IEEE STANDARDS ASSOCIATION

IEEE Standard for Ethernet

Amendment 5: Specification and Management Parameters for Interspersing Express Traffic

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

IEEE Std 802.3br™-2016
(Amendment to
IEEE Std 802.3™-2015
as amended by
IEEE Std 802.3bw™-2015,
IEEE Std 802.3by™-2016,
IEEE Std 802.3bq™-2016, and
IEEE Std 802.3bp™-2016)

IEEE Std 802.3br™-2016
(Amendment to
IEEE Std 802.3™-2015
as amended by
IEEE Std 802.3bw™-2015,
IEEE Std 802.3by™-2016,
IEEE Std 802.3bq™-2016, and
IEEE Std 802.3bp™-2016)

IEEE Standard for Ethernet

Amendment 5: Specification and Management Parameters for Interspersing Express Traffic

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 30 June 2016

IEEE-SA Standards Board

Abstract: This amendment to IEEE Std 802.3-2015 specifies additions to and appropriate modifications to add support for interspersing express traffic over a single physical link.

Keywords: Ethernet, express traffic, IEEE 802®, IEEE 802.3™, IEEE 802.3br™, preemption

---

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

Print: ISBN 978-1-5044-2335-9 STD21119
PDF: ISBN 978-1-5044-2336-6 STDPD21119

Important Notices and Disclaimers Concerning IEEE Standards Documents

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

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

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

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

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

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

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

Translations

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

Official statements

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

Comments on standards

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

Comments on standards should be submitted to the following address:

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

Laws and regulations

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

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

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

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

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

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://ieeexplore.ieee.org/Xplore/home.jsp or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

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

Patents

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

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

The following individuals were officers and members of the IEEE 802.3 Working Group at the beginning of the IEEE P802.3br Working Group ballot. Individuals may have not voted, voted for approval, disapproval or abstained on this standard.

David J. Law, *IEEE 802.3 Working Group Chair*
Adam Healey, *IEEE 802.3 Working Group Vice Chair*
Peter Anslow, *IEEE 802.3 Working Group Secretary*
Steven B. Carlson, *IEEE 802.3 Working Group Executive Secretary*
Valerie Maguire, *IEEE 802.3 Working Group Treasurer*

Ludwig Winkel, *IEEE P802.3br Interspersing Express Traffic Task Force Chair*
Patricia Thaler, *IEEE P802.3br Interspersing Express Traffic Task Force Editor-in-Chief*

John Abbott
David Abramson
Shadi Abughazaleh
Faisal Ahmad
Dale Amason
J. Michael Andrewartha
Oleksandr Babenko
Kwang-Hyun Baek
Amrik Bains
Koussalya Balasubramanian
Thananya Baldwin
Denis Beaudoin
Christian Beia
Yakov Belopolsky
Michael Bennett
Vipul Bhatt
William Bliss
Brad Booth
Martin Bouda
David Brandt
Ralf-Peter Braun
Theodore Brillhart
Paul Brooks
David Brown
Matthew Brown
Thomas Brown
Phillip Brownlee
Mark Bugg
Juan-Carlos Calderon
J. Martin Carroll
Clark Carty
Mandeep Chadha
David Chalupsky
Jacky Chang
Xin Chang
David Chen
Wheling Cheng
Ahmad Chini
Golam Choudhury
Peter Cibula
Christopher R. Cole
Keith Conroy
Eugene Dai
Shaoan Dai
John D'Ambrosia
Mike Darling
Yair Darshan
Piers Dawe
Fred Dawson
Ian Dedic
Chris Diminico
Thuyen Dinh
Curtis Donahue
Dan Dove
Mike Dudek
David Dwelley
Hesham Elbakoury
David Estes
John Ewen
Josef Faller
Shahar Feldman
German Feyh
Alan Flatman
Howard Frazier
Richard Frosch
Michael Furlong
Andrew Gardner
Mike Gardner
Ali Ghiasi
Joel Goergen
Zhigang Gong
Steven Gorshe
James Graba
Robert Grow
Mark Gustlin
Marek Hajduczenia
Bernie Hammond
Jeffrey Heath
Carl Herman
David Hess
Yasuo Hidaka
Riu Hirai
Thomas Hogenmueller
Brian Holden
Rita Horner
Bernd Horrmeyer
Victor Hou
Rui Hua
Liang-wei Huang
Scott Irwin
Kazuhiko Ishibe
Hideki Isono
Tom Issenhuth
Kenneth Jackson
Jack Jewell
Wenbin Jiang
Andrew Jimenez
Chad Jones
Antony Joseph
Yasuaki Kawatsu
Michael Kelsen
Yongbum Kim
Jonathan King
Scott Kipp
Michael Klempa
Avi Kliger
Curtis Knittle
Shigeru Kobayashi
Keisuke Kojima
Paul Kolesar
Tom Kolze
Glen Kramer
Albert Kuo
Hans Lackner
Efstathios Larios
Wayne Larsen
Mark Laubach
Greg Le Cheminant
Arthur Lee
David Lewis
Lei Li
Mike Peng Li
Shaohua Li

Thomas Lichtenegger
Ru Jian Lin
Robert Lingle
James Liu
Zhenyu Liu
William Lo
Miklos Lukacs
Kent Lusted
Jeffery Maki
James Malkemus
Yonatan Malkiman
Edwin Mallette
Arthur Marris
Chris Mash
Kirsten Matheus
Erdem Matoglu
Laurence Matola
Brett Mcclellan
Thomas Mcdermott
John McDonough
Richard Mei
Richard Mellitz
Bryan Moffitt
Leo Montreuil
Paul Mooney
Charles Moore
Andy Moorwood
Thomas Mueller
Ron Muir
Dale Murray
Henry Muyshondt
Edward Nakamoto
Gary Nicholl
Paul Nikolich
Kevin Noll
Ronald Nordin
Mark Nowell
David Ofelt
Ichiro Ogura
Tom Palkert
Sujan Pandey
Sesha Panguluri
Carlos Pardo
Moon Park
Pravin Patel
Petar Pepeljugoski
Gerald Pepper
Ruben Perez De Aranda Alonso
Michael Peters
John Petrilla
Rick Pimpinella
Neven Pischl
Rainer Poehmerer
William Powell
Richard Prodan
Rick Rabinovich
Saifur Rahman
Adee Ran
Ram Rao
Alon Regev
Duane Remein
Victor Renteria
Michael Ressl
Poldi (Pavlick) Rimboim
Christopher Roth
Salvatore Rotolo
Hisaya Sakamoto
Vineet Salunke
Sam Sambasivan
Yasuo Sasaki
Fred Schindler
Stefan Schneele
Peter Scruton
Alexander Seiger
Naoshi Serizawa
Megha Shanbhag
Masood Shariff
Stephen Shellhammer
Bazhong Shen
Mizuki Shirao
Kapil Shrikhande
Jeff Slavick
Scott Sommers
Yoshiaki Sone
Xiaolu Song
Tom Souvignier
Bryan Sparrowhawk
Edward Sprague
Peter Stassar
Leonard Stencel
Robert Stone
Steve Swanson
Andre Szczepanek
William Szeto
Bharat Tailor
Akio Tajima
Takayuki Tajima
Tomoo Takahara
Satoshi Takahashi
Kiyoto Takahata
Alexander Tan
Toshiki Tanaka
Mehmet Tazebay
Brian Teipen
Geoffrey Thompson
Alan Tipper
Pirooz Tooyserkani
Nathan Tracy
David Tremblay
Albert Tretter
Stephen Trowbridge
Wen-Cheng Tseng
Yoshihiro Tsukamoto
Mike Tu
Alan Ugolini
John Ulm
Ed Ulrichs
Sterling A. Vaden
Stefano Valle
Paul Vanderlaan
Robert Wagner
Robert Wang
Tongtong Wang
Xiaofeng Wang
Xinyuan Wang
Zhong Feng Wang
Markus Weber
Brian Welch
Yang Wen
Matthias Wendt
Oded Wertheim
Natalie Wienckowski
Peter Wu
Yu Xu
Lennart Yseboodt
Liquan Yuan
Hayato Yuki
Garold Yurko
Andrew Zambell
Jin Zhang
Yan Zhuang
George Zimmerman
Helge Zinner
Pavel Zivny
Gaoling Zou

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

Thomas Alexander
Richard Alfvin
Dale Amason
Peter Anslow
Butch Anton
Stefan Aust
Eric Baden
Amrik Bains
Michael Bennett
Gennaro Boggia
Christian Boiger
Ralf-Peter Braun
Nancy Bravin
Theodore Brillhart
Jairo Bustos Heredia
William Byrd
Steven B. Carlson
Juan Carreon
Mandeep Chadha
Minho Cheong
Keith Chow
Keng Hua Chuang
Charles Cook
Rodney Cummings
Sourav Dutta
Janos Farkas
German Feyh
Norman Finn
Matthias Fritsche
Yukihiro Fujimoto
Eric W. Gray
David Gregson
Randall Groves
Craig Gunther
Stephen Haddock
Marek Hajduczenia
Mark Hantel
Adam Healey
Jerome Henry
Marco Hernandez
David Hess
Werner Hoelzl
Rita Horner
Victor Hou
Noriyuki Ikeuchi
Sergiu Iordanescu
Osamu Ishida
Atsushi Ito
Raj Jain
Anthony Jeffree
Michael Johas Teener
Peter Jones
Adri Jovin
Piotr Karocki
Stuart Kerry
Yongbum Kim
Jonathan King
Scott Kipp
Jouni Korhonen
Bruce Kraemer
Mark Laubach
David J. Law
David Lewis
Jon Lewis
Arthur H. Light
William Lo
Michael Lynch
Elvis Maculuba
Valerie Maguire
Jeffery Maki
Arthur Marris
Brett Mcclellan
Thomas Mcdermott
Richard Mellitz
John Messenger
Bryan Moffitt
Charles Moorwood
Shimon Muller
Michael Newman
Nick S. A. Nikjoo
Satoshi Obara
Satoshi Oyama
Arumugam Paventhan
Ruben Perez De Aranda Alonso
Michael Peters
Adee Ran
R. K. Rannow
Alon Regev
Duane Remein
Maximilian Riegel
Robert Robinson
Benjamin Rolfe
Dan Romascanu
Bartien Sayogo
Frank Schewe
Dieter Schicketanz
Stefan Schneele
Michael Seaman
Shusaku Shimada
Ju-Hyung Son
Thomas Starai
Peter Stassar
Eugene Stoudenmire
Walter Struppler
Mitsutoshi Sugawara
Patricia Thaler
Mark-Rene Uchida
Lorenzo Vangelista
Dmitri Varsanofiev
Prabodh Varshney
George Vlantis
Stephen Webb
Hung-Yu Wei
Ludwig Winkel
Andreas Wolf
Peter Wu
Oren Yuen
Andrew Zambell
Zhen Zhou

When the IEEE-SA Standards Board approved this standard on 30 June 2016, it had the following membership:

Introduction

This introduction is not part of IEEE Std 802.3br™-2016, IEEE Standard for Ethernet—Amendment 5: Specification and Management Parameters for Interspersing Express Traffic.

IEEE Std 802.3 was first published in 1985. Since the initial publication, many projects have added functionality or provided maintenance updates to the specifications and text included in the standard. Each IEEE 802.3 project/amendment is identified with a suffix (e.g., IEEE Std 802.3ba™-2010).

