNTAX      IfMauPerPCSLaneStatsEntry
22         MAX-ACCESS   not-accessible
23         STATUS      current
24         DESCRIPTION "An entry in the table, containing information
25                     about a single PCS lane."
26         INDEX       { ifMauIfIndex,
27                     ifMauIndex,
28                     ifPCSLaneIndex
29                     }
30         ::= { ifMauPerPCSLaneStatsTable 1 }
31
32
33     IfMauPerPCSLaneStatsEntry ::=
34         SEQUENCE {
35             ifPCSLaneIndex                Unsigned32,
36             ifMauPPLFECCorrectedBlocks    Counter64,
37             ifMauPPLFECUncorrectableBlocks Counter64,
38             ifMauBIPErrorCount            Counter32,
39             ifMauPCStoPHYLaneMapping      Unsigned32
40         }
41
42
43     ifPCSLaneIndex OBJECT-TYPE
44         SYNTAX      Unsigned32 (0..255)
45         MAX-ACCESS   not-accessible
46         STATUS      current
47         DESCRIPTION "This object provides the identification of the
48                     PCS lane for which this ifMauPerPCSLaneStatsEntry
49                     is applicable. This object can hold an integer value
50                     from 0 to N-1, where N is the total number of PCS
51                     lanes supported by the given PCS. "
52         ::= { ifMauPerPCSLaneStatsEntry 1 }
53
54
55     ifMauPPLFECCorrectedBlocks OBJECT-TYPE
56         SYNTAX      Counter64
57         MAX-ACCESS   read-only
58         STATUS      current
59         DESCRIPTION "Generalized nonresettable counter. This counter has a
60                     maximum increment rate of 1 200 000 counts per second
61                     for 1000 Mb/s implementations, 5 000 000 counts per
62                     second for 10 Gb/s and 40 Gb/s implementations, and
63                     2 500 000 counts per second for 100 Gb/s implementations.
64
65

```


1 For 1000BASE-PX, 10/40/100GBASE-R PHYs, a count of
2 corrected FEC blocks received on the PSC lane identified
3 by ifPCSLaneIndex object. This counter will not increment
4 for other PHY types.
5
6 Increment the counter by one for each received block that
7 is corrected by the FE C function in the PHY for the
8 corresponding lane identified by the ifPCSLaneIndex
9 object.
10
11 If a Clause 45 MDIO Interface to the PCS is present,
12 then this object will map to the FEC corrected blocks
13 counter for PCS lane number n, identified by the
14 ifPCSLaneIndex object
15 (see IEEE Std 802.3 45.2.8.5, 45.2.1.91 , and 45.2.1.93)."
16
17 REFERENCE "IEEE Std 802.3 30.5.1.1.17"
18 ::= { ifMauPerPCSLaneStatsEntry 2 }
19
20
21 ifMauPPLFECUncorrectableBlocks OBJECT-TYPE
22 SYNTAX Counter64
23 MAX-ACCESS read-only
24 STATUS current
25 DESCRIPTION "Generalized nonresettable counter. This counter has a
26 maximum increment rate of 1 200 000 counts per second
27 for 1000 Mb/s implementations, 5 000 000 counts
28 per second for 10 Gb/s and 40 Gb/s implementations,
29 and 2 500 000 counts per second for 100 Gb/s
30 implementations.
31
32 For 1000BASE-PX, 10/40/100GBASE-R PHYs, a count of
33 uncorrectable FEC blocks received on the PSC lane
34 identified by ifPCSLaneIndex object. This counter will
35 not increment for other PHY types.
36
37 Increment the counter by one for each FEC block that
38 is determined to be uncorrectable by the FEC function
39 in the PHY for the corresponding lane identified by
40 the ifPCSLaneIndex object.
41
42 If a Clause 45 MDIO Interface to the PCS is present,
43 then this object will map to the FEC uncorrectable
44 blocks counter for PSC lane number n, identified by
45 the ifPCSLaneIndex object
46 (see IEEE Std 802.3 45.2.8.6, 45.2.1.92, and 45.2.1.94)."
47
48 REFERENCE "IEEE Std 802.3 30.5.1.1.18"
49 ::= { ifMauPerPCSLaneStatsEntry 3 }
50
51
52
53 ifMauBIPErrorCount OBJECT-TYPE
54 SYNTAX Counter32
55 MAX-ACCESS read-only
56 STATUS current
57 DESCRIPTION "Generalized nonresettable counter. This counter
58 has a maximum increment rate of 10 000 counts per
59 second for 40 Gb/s implementations and 5 000 counts
60 per second for 100 Gb/ s implementations.
61
62 For 40/100GBASE-R PHYs, a count of BIP errors on the
63 PCS lane identified by ifPCSLaneIndex object. This
64 counter will not increment for other PHY types.
65

```
1
2      Increment the counter by one for each BIP error
3      detected during alignment marker removal in the
4      PCS identified by the ifPCSLaneIndex object.
5
6      If a Clause 45 MDIO Interface to the PCS is
7      present, then this object will map to the BIP error
8      counter for PCS lane number n, identified by the
9      ifPCSLaneIndex object
10     (see IEEE Std 802.3, 45.2.3.44 and 45.2.3.45)."
```

REFERENCE "IEEE Std 802.3, 30.5.1.1.11"

::= { ifMauPerPCSLaneStatsEntry 4 }

ifMauPCStoPHYLaneMapping OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "For 40/100GBASE-R PHYs, an array of PCS lane
identifiers. The indices of this array (0 to n-1)
denote the service interface lane number where n is
the number of PCS lanes in use. Each element of
this array contains the PCS lane number for the PCS
lane that has been detected in the corresponding
service interface lane.

If a Clause 45 MDIO Interface to the PCS is
present, then this object will map to the Lane
mapping register for PCS lane number n, identified
by the ifPCSLaneIndex object
(see IEEE Std 802.3 45.2.3.46 and 45.2.3.47)."

REFERENCE "IEEE Std 802.3 30.5.1.1.12"

::= { ifMauPerPCSLaneStatsEntry 5 }

--

-- The MAU Auto-Negotiation Table

--

ifMauAutoNegTable OBJECT-TYPE

SYNTAX SEQUENCE OF IfMauAutoNegEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "Configuration and status objects for the
Auto-Negotiation function of MAUs attached to
interfaces.

The ifMauAutoNegTable applies to systems in
which Auto-Negotiation is supported on one or
more MAUs attached to interfaces. Note that if
Auto-Negotiation is present and enabled, the
ifMauType object reflects the result of the
Auto-Negotiation function."

::= { dot3IfMauAutoNegGroup 1 }

ifMauAutoNegEntry OBJECT-TYPE

SYNTAX IfMauAutoNegEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "An entry in the table, containing configuration
and status information for the Auto-Negotiation

```

1           function of a particular MAU."
2     INDEX      { ifMauIfIndex,
3                 ifMauIndex
4                 }
5     ::= { ifMauAutoNegTable 1 }
6
7     IfMauAutoNegEntry ::=
8     SEQUENCE {
9         ifMauAutoNegAdminStatus      INTEGER,
10        ifMauAutoNegRemoteSignaling  INTEGER,
11        ifMauAutoNegConfig            INTEGER,
12        ifMauAutoNegRestart           INTEGER,
13        ifMauAutoNegCapabilityBits    IANAifMauAutoNegCapBits,
14        ifMauAutoNegCapAdvertisedBits IANAifMauAutoNegCapBits,
15        ifMauAutoNegCapReceivedBits   IANAifMauAutoNegCapBits,
16        ifMauAutoNegRemoteFaultAdvertised INTEGER,
17        ifMauAutoNegRemoteFaultReceived INTEGER
18    }
19
20 ifMauAutoNegAdminStatus OBJECT-TYPE
21     SYNTAX      INTEGER {
22         enabled(1),
23         disabled(2)
24     }
25
26     MAX-ACCESS  read-write
27     STATUS      current
28     DESCRIPTION "Setting this object to enabled(1) will cause
29                 the interface that has the Auto-Negotiation
30                 signaling ability to be enabled.
31
32                 If the value of this object is disabled(2) then
33                 the interface will act as it would if it had no
34                 Auto-Negotiation signaling. Under these
35                 conditions, an IEEE 802.3 MAU will immediately
36                 be forced to the state indicated by the value of
37                 the object ifMauDefaultType.
38
39                 When ifMauAutoNegAdminStatus transitions from enabled
40                 to disabled, the agent implementation shall
41                 verify that the operational type of the MAU (as
42                 reported by ifMauType) correctly transitions to
43                 the value specified by the ifMauDefaultType
44                 object, rather than continuing to operate at the
45                 value earlier determined by the Auto-Negotiation
46                 function."
47
48     REFERENCE   "IEEE Std 802.3, 30.6.1.1.2, aAutoNegAdminState,
49                 and 30.6.1.2.2, acAutoNegAdminControl."
50
51     ::= { ifMauAutoNegEntry 1 }
52
53 ifMauAutoNegRemoteSignaling OBJECT-TYPE
54     SYNTAX      INTEGER {
55         detected(1),
56         notdetected(2)
57     }
58
59     MAX-ACCESS  read-only
60     STATUS      current
61     DESCRIPTION "A value indicating whether the remote end of
62                 the link is using Auto-Negotiation signaling. It
63                 takes the value detected(1) if and only if,
64                 during the previous link negotiation, FLP Bursts
65

```

```

1           were received."
2     REFERENCE "IEEE Std 802.3, 30.6.1.1.3,
3               aAutoNegRemoteSignaling."
4     ::= { ifMauAutoNegEntry 2 }
5
6   ifMauAutoNegConfig OBJECT-TYPE
7     SYNTAX      INTEGER {
8
9               other(1),
10              configuring(2),
11              complete(3),
12              disabled(4),
13              parallelDetectFail(5)
14            }
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION "A value indicating the current status of the
18                 Auto-Negotiation process. The enumeration
19                 parallelDetectFail(5) maps to a failure in
20                 parallel detection as defined in 28.2.3.1 of
21                 IEEE Std 802.3."
22     REFERENCE  "IEEE Std 802.3, 30.6.1.1.4, aAutoNegAutoConfig."
23     ::= { ifMauAutoNegEntry 4 }
24
25
26   ifMauAutoNegRestart OBJECT-TYPE
27     SYNTAX      INTEGER {
28
29               restart(1),
30               norestart(2)
31            }
32     MAX-ACCESS  read-write
33     STATUS      current
34     DESCRIPTION "If the value of this object is set to
35                 restart(1) then this will force Auto-Negotiation
36                 to begin link renegotiation. If Auto-Negotiation
37                 signaling is disabled, a write to this object
38                 has no effect.
39                 Setting the value of this object to norestart(2)
40                 has no effect."
41     REFERENCE  "IEEE Std 802.3, 30.6.1.2.1,
42                 acAutoNegRestartAutoConfig."
43     ::= { ifMauAutoNegEntry 5 }
44
45
46   ifMauAutoNegCapabilityBits OBJECT-TYPE
47     SYNTAX      IANAifMauAutoNegCapBits
48     MAX-ACCESS  read-only
49     STATUS      current
50     DESCRIPTION "A value that uniquely identifies the set of
51                 capabilities of the local Auto-Negotiation
52                 entity. Note that interfaces that support this
53                 MIB may have capabilities that extend beyond the
54                 scope of this MIB.
55
56
57                 Note that the local Auto-Negotiation entity may
58                 support some capabilities beyond the scope of
59                 this MIB. This is indicated by returning the
60                 bit value bOther in addition to any bit values
61                 for standard capabilities that are listed in the
62                 IANAifMauAutoNegCapBits TC."
63     REFERENCE  "IEEE Std 802.3, 30.6.1.1.5,
64                 aAutoNegLocalTechnologyAbility."
65

```

```
1      ::= { ifMauAutoNegEntry 6 }
2
3  ifMauAutoNegCapAdvertisedBits OBJECT-TYPE
4      SYNTAX      IANAifMauAutoNegCapBits
5      MAX-ACCESS  read-write
6      STATUS      current
7      DESCRIPTION "A value that uniquely identifies the set of
8                  capabilities advertised by the local
9                  Auto-Negotiation entity.
10
11                  Capabilities in this object that are not
12                  available in ifMauAutoNegCapabilityBits cannot
13                  be enabled.
14
15                  Note that the local Auto-Negotiation entity may
16                  advertise some capabilities beyond the scope of
17                  this MIB. This is indicated by returning the
18                  bit value bOther in addition to any bit values
19                  for standard capabilities that are listed in the
20                  IANAifMauAutoNegCapBits TC."
21      REFERENCE   "IEEE Std 802.3, 30.6.1.1.6,
22                  aAutoNegAdvertisedTechnologyAbility."
23      ::= { ifMauAutoNegEntry 7 }
24
25  ifMauAutoNegCapReceivedBits OBJECT-TYPE
26      SYNTAX      IANAifMauAutoNegCapBits
27      MAX-ACCESS  read-only
28      STATUS      current
29      DESCRIPTION "A value that uniquely identifies the set of
30                  capabilities received from the remote
31                  Auto-Negotiation entity.
32
33                  Note that interfaces that support this MIB may
34                  be attached to remote Auto-Negotiation entities
35                  that have capabilities beyond the scope of this
36                  MIB. This is indicated by returning the bit
37                  value bOther in addition to any bit values for
38                  standard capabilities that are listed in the
39                  IANAifMauAutoNegCapBits TC."
40      REFERENCE   "IEEE Std 802.3, 30.6.1.1.7,
41                  aAutoNegReceivedTechnologyAbility."
42      ::= { ifMauAutoNegEntry 8 }
43
44  ifMauAutoNegRemoteFaultAdvertised OBJECT-TYPE
45      SYNTAX      INTEGER {
46          noError(1),
47          offline(2),
48          linkFailure(3),
49          autoNegError(4)
50      }
51      MAX-ACCESS  read-write
52      STATUS      current
53      DESCRIPTION "A value that identifies any local fault
54                  indications that this MAU has detected and will
55                  advertise at the next Auto-Negotiation
56                  interaction for 1000 Mb/s MAUs."
57      REFERENCE   "IEEE Std 802.3, 30.6.1.1.6,
58                  aAutoNegAdvertisedTechnologyAbility."
59      ::= { ifMauAutoNegEntry 9 }
60
61
62
63
64
65
```

```
1      ifMauAutoNegRemoteFaultReceived OBJECT-TYPE
2          SYNTAX      INTEGER {
3              noError(1),
4              offline(2),
5              linkFailure(3),
6              autoNegError(4)
7          }
8
9      MAX-ACCESS      read-only
10     STATUS          current
11     DESCRIPTION     "A value that identifies any fault indications
12                     received from the far end of a link by the
13                     local Auto-Negotiation entity for 1000 Mb/s
14                     MAUs."
15     REFERENCE       "IEEE Std 802.3, 30.6.1.1.7,
16                     aAutoNegReceivedTechnologyAbility."
17     ::= { ifMauAutoNegEntry 10 }
18
19
20     -- Placeholder to preserve module structure and assignments
21     dot3Placeholder OBJECT-TYPE
22         SYNTAX      INTEGER {
23             placeholder(1)
24         }
25         MAX-ACCESS      read-only
26         STATUS          current
27         DESCRIPTION     "A placeholder object to preserve the assignments
28                         that follow in the module. The assignment was given
29                         to the object broadMauBasicTable in earlier
30                         versions of this module. Preserving the assignments that
31                         follow is considered important because they are used for
32                         the IANA-MAU-MIB to assign as MAU type values."
33         REFERENCE       "none"
34         ::= { dot3PlaceholderGroup 1 }
35
36
37     -- Notifications for use by 802.3 MAUs
38
39     snmpDot3MauTraps OBJECT IDENTIFIER ::= { ieee8023snmpDot3MauMgt 0 }
40
41
42     rpMauJabberTrap NOTIFICATION-TYPE
43         OBJECTS      { rpMauJabberState }
44         STATUS          current
45         DESCRIPTION     "This trap is sent whenever a managed repeater
46                         MAU enters the jabber state.
47
48                         The agent shall limit the generation of
49                         consecutive rpMauJabberTraps so that there is at
50                         least a five-second gap between them."
51         REFERENCE       "IEEE Std 802.3, 30.5.1.3.1, nJabber notification."
52         ::= { snmpDot3MauTraps 1 }
53
54
55     ifMauJabberTrap NOTIFICATION-TYPE
56         OBJECTS      { ifMauJabberState }
57         STATUS          current
58         DESCRIPTION     "This trap is sent whenever a managed interface
59                         MAU enters the jabber state.
60
61                         The agent shall limit the generation of
62                         consecutive ifMauJabberTraps so that there is at
63                         least a five-second gap between them."
64
65
```

```
1      REFERENCE      "IEEE Std 802.3, 30.5.1.3.1, nJabber notification."
2      ::= { snmpDot3MauTraps 2 }
3
4  -- Conformance statements
5
6  mauModConf
7      OBJECT IDENTIFIER ::= { ieee8023mauMIB 2 }
8  mauModCompls
9      OBJECT IDENTIFIER ::= { mauModConf 1 }
10 mauModObjGrps
11     OBJECT IDENTIFIER ::= { mauModConf 2 }
12 mauModNotGrps
13     OBJECT IDENTIFIER ::= { mauModConf 3 }
14
15
16 -- Object groups
17 mauRpGrpBasic OBJECT-GROUP
18     OBJECTS      { rpMauType,
19                   rpMauStatus,
20                   rpMauMediaAvailable,
21                   rpMauMediaAvailableStateExits,
22                   rpMauJabberState,
23                   rpMauJabberingStateEnters
24                 }
25     STATUS      current
26     DESCRIPTION "Basic conformance group for MAUs attached to
27                 repeater ports. This group is also the
28                 conformance specification for RFC 1515
29                 implementations."
30     ::= { mauModObjGrps 1 }
31
32
33 mauRpGrp100Mbs OBJECT-GROUP
34     OBJECTS      { rpMauFalseCarriers }
35     STATUS      current
36     DESCRIPTION "Conformance group for MAUs attached to
37                 repeater ports with 100 Mb/s or greater
38                 capability."
39     ::= { mauModObjGrps 2 }
40
41
42 mauRpGrpJack OBJECT-GROUP
43     OBJECTS      { rpJackType }
44     STATUS      current
45     DESCRIPTION "Conformance group for MAUs attached to
46                 repeater ports with managed jacks."
47     ::= { mauModObjGrps 3 }
48
49
50 mauIfGrpBasic OBJECT-GROUP
51     OBJECTS      { ifMauType,
52                   ifMauStatus,
53                   ifMauMediaAvailable,
54                   ifMauMediaAvailableStateExits,
55                   ifMauJabberState,
56                   ifMauJabberingStateEnters,
57                   dot3Placeholder
58                 }
59     STATUS      current
60     DESCRIPTION "Basic conformance group for MAUs attached to
61                 interfaces. This group also provides a
62                 conformance specification for RFC 1515
63                 implementations."
64
65
```

```
1      ::= { mauModObjGrps 4 }
2
3  mauIfGrpJack OBJECT-GROUP
4      OBJECTS      { ifJackType }
5      STATUS       current
6      DESCRIPTION  "Conformance group for MAUs attached to
7                  interfaces with managed jacks."
8      ::= { mauModObjGrps 5 }
9
10
11  mauIfGrpHighCapacity OBJECT-GROUP
12      OBJECTS      { ifMauFalseCarriers,
13                  ifMauTypeListBits,
14                  ifMauDefaultType,
15                  ifMauAutoNegSupported
16                  }
17      STATUS       current
18      DESCRIPTION  "Conformance group for MAUs attached to
19                  interfaces with 100 Mb/s or greater capability."