The half-duplex Media Access Control (MAC) protocol specified in IEEE Std 802.3-1985 is Carrier Sense Multiple Access with Collision Detection (CSMA/CD). This MAC protocol was key to the experimental Ethernet developed at Xerox Palo Alto Research Center, which had a 2.94 Mb/s data rate. Ethernet at 10 Mb/s was jointly released as a public specification by Digital Equipment Corporation (DEC), Intel, and Xerox in 1980. Ethernet at 10 Mb/s was approved as an IEEE standard by the IEEE Standards Board in 1983 and subsequently published in 1985 as IEEE Std 802.3-1985. Since 1985, new media options, new speeds of operation, and new capabilities have been added to IEEE Std 802.3. A full duplex MAC protocol was added in 1997.

Some of the major additions to IEEE Std 802.3 are identified in the marketplace with their project number. This is most common for projects adding higher speeds of operation or new protocols. For example, IEEE Std 802.3u™ added 100 Mb/s operation (also called Fast Ethernet), IEEE Std 802.3z™ added 1000 Mb/s operation (also called Gigabit Ethernet), IEEE Std 802.3ae™ added 10 Gb/s operation (also called 10 Gigabit Ethernet), IEEE Std 802.3ah™ specified access network Ethernet (also called Ethernet in the First Mile), and IEEE Std 802.3ba added 40 Gb/s operation (also called 40 Gigabit Ethernet) and 100 Gb/s operation (also called 100 Gigabit Ethernet). These major additions are all now included in and are superseded by IEEE Std 802.3-2015 and are not maintained as separate documents.

At the date of IEEE Std 802.3br-2016 publication, IEEE Std 802.3 is composed of the following documents:

IEEE Std 802.3-2015

Section One—Includes Clause 1 through Clause 20 and Annex A through Annex H and Annex 4A. Section One includes the specifications for 10 Mb/s operation and the MAC, frame formats, and service interfaces used for all speeds of operation.

Section Two—Includes Clause 21 through Clause 33 and Annex 22A through Annex 33E. Section Two includes management attributes for multiple protocols and speed of operation as well as specifications for providing power over twisted-pair cabling for multiple operational speeds. It also includes general information on 100 Mb/s operation as well as most of the 100 Mb/s Physical Layer specifications.

Section Three—Includes Clause 34 through Clause 43 and Annex 36A through Annex 43C. Section Three includes general information on 1000 Mb/s operation as well as most of the 1000 Mb/s Physical Layer specifications.

Section Four—Includes Clause 44 through Clause 55 and Annex 44A through Annex 55B. Section Four includes general information on 10 Gb/s operation as well as most of the 10 Gb/s Physical Layer specifications.

Section Five—Includes Clause 56 through Clause 77 and Annex 57A through Annex 76A. Clause 56 through Clause 67 and Clause 75 through Clause 77, as well as associated annexes, specify subscriber access and other Physical Layers and sublayers for operation from 512 kb/s to 10 Gb/s, and defines services and protocol elements that enable the exchange of IEEE 802.3 format frames between stations in a subscriber access network. Clause 68 specifies a 10 Gb/s Physical Layer specification. Clause 69 through Clause 74 and associated annexes specify Ethernet operation over electrical backplanes at speeds of 1000 Mb/s and 10 Gb/s.

Section Six—Includes Clause 78 through Clause 95 and Annex 83A through Annex 93C. Clause 78 specifies Energy-Efficient Ethernet. Clause 79 specifies IEEE 802.3 Organizationally Specific Link Layer Discovery Protocol (LLDP) type, length, and value (TLV) information elements. Clause 80 through Clause 95 and associated annexes include general information on 40 Gb/s and 100 Gb/s operation as well the 40 Gb/s and 100 Gb/s Physical Layer specifications. Clause 90 specifies Ethernet support for time synchronization protocols.

IEEE Std 802.3bw-2015

Amendment 1—This amendment includes changes to IEEE Std 802.3-2015 and adds Clause 96. This amendment adds 100 Mb/s Physical Layer (PHY) specifications and management parameters for operation on a single balanced twisted-pair copper cable.

IEEE Std 802.3by-2016

Amendment 2—This amendment includes changes to IEEE Std 802.3-2015 and adds Clause 105 through Clause 112, Annex 109A, Annex 109B, Annex 109C, Annex 110A, Annex 110B, and Annex 110C. This amendment adds MAC parameters, Physical Layers, and management parameters for the transfer of IEEE 802.3 format frames at 25 Gb/s.

IEEE Std 802.3bq-2016

Amendment 3—This amendment includes changes to IEEE Std 802.3-2015 and adds Clause 113 and Annex 113A. This amendment adds new Physical Layers for 25 Gb/s and 40 Gb/s operation over balanced twisted-pair structured cabling systems.

IEEE Std 802.3bp-2016

Amendment 4—This amendment includes changes to IEEE Std 802.3-2015 and adds Clause 97 and Clause 98. This amendment adds point-to-point 1 Gb/s Physical Layer (PHY) specifications and management parameters for operation on a single balanced twisted-pair copper cable in automotive and other applications not utilizing the structured wiring plant.

IEEE Std 802.3br-2016

Amendment 5—This amendment includes changes to IEEE Std 802.3-2015 and adds Clause 99. This amendment adds a MAC Merge sublayer and a MAC Merge Service Interface to support for Interspersing Express Traffic over a single link.

A companion document IEEE Std 802.3.1 describes Ethernet management information base (MIB) modules for use with the Simple Network Management Protocol (SNMP). IEEE Std 802.3.1 is updated to add management capability for enhancements to IEEE Std 802.3 after approval of the enhancements.

IEEE Std 802.3 will continue to evolve. New Ethernet capabilities are anticipated to be added within the next few years as amendments to this standard.

Contents

IEEE Standard for Ethernet

Amendment 5: Specification and Management Parameters for Interspersing Express Traffic

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.

(This amendment is based on IEEE Std 802.3™-2015 as amended by IEEE Std 802.3bw™-2015, IEEE Std 802.3by™-2016, IEEE Std 802.3bq™-2016, and IEEE Std 802.3bp™-2016.

NOTE—The editing instructions contained in this amendment define how to merge the material contained therein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in ***bold italic***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Deletions and insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this NOTE will not be carried over into future editions because the changes will be incorporated into the base standard.[1]

Cross references that refer to clauses, tables, equations, or figures not covered by this amendment are highlighted in green.

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

1. Introduction

1.4 Definitions

Insert the following two new definitions into the list after 1.4.197 Exception Window:

1.4.197a express Media Access Control (eMAC): The instance of a Media Access Control sublayer (IEEE Std 802.3, Annex 4A) that is the client of a MAC Merge sublayer service interface that handles express traffic. (See IEEE Std 802.3, Clause 99.)

1.4.197b express traffic: Frames transmitted through an express Media Access Control (eMAC) sublayer. (See IEEE Std 802.3, Clause 99.)

Insert the following new definition into the list after 1.4.261 MAC frame:

1.4.261a MAC Merge sublayer: An optional sublayer that supports interspersing express traffic with preemptable traffic by attaching an express Media Access Control (eMAC) and a preemptable Media Access Control (pMAC) to a single Physical Signaling Sublayer (PLS) service. (See IEEE Std 802.3, Clause 99.)

Insert the following two new definitions into the list after 1.4.339 Powered Device (PD):

1.4.339a preemptable Media Access Control (pMAC): The instance of a Media Access Control sublayer (IEEE Std 802.3, Annex 4A) that is the client of a MAC Merge sublayer service interface that handles preemptable traffic. (See IEEE Std 802.3, Clause 99.)

1.4.339b preemptable traffic: Frames transmitted through a preemptable Media Access Control (pMAC) sublayer (See IEEE Std 802.3, Clause 99.)

1.5 Abbreviations

Insert the following new abbreviations into the list, in alphabetical order:

eMAC	express Media Access Control
MMSI	MAC Merge service interface
pMAC	preemptable Media Access Control

30. Management

30.2 Managed objects

30.2.2 Overview of managed objects

30.2.2.1 Text description of managed objects

Change the description of oMACEntity as follows:

oMACEntity If oMACControlEntity is implemented, oMACEntity is contained within oMACControlEntity. Otherwise, if oOAM is implemented, oMACEntity is contained within oOAM. Otherwise, if oAggregator is implemented, oMACEntity is contained within oAggregator. Otherwise, oMACEntity becomes the top-most managed object class of the DTE containment tree shown in Figure 30–3. Note that this managed object class may be contained within another superior managed object class. Such containment is expected, but is outside the scope of this International Standard. If oMACMergeEntity is implemented, the oMACEntity for the express MAC (eMAC) and the oMACEntity for the preemptable MAC (pMAC) are connected to an instance of oMACMergeEntity.

Insert the following new description into the list in alphabetical order:

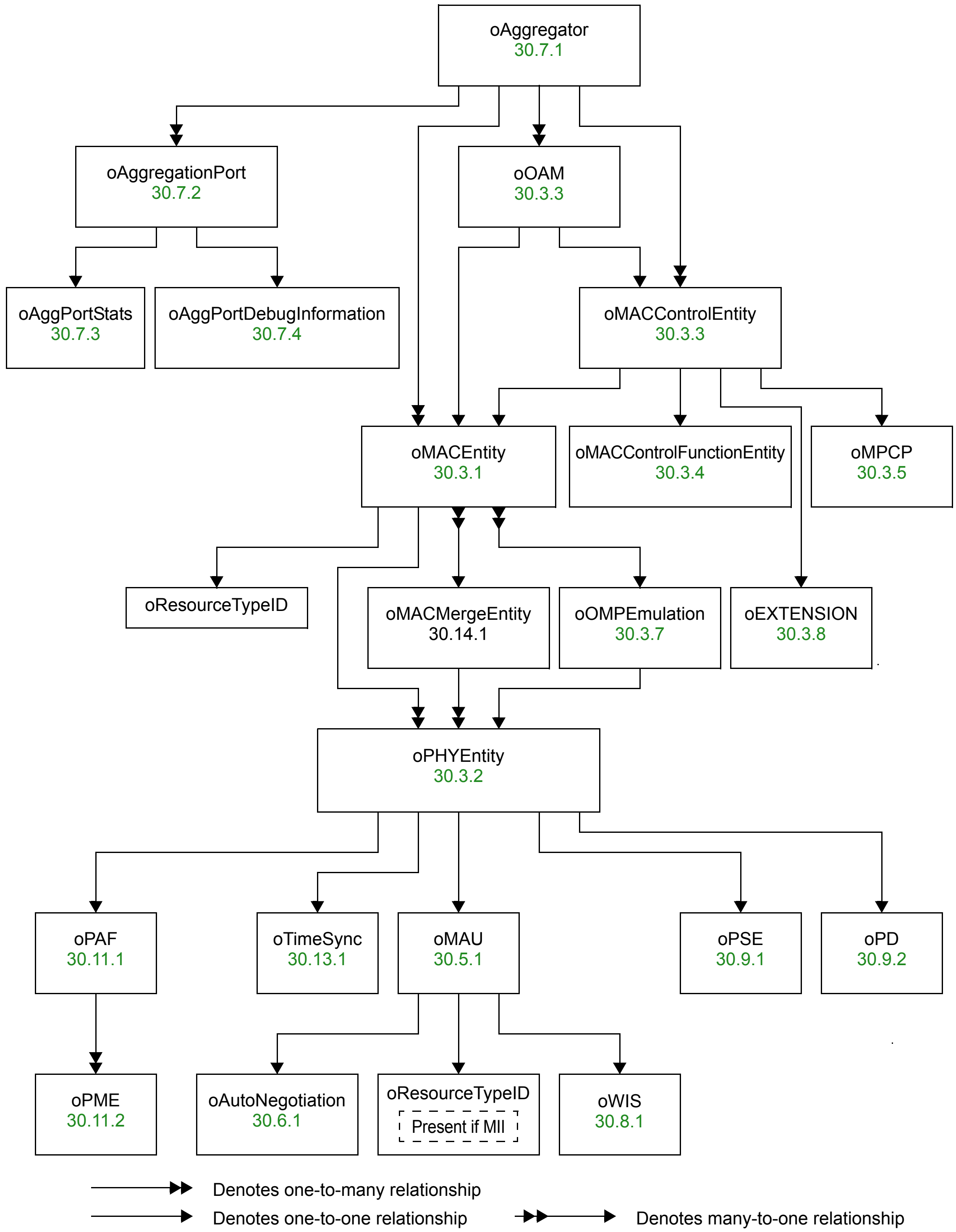
oMACMergeEntity If implemented, a single instance of oMACMergeEntity is contained within oMACEntity for an express MAC (eMAC) and oMACEntity for a preemptable MAC (pMAC) (see Clause 99). oMACMergeEntity managed object class provides the management controls necessary for the MAC Merge sublayer.

Change the description of oPHYEntity as follows:

oPHYEntity If oOMPEmulation is implemented, oPHYEntity is contained within oOMPEmulation. If oMACMergeEntity is implemented, oPHYEntity is contained within oPHYEntity. Otherwise oPHYEntity is contained within oMACEntity. Many instances of oPHYEntity may coexist within one instance of oMACEntity or oMACMergeEntity; however, only one PHY may be active for data transfer to and from the MAC at any one time. oPHYEntity is the managed object that contains the MAU, PAF, and PSE managed objects in a DTE.

30.2.3 Containment

Replace Figure 30-3 with a new Figure 30–3 as follows:



NOTE—The objects oAggregator, oAggregationPort, oAggPortStats, and oAggPortDebugInformation are deprecated by IEEE Std 802.1AX™-2008.

Figure 30–3—DTE System entity relationship diagram

30.2.5 Capabilities

Change the first paragraph in 30.2.5 and insert Table 30–8 and Table 30–9 as follows:

This standard makes use of the concept of *packages* as defined in ISO/IEC 10165-4:1992 as a means of grouping behaviour, attributes, actions, and notifications within a managed object class definition. Packages may either be mandatory, or be conditional, that is to say, present if a given condition is true. Within this standard *capabilities* are defined, each of which corresponds to a set of packages, which are components of a number of managed object class definitions and which share the same condition for presence. Implementation of the appropriate Basic and Mandatory packages is the minimum requirement for claiming conformance to IEEE 802.3 Management. Implementation of an entire optional capability is required in order to claim conformance to that capability. The capabilities and packages for IEEE 802.3 Management are specified in Table 30–1a through ~~Table 30–7~~Table 30–9.

Table 30–8—LLDP capabilities (additional packages)

			LLDP MAC Merge Package (optional)
oLldpXdot3LocSystemsGroup managed object class (30.14)			
aLldpXdot3LocPreemptSupported	ATTRIBUTE	GET	X
aLldpXdot3LocPreemptEnabled	ATTRIBUTE	GET	X
aLldpXdot3LocPreemptActive	ATTRIBUTE	GET	X
aLldpXdot3LocAddFragSize	ATTRIBUTE	GET-SET	X
aLldpXdot3RemPreemptSupported	ATTRIBUTE	GET	X
aLldpXdot3RemPreemptEnabled	ATTRIBUTE	GET	X
aLldpXdot3RemPreemptActive	ATTRIBUTE	GET	X
aLldpXdot3RemAddFragSize	ATTRIBUTE	GET	X

Table 30–9—MAC Merge sublayer capabilities

			MAC Merge Basic Package (mandatory)
oMACMergeEntity managed object class			
aMACMergeSupport	ATTRIBUTE	GET	X
aMACMergeStatusVerify	ATTRIBUTE	GET	X
aMACMergeEnableTx	ATTRIBUTE	GET-SET	X
aMACMergeVerifyDisableTx	ATTRIBUTE	GET-SET	X
aMACMergeStatusTx	ATTRIBUTE	GET	X
aMACMergeVerifyTime	ATTRIBUTE	GET-SET	X
aMACMergeAddFragSize	ATTRIBUTE	GET	X
aMACMergeFrameAssErrorCount	ATTRIBUTE	GET	X
aMACMergeFrameSmdErrorCount	ATTRIBUTE	GET	X
aMACMergeFrameAssOkCount	ATTRIBUTE	GET	X
aMACMergeFragCountRx	ATTRIBUTE	GET	X
aMACMergeFragCountTx	ATTRIBUTE	GET	X
aMACMergeHoldCount	ATTRIBUTE	GET	X

30.12 Layer Management for Link Layer Discovery Protocol (LLDP)

30.12.1 LLDP Configuration managed object class

30.12.1.1 LLDP Configuration attributes

30.12.1.1.1 aLldpXdot3PortConfigTLVsTxEnable

Change 30.12.1.1.1 as follows:

ATTRIBUTE

APPROPRIATE SYNTAX:
BITSTRING

BEHAVIOUR DEFINED AS:
A read-write string of ~~6~~7 bits indicating, for each of the IEEE 802.3 optional LLDP TLVs, if transmit is enabled on the local LLDP agent by the network management. A “1” in the bitstring indicates transmit of the TLV is enabled, “0” indicates transmit of the TLV is disabled. The value of this attribute is preserved across reset including loss of power.

~~The first bit indicates if MAC/PHY configuration/status TLV transmit is enabled, the second bit indicates if Power via MDI TLV transmit is enabled, the third bit indicates if the deprecated Link Aggregation TLV transmit is enabled, the fourth bit indicates if the Maximum Frame Size TLV transmit is enabled, the fifth bit indicates if the EEE TLV is enabled, and the sixth bit indicates if the EEE Fast Wake TLV is enabled.;~~ Each bit of the bit string indicates whether transmit is enabled for the TLV corresponding to the bit. The mapping of bits to TLVs is:

first	MAC/PHY configuration/status TLV
second	Power via MDI TLV
third	Link Aggregation TLV (deprecated)
fourth	Maximum Frame Size TLV
fifth	EEE TLV
sixth	EEE Fast Wake TLV
seventh	Additional Ethernet Capabilities TLV.;

30.12.2 LLDP Local System Group managed object class

30.12.2.1 LLDP Local System Group attributes

Insert 30.12.2.1.34 through 30.12.2.1.37 after 30.12.2.1.33 as follows:

30.12.2.1.34 aLldpXdot3LocPreemptSupported

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
A read-only Boolean value used to indicate whether the given port (associated with the local System) supports the preemption capability.;

30.12.2.1.35 aLldpXdot3LocPreemptEnabled

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
A read-only Boolean value used to indicate whether the preemption capability is enabled on the given port associated with the local System.;

30.12.2.1.36 aLldpXdot3LocPreemptActive

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
A read-only Boolean value used to indicate whether the preemption capability is active on the given port associated with the local System.;

30.12.2.1.37 aLldpXdot3LocAddFragSize

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
A 2-bit integer value used to indicate the minimum size of non-final fragments supported by the receiver on the given port associated with the local System. This value is expressed in units of 64 octets of additional fragment length. The minimum non-final fragment size is (aLldpXdot3LocAddFragSize + 1) × 64 octets.;

30.12.3 LLDP Remote System Group managed object class

30.12.3.1 LLDP Remote System Group attributes

Insert 30.12.3.1.28 through 30.12.3.1.31 after 30.12.3.1.27 as follows:

30.12.3.1.28 aLldpXdot3RemPreemptSupported

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
A read-only Boolean value used to indicate whether the given port (associated with the remote system) supports the preemption capability.;

30.12.3.1.29 aLldpXdot3RemPreemptEnabled

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
A read-only Boolean value used to indicate whether the preemption capability is enabled on the given port associated with the remote system.;

30.12.3.1.30 aLldpXdot3RemPreemptActive

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
A read-only Boolean value used to indicate whether the preemption capability is active on the given port associated with the remote system.;

30.12.3.1.31 aLldpXdot3RemAddFragSize

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:

A 2-bit integer value used to indicate, in units of 64 octets, the minimum number of octets over 64 octets required in non-final fragments by the receiver on the given port associated with the remote system. The minimum non-final fragment size is (aLldpXdot3LocAddFragSize + 1) × 64 octets.;

Insert a new subclause 30.14 after 30.13 as follows:

30.14 Management for MAC Merge Sublayer

30.14.1 oMACMergeEntity managed object class

This subclause formally defines the behaviours for the oMACMergeEntity managed object class attributes.

30.14.1.1 aMACMergeSupport

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

supported	MAC Merge sublayer is supported on the device
not supported	MAC Merge sublayer is not supported on the device

BEHAVIOUR DEFINED AS:

This attribute indicates (when accessed via a GET operation) whether the given device supports a MAC Merge sublayer. The SET operation shall have no effect on a device.;

30.14.1.2 aMACMergeStatusVerify

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

unknown	verification status is unknown
initial	the Verify State diagram (Figure 99–8) is in the state INIT_VERIFICATION
verifying	the Verify State diagram is in the state VERIFICATION_IDLE, SEND_VERIFY or WAIT_FOR_RESPONSE
succeeded	indicates that the Verify State diagram is in the state VERIFIED
failed	the Verify State diagram is in the state VERIFY_FAIL
disabled	verification of preemption operation is disabled

BEHAVIOUR DEFINED AS:

This attribute indicates (when accessed via a GET operation) the status of the MAC Merge sublayer verification on the given device. The SET operation shall have no effect on a device.;

30.14.1.3 aMACMergeEnableTx

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

disabled	transmit preemption is disabled
enabled	transmit preemption is enabled

BEHAVIOUR DEFINED AS:

This attribute indicates (when accessed via a GET operation) the status of the MAC Merge sublayer on the given device in the transmit direction. The status of the MAC Merge sublayer may be modified to the indicated value via a SET operation. This attribute maps to the variable pEnable (see 99.4.7.3).;

30.14.1.4 aMACMergeVerifyDisableTx

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

disabled	verify is disabled
enabled	verify is enabled

BEHAVIOUR DEFINED AS:

This attribute indicates (when accessed via a GET operation) the status of the Verify function of MAC Merge sublayer on the given device in the transmit direction. The status of the Verify function may be modified to the indicated value via a SET operation. This attribute maps to the variable disableVerify (see 99.4.7.3).;

30.14.1.5 aMACMergeStatusTx

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

unknown	transmit preemption status is unknown
inactive	transmit preemption is inactive
active	transmit preemption is active

BEHAVIOUR DEFINED AS:

This attribute indicates (when accessed via a GET operation) the status of the MAC Merge sublayer on the given device in the transmit direction. The SET operation shall have no effect on a device. This attribute maps to the variable preempt (see 99.4.7.3).;

30.14.1.6 aMACMergeVerifyTime

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The value of this attribute defines the nominal wait time between verification attempts in milliseconds. Valid range is 1 to 128 inclusive. The default value is 10. This attribute maps to the variable verifyTime (see 99.4.7.3).;

30.14.1.7 aMACMergeAddFragSize

ATTRIBUTE

APPROPRIATE SYNTAX

INTEGER

BEHAVIOUR DEFINED AS:

A 2-bit integer value used to indicate the value of addFragSize variable used by the Transmit Processing State Diagram (see Figure 99–5).;

30.14.1.8 aMACMergeFrameAssErrorCount

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter.
This counter has a maximum increment rate of 160 000 counts per second at 100 Mb/s.

BEHAVIOUR DEFINED AS:

A count of MAC frames with reassembly errors. The counter is incremented by one every time the ASSEMBLY_ERROR state in the Receive Processing State Diagram is entered (see Figure 99–6).;

30.14.1.9 aMACMergeFrameSmdErrorCount

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter.
This counter has a maximum increment rate of 160 000 counts per second at 100 Mb/s.