20      ::= { mauModObjGrps 6 }
21
22
23  mauIfGrpAutoNeg2 OBJECT-GROUP
24      OBJECTS      { ifMauAutoNegAdminStatus,
25                  ifMauAutoNegRemoteSignaling,
26                  ifMauAutoNegConfig,
27                  ifMauAutoNegCapabilityBits,
28                  ifMauAutoNegCapAdvertisedBits,
29                  ifMauAutoNegCapReceivedBits,
30                  ifMauAutoNegRestart
31                  }
32      STATUS       current
33      DESCRIPTION  "Conformance group for MAUs attached to
34                  interfaces with managed Auto-Negotiation."
35      ::= { mauModObjGrps 7 }
36
37
38  mauIfGrpAutoNeg1000Mbps OBJECT-GROUP
39      OBJECTS      { ifMauAutoNegRemoteFaultAdvertised,
40                  ifMauAutoNegRemoteFaultReceived
41                  }
42      STATUS       current
43      DESCRIPTION  "Conformance group for 1000 Mb/s MAUs attached to
44                  interfaces with managed Auto-Negotiation."
45      ::= { mauModObjGrps 8 }
46
47
48  mauIfGrpHCStats OBJECT-GROUP
49      OBJECTS      { ifMauHCFALSECarriers,
50                  ifMauPCSCodingViolations
51                  }
52      STATUS       current
53      DESCRIPTION  "Conformance for high capacity statistics for
54                  MAUs attached to interfaces."
55      ::= { mauModObjGrps 9 }
56
57
58  mauIfGrpFEC OBJECT-GROUP
59      OBJECTS      { ifMauFECAbility,
60                  ifMauFECMode,
61                  ifMauFECCorrectedBlocks,
62                  ifMauFECUncorrectableBlocks
63                  }
64      STATUS       current
65
```



```

1      DESCRIPTION "Conformance for FEC capable
2                  MAUs attached to interfaces."
3      ::= { mauModObjGrps 10 }
4
5  mauIfGrpSNR OBJECT-GROUP
6      OBJECTS      { ifMauSNROpMarginChnlA,
7                    ifMauSNROpMarginChnlB,
8                    ifMauSNROpMarginChnlC,
9                    ifMauSNROpMarginChnlD
10                   }
11
12      STATUS      current
13      DESCRIPTION "Conformance for SNR operating margin reporting
14                  MAUs attached to interfaces."
15      ::= { mauModObjGrps 11 }
16
17  mauIfGrpEEE OBJECT-GROUP
18      OBJECTS      { ifMauEEESupportList,
19                    ifMauEEELDFastRetrainCount,
20                    ifMauEEELPFastRetrainCount
21                   }
22
23      STATUS      current
24      DESCRIPTION "Conformance EEE support and Fast Retrain count
25                  reporting MAUs attached to interfaces."
26      ::= { mauModObjGrps 12 }
27
28  mauIfGrpTimeSync OBJECT-GROUP
29      OBJECTS      { ifMauTimeSyncCapabilityTX,
30                    ifMauTimeSyncCapabilityRX,
31                    ifMauTimeSyncDelayTXmax,
32                    ifMauTimeSyncDelayTXmin,
33                    ifMauTimeSyncDelayRXmax,
34                    ifMauTimeSyncDelayRXmin
35                   }
36
37      STATUS      current
38      DESCRIPTION "Conformance Time Sync support and delay
39                  reporting MAUs attached to interfaces."
40      ::= { mauModObjGrps 13 }
41
42  mauIfGrpPerPCSLaneStats OBJECT-GROUP
43      OBJECTS      { ifMauPPLFECCorrectedBlocks,
44                    ifMauPPLFECUncorrectableBlocks,
45                    ifMauBIPErrorCount,
46                    ifMauPCStoPHYLaneMapping
47                   }
48
49      STATUS      current
50      DESCRIPTION "Conformance Per-PCS lane statistics
51                  reporting MAUs attached to interfaces."
52      ::= { mauModObjGrps 14 }
53
54  -- Notification groups
55
56  rpMauNotifications NOTIFICATION-GROUP
57      NOTIFICATIONS { rpMauJabberTrap }
58
59      STATUS      current
60      DESCRIPTION "Notifications for repeater MAUs."
61      ::= { mauModNotGrps 1 }
62
63  ifMauNotifications NOTIFICATION-GROUP
64      NOTIFICATIONS { ifMauJabberTrap }
65

```

```
1      STATUS      current
2      DESCRIPTION "Notifications for interface MAUs."
3      ::= { mauModNotGrps 2 }
4
5  -- Compliance statements
6
7  mauModRpCompl2 MODULE-COMPLIANCE
8      STATUS      current
9      DESCRIPTION "Compliance for MAUs attached to repeater
10                 ports.
11
12                 Note that compliance with this compliance
13                 statement requires compliance with the
14                 snmpRpPtrModCompl MODULE-COMPLIANCE statement of
15                 the IEEE8023-SNMP-REPEATER-MIB defined in Clause 7."
16
17  MODULE -- this module
18      MANDATORY-GROUPS { mauRpGrpBasic }
19
20      GROUP      mauRpGrp100Mbs
21      DESCRIPTION "Implementation of this optional group is
22                 recommended for MAUs that have 100 Mb/s or
23                 greater capability."
24
25      GROUP      mauRpGrpJack
26      DESCRIPTION "Implementation of this optional group is
27                 recommended for MAUs that have one or more
28                 external jacks."
29
30      GROUP      rpMauNotifications
31      DESCRIPTION "Implementation of this group is recommended
32                 for MAUs attached to repeater ports."
33
34      OBJECT      rpMauStatus
35      MIN-ACCESS  read-only
36      DESCRIPTION "Write access is not required."
37      ::= { mauModCompls 1 }
38
39  mauModIfCompl3 MODULE-COMPLIANCE
40      STATUS      current
41      DESCRIPTION "Compliance for MAUs attached to interfaces.
42
43                 Note that compliance with this compliance
44                 statement requires compliance with the
45                 ifCompliance3 MODULE-COMPLIANCE statement of the
46                 IF-MIB (RFC 2863) and the dot3Compliance2
47                 MODULE-COMPLIANCE statement of the
48                 IEEE8023-EtherLike-MIB defined in Clause 10."
49
50  MODULE -- this module
51      MANDATORY-GROUPS { mauIfGrpBasic }
52
53      GROUP      mauIfGrpHighCapacity
54      DESCRIPTION "Implementation of this optional group is
55                 recommended for MAUs that have 100 Mb/s
56                 or greater capability."
57
58      GROUP      mauIfGrpHCStats
59      DESCRIPTION "Implementation of this group is mandatory
```

```
1          for MAUs that have 1000 Mb/s capacity, and
2          is recommended for MAUs that have 100 Mb/s
3          capacity."
4
5
6          GROUP      mauIfGrpJack
7          DESCRIPTION "Implementation of this optional group is
8                      recommended for MAUs that have one or more
9                      external jacks."
10
11         GROUP      mauIfGrpAutoNeg2
12         DESCRIPTION "Implementation of this group is mandatory
13                     for MAUs that support managed
14                     Auto-Negotiation."
15
16         GROUP      mauIfGrpAutoNeg1000Mbps
17         DESCRIPTION "Implementation of this group is mandatory
18                     for MAUs that have 1000 Mb/s or greater
19                     capability and support managed
20                     Auto-Negotiation."
21
22         GROUP      ifMauNotifications
23         DESCRIPTION "Implementation of this group is recommended
24                     for MAUs attached to interfaces."
25
26         OBJECT      ifMauStatus
27         MIN-ACCESS   read-only
28         DESCRIPTION "Write access is not required."
29
30         GROUP      mauIfGrpFEC
31         DESCRIPTION "Implementation of this optional group is
32                     recommended for MAUs that incorporate FEC."
33
34         GROUP      mauIfGrpSNR
35         DESCRIPTION "Implementation of this optional group is
36                     recommended for MAUs that report SNR operating
37                     margin."
38
39         GROUP      mauIfGrpEEE
40         DESCRIPTION "Implementation of this group is
41                     mandatory for MAUs that support EEE."
42
43         GROUP      mauIfGrpTimeSync
44         DESCRIPTION "Implementation of this group is
45                     mandatory for MAUs that support Time Sync"
46
47         GROUP      mauIfGrpPerPCSLaneStats
48         DESCRIPTION "Implementation of this group is
49                     mandatory for MAUs that report per-PCS lane
50                     statistics."
51
52 ::= { mauModCompls 2 }
```

END

Annex 12A

(informative)

Collection of performance data using WIS MDIO registers

The purpose of this annex is to illustrate how the WIS MDIO registers specified in 45.2.2 of IEEE Std 802.3 (and more specifically the subset required by 50.3.11 of IEEE Std 802.3) can be used to collect performance data either according to the conventions adopted by this document or according to the conventions specified in Clause 30 of IEEE Std 802.3.

For an agent implementing the SNMP managed objects required by this document, the first step in collecting WIS performance data would be to poll the 10G WIS status 3 register and the various error count registers (10G WIS section BIP error count, 10G WIS line BIP errors, 10G WIS far end line BIP errors, 10G WIS path block error count, and 10G WIS far end path block error count) once per second. The 10G WIS status 3 register bits are all latched until read and so would indicate whether a given defect occurred any time during the previous second. The error count registers roll over modulo 2^{16} or 2^{32} , and so to find the number of errors within the previous second, the agent would need to subtract (modulo 2^{16} or 2^{32}) the current reading from the reading taken 1 second ago. Armed with that information, the agent could determine for any layer whether the 1-second interval was an errored second, a severely errored second (that requires comparison with a threshold unless a defect is present), or a severely errored frame second. Determining whether a given second is or is not part of unavailable time requires additional logic; the most straightforward and accurate method is the delay-line approach outlined in Appendix A of IETF RFC 3592. With that information available, the agent would be able to determine by how much each current count should be incremented (including effects of inhibiting). Implementations that conform to ANSI T1.231-1997 would end each 15-minute interval on time-of-day clock 1/4 hour boundaries; if the delay-line approach is used, then a time-of-day timestamp would accompany the 1-second statistics. At the end of each interval, the current registers would be pushed onto the history stack and then would be cleared. The `xyxIntervalValidData` flags would be set to False(2) if the number of samples was not between 890 and 910 or, in the case of far-end counts, if a near-end defect occurred during the just-completed interval (see Section 9.1.2.2 of ANSI T1.231-1997 for details).