BEHAVIOUR DEFINED AS:

A count of received MAC frames / MAC frame fragments rejected due to unknown SMD value or arriving with an SMD-C when no frame is in progress. The counter is incremented by one every time the BAD_FRAG state in the Receive Processing State Diagram is entered and every time the WAIT_FOR_DV_FALSE state is entered due to the invocation of the SMD_DECODE function returning the value “ERR” (see Figure 99–6).;

30.14.1.10 aMACMergeFrameAssOkCount

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter.
This counter has a maximum increment rate of 160 000 counts per second at 100 Mb/s.

BEHAVIOUR DEFINED AS:

A count of MAC frames that were successfully reassembled and delivered to MAC. The counter is incremented by one every time the FRAME_COMPLETE state in the Receive Processing state diagram (see Figure 99–6) is entered if the state CHECK_FOR_RESUME was previously entered while processing the packet.;

30.14.1.11 aMACMergeFragCountRx

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter.
This counter has a maximum increment rate of 160 000 counts per second at 100 Mb/s.

BEHAVIOUR DEFINED AS:

A count of the number of additional mPackets received due to preemption. The counter is incremented by one every time the state CHECK_FRAG_CNT in the Receive Processing State Diagram (see Figure 99–6) is entered.;

30.14.1.12 aMACMergeFragCountTx

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter.
This counter has a maximum increment rate of 160 000 counts per second at 100 Mb/s

BEHAVIOUR DEFINED AS:
A count of the number of additional mPackets transmitted due to preemption. This counter is incremented by one every time the SEND_SMD_C state in the Transmit Processing State Diagram (see Figure 99–5) is entered.;

30.14.1.13 aMACMergeHoldCount

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter.
This counter has a maximum increment rate of 160 000 counts per second at 100 Mb/s.

BEHAVIOUR DEFINED AS:
A count of the number of times the variable hold (see 99.4.7.3) transitions from FALSE to TRUE.;

79. IEEE 802.3 Organizationally Specific Link Layer Discovery Protocol (LLDP) type, length, and value (TLV) information elements

79.3 IEEE 802.3 Organizationally Specific TLVs

Insert the following row in Table 79–1 and change the range in the subtype column of the last row to remove the assigned subtype value.

Table 79–1—IEEE 802.3 Organizationally Specific TLVs

IEEE 802.3 subtype	TLV name	Subclause reference
7	Additional Ethernet Capabilities	79.3.7
~~7~~ 8 to 255	Reserved	—

Insert a new subclause 79.3.7 after 79.3.6.4 as follows:

79.3.7 Additional Ethernet Capabilities TLV

The Additional Ethernet Capabilities TLV is an optional TLV that indicates which additional capabilities are supported. Figure 79–8 shows the format of this TLV.

NOTE—MAC and PHY support for a given frame size does not necessarily mean that the upper layers support that frame size.

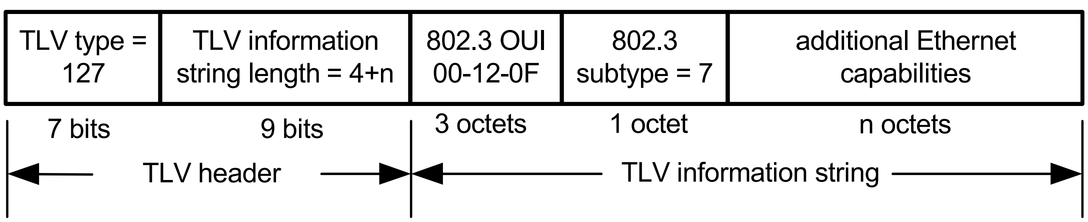


Figure 79–8—Additional Ethernet Capabilities TLV

79.3.7.1 Additional Ethernet capabilities

The additional Ethernet capabilities field shall contain a bitmap that identifies the support and current status of additional Ethernet capabilities on the local IEEE 802.3 LAN station. The first two octets of the field are defined in Table 79–7a. Additional octets are reserved.

An implementation shall transmit all Reserved bits as zero and ignore received Reserved bits. Reserved octets shall not be transmitted and if more octets are received than were defined as other than Reserved, the additional octet(s) shall be ignored. If fewer octet(s) are received than defined, the implementation shall act as if the additional octet(s) were received as zero.

79.3.7.2 Additional Ethernet Capabilities TLV usage rules

An LLDPDU should contain no more than one Additional Ethernet Capabilities TLV. Since this TLV is intended to inform a link partner of capabilities, the Additional Ethernet Capabilities TLV should be sent in an LLDPDU addressed to the Nearest Bridge group address (see IEEE Std 802.1Q).

Table 79–7a—Additional Ethernet capabilities

Bit	Function	Value/meaning	Reference
0	preemption capability support	1 = supported 0 = not supported	99.4.2
1	preemption capability status	1 = enabled 0 = not enabled	99.4.2
2	preemption capability active	1 = active 0 = not active	99.4.3
4:3	additional fragment size	A 2-bit integer value indicating, in units of 64 octets, the minimum number of octets over 64 octets required in non-final fragments by the receiver	99.4.4
15:5	Reserved	—	—

79.4 IEEE 802.3 Organizationally Specific TLV selection management

79.4.2 IEEE 802.3 Organizationally Specific TLV/LLDP Local and Remote System group managed object class cross references

Insert new rows at the bottom of Table 79–9 and Table 79–10.

Table 79–9—IEEE 802.3 Organizationally Specific TLV/LLDP Local System Group managed object class cross references

TLV name	TLV variable	LLDP Local System Group managed object class attribute
Additional Ethernet Capabilities	preemption capability support	aLldpXdot3LocPreemptSupported
	preemption capability status	aLldpXdot3LocPreemptEnabled
	preemption capability active	aLldpXdot3LocPreemptActive
	additional fragment size	aLldpXdot3LocAddFragSize

Table 79–10—IEEE 802.3 Organizationally Specific TLV/LLDP Remote System Group managed object class cross references

TLV name	TLV variable	LLDP Remote System Group managed object class attribute
Additional Ethernet Capabilities	preemption capability support	aLldpXdot3RemPreemptSupported
	preemption capability status	aLldpXdot3RemPreemptEnabled
	preemption capability active	aLldpXdot3RemPreemptActive
	additional fragment size	aLldpXdot3RemAddFragSize

79.5 Protocol implementation conformance statement (PICS) proforma for IEEE 802.3 Organizationally Specific Link Layer Discovery Protocol (LLDP) type, length, and value (TLV) information elements[2]

79.5.3 Major capabilities/options

Insert one row at the bottom of the table in 79.5.3:

Item	Feature	Subclause	Value/Comment	Status	Support
*AE	Additional Ethernet Capabilities TLV	79.3.7		O	Yes [ ] No [ ]

Insert a new subclause 79.5.11 after 79.5.10 as follows:

79.5.11 Additional Ethernet Capabilities TLV

Item	Feature	Subclause	Value/Comment	Status	Support
AET1	Ethernet capabilities field	79.3.7.1	Contains a bitmap that identifies the support and current status of additional Ethernet capabilities	AE:M	Yes [ ] N/A [ ]
AET2	Reserved bits	79.3.7.1	Set to zero and ignored on receipt	AE:M	Yes [ ] N/A [ ]
AET3	Reserved octets	79.3.7.1	Not transmitted and ignored on receipt	AE:M	Yes [ ] N/A [ ]
AET4	octets not received	79.3.7.1	Treated as all zero	AE:M	Yes [ ] N/A [ ]
AET5	usage rule—at most one	79.3.7.2	LLDPDU should contain no more than one Additional Ethernet Capabilities TLV	AE:O	Yes [ ] No [ ] N/A [ ]
AET6	usage rule—address	79.3.7.2	Additional Ethernet Capabilities TLV should be sent in an LLDPDU addressed to Nearest Bridge group address	AE:O	Yes [ ] No [ ] N/A [ ]

[2]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

90. Ethernet support for time synchronization protocols

90.4 Time Synchronization Service Interface (TSSI)

90.4.3 Detailed service specification

90.4.3.1 TS_TX.indication primitive

Change 90.4.3.1.1 as follows:

90.4.3.1.1 Semantics

The semantics of the primitive are as follows:

TS_TX.indication(SFD, MM)

The SFD parameter can take only one possible value, DETECTED. When asserted (SFD = DETECTED), the TimeSync Client is notified that a valid SFD was detected by the gRS sublayer TS_SFD_Detect_TX function (see 90.5.1) in the xMII transmit signals.

The MM parameter is mandatory when the MAC Merge sublayer (see Clause 99) is instantiated. The MM parameter, when present, can take one of two possible values, i.e., PMAC or EMAC. The value EMAC indicates the SMD-E (SFD) value has been detected at the xMII. The value PMAC indicates that an SMD-S value has been detected at the xMII (see Table 99–1). The MM parameter is not provided when MAC Merge sublayer is not instantiated.

90.4.3.2 TS_RX.indication primitive

Change 90.4.3.2.1 as follows:

90.4.3.2.1 Semantics

The semantics of the primitive are as follows:

TS_RX.indication(SFD, MM)

The SFD parameter can take only one possible value, DETECTED. When asserted (SFD = DETECTED), the TimeSync Client is notified that a valid SFD was detected by the gRS sublayer TS_SFD_Detect_RX function (see 90.5.2) in the xMII receive signals.

The MM parameter is mandatory when the MAC Merge sublayer (see Clause 99) is instantiated. The MM parameter, when present, can take one of two possible values, i.e., PMAC or EMAC. The value EMAC indicates the SMD-E (SFD) value has been detected at the xMII. The value PMAC indicates that an SMD-S value has been detected at the xMII (see Table 99–1). The MM parameter is not provided when MAC Merge sublayer is not instantiated.

90.5 generic Reconciliation Sublayer (gRS)

Change the first paragraph of 90.5 as follows:

Within the scope of this clause, the term generic Reconciliation Sublayer (gRS) is used to denote any IEEE 802.3 Reconciliation Sublayer (RS) used to interface a MAC with any PHY supporting the TimeSync capability through the xMII. The behavior of the gRS depends on whether a MAC Merge sublayer (see Clause 99) is instantiated above the gRS.

Change 90.5.1 and 90.5.2 as follows:

90.5.1 TS_SFD_Detect_TX function

The TS_SFD_Detect_TX function observes the xMII transmit signals.

When the MAC Merge sublayer is not instantiated, the TS_SFD_Detect_TX function ~~and~~ detects the occurrence of the Start Frame Delimiter (SFD, see 3.1.1 and 3.2.2) in compliance with the specifications of the given type of the instantiated xMII. The service primitive across the TSSI, i.e., TS_TX.indication shall be generated only when the SFD is detected on the transmit signals of the xMII (SFD=DETECTED).

When the MAC Merge sublayer is instantiated, the TS_SFD_Detect_TX function detects the occurrence of the Start mPacket Delimiter for an express packet or preemptable packet start (SMD-E or SMD-S, see 99.3.3) in compliance with the specifications of the given type of the instantiated xMII. The service primitive across the TSSI, i.e., TS_TX.indication, shall be generated only when the SMD-E or SMD-S is detected on the transmit signals of the xMII (SFD=DETECTED). The value of MM shall indicate whether an SMD-E (MM=EMAC) or an SMD-S (MM=PMAC) was detected.

90.5.2 TS_SFD_Detect_RX function

The TS_SFD_Detect_RX function observes the xMII receive signals.

When the MAC Merge sublayer is not instantiated, the TS_SFD_Detect_RX function ~~and~~ detects the occurrence of the Start Frame (SFD, see 3.1.1 and 3.2.2) in compliance with the specifications of the given type of the instantiated xMII. The service primitive across the TSSI, i.e., TS_RX.indication, shall be generated only when the SFD is detected on the receive signals of the xMII (SFD=DETECTED).