An agent implementing the oWIS objects of Clause 30 of IEEE Std 802.3 could also start by polling the 10G WIS status 3 register and the various error count registers to find the defects and error counts for the previous second, and it could determine the number of errors and whether the second was an errored second, a severely errored second, or a severely errored frame second in the same manner as above. The rest of the process would simply be to increment the generalized non-resettable counters without consideration of any inhibiting rules.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Annex A

(informative)

Bibliography

[B1] Beili, E., "Interface Stack Capability MIB."²²

[B2] Case, J., and C. Partridge, "Case diagrams: A first step to diagrammed management information bases," *Computer Communications Review*, vol. 19, no. 1, pp. 13–16, Jan. 1989.

[B3] Faulk, Jr., R. L., C. A. Black, and A. S. Fraley, "Detection on a network by a mapping application of a relative location of a first device to a second device," U.S. Patent 5,293,635, filed Apr. 30, 1991 and issued Mar. 8, 1994.

[B4] Faulk, Jr. R. L., and C. A. Black, "Detection of a relative location of a network device using a multicast packet processed only by hubs," U.S. Patent 5,421,024, filed Aug. 30, 1993 and issued May 30, 1995.

Editor's Note (to be removed prior to publication):

Reference to IEEE Std 802.1Q was removed per Maintenance Request 1383 (see https://www.ieee802.org/3/maint/requests/maint_1383.pdf) and remaining references were renumbered.

[B5] IEEE Std 802.9TM, IEEE Standard for Local and Metropolitan Area Networks: Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers.^{23, 24}

[B6] IETF RFC 1157, Simple Network Management Protocol, Case, J., Fedor, M., Schoffstall, M., and Davin, J., May 1990.²⁵

[B7] IETF RFC 1573, Evolution of the Interfaces Group of MIB-II, McCloghrie, K., and Kastenholz, F., Jan. 1994.

[B8] IETF RFC 1212, Concise MIB Definitions, Rose, M., and McCloghrie, K., Mar. 1991.

[B9] IETF RFC 1284, Definitions of Managed Objects for Ethernet-Like Interface Types, Cook, J., Dec. 1991.

[B10] IETF RFC 1368, Definitions of Managed Objects for IEEE 802.3 Repeater Devices, McMaster, D., and McCloghrie, K., Oct. 1992.

[B11] IETF RFC 1369, Implementation Notes and Experience for The Internet Ethernet MIB, Kastenholz, F., Oct. 1992.

[B12] IETF RFC 1398, Definitions of Managed Objects for the Ethernet-like Interface Types, Kastenholz, F., Jan. 1993.

[B13] IETF RFC 1515, Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs), McMaster, D., McCloghrie, K., and Roberts, S., Sept. 1993.

²²This document is available at the following URL: <https://datatracker.ietf.org/doc/draft-ietf-opsawg-rfc5066bis/>.

²³IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

²⁴The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

²⁵ETF documents (i.e., RFCs) are available for download at <http://www.rfc-archive.org/>.

- [B14] IETF RFC 1525, Definitions of Managed Objects for Source Routing Bridges, Decker, E., McCloghrie, K., Langille, P., and Rijssinghani, A., Sept. 1993.
- [B15] IETF RFC 1643, Definitions of Managed Objects for the Ethernet-like Interface Types, Kastenholz, F., July 1994.
- [B16] IETF RFC 1650, Definitions of Managed Objects for the Ethernet-like Interface Types using SMIPv2, Kastenholz, F., Aug. 1994.
- [B17] IETF RFC 1905, Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2), Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., Jan. 1996.
- [B18] IETF RFC 2026, The Internet Standards Process - Revision 3, Bradner, S., Oct. 1996.
- [B19] IETF RFC 2108, Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIPv2, de Graaf, K., Romascanu, D., McMaster, D., and McCloghrie, K., Feb. 1997.
- [B20] IETF RFC 2239, Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs) using SMIPv2, de Graaf, K., Romascanu, D., McMaster, D., McCloghrie, K., and Roberts, S., Nov. 1997.
- [B21] IETF RFC 2358, Definitions of Managed Objects for the Ethernet-like Interface Types, Flick, J., and Johnson, J., June 1998.
- [B22] IETF RFC 2665, Definitions of Managed Objects for the Ethernet-like Interface Types, Flick, J., and Johnson, J., Aug. 1999.
- [B23] IETF RFC 2666, Definitions of Object Identifiers for Identifying Ethernet Chip Sets, Flick, J., Aug. 1999.
- [B24] IETF RFC 2668, Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs), Smith, A., Flick, J., de Graaf, K., Romascanu, D., McMaster, D., McCloghrie, K., and Roberts, S., Aug. 1999.
- [B25] IETF RFC 3418, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), Presuhn, R., Dec. 2002.
- [B26] IETF RFC 3621, Power Ethernet MIB, Berger, A., and Romascanu, D., Dec. 2003.
- [B27] IETF RFC 3636, Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs), Flick, J., Sept. 2003.
- [B28] IETF RFC 3635, Definitions of Managed Objects for the Ethernet-like Interface Types, Flick, J., Sept. 2003.
- [B29] IETF RFC 3637, Definitions of Managed Objects for the Ethernet WAN Interface Sublayer, Heard, C., Sept. 2003.
- [B30] IETF RFC 4070, Definitions of Managed Object Extensions for Very High Speed Digital Subscriber Lines (VDSL) Using Multiple Carrier Modulation (MCM) Line Coding, Dodge, M., and Ray, B., May 2005.
- [B31] IETF RFC 4181, Guidelines for Authors and Reviewers of MIB Documents, Heard, C., Sept. 2005.
- [B32] IETF RFC 4188, Definitions of Managed Objects for Bridges, Norseth, K., and Bell, E., Sept. 2005.

- [B33] IETF RFC 4319, Definitions of Managed Objects for High Bit-Rate DSL—2nd generation (HDSL2) and Single-Pair High-Speed Digital Subscriber Line (SHDSL) Lines, Sikes, C., Ray, B., and Abbi, R., Dec. 2005.
- [B34] IETF RFC 4836, Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs), Beili, E., Apr. 2007.
- [B35] IETF RFC 4837, Managed Objects of Ethernet Passive Optical Networks (EPON), Kharmosh, L., July 2007.
- [B36] IETF RFC 4878, Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces, Squire, M., June 2007.
- [B37] IETF RFC 5066, Ethernet in the First Mile Copper (EFMCu) Interfaces MIB, Beili, E., Nov. 2007.
- [B38] IETF, “Ethernet Interfaces and Hub MIB (hubmib) Charter.”²⁶
- [B39] Internet Assigned Numbers Authority (IANA), “IANAifType Textual Convention definition.”²⁷
- [B40] ISO/IEC 8824:1990, Information technology—Abstract Syntax Notation One (ASN.1).²⁸
- [B41] ISO/IEC 10165-4:1992, Information technology—Open Systems Interconnection—Structure of management information—Part 4: Guidelines for the definition of managed objects.
- [B42] ISO/IEC 10165-5:1994, Information technology—Open Systems Interconnection—Structure of management information: Generic management information.
- [B43] Kastenholz, F., “IEEE 802.3 Layer Management Draft compatible MIB for TCP/IP Networks,” electronic mail message to mib-wg@nnsf.net, 9 June 1989.
- [B44] Network Interoperability Consultative Committee (NICC), “Specification of the Access Network Frequency Plan (ANFP) applicable to transmission systems used on the BT Access Network,” NICC Document ND1602:2005/08, Aug. 2005.²⁹

²⁶This document is available at the following URL: <http://www.ietf.org/html.charters/OLD/hubmib-charter.html>.

²⁷This document is available at the following URL: <http://www.iana.org/assignments/ianaiftype-mib>.

²⁸ISO/IEC publications are available from the ISO Central Secretariat (<http://www.iso.org/>). ISO publications are also available in the United States from the American National Standards Institute (<http://www.ansi.org/>).

²⁹NICC publications are available from the Network Interoperability Consultative Committee (<http://www.nicstandards.org.uk/>).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Annex B

(normative)

Branch and leaf assignments for IEEE 802.3 and IEEE 802.3.1 managed objects

This annex formally defines the branch and leaf assignments for the IEEE 802.3 and IEEE 802.3.1 managed objects. The branch and leaf assignments currently specified in this annex supercede any object identifiers (OIDs) formerly specified in this annex.

One use for these branch and leaf assignments can be found in Clause 57 of IEEE Std 802.3, which defines OAM, for example the variable descriptor format found in 57.6.1 of IEEE Std 802.3.

B.1 Branch and leaf table

An ASCII machine readable extract of Table B-1 can be obtained at the following URL³⁰:

http://www.ieee802.org/3/1/public/mib_modules/20130411/802dot3dot1AB.txt

Table B-1 contains the branch and leaf assignments for Ethernet managed objects. The branch and leaf assignments are provided for use in the variable descriptors found in Clause 57 of IEEE Std 802.3.

Table B-1—Branch and leaf assignments for managed objects

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aResourceTypeIDName	ATTRIBUTE	GET	7	1
aResourceInfo	ATTRIBUTE	GET	7	2
oMACEntity	OBJECT	GET	3	1
aMACID	ATTRIBUTE	GET	7	1
aFramesTransmittedOK	ATTRIBUTE	GET	7	2
aSingleCollisionFrames	ATTRIBUTE	GET	7	3
aMultipleCollisionFrames	ATTRIBUTE	GET	7	4
aFramesReceivedOK	ATTRIBUTE	GET	7	5
aFrameCheckSequenceErrors	ATTRIBUTE	GET	7	6
aAlignmentErrors	ATTRIBUTE	GET	7	7
aOctetsTransmittedOK	ATTRIBUTE	GET	7	8

³⁰Copyright release for branch and leaf table: Users of this standard may freely reproduce the branch and leaf table contained in this subclause so that it can be used for its intended purpose.

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aFramesWithDeferredXmissions	ATTRIBUTE	GET	7	9
aLateCollisions	ATTRIBUTE	GET	7	10
aFramesAbortedDueToXSColls	ATTRIBUTE	GET	7	11
aFramesLostDueToIntMACXmitError	ATTRIBUTE	GET	7	12
aCarrierSenseErrors	ATTRIBUTE	GET	7	13
aOctetsReceivedOK	ATTRIBUTE	GET	7	14
aFramesLostDueToIntMACRcvError	ATTRIBUTE	GET	7	15
aPromiscuousStatus	ATTRIBUTE	GET-SET	7	16
aReadMulticastAddressList	ATTRIBUTE	GET	7	17
aMaxFrameLength	ATTRIBUTE	GET	7	357
aSlowProtocolFrameLimit	ATTRIBUTE	GET	7	426
aMulticastFramesXmittedOK	ATTRIBUTE	GET	7	18
aBroadcastFramesXmittedOK	ATTRIBUTE	GET	7	19
aFramesWithExcessiveDeferral	ATTRIBUTE	GET	7	20
aMulticastFramesReceivedOK	ATTRIBUTE	GET	7	21
aBroadcastFramesReceivedOK	ATTRIBUTE	GET	7	22
aInRangeLengthErrors	ATTRIBUTE	GET	7	23
aOutOfRangeLengthField	ATTRIBUTE	GET	7	24
aFrameTooLongErrors	ATTRIBUTE	GET	7	25
aMACEnableStatus	ATTRIBUTE	GET-SET	7	26
aTransmitEnableStatus	ATTRIBUTE	GET-SET	7	27
aMulticastReceiveStatus	ATTRIBUTE	GET-SET	7	28
aReadWriteMACAddress	ATTRIBUTE	GET-SET	7	29
aCollisionFrames	ATTRIBUTE	GET	7	30
aMACCapabilities	ATTRIBUTE	GET	7	89
aDuplexStatus	ATTRIBUTE	GET-SET	7	90
aRateControlAbility	ATTRIBUTE	GET	7	179

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aRateControlStatus	ATTRIBUTE	GET-SET	7	180
aDeferControlAbility	ATTRIBUTE	GET	7	311
aDeferControlStatus	ATTRIBUTE	GET-SET	7	312
acInitializeMAC	ACTION	—	9	1
acAddGroupAddress	ACTION	—	9	2
acDeleteGroupAddress	ACTION	—	9	3
acExecuteSelfTest	ACTION	—	9	4
oPHYEntity	OBJECT	GET	3	2
aPHYID	ATTRIBUTE	GET	7	31
aPHYType	ATTRIBUTE	GET	7	32
aPHYTypeList	ATTRIBUTE	GET	7	33
aSQETestErrors	ATTRIBUTE	GET	7	34
aSymbolErrorDuringCarrier	ATTRIBUTE	GET	7	35
aMIIDetect	ATTRIBUTE	GET	7	36
aPHYAdminState	ATTRIBUTE	GET	7	37
acPHYAdminControl	ACTION	—	9	5
oMACControlEntity	OBJECT	GET	3	8
aMACControlIID	ATTRIBUTE	GET	7	92
aMACControlFunctionsSupported	ATTRIBUTE	GET-SET	7	93
aMACControlFramesTransmitted	ATTRIBUTE	GET	7	94
aMACControlFramesReceived	ATTRIBUTE	GET	7	95
aUnsupportedOpcodesReceived	ATTRIBUTE	GET	7	96
aPFCEnableStatus	ATTRIBUTE	GET	7	415
oMACControlFunctionEntity	OBJECT	GET	3	9
aPAUSELinkDelayAllowance	ATTRIBUTE	GET-SET	7	97
aPAUSEMACCtrlFramesTransmitted	ATTRIBUTE	GET	7	98
aPAUSEMACCtrlFramesReceived	ATTRIBUTE	GET	7	99

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
oMPCP	OBJECT	GET	3	21
aMPCPID	ATTRIBUTE	GET	7	351
aMPCPAdminState	ATTRIBUTE	GET	7	278
aMPCPMode	ATTRIBUTE	GET	7	279
aMPCPLinkID	ATTRIBUTE	GET	7	282
aMPCPRemoteMACAddress	ATTRIBUTE	GET	7	283
aMPCPRegistrationState	ATTRIBUTE	GET	7	284
aMPCPMACCtrlFramesTransmitted	ATTRIBUTE	GET	7	280
aMPCPMACCtrlFramesReceived	ATTRIBUTE	GET	7	281
aMPCPTxGate	ATTRIBUTE	GET	7	315
aMPCPTxRegAck	ATTRIBUTE	GET	7	316
aMPCPTxRegister	ATTRIBUTE	GET	7	317
aMPCPTxRegRequest	ATTRIBUTE	GET	7	318
aMPCPTxReport	ATTRIBUTE	GET	7	319
aMPCPRxGate	ATTRIBUTE	GET	7	320
aMPCPRxRegAck	ATTRIBUTE	GET	7	321
aMPCPRxRegister	ATTRIBUTE	GET	7	322
aMPCPRxRegRequest	ATTRIBUTE	GET	7	318
aMPCPRxReport	ATTRIBUTE	GET	7	324
aMPCPTransmitElapsed	ATTRIBUTE	GET	7	285
aMPCPReceiveElapsed	ATTRIBUTE	GET	7	286
aMPCPRoundTripTime	ATTRIBUTE	GET	7	287
aMPCPDiscoveryWindowsSent	ATTRIBUTE	GET	7	288
aMPCPDiscoveryTimeout	ATTRIBUTE	GET	7	290
aMPCPMaximumPendingGrants	ATTRIBUTE	GET	7	291
acMPCPAdminControl	ACTION	—	9	16
oOAM	OBJECT	GET	3	20

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aOAMID	ATTRIBUTE	GET	7	236
aOAMAdminState	ATTRIBUTE	GET	7	237
aOAMMode	ATTRIBUTE	GET-SET	7	238
aOAMDiscoveryState	ATTRIBUTE	GET	7	333
aOAMRemoteMACAddress	ATTRIBUTE	GET	7	239
aOAMLocalConfiguration	ATTRIBUTE	GET	7	334
aOAMRemoteConfiguration	ATTRIBUTE	GET	7	240
aOAMLocalPDUConfiguration	ATTRIBUTE	GET	7	335
aOAMRemotePDUConfiguration	ATTRIBUTE	GET	7	241
aOAMLocalFlagsField	ATTRIBUTE	GET	7	242
aOAMRemoteFlagsField	ATTRIBUTE	GET	7	243
aOAMLocalRevision	ATTRIBUTE	GET	7	336
aOAMRemoteRevision	ATTRIBUTE	GET	7	244
aOAMLocalState	ATTRIBUTE	GET	7	337
aOAMRemoteState	ATTRIBUTE	GET	7	245
aOAMRemoteVendorOUI	ATTRIBUTE	GET	7	246
aOAMRemoteVendorSpecificInfo	ATTRIBUTE	GET	7	247
aOAMUnsupportedCodesTx	ATTRIBUTE	GET	7	338
aOAMUnsupportedCodesRx	ATTRIBUTE	GET	7	250
aOAMInformationTx	ATTRIBUTE	GET	7	251
aOAMInformationRx	ATTRIBUTE	GET	7	252
aOAMUniqueEventNotificationTx	ATTRIBUTE	GET	7	339
aOAMDuplicateEventNotificationTx	ATTRIBUTE	GET	7	340
aOAMUniqueEventNotificationRx	ATTRIBUTE	GET	7	254
aOAMDuplicateEventNotificationRx	ATTRIBUTE	GET	7	255
aOAMLoopbackControlTx	ATTRIBUTE	GET	7	256
aOAMLoopbackControlRx	ATTRIBUTE	GET	7	257

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aOAMVariableRequestTx	ATTRIBUTE	GET	7	258
aOAMVariableRequestRx	ATTRIBUTE	GET	7	259
aOAMVariableResponseTx	ATTRIBUTE	GET	7	260
aOAMVariableResponseRx	ATTRIBUTE	GET	7	261
aOAMOrganizationSpecificTx	ATTRIBUTE	GET	7	262
aOAMOrganizationSpecificRx	ATTRIBUTE	GET	7	263
aOAMLocalErrSymPeriodConfig	ATTRIBUTE	GET	7	264
aOAMLocalErrSymPeriodEvent	ATTRIBUTE	GET	7	265
aOAMLocalErrFrameConfig	ATTRIBUTE	GET	7	266
aOAMLocalErrFrameEvent	ATTRIBUTE	GET	7	267
aOAMLocalErrFramePeriodConfig	ATTRIBUTE	GET	7	268
aOAMLocalErrFramePeriodEvent	ATTRIBUTE	GET	7	269
aOAMLocalErrFrameSecsSummaryConfig	ATTRIBUTE	GET	7	270
aOAMLocalErrFrameSecsSummaryEvent	ATTRIBUTE	GET	7	271
aOAMRemoteErrSymPeriodEvent	ATTRIBUTE	GET	7	272
aOAMRemoteErrFrameEvent	ATTRIBUTE	GET	7	273
aOAMRemoteErrFramePeriodEvent	ATTRIBUTE	GET	7	274