When the MAC Merge sublayer is instantiated, the TS_SFD_Detect_RX function detects the occurrence of the Start mPacket Delimiter for an express packet or preemptable packet start (SMD-E or SMD-S, see 99.3.3) in compliance with the specifications of the given type of the instantiated xMII. The service primitive across the TSSI, i.e., TS_RX.indication, shall be generated only when the SMD-E or SMD-S is detected on the transmit signals of the xMII (SFD=DETECTED). The value of MM shall indicate whether an SMD-E (MM=EMAC) or an SMD-S (MM=PMAC) was detected.

90.8 Protocol implementation conformance statement (PICS) proforma for Clause 90, Ethernet support for time synchronization protocols[3]

Change 90.8.3 as follows:

90.8.3 TSSI indication

Item	Feature	Subclause	Value/Comment	Status	Support
*MM	MAC Merge	90.5	A MAC Merge sublayer is instantiated above gRS	O	Yes [ ] No [ ]
TS_TX	TS_TX.indication generation	90.5.1	TS_TX.indication is generated only when the SFD is detected on the transmit signals of the xMII.	!MM:M	Yes [ ] N/A [ ]
TS_RX	TS_RX.indication generation	90.5.2	TS_RX.indication is generated only when the SFD is detected on the receive signals of the xMII.	!MM:M	Yes [ ] N/A [ ]
TS_T2	TS_TX.indication generation with MAC Merge	90.5.1	TS_TX.indication is generated only when SMD-E or SMD-S is detected on the transmit signals of the xMII	MM:M	Yes [ ] N/A [ ]
TS_T3	MM parameter for TS_TX	90.5.1	Value of MM parameter indicates whether an SMD-E or SMD-S is detected.	MM:M	Yes [ ] N/A [ ]
TS_R2	TS_RX.indication generation with MAC Merge	90.5.2	TS_RX.indication is generated only when SMD-E or SMD-S is detected on the receive signals of the xMII	MM:M	Yes [ ] N/A [ ]
TS_R3	MM parameter for TS_RX	90.5.2	Value of MM parameter indicates whether an SMD-E or SMD-S is detected.	MM:M	Yes [ ] N/A [ ]

[3]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

Insert a new Clause 99 as follows:

99. MAC Merge sublayer

99.1 Introduction

This clause specifies an optional MAC Merge sublayer for use with a pair of full-duplex MACs and a single PHY operating at 100 Mb/s or higher on a point-to-point link. The two MACs are:

- a preemptable MAC (pMAC), which carries the preemptable traffic, and
- an express MAC (eMAC), which carries the express traffic.

The MAC Merge sublayer supports interspersing express traffic with preemptable traffic. This is achieved by using a MAC Merge sublayer to attach an express Media Access Control (MAC) and a preemptable MAC to a single Reconciliation Sublayer (RS) service. The MAC Merge sublayer supports two ways to hold transmission of preemptable traffic in the presence of express traffic:

- the MAC Merge sublayer may preempt (interrupt) preemptable traffic being currently transmitted, and
- the MAC Merge sublayer may prevent starting the transmission of preemptable traffic.

This clause also specifies a MAC Merge Service Interface (MMSI) providing a primitive that causes the transmission of preemptable traffic to be held or released, minimizing the latency for express traffic. For example, when the MAC Client supports scheduled traffic as defined in IEEE Std 802.1Q, transmission of preemptable packets can be held before express traffic is scheduled to be transmitted.[4]

When the preemption capability is active, the MAC Merge sublayer allows frames provided over the express MAC service interface (express traffic) or the MMSI service primitive to interrupt transmission of a frame provided over the preemptable MAC service interface.

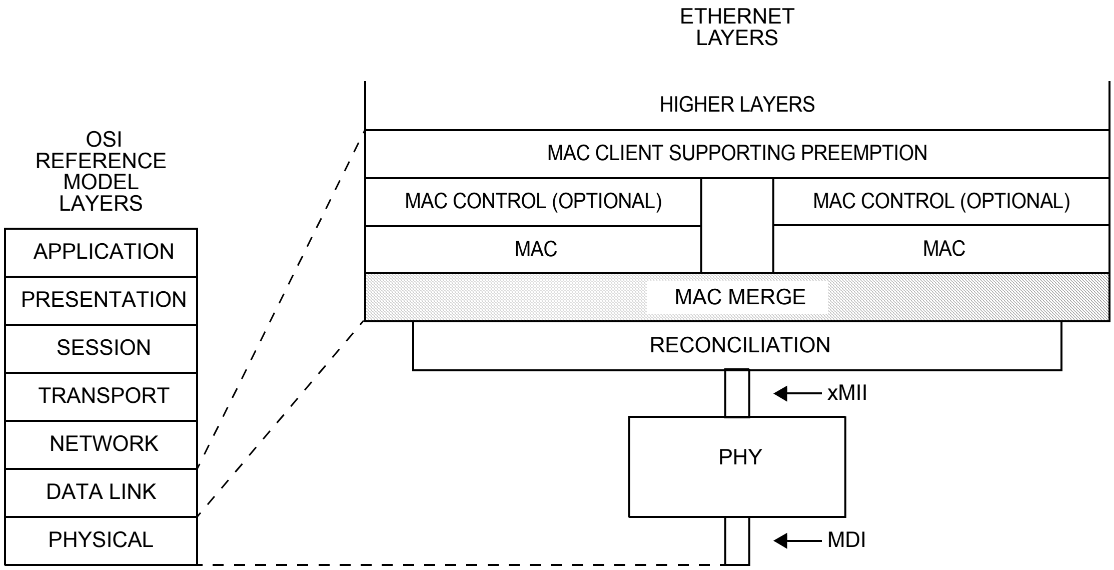
When the preemption capability is inactive, the MAC Merge sublayer does not allow express traffic or the MMSI service primitive to interrupt transmission of a frame provided over the preemptable MAC service interface.

Regardless of whether the preemption capability is active, the MAC Merge sublayer allows express traffic or the MMSI service primitive to prevent the start of transmission of frames provided over the preemptable MAC service interface (preemptable traffic).

The preemption capability is most useful at lower operating speeds. For example, the duration of a 2000 octet packet on a 100 Mb/s link is 160 µs and on a 1 Gb/s link is 16 µs. The time to transmit a maximum length packet is an upper bound on the additional delay before a MAC Client can send an Express frame when the preemption capability is not used. At higher operating speeds this additional delay gets smaller in proportion to the speed, reducing the advantage of the preemption mechanism.

Figure 99–1 shows the relationship of MAC Merge sublayer to the other sublayers and the ISO/IEC Open Systems Interconnection (OSI) reference model. The MAC Merge sublayer has two clients that are MAC sublayer instances: the eMAC and the pMAC.

[4]At the time of publication of this amendment, the content of IEEE Std 802.1Q relevant to this clause was separately published as IEEE Std 802.1Qbu™-2016, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Amendment 26: Frame Preemption, and IEEE Std 802.1Qbv™-2015, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Amendment 25: Enhancements for Scheduled Traffic.



NOTE—In this figure, the xMII is used as a generic term for the Media Independent Interfaces for implementations of 100 Mb/s and above. For example: for 100 Mb/s implementations this interface is called MII; for 1 Gb/s implementations it is called GMII; for 10 Gb/s implementations it is called XGMII; etc.

MAC = MEDIA ACCESS CONTROL
xMII = MEDIA INDEPENDENT INTERFACE
MDI = MEDIUM DEPENDENT INTERFACE
PHY = PHYSICAL LAYER DEVICE

Figure 99–1—Relationship of MAC Merge sublayer to the ISO/IEC Open Systems Interconnection (OSI) reference model and the IEEE 802.3 Ethernet model

Figure 99–2 shows the service interfaces of the MAC Merge sublayer and its associated MACs and Reconciliation Sublayer.

When attached to an eMAC or a pMAC, the MAC Control Sublayer shall not generate PAUSE (see Annex 31B).

Devices that implement the MAC Merge sublayer interoperate with devices that do not implement the MAC Merge sublayer. The preemption capability is enabled only after it has been determined that the link partner supports it (see 99.4.2).

99.1.1 Relationship to other IEEE standards

MAC Merge sublayer and its MACs provide support for Frame Preemption as defined in IEEE Std 802.1Q.

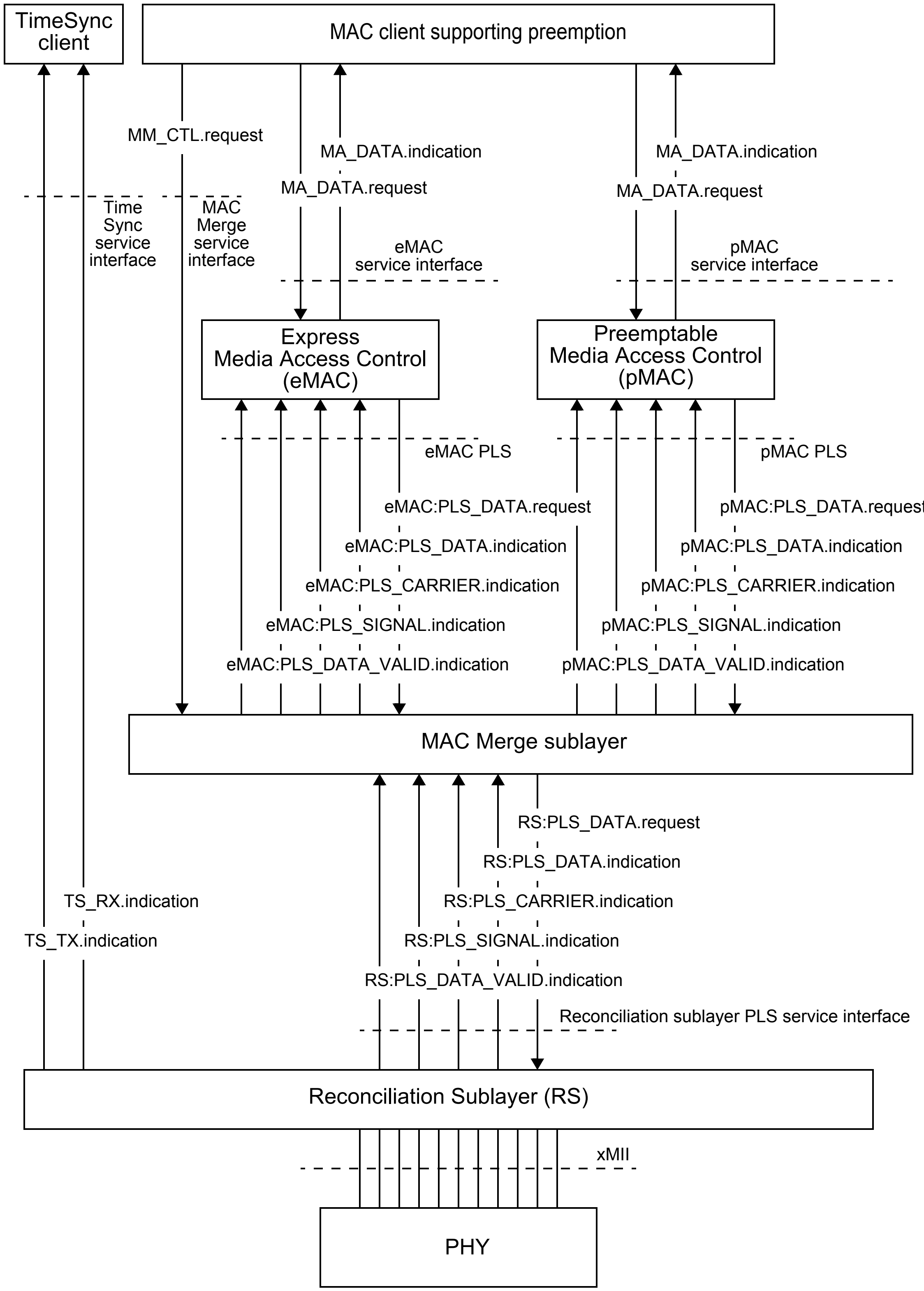


Figure 99–2—MAC Merge sublayer service interfaces diagram

99.1.2 Functional Block Diagram

Figure 99–3 provides a functional block diagram of the MAC Merge sublayer.