aOAMRemoteErrFrameSecsSummaryEvent	ATTRIBUTE	GET	7	275
aFramesLostDueToOAMError	ATTRIBUTE	GET	7	276
acOAMAdminControl	ACTION	—	9	15
oOMPEmulation	OBJECT	GET	3	19
aOMPEmulationID	ATTRIBUTE	GET	7	231
aOMPEmulationType	ATTRIBUTE	GET	7	232
aSLDErrors	ATTRIBUTE	GET	7	233
aCRC8Errors	ATTRIBUTE	GET	7	234
aGoodLLID	ATTRIBUTE	GET	7	341
aONUPONcastLLID	ATTRIBUTE	GET	7	342

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aOLTPONcastLLID	ATTRIBUTE	GET	7	343
aBadLLID	ATTRIBUTE	GET	7	235
oRepeater	OBJECT	GET	3	3
aRepeaterID	ATTRIBUTE	GET	7	38
aRepeaterType	ATTRIBUTE	GET	7	39
aRepeaterGroupCapacity	ATTRIBUTE	GET	7	40
aGroupMap	ATTRIBUTE	GET	7	41
aRepeaterHealthState	ATTRIBUTE	GET	7	42
aRepeaterHealthText	ATTRIBUTE	GET	7	43
aRepeaterHealthData	ATTRIBUTE	GET	7	44
aTransmitCollisions	ATTRIBUTE	GET	7	45
acResetRepeater	ACTION	—	9	6
acExecuteNonDisruptiveSelfTest	ACTION	—	9	7
nRepeaterHealth	NOTIFICATION	—	10	1
nRepeaterReset	NOTIFICATION	—	10	2
nGroupMapChange	NOTIFICATION	—	10	3
oGroup	OBJECT	GET	3	4
aGroupID	ATTRIBUTE	GET	7	46
aGroupPortCapacity	ATTRIBUTE	GET	7	47
aPortMap	ATTRIBUTE	GET	7	48
nPortMapChange	NOTIFICATION	—	10	4
oRepeaterPort	OBJECT	GET	3	5
aPortID	ATTRIBUTE	GET	7	49
aPortAdminState	ATTRIBUTE	GET	7	50
aAutoPartitionState	ATTRIBUTE	GET	7	51
aReadableFrames	ATTRIBUTE	GET	7	52
aReadableOctets	ATTRIBUTE	GET	7	53

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aFrameCheckSequenceErrors	ATTRIBUTE	GET	7	54
aAlignmentErrors	ATTRIBUTE	GET	7	55
aFramesTooLong	ATTRIBUTE	GET	7	56
aShortEvents	ATTRIBUTE	GET	7	57
aRunts	ATTRIBUTE	GET	7	58
aCollisions	ATTRIBUTE	GET	7	59
aLateEvents	ATTRIBUTE	GET	7	60
aVeryLongEvents	ATTRIBUTE	GET	7	61
aDataRateMismatches	ATTRIBUTE	GET	7	62
aAutoPartitions	ATTRIBUTE	GET	7	63
aIsolates	ATTRIBUTE	GET	7	64
aSymbolErrorDuringPacket	ATTRIBUTE	GET	7	65
aLastSourceAddress	ATTRIBUTE	GET	7	66
aSourceAddressChanges	ATTRIBUTE	GET	7	67
aBursts	ATTRIBUTE	GET	7	100
acPortAdminControl	ACTION	—	9	8
oMAU	OBJECT	GET	3	6
aMAUID	ATTRIBUTE	GET	7	68
aMAUType	ATTRIBUTE	GET-SET	7	69
aMAUTypeList	ATTRIBUTE	GET	7	70
aMediaAvailable	ATTRIBUTE	GET	7	71
aLoseMediaCounter	ATTRIBUTE	GET	7	72
aJabber	ATTRIBUTE	GET	7	73
aMAUAdminState	ATTRIBUTE	GET	7	74
aBbMAUXmitRcvSplitType	ATTRIBUTE	GET	7	75
aBroadbandFrequencies	ATTRIBUTE	GET	7	76
aFalseCarriers	ATTRIBUTE	GET	7	77

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aIdleErrorCount	ATTRIBUTE	GET	7	91
aSNROpMarginChnlA	ATTRIBUTE	GET	7	353
aSNROpMarginChnlB	ATTRIBUTE	GET	7	354
aSNROpMarginChnlC	ATTRIBUTE	GET	7	355
aSNROpMarginChnlD	ATTRIBUTE	GET	7	356
aPCSCodingViolation	ATTRIBUTE	GET	7	292
aFECAbility	ATTRIBUTE	GET	7	313
aFECmode	ATTRIBUTE	GET-SET	7	314
aFECCorrectedBlocks	ATTRIBUTE	GET	7	293
aFECUncorrectableBlocks	ATTRIBUTE	GET	7	294
acResetMAU	ACTION	—	9	9
acMAUAdminControl	ACTION	—	9	10
nJabber	NOTIFICATION	—	10	5
oAutoNegotiation	OBJECT	GET	3	7
aAutoNegID	ATTRIBUTE	GET	7	78
aAutoNegAdminState	ATTRIBUTE	GET	7	79
aAutoNegRemoteSignaling	ATTRIBUTE	GET	7	80
aAutoNegAutoConfig	ATTRIBUTE	GET-SET	7	81
aAutoNegLocalTechnologyAbility	ATTRIBUTE	GET	7	82
aAutoNegAdvertisedTechnologyAbility	ATTRIBUTE	GET-SET	7	83
aAutoNegReceivedTechnologyAbility	ATTRIBUTE	GET	7	84
aAutoNegLocalSelectorAbility	ATTRIBUTE	GET	7	85
aAutoNegAdvertisedSelectorAbility	ATTRIBUTE	GET-SET	7	86
aAutoNegReceivedSelectorAbility	ATTRIBUTE	GET	7	87
acAutoNegRestartAutoConfig	ACTION	—	9	11
acAutoNegAdminControl	ACTION	—	9	12
oAggregator	OBJECT	GET	3	10

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aAggID	ATTRIBUTE	GET	7	101
aAggDescription	ATTRIBUTE	GET	7	102
aAggName	ATTRIBUTE	GET-SET	7	103
aAggActorSystemID	ATTRIBUTE	GET-SET	7	104
aAggActorSystemPriority	ATTRIBUTE	GET-SET	7	105
aAggAggregateOrIndividual	ATTRIBUTE	GET	7	106
aAggActorAdminKey	ATTRIBUTE	GET-SET	7	107
aAggActorOperKey	ATTRIBUTE	GET	7	108
aAggMACAddress	ATTRIBUTE	GET	7	109
aAggPartnerSystemID	ATTRIBUTE	GET	7	110
aAggPartnerSystemPriority	ATTRIBUTE	GET	7	111
aAggPartnerOperKey	ATTRIBUTE	GET	7	112
aAggAdminState	ATTRIBUTE	GET-SET	7	113
aAggOperState	ATTRIBUTE	GET	7	114
aAggTimeOfLastOperChange	ATTRIBUTE	GET	7	115
aAggDataRate	ATTRIBUTE	GET	7	116
aAggOctetsTxOK	ATTRIBUTE	GET	7	117
aAggOctetsRxOK	ATTRIBUTE	GET	7	118
aAggFramesTxOK	ATTRIBUTE	GET	7	119
aAggFramesRxOK	ATTRIBUTE	GET	7	120
aAggMulticastFramesTxOK	ATTRIBUTE	GET	7	121
aAggMulticastFramesRxOK	ATTRIBUTE	GET	7	122
aAggBroadcastFramesTxOK	ATTRIBUTE	GET	7	123
aAggBroadcastFramesRxOK	ATTRIBUTE	GET	7	124
aAggFramesDiscardedOnTx	ATTRIBUTE	GET	7	125
aAggFramesDiscardedOnRx	ATTRIBUTE	GET	7	126
aAggFramesWithTxErrors	ATTRIBUTE	GET	7	127

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aAggFramesWithRxErrors	ATTRIBUTE	GET	7	128
aAggUnknownProtocolFrames	ATTRIBUTE	GET	7	129
aAggLinkUpDownNotificationEnable	ATTRIBUTE	GET-SET	7	130
nAggLinkUpNotification	NOTIFICATION	—	10	6
nAggLinkDownNotification	NOTIFICATION	—	10	7
aAggPortList	ATTRIBUTE	GET	7	131
aAggCollectorMaxDelay	ATTRIBUTE	GET-SET	7	132
oAggregationPort	OBJECT	GET	3	11
aAggPortID	ATTRIBUTE	GET	7	133
aAggPortActorSystemPriority	ATTRIBUTE	GET-SET	7	134
aAggPortActorSystemID	ATTRIBUTE	GET	7	135
aAggPortActorAdminKey	ATTRIBUTE	GET-SET	7	136
aAggPortActorOperKey	ATTRIBUTE	GET	7	137
aAggPortPartnerAdminSystemPriority	ATTRIBUTE	GET-SET	7	138
aAggPortPartnerOperSystemPriority	ATTRIBUTE	GET	7	139
aAggPortPartnerAdminSystemID	ATTRIBUTE	GET-SET	7	140
aAggPortPartnerOperSystemID	ATTRIBUTE	GET	7	141
aAggPortPartnerAdminKey	ATTRIBUTE	GET-SET	7	142
aAggPortPartnerOperKey	ATTRIBUTE	GET	7	143
aAggPortSelectedAggID	ATTRIBUTE	GET	7	144
aAggPortAttachedAggID	ATTRIBUTE	GET	7	145
aAggPortActorPort	ATTRIBUTE	GET	7	146
aAggPortActorPortPriority	ATTRIBUTE	GET-SET	7	147
aAggPortPartnerAdminPort	ATTRIBUTE	GET-SET	7	148
aAggPortPartnerOperPort	ATTRIBUTE	GET	7	149
aAggPortPartnerAdminPortPriority	ATTRIBUTE	GET-SET	7	150
aAggPortPartnerOperPortPriority	ATTRIBUTE	GET	7	151

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aAggPortActorAdminState	ATTRIBUTE	GET-SET	7	152
aAggPortActorOperState	ATTRIBUTE	GET	7	153
aAggPortPartnerAdminState	ATTRIBUTE	GET-SET	7	154
aAggPortPartnerOperState	ATTRIBUTE	GET	7	155
aAggPortAggregateOrIndividual	ATTRIBUTE	GET	7	156
oAggPortStats	OBJECT	GET	3	12
aAggPortStatsID	ATTRIBUTE	GET	7	157
aAggPortStatsLACPDUssRx	ATTRIBUTE	GET	7	158
aAggPortStatsMarkerPDUsRx	ATTRIBUTE	GET	7	159
aAggPortStatsMarkerResponsePDUsRx	ATTRIBUTE	GET	7	160
aAggPortStatsUnknownRx	ATTRIBUTE	GET	7	161
aAggPortStatsIllegalRx	ATTRIBUTE	GET	7	162
aAggPortStatsLACPDUssTx	ATTRIBUTE	GET	7	163
aAggPortStatsMarkerPDUsTx	ATTRIBUTE	GET	7	164
aAggPortStatsMarkerResponsePDUsTx	ATTRIBUTE	GET	7	165
oAggPortDebugInformation	OBJECT	GET	3	13
aAggPortDebugInformationID	ATTRIBUTE	GET	7	166
aAggPortDebugRxState	ATTRIBUTE	GET	7	167
aAggPortDebugLastRxTime	ATTRIBUTE	GET	7	168
aAggPortDebugMuxState	ATTRIBUTE	GET	7	169
aAggPortDebugMuxReason	ATTRIBUTE	GET	7	170
aAggPortDebugActorChurnState	ATTRIBUTE	GET	7	171
aAggPortDebugPartnerChurnState	ATTRIBUTE	GET	7	172
aAggPortDebugActorChurnCount	ATTRIBUTE	GET	7	173
aAggPortDebugPartnerChurnCount	ATTRIBUTE	GET	7	174
aAggPortDebugActorSyncTransitionCount	ATTRIBUTE	GET	7	175
aAggPortDebugPartnerSyncTransitionCount	ATTRIBUTE	GET	7	176

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aAggPortDebugActorChangeCount	ATTRIBUTE	GET	7	177
aAggPortDebugPartnerChangeCount	ATTRIBUTE	GET	7	178
oWIS	OBJECT	GET	3	14
aWISID	ATTRIBUTE	GET	7	181
aSectionStatus	ATTRIBUTE	GET	7	182
aSectionSESThreshold	ATTRIBUTE	GET-SET	7	183
aSectionSESSs	ATTRIBUTE	GET	7	184
aSectionESs	ATTRIBUTE	GET	7	185
aSectionSEFSs	ATTRIBUTE	GET	7	186
aSectionCVs	ATTRIBUTE	GET	7	187
aJ0ValueTX	ATTRIBUTE	GET-SET	7	188
aJ0ValueRX	ATTRIBUTE	GET	7	189
aLineStatus	ATTRIBUTE	GET	7	190
aLineSESThreshold	ATTRIBUTE	GET-SET	7	191
aLineSESSs	ATTRIBUTE	GET	7	192
aLineESs	ATTRIBUTE	GET	7	193
aLineCVs	ATTRIBUTE	GET	7	194
aFarEndLineSESSs	ATTRIBUTE	GET	7	195
aFarEndLineESs	ATTRIBUTE	GET	7	196
aFarEndLineCVs	ATTRIBUTE	GET	7	197
aPathStatus	ATTRIBUTE	GET	7	198
aPathSESThreshold	ATTRIBUTE	GET-SET	7	199
aPathSESSs	ATTRIBUTE	GET	7	200
aPathESs	ATTRIBUTE	GET	7	201
aPathCVs	ATTRIBUTE	GET	7	202
aJ1ValueTX	ATTRIBUTE	GET-SET	7	203
aJ1ValueRX	ATTRIBUTE	GET	7	204

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aFarEndPathStatus	ATTRIBUTE	GET	7	205
aFarEndPathSESs	ATTRIBUTE	GET	7	206
aFarEndPathESs	ATTRIBUTE	GET	7	207
aFarEndPathCVs	ATTRIBUTE	GET	7	208
oPSE	OBJECT	GET	3	15
aPSEID	ATTRIBUTE	GET	7	209
aPSEAdminState	ATTRIBUTE	GET	7	210
aPSEPowerPairsControlAbility	ATTRIBUTE	GET	7	211
aPSEPowerPairs	ATTRIBUTE	GET-SET	7	212
aPSEPowerDetectionStatus	ATTRIBUTE	GET	7	214
aPSEPowerClassification	ATTRIBUTE	GET	7	215
aPSEInvalidSignatureCounter	ATTRIBUTE	GET	7	227
aPSEPowerDeniedCounter	ATTRIBUTE	GET	7	228
aPSEOverLoadCounter	ATTRIBUTE	GET	7	229
aPSEShortCounter	ATTRIBUTE	GET	7	230
aPSEMPSAbsentCounter	ATTRIBUTE	GET	7	217
acPSEAdminControl	ACTION	—	9	13
aPSEActualPower	ATTRIBUTE	GET	7	427
aPSEPowerAccuracy	ATTRIBUTE	GET	7	428
aPSECumulativeEnergy	ATTRIBUTE	GET	7	429
oMidSpan	OBJECT	GET	3	17
aMidSpanID	ATTRIBUTE	GET	7	221
aMidSpanPSEGroupCapacity	ATTRIBUTE	GET	7	222
aMidSpanPSEGroupMap	ATTRIBUTE	GET	7	223
nMidSpanPSEGroupMapChange	NOTIFICATION	—	10	8
oPSEGroup	OBJECT	GET	3	18
aPSEGroupID	ATTRIBUTE	GET	7	224

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aPSECapacity	ATTRIBUTE	GET	7	225
aPSEMap	ATTRIBUTE	GET	7	226
nPSEMapChange	NOTIFICATION	—	10	9
oPAF	OBJECT	GET	3	24
aPAFID	ATTRIBUTE	GET	7	344
aPhyEnd	ATTRIBUTE	GET	7	326
aPHYCurrentStatus	ATTRIBUTE	GET	7	296
aPAFSupported	ATTRIBUTE	GET	7	304
aPAFAdminState	ATTRIBUTE	GET-SET	7	305
aLocalPAFCapacity	ATTRIBUTE	GET	7	327
aLocalPMEAvailable	ATTRIBUTE	GET	7	306
aLocalPMEAggregate	ATTRIBUTE	GET	7	307
aRemotePAFSupported	ATTRIBUTE	GET	7	328
aRemotePAFCapacity	ATTRIBUTE	GET	7	329
aRemotePMEAggregate	ATTRIBUTE	GET	7	310
oPME	OBJECT	GET	3	25
aPMEID	ATTRIBUTE	GET	7	330
aPMEAdminState	ATTRIBUTE	GET-SET	7	345
aPMEStatus	ATTRIBUTE	GET	7	346
aPMESNRMgn	ATTRIBUTE	GET	7	331
aTCCodingViolations	ATTRIBUTE	GET	7	332
aProfileSelect	ATTRIBUTE	GET-SET	7	347
aOperatingProfile	ATTRIBUTE	GET	7	348
aPMEFECCorrectedBlocks	ATTRIBUTE	GET	7	349
aPMEFECCorrectableBlocks	ATTRIBUTE	GET	7	350
aTCCRCErrors	ATTRIBUTE	GET	7	352
aCMCounter	ATTRIBUTE	GET	7	88

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
oMACEntity	OBJECT	GET	—	—
Basic Package (Mandatory)	PACKAGE	—	—	—
Mandatory Package (Mandatory)	PACKAGE	—	4	1
Recommended Package (Conditional)	PACKAGE	—	4	2
Conditional Package (Conditional)	PACKAGE	—	4	3
Array Package (Conditional)	PACKAGE	—	4	4
ExcessiveDeferral Package (Conditional)	PACKAGE	—	4	5
PHYRecommended Package (Conditional)	PACKAGE	—	4	6
oPHYEntity	OBJECT	GET	—	—
Basic Package (Mandatory)	PACKAGE	—	—	—
MultiplePHY Package (Conditional)	PACKAGE	—	4	7
100MbsMonitor Capability (Conditional)	PACKAGE	—	4	8
oMACControlEntity	OBJECT	GET	—	—
Mandatory Package (Mandatory)	PACKAGE	—	—	—
Recommended Package (Conditional)	PACKAGE	—	4	17
oMACControlFunctionEntity	OBJECT	GET	—	—
Mandatory Package (Mandatory)	PACKAGE	—	—	—
Recommended Package (Conditional)	PACKAGE	—	4	25
oAggregator	OBJECT	GET	—	—
AggregatorBasic Package (Mandatory)	PACKAGE	—	—	—
AggregatorMandatory Package (Mandatory)	PACKAGE	—	4	19
AggregatorRecommended Package (Conditional)	PACKAGE	—	4	20
AggregatorConditional Package (Conditional)	PACKAGE	—	4	21
oAggregationPort	OBJECT	GET	—	—
AggregationPortBasic Package (Mandatory)	PACKAGE	—	—	—
AggregationPortMandatory Package (Mandatory)	PACKAGE	—	4	22
oAggPortStats	OBJECT	GET	—	—

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
AggPortStatsPackage (Conditional)	PACKAGE	—	4	23
oAggPortDebugInformation	OBJECT	GET	—	—
AggPortDebugInformation Package (Conditional)	PACKAGE	—	4	24
oOMP	OBJECT	GET	—	—
Optical Multipoint Emulation Package (Conditional)	PACKAGE	—	—	—
Optical Multipoint Emulation Error Package (Conditional)	PACKAGE	—	4	37
oRepeater	OBJECT	GET	—	—
RepeaterBasicControl Package (Mandatory)	PACKAGE	—	—	—
RepeaterPerfMonitor Package (Conditional)	PACKAGE	—	4	9
oGroup	OBJECT	GET	—	—
GroupBasicControl Package (Mandatory)	PACKAGE	—	—	—
oRepeaterPort	OBJECT	GET	—	—