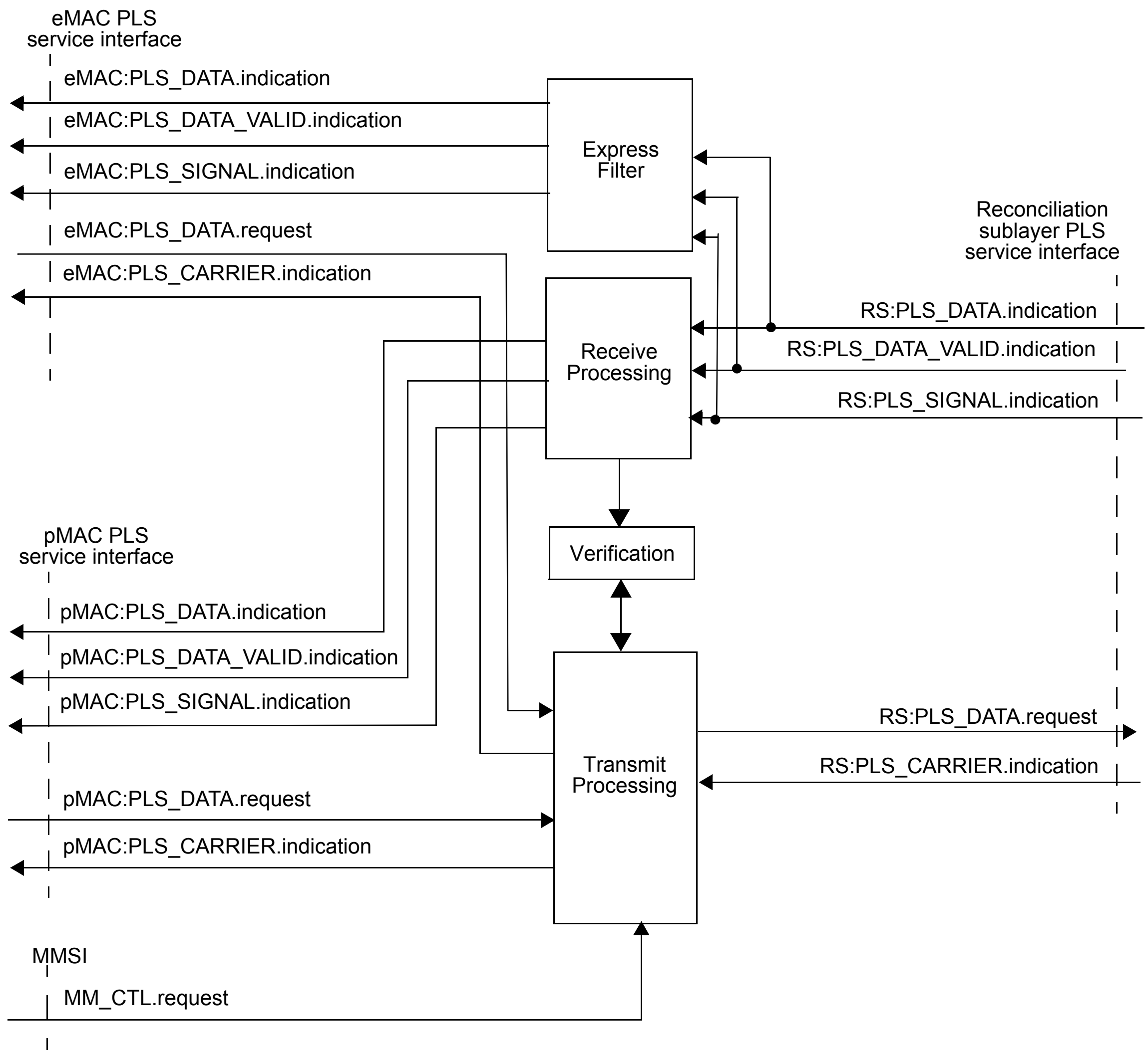


Figure 99–3—MAC Merge sublayer Functional Block Diagram

99.2 MAC Merge Service Interface (MMSI)

This subclause specifies the services provided by the MAC Merge sublayer to a MAC Client. The service interface is described in an abstract manner and does not imply any particular implementation. The model used in this service specification is identical to that used in 1.2.2.

The following primitive is defined:

— MM_CTL.request

99.2.1 MM_CTL.request

This primitive defines a request from a MAC Client to MAC Merge sublayer to hold or release transmission of preemptable traffic.

99.2.1.1 Semantics

The semantics of the primitive are as follows:

MM_CTL.request (hold_req)

The hold_req parameter takes one of two values, HOLD or RELEASE. The value of HOLD causes the MAC Merge sublayer to hold transmission of preemptable traffic. The value of RELEASE causes the MAC Merge sublayer to release transmission of preemptable traffic.

99.2.1.2 When generated

The generation of this primitive is out of scope of this standard.

99.2.1.3 Effect of receipt

Receipt of the primitive with the hold_req parameter set to the value HOLD causes preemption when the conditions allow preemption and prevents starting transmission of pMAC packets until this primitive is received with the value RELEASE.

Receipt of the primitive with the hold_req parameter set to the value RELEASE allows MAC Merge sublayer to transmit packets from the pMAC when the eMAC does not have a packet to transmit.

99.3 MAC Merge Packet (mPacket)

When the preemption capability is active, MAC Merge sublayer sends mPackets to the Reconciliation sublayer (RS). An mPacket contains one of the following:

- a complete express packet,
- a complete preemptable packet
- an initial fragment of a preemptable packet, or
- a continuation fragment of a preemptable packet.

99.3.1 mPacket format

Part (a) of Figure 99–4 shows the format of an mPacket containing an express packet, a complete preemptable packet, or the initial fragment of a preemptable packet. Part (b) of Figure 99–4 shows the format of an mPacket containing a continuation fragment of a packet.

The mPacket format is indicated by the value of the SMD (see 99.3.3).

99.3.2 Preamble

The preamble in the mPacket shown in part (a) of Figure 99–4 contains seven octets. The preamble in the mPacket shown in part (b) of Figure 99–4 contains six octets. Each octet contains the value of 0x55 (transmitted in order from left to right 10101010).

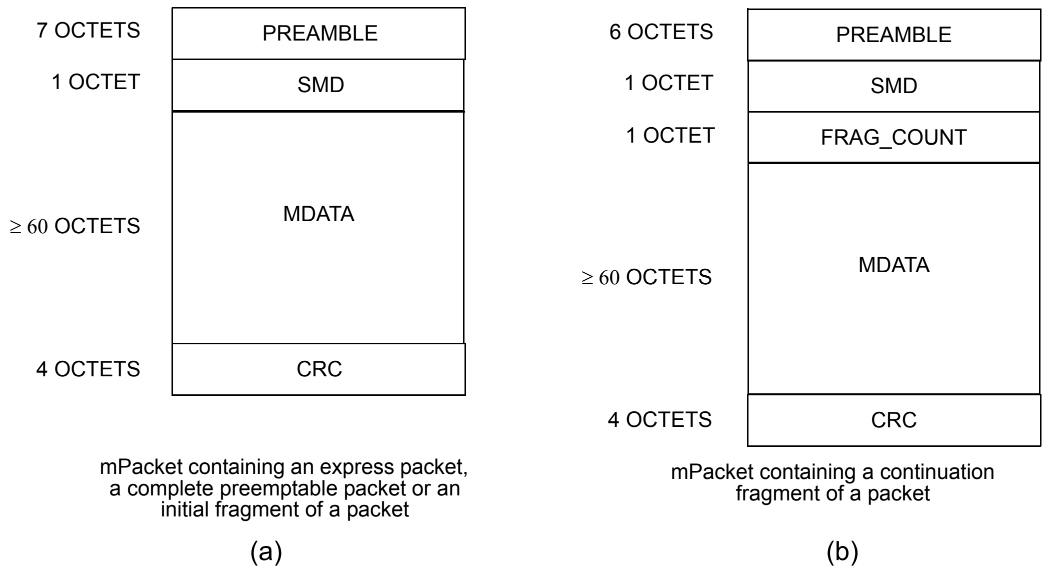


Figure 99–4—mPacket format

99.3.3 Start mPacket Delimiter (SMD)

The value of the SMD indicates whether the mPacket contains an express packet, the start of a preemptable packet (initial fragment or complete packet), or any of continuation fragments of a preemptable packet. All valid SMD values are defined in Table 99–1.

An mPacket carrying an express packet (transmitted by eMAC) has the same format as the express packet, because SMD-E (i.e., SMD value for an express packet) is same as the SFD value. An mPacket carrying a complete (non-fragmented) preemptable packet or the initial fragment of a preempted preemptable packet (transmitted by pMAC) has the SFD octet replaced with the appropriate SMD value. An mPacket carrying any of the continuation fragments of a preempted preemptable packet (transmitted by pMAC) has a continuation fragment SMD value, and includes an additional fragment counter octet (frag_count) following the SMD.

The SMD in an mPacket carrying a complete (non-fragmented) preemptable packet or any of the fragments of a preemptable packet also indicates the frame count. Information about the frame count prevents reassembling an invalid packet if the final mPacket of one preemptable packet and the initial fragment of the next preemptable packet are lost. The frame count is a modulo-4 count.

The term “SMD-S” refers to any of the four SMD values (SMD-S0, SMD-S1, SMD-S2, and SMD-S3) in an mPacket carrying the start of a preemptable packet. The term “SMD-C” refers to any of the four SMD values (SMD-C0, SMD-C1, SMD-C2, and SMD-C3) in an mPacket carrying a continuation fragment of a preemptable packet.

Two additional SMD values, SMD-V and SMD-R, identify mPackets used to verify that a link can support the preemption capability.

99.3.4 frag_count

The frag_count is a modulo-4 counter that increments for each continuation fragment of the preemptable packet. The frag_count protects against mPacket reassembly errors by enabling detection of the loss of up to 3 packet fragments.

Table 99–1—SMD values

mPacket type	Notation	Frame count	Value
verify packet	SMD-V	—	0x07
respond packet	SMD-R	—	0x19
express packet	SMD-E	—	0xD5
preemptable packet start	SMD-S0	0	0xE6
	SMD-S1	1	0x4C
	SMD-S2	2	0x7F
	SMD-S3	3	0xB3
continuation fragment	SMD-C0	0	0x61
	SMD-C1	1	0x52
	SMD-C2	2	0x9E
	SMD-C3	3	0x2A

The frag_count field is present only in mPackets with SMD-C. The frag_count is zero in the first continuation fragment of each preemptable packet.

The valid frag_count values are shown in Table 99–2.

Table 99–2—Frag_count values

Frag_count	Value
0	0xE6
1	0x4C
2	0x7F
3	0xB3

99.3.5 mData

The contents of the frame from the MAC from the first octet after the SFD to the last octet before the FCS are sent in the mData fields of the one or more mPackets for that frame. The minimum size of the mData field is 60 octets.

NOTE—The Transmit Processing State Diagram ensures that the minimum mData field size is met. There is no pad field.

99.3.6 CRC

The CRC field contains a cyclic redundancy check (CRC) and an indication of whether this is the final mPacket of a frame.

In the final mPacket of a frame, the CRC field contains the last 4 octets of the MAC frame (the FCS field).

For other mPackets, the CRC field contains an mCRC value. This includes mPackets used to verify that a link can support the preemption capability. The mCRC shall be calculated on the octets of the frame from the first octet of the frame (i.e., the octet following the SFD sent by the pMAC) to the last octet of the frame transmitted in that mPacket by:

- performing steps a) through d) in 3.2.9, and then
- XORing the calculated 32 bits with 0x0000FFFF.

NOTE—0x0000 is XORed with two octets that contain the higher order coefficients of the CRC, and 0xFFFF is XORed with the two octets that contain the lower order coefficients of the CRC.

99.4 MAC Merge sublayer operation

The MAC Merge sublayer receiver always operates the same way regardless of whether preemption is active. This allows MAC Merge sublayers to enable and use preemption once the link partner has indicated support for it, without sychronizing the transition between the two ends of the link.

The express mPacket format is the same as the MAC packet format. As a result, any packet received from a device that does not implement the MAC Merge sublayer or that has preemption disabled is received through the eMAC.

99.4.1 MAC Merge sublayer transmit behavior when preemption is disabled

When preemption is disabled, the packets presented by the pMAC and eMAC pass through the MAC Merge sublayer without alteration, i.e., the MAC Merge sublayer transmits packets rather than mPackets. If both the eMAC and pMAC have packets ready to transmit and no packet is being transmitted, the eMAC packet is transmitted. If a pMAC packet is being transmitted and the eMAC has a packet to transmit, the packet from the eMAC is transmitted after transmission of the pMAC packet completes.

99.4.2 Determining that the link partner supports preemption

The preemption capability is enabled in the transmit direction only if it is determined that the link partner supports the preemption capability.

The process of discovering the support for the preemption capability on the link partner relies on the exchange of the Additional Ethernet Capabilities TLV (see 79.3.7).

The preemption capability shall be enabled only if the link partner announces its support for the preemption capability via an Additional Ethernet Capabilities TLV in an LLDPDU addressed to the Nearest Bridge group address (see IEEE Std 802.1Q). The preemption capability shall be disabled if the MAC Merge sub-layer receives indication of link failure.

NOTE—Indication of link failure to the MAC Merge sublayer is implementation dependent.

99.4.3 Verifying preemption operation

Verification (see Figure 99–3) checks that the link can support the preemption capability.

If verification is enabled, the preemption capability shall be active only after verification has completed successfully.

If the preemption capability is enabled but has not been verified yet, the MAC Merge sublayer initiates verification. Verification relies on the transmission of a verify mPacket and receipt of a respond mPacket to confirm that the remote station supports the preemption capability. The format of a verify mPacket and a respond mPacket is as shown in part (a) of Figure 99–4, with the SMD values defined in Table 99–1 and an mData field containing 60 octets of 0x00.

When an mPacket with an SMD-V and a correct mCRC is received, a respond mPacket is sent. A respond mPacket has 7 octets of preamble (0x55), an SMD-R, 60 octets of 0x00, and an mCRC.

When an mPacket with an SMD-R and a correct mCRC is received, the preemption capability is verified.

Verification may be disabled. Verification disable is intended for engineered closed networks (i.e., where it is ensured by design that links can support preemption capability). Verification disable does not affect the transmission of respond mPackets.

If verification is enabled, it shall be performed as specified in part (a) of Figure 99–8.

Respond shall be performed as in part (b) of Figure 99–8.

99.4.4 Transmit processing

Transmit processing (see Figure 99–3) receives packets from the eMAC and pMAC. Transmit processing preempts a preemptable packet when a MM_CTL.request primitive is received with a hold_req parameter value of HOLD or the eMAC has a packet to transmit if that can be done while meeting minimum mPacket mData field size. Therefore, preemption occurs only if at least 60 octets of the preemptable frame have been transmitted and at least 64 octets (including the frame CRC) remain to be transmitted.

The earliest starting position of preemption is controlled by the addFragSize variable. Preemption does not occur until at least $64 \times (1+ \text{addFragSize}) - 4$ octets of the preemptable frame have been sent. The addFragSize variable is set to the value of the addFragSize field in the received Additional Ethernet Capabilities TLV (see 79.3.7).

Transmit processing includes a CRC generator to calculate an mCRC as specified in 99.3.6.

When the preemption capability is active, transmit processing replaces the SFD of a pMAC packet with an SMD-S value. A 2-bit rolling frame count is encoded in the SMD-S value. The SMD-E value is the same as the SFD value so the SFD of an eMAC packet does not need to be replaced.

When a packet is preempted, transmit processing appends the mCRC to the non-final mPackets. For the final mPacket of a preempted frame, the CRC field contains the CRC of the preempted MAC frame (the FCS field).

Transmit processing resumes transmission of the remainder of a preempted packet by sending preamble followed by an SMD-C and frag_count before continuing transmission of the packet. The SMD-C encodes the same frame count value as the SMD-S of the initial fragment.

Transmit processing shall be performed as specified in Figure 99–5.

If a PLS_CARRIER.indication is received from the PLS, PLS_CARRIER.indications with the same CARRIER_STATUS shall be sent to the pMAC and the eMAC.

99.4.5 Receive processing

Receive processing (see Figure 99–3) checks the SMD of each received mPacket. If an mPacket contains an SMD-E, Receive processing ignores the mPacket.

An SMD containing an SMD-S indicates the initial mPacket of a preemptable packet.

If an mPacket containing an SMD-S is received when Receive processing has not completed receiving the previous preempted packet, Receive processing shall ensure that the MAC detects a FrameCheckError in the partially received frame. This requirement may be met by producing a received frame data sequence delivered to the MAC sublayer that is guaranteed to not yield a valid CRC result, as specified by the frame check sequence algorithm (see 3.2.8). This data sequence may be produced by substituting data delivered to the MAC. It can do this by checking that the prior four octets sent to the MAC did not match the CRC of the data sent before them or by sending eight additional PLS_DATA.indication primitives to the pMAC or by implementation dependent means. Other techniques may be employed to respond to an incomplete packet provided that the result is that the MAC sublayer behaves as though a FrameCheckError occurred in the received frame.

When an SMD-S is detected, receive processing records the frame count indicated by the SMD. Receive processing sends an SFD to the pMAC in place of the SMD and then begins sending PLS_DATA.indication primitives to the pMAC.

Receive processing checks whether the last four octets of the mPacket match the mCRC. If the last four octets of the mPacket do not match, that indicates the end of the packet. The PLS_DATA.indications for the last four octets are sent to the pMAC followed by PLS_DATA_VALID.indication(DATA_NOT_VALID). If the last for octets of the mPacket match, that indicates that the packet was preempted and the PLS_DATA.indications for the last four octets are not sent to the pMAC. No further primitives are sent to the pMAC until the next mPacket of the packet is received.

An SMD containing an SMD-C indicates an mPacket that continues the data for a preempted packet.

Upon receiving an SMD value of SMD-C, Receive processing checks that:

a) There is a preempted packet in progress,
b) The frame count indicated by the SMD matches the frame count of the packet in progress, and
c) The frag_count value indicates the next fragment count.

If any of the checks fail, the mPacket is discarded and Receive processing ensures that the pMAC detects a FrameCheckError as previously described. Then PLS_DATA_VALID.indication(DATA_NOT_VALID) is sent to the pMAC.

If all the checks pass, the next fragment count is incremented modulo 4 and the PLS_DATA.indications after the frag_count field are sent to the pMAC.

Receive processing handles the last four octets of the mPacket as previously described.

Receive processing shall be performed as specified in Figure 99–6.

PLS_SIGNAL.indication is never produced by Receive processing since it does not apply to full duplex PHYs.

99.4.6 Express filter

The Express filter (see Figure 99–3) checks the SMD of each received mPacket. If an mPacket contains an SMD-E, the Express filter passes the RS PLS_DATA.request to the eMAC. If the SMD contains any other value, Express filter ignores the mPacket.

Express processing shall be performed as specified in Figure 99–7.

PLS_SIGNAL.indication is never produced by Express filter since it does not apply to full duplex PHYs.

99.4.7 Detailed functions and state diagrams

99.4.7.1 State diagram conventions

The body of this subclause is comprised of state diagrams, including the associated definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

The notation used in the state diagrams follows the conventions of 21.5. State diagram timers follow the conventions of 14.2.3.2. The notation ++ after a counter or integer variable indicates that its value is to be incremented.

PLS service interface primitives are prefixed by a letter to designate the PLS service interface as follows:

- e PLS service interface between the eMAC and MAC Merge sublayer
- p PLS service interface between the pMAC and MAC Merge sublayer
- r PLS service interface between MAC Merge sublayer and RS

For example, ePLS_DATA.request is a PLS_DATA.request primitive from the eMAC.

99.4.7.2 Constants

minFrag
: the integer 64, the number of octets required for a minimum size fragment.

PREAMBLE
: 0x55

SFD
: 0xD5

verifyLimit
: the integer 3, the number of verification attempts

99.4.7.3 Variables

addFragSize
: An integer in the range 0:3 that controls the minimum non-final mPacket length, as specified in 99.4.4. Set to the value of the addFragSize field in the received Additional Ethernet Capabilities TLV (see 79.3.7).

begin
: A Boolean variable that is set TRUE by implementation dependent means to initialize the state machines.

cFrameCnt
: An integer in the range 0:3 set by the SMD_DECODE function invoked on the SMD-C of a continuation packet that indicates the frame count of the continuation mPacket.

disableVerify
: A Boolean variable that is set by management to control verification of preemption operation (see 99.4.3). TRUE disables verification and FALSE enables verification.

eTx
: Boolean variable that is TRUE when an ePLS_DATA.request has been received and a corresponding rPLS_DATA.request has not yet been generated. FALSE otherwise.

eTxCplt
: Boolean variable that is set TRUE when ePLS_DATA.request with the value DATA_COMPLETE is received and set FALSE when the end of frame has been processed.

keepSafterD
: Boolean variable that indicates whether an implementation is able to process the start of a packet while discarding an errored packet. When an SMD-S is received while Receive processing is waiting for the next mPacket of a preempted packet, the preempted packet is discarded. The value TRUE indicates that an implementation is able to process the new packet when this occurs.

hold
: Boolean variable that is set TRUE when MM_CTL.request is received with a hold_req parameter value of HOLD and FALSE when MM_CTL.request is received with a hold_req parameter value of RELEASE.

link_fail
: Boolean variable that is TRUE when the MAC Merge sublayer receives indication of link failure. FALSE otherwise.