PortBasicControl Package (Mandatory)	PACKAGE	—	—	—
PortPerfMonitor Package (Conditional)	PACKAGE	—	4	10
PortAddrTracking Package (Conditional)	PACKAGE	—	4	11
100MbpsMonitor Package (Conditional)	PACKAGE	—	4	12
Burst Package (Conditional)	PACKAGE	—	4	18
oMAU	OBJECT	GET	—	—
Basic Package (Mandatory)	PACKAGE	—	—	—
MAUControl Package (Conditional)	PACKAGE	—	4	13
MediaLossTracking Package (Conditional)	PACKAGE	—	4	14
BroadbandDTEMAU Package (Conditional)	PACKAGE	—	4	15
100MbsMonitor Capability (Conditional)	PACKAGE	—	4	16
10GBASE-T Operating Margin package (Conditional)	PACKAGE	—	4	39
PCS Code Error Monitor Capability (Optional)	PACKAGE	—	4	35
Forward Error Correction Capability (Conditional)	PACKAGE	—	4	30
oWIS	OBJECT	GET	—	—

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
WIS Basic Package (Mandatory)	PACKAGE	—	—	—
WIS Recommended Package (Conditional)	PACKAGE	—	4	26
oAutoNegotiation	OBJECT	GET	—	—
Auto Negotiation Package (Mandatory)	PACKAGE	—	—	—
oPSE	OBJECT	GET	—	—
PSEBasic Package (Mandatory)	PACKAGE	—	—	—
PSERecommended Package (Conditional)	PACKAGE	—	4	27
oMidSpan	OBJECT	GET	—	—
MidSpanBasic (Mandatory)	PACKAGE	—	—	—
oPAF	OBJECT	GET	—	—
Basic Package (Mandatory)	PACKAGE	—	—	—
PME Aggregation Package (Optional)	PACKAGE	—	4	38
oPME	OBJECT	GET	—	—
10P/2B Package (Mandatory)	PACKAGE	—	—	—
oResourceTypeID	OBJECT	GET	—	—
MII Package (Conditional)	PACKAGE	—	—	—
oTimeSync	OBJECT	GET	—	—
Support for Time Sync (Mandatory)	PACKAGE	—	4	40
aLldpXdot3LocPowerType	ATTRIBUTE	GET	7	358
aLldpXdot3LocPowerSource	ATTRIBUTE	GET	7	359
aLldpXdot3LocPowerPriority	ATTRIBUTE	GET	7	360
aLldpXdot3LocPDRrequestedPowerValue	ATTRIBUTE	GET	7	361
aLldpXdot3LocPSEAllocatedPowerValue	ATTRIBUTE	GET	7	362
aLldpXdot3LocResponseTime	ATTRIBUTE	GET	7	363
aLldpXdot3LocReady	ATTRIBUTE	GET	7	364
aLldpXdot3LocReducedOperationPowerValue	ATTRIBUTE	GET	7	365
aLldpXdot3RemPowerType	ATTRIBUTE	GET	7	366

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aLdpXdot3RemPowerSource	ATTRIBUTE	GET	7	367
aLdpXdot3RemPowerPriority	ATTRIBUTE	GET	7	368
aLdpXdot3RemPDRRequestedPowerValue	ATTRIBUTE	GET	7	369
aLdpXdot3RemPSEAllocatedPowerValue	ATTRIBUTE	GET	7	370
aTransmitLPIMicroseconds	ATTRIBUTE	GET	7	371
aReceiveLPIMicroseconds	ATTRIBUTE	GET	7	372
aTransmitLPITransitions	ATTRIBUTE	GET	7	373
aReceiveLPITransitions	ATTRIBUTE	GET	7	374
aLDFastRetrainCount	ATTRIBUTE	GET	7	375
aLPFastRetrainCount	ATTRIBUTE	GET	7	376
aEEESupportList	ATTRIBUTE	GET	7	377
oLdpXdot3Config	OBJECT	GET	3	27
LLDP Basic Package (mandatory)	PACKAGE	—	4	41
oLdpXdot3LocSystemsGroup	OBJECT	GET	3	28
LLDP MAC/PHY Configuration/Status Local Package (conditional)	PACKAGE	—	4	42
LLDP Power via MDI Local Package (conditional)	PACKAGE	—	4	44
LLDP Link Aggregation Local Package (conditional)	PACKAGE	—	4	46
LLDP Maximum Frame Size Local Package (conditional)	PACKAGE	—	4	48
LLDP EEE Local Package (optional)	PACKAGE	—	4	50
oLdpXdot3RemSystemsGroup	OBJECT	GET	3	29
LLDP MAC/PHY Configuration/Status Remote Package (conditional)	PACKAGE	—	4	43
LLDP Power via MDI Remote Package (conditional)	PACKAGE	—	4	45
LLDP Link Aggregation Remote Package (conditional)	PACKAGE	—	4	47
LLDP Maximum Frame Size Remote Package (conditional)	PACKAGE	—	4	49
LLDP EEE Remote Package (optional)	PACKAGE	—	4	51
aLdpXdot3LocTxTwSys	ATTRIBUTE	GET	7	378

Table B-1—Branch and leaf assignments for managed objects *(continued)*

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aLdpXdot3LocTxTwSysEcho	ATTRIBUTE	GET	7	379
aLdpXdot3LocRxTwSys	ATTRIBUTE	GET	7	380
aLdpXdot3LocRxTwSysEcho	ATTRIBUTE	GET	7	381
aLdpXdot3LocFbTwSys	ATTRIBUTE	GET	7	382
aLdpXdot3TxDllReady	ATTRIBUTE	GET	7	383
aLdpXdot3RxDllReady	ATTRIBUTE	GET	7	384
aLdpXdot3LocDllEnabled	ATTRIBUTE	GET	7	385
aLdpXdot3RemTxTwSys	ATTRIBUTE	GET	7	386
aLdpXdot3RemTxTwSysEcho	ATTRIBUTE	GET	7	387
aLdpXdot3RemRxTwSys	ATTRIBUTE	GET	7	388
aLdpXdot3RemRxTwSysEcho	ATTRIBUTE	GET	7	389
aLdpXdot3RemFbTwSys	ATTRIBUTE	GET	7	390
aLdpXdot3PortConfigTLVsTxEnable	ATTRIBUTE	GET	7	391
aLdpXdot3LocPortAutoNegSupported	ATTRIBUTE	GET	7	392
aLdpXdot3LocPortAutoNegEnabled	ATTRIBUTE	GET	7	393
aLdpXdot3LocPortAutoNegAdvertisedCap	ATTRIBUTE	GET	7	394
aLdpXdot3LocPortOperMauType	ATTRIBUTE	GET	7	395
aLdpXdot3LocPowerPortClass	ATTRIBUTE	GET	7	396
aLdpXdot3LocPowerMDISupported	ATTRIBUTE	GET	7	397
aLdpXdot3LocPowerMDIEnabled	ATTRIBUTE	GET	7	398
aLdpXdot3LocPowerPairControlable	ATTRIBUTE	GET	7	399
aLdpXdot3LocPowerPairs	ATTRIBUTE	GET	7	400
aLdpXdot3LocPowerClass	ATTRIBUTE	GET	7	401
aLdpXdot3LocMaxFrameSize	ATTRIBUTE	GET	7	402
aLdpXdot3RemPortAutoNegSupported	ATTRIBUTE	GET	7	403
aLdpXdot3RemPortAutoNegEnabled	ATTRIBUTE	GET	7	404
aLdpXdot3RemPortAutoNegAdvertisedCap	ATTRIBUTE	GET	7	405

Table B-1—Branch and leaf assignments for managed objects (continued)

IEEE Std 802.3 Clause 30 object name	Type	Access	BRANCH	LEAF
aLdpXdot3RemPortOperMauType	ATTRIBUTE	GET	7	406
aLdpXdot3RemPowerPortClass	ATTRIBUTE	GET	7	407
aLdpXdot3RemPowerMDISupported	ATTRIBUTE	GET	7	408
aLdpXdot3RemPowerMDIEnabled	ATTRIBUTE	GET	7	409
aLdpXdot3RemPowerPairControlable	ATTRIBUTE	GET	7	410
aLdpXdot3RemPowerPairs	ATTRIBUTE	GET	7	411
aLdpXdot3RemPowerClass	ATTRIBUTE	GET	7	412
aLdpXdot3RemMaxFrameSize	ATTRIBUTE	GET	7	413
oTimeSync	OBJECT	GET	3	26
aTimeSyncCapabilityTX	ATTRIBUTE	GET	7	416
aTimeSyncCapabilityRX	ATTRIBUTE	GET	7	417
aTimeSyncDelayTXmax	ATTRIBUTE	GET	7	418
aTimeSyncDelayTXmin	ATTRIBUTE	GET	7	419
aTimeSyncDelayRXmax	ATTRIBUTE	GET	7	420
aTimeSyncDelayRXmin	ATTRIBUTE	GET	7	421
oEXTENSION	OBJECT	GET	3	30
aEXTENSIONMACCtrlStatus	ATTRIBUTE	GET	7	422
aEXTENSIONMACCtrlFramesTransmitted	ATTRIBUTE	GET	7	423
aEXTENSIONMACCtrlFramesReceived	ATTRIBUTE	GET	7	424
aMPCPRecognizedMulticastIDs	ATTRIBUTE	GET	7	425
ifMauPerPCSLaneStatsTable	ATTRIBUTE	GET	7	430
ifMauPerPCSLaneStatsEntry	ATTRIBUTE	GET	7	431
ifPCSLaneIndex	ATTRIBUTE	GET	7	432
ifMauPPLFECCorrectedBlocks	ATTRIBUTE	GET	7	433
ifMauPPLFECUncorrectableBlocks	ATTRIBUTE	GET	7	434
ifMauBIPErrorsCount	ATTRIBUTE	GET	7	435
ifMauPCStoPHYLaneMapping	ATTRIBUTE	GET	7	436

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65