NOTE—Indication of link failure to the MAC Merge sublayer is implementation dependent.

rRxDv
: Boolean variable that is set TRUE when rPLS_DATA_VALID.indication is received with the value DATA_VALID and set FALSE when rPLS_DATA_VALID.indication is received with the value DATA_NOT_VALID.

pActive
: Boolean variable that is TRUE when the preemption capability is active and FALSE otherwise. The value of pActive is pEnable ∗ (verified + disableVerify).

pAllow
: Boolean variable that is set to the value of pActive when SEND_SMD_S state is entered.

pEnable
: Boolean variable that is set TRUE to enable the preemption capability and set FALSE to disable the preemption capability.

preempt
: Boolean that is TRUE when a preemptable packet is to be preempted. The value of preempt is: pAllow ∗ (eTx + hold) ∗ preemptableFragSize ∗ MIN_REMAIN.

preemptableFragSize:
: Boolean variable that is TRUE when a enough bits of the current preemptable packet have been transmitted to allow it to be preempted. The value of preemptableFragSize is:
fragSize ≥ (minFrag × (1 + addFragSize) – 4).

pTX
: Boolean variable that is TRUE when a pPLS_DATA.request has been received and a corresponding rPLS_DATA.request has not yet been generated. FALSE otherwise.

pTxCplt
: Boolean variable that is set TRUE when pPLS_DATA.request with the value DATA_COMPLETE is received and set FALSE when the end of packet has been processed.

rcv_r
: Boolean that is set TRUE when an mPacket with SMD-R and a correct mCRC is received and set FALSE when the INIT_VERIFICATION state of the VERIFY State diagram is entered.

rcv_v
: Boolean that is set TRUE when an mPacket with SMD-V and a correct mCRC is received and set FALSE when the INIT_VERIFICATION state of the VERIFY State diagram or the SEND_RESPOND state of the Respond State diagram is entered.

resumeTx
: Boolean variable that is TRUE if a preemptable packet is waiting to resume transmission.

rxFragCnt
: An integer in the range 0:4 set by the FRAG_DECODE function invoked on the frag_count of a continuation packet that indicates the fragment count of the continuation mPacket.

rxFrameCnt
An integer in the range 0:3 set by the SMD_DECODE function invoked on the SMD-S of a continuation packet that indicates the frame count of the initial mPacket.

send_r
Boolean that is set TRUE to initiate sending a respond mPacket and set FALSE when a respond mPacket is sent.

send_v
Boolean that is set TRUE to initiate sending a verification mPacket and set FALSE when a verify mPacket is sent.

verified
Boolean that is set TRUE when the ability of the link to support the preemption capability has been verified and FALSE when the INIT_VERIFICATION state of the VERIFY State diagram is entered.

verify_fail
Boolean that is set TRUE when verification attempts have failed and FALSE when the INIT_VERIFICATION state of the VERIFY State diagram is entered.

verifyTime
An integer in the range 1:128 used to configure the number of ms after which the verify_timer expires (see 99.4.7.6). The default value of verifyTime is 10.

99.4.7.4 Functions

For functions that produce primitives based on an 8-bit vector, a primitive value is ONE if the corresponding bit is 1 and ZERO if the corresponding bit is 0. The primitives are produced from bit 0 to bit 7 in sequence.

For functions that produce an 8-bit vector based on eight primitives, a bit is 1 if the corresponding primitive value is ONE and 0 if the corresponding primitive is ZERO. The primitives are mapped to bit 0 to bit 7 in sequence.

When the value of a vector is expressed as a hexadecimal number, the LSB corresponds to bit 0 and the MSB corresponds to bit 7.

DISCARD
Ensures that the MAC detects a FrameCheckError in that frame (see 99.4.5) and then invokes pRX_DV(FALSE). Used when Receive processing detects that the packet cannot be continued after it was preempted.

eRX_DATA(data<7:0>)
Produces eight ePLS_DATA.indication primitives based on the 8-bit vector data<7:0>.

eRX_DV(data_valid)
Produces an ePLS_DATA_VALID.indication. If the Boolean parameter data_valid is TRUE, the primitive value is DATA_VALID. If data_valid is FALSE, the primitive value is DATA_NOT_VALID.

eTX_DATA
Returns an 8-bit vector based on eight ePLS_DATA.request primitives.

FRAG_ENCODE(frag_cnt)
Creates an 8-bit vector with the frag_count encoding for a fragment count of frag_cnt (see Table 99–2). Produces eight rPLS_DATA.request primitives based on the 8-bit vector.

FRAG_DECODE
Decodes eight rPLS_DATA.indication primitives by producing an 8-bit vector and comparing it to encoded frag_count values (see Table 99–2). If frag_count contains a valid value, places the fragment count decoded in rxFragCnt. Otherwise, it sets rxFragCnt to 4.

MIN_REMAIN

Prescient function to check if enough octets of the current pMAC packet remain meet the minimum fragment requirement after preemption. Produces a Boolean value as follows:

TRUE ≥ minFrag octets are left to transmit
FALSE Otherwise

rRX_DATA

Returns an 8-bit vector based on eight rPLS_DATA.indication primitives.

rTX_CPLT

Produces an rPLS_DATA.request primitive with the value DATA_COMPLETE.

rTX_DATA(data<7:0>)

Produces eight rPLS_DATA.request primitives based on the 8-bit vector data<7:0>.

pRX_DATA(data<7:0>)

Produces eight pPLS_DATA.indication primitives based on the 8-bit vector data<7:0>.

pRX_DV(data_valid)

Produces a pPLS_DATA_VALID.indication. If the Boolean parameter data_valid is TRUE, the primitive value is DATA_VALID. If data_valid is FALSE, the primitive value is DATA_NOT_VALID.

pTX_DATA

Returns an 8-bit vector based on eight pPLS_DATA.request primitives.

RX_MCRC_CK

Prescient function returning a Boolean value. The value is TRUE if rPLS_DATA_VALID.indication with a value of DATA_NOT_VALID will be received after the next 32 rPLS_DATA.indication primitives and the next 32 rPLS_DATA.indications equal the computed mCRC result for the preemptable packet being received. It is FALSE otherwise.

SFD_DET

Prescient function returning a Boolean value. The value is TRUE if an 8-bit vector produced from the next eight pPLS_DATA.request primitives contains an SFD.

SMD_DECODE

Decodes the value of 8 rPLS_DATA.indication primitives by producing an 8-bit vector and returning one of the following values based on the value of the vector (see Table 99–1):

Preamble 0x55
E SMD-E encoding
S SMD-S encoding
C SMD-C encoding
V SMD-V encoding
R SMD-R encoding
ERR Any other value - error.

If S is returned, the function sets rxFrameCnt equal to the frame count indicated by the SMD-S. If C is returned, the function sets cFrameCnt equal to the frame count indicated by the SMD-C.

SMDC_ENCODE(frame_cnt)

Creates an 8-bit vector with the SMD encoding for an SMD-C with frame count of frame_cnt. Produces eight rPLS_DATA.request primitives based on the 8-bit vector.

SMDS_ENCODE(frame_cnt)

Returns an 8-bit vector with the SMD encoding for an SMD-S with frame count of frame_cnt if pAllow is TRUE. Otherwise it returns a vector containing 0xD5 (i.e., SFD). Consumes 8 pPLS_DATA.request primitives containing the SFD.

TX_MCRC

Produces 32 rPLS_DATA.requests that transmit the mCRC computation for the mPacket.

TX_R

Produces 576 rPLS_DATA.requests to send a respond mPacket: 7 octets of 0x55, 1 octet SMD-R, 60 octets of 0x00, mCRC followed by an rPLS_DATA.request with the value DATA_COMPLETE.

TX_V
: Produces 576 rPLS_DATA.requests to send a verify mPacket: 7 octets of 0x55, 1 octet SMD-V, 60 octets of 0x00, mCRC followed by an rPLS_DATA.request with the value DATA_COMPLETE.

99.4.7.5 Counters

fragSize
: a count of the number of octets of mData transmitted in the current preemptable mPacket.

nxtRxFrag
: a modulo-4 counter containing the next expected fragment count for the next incoming continuation packet.

preambleCnt
: a count of the number of preamble octets sent for a continuation fragment.

txFrag
: a modulo-4 counter containing the fragment count for the next outgoing continuation mPacket.

txFrame
: a modulo-4 counter containing the frame count for the next outgoing preemptable packet.

verifyCnt
: a count of the number of verification attempts that have completed.

99.4.7.6 Timers

ipg_timer
: A timer counting bit times since the end of the prior packet. The timer expires 96 bit times after being started.

verify_timer
: A timer of time from when a verification mPacket was sent to initiating the next attempt. The timer expires verifyTime ms ± 20% after being started.

99.4.7.7 State diagrams

The Transmit Processing State Diagram is shown in Figure 99–5. The Receive Processing State Diagram is shown in Figure 99–6. The Express Filter State Diagram is shown in Figure 99–7. The Verify State Diagram and Respond State Diagram are shown in Figure 99–8.

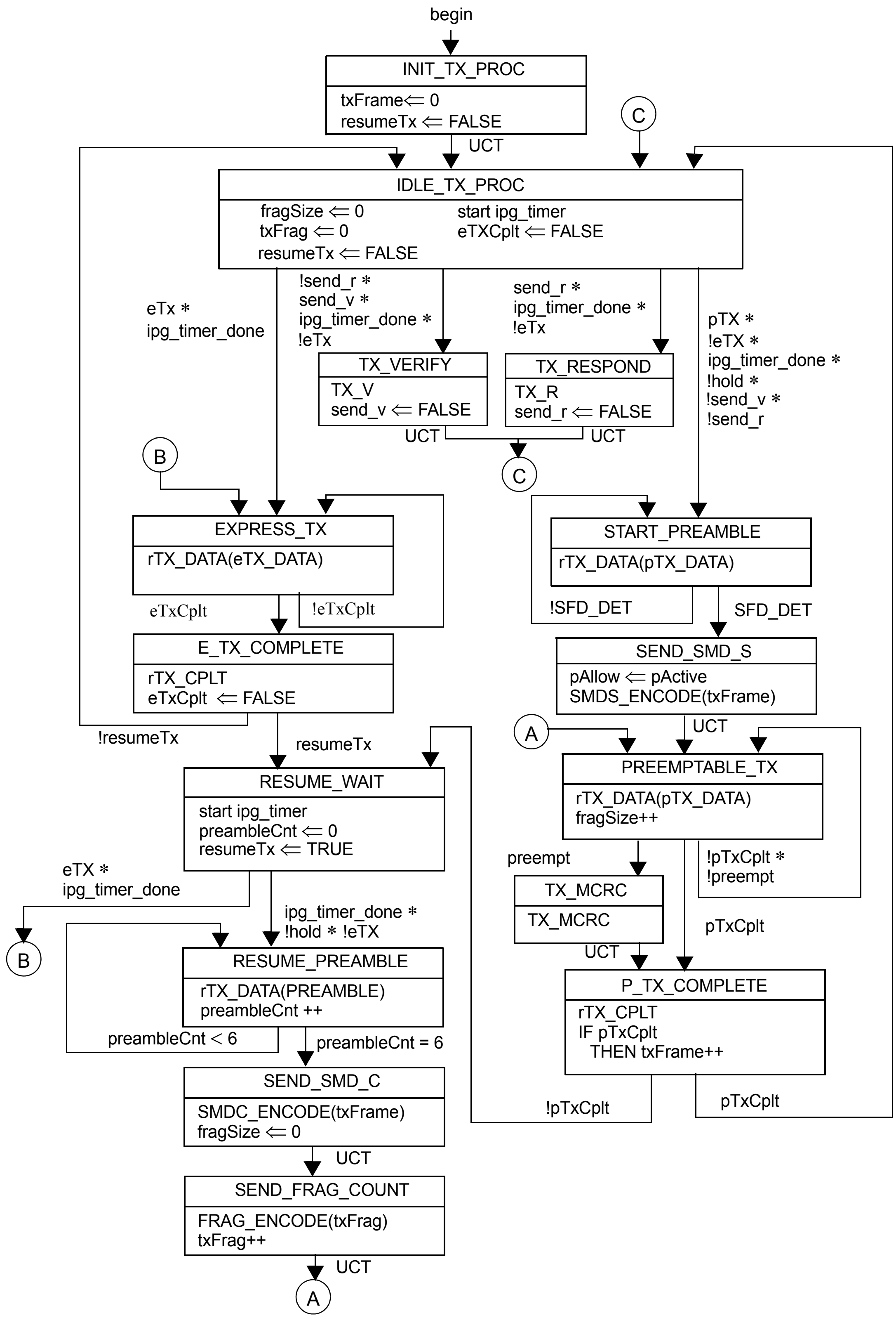


Figure 99–5—Transmit Processing State Diagram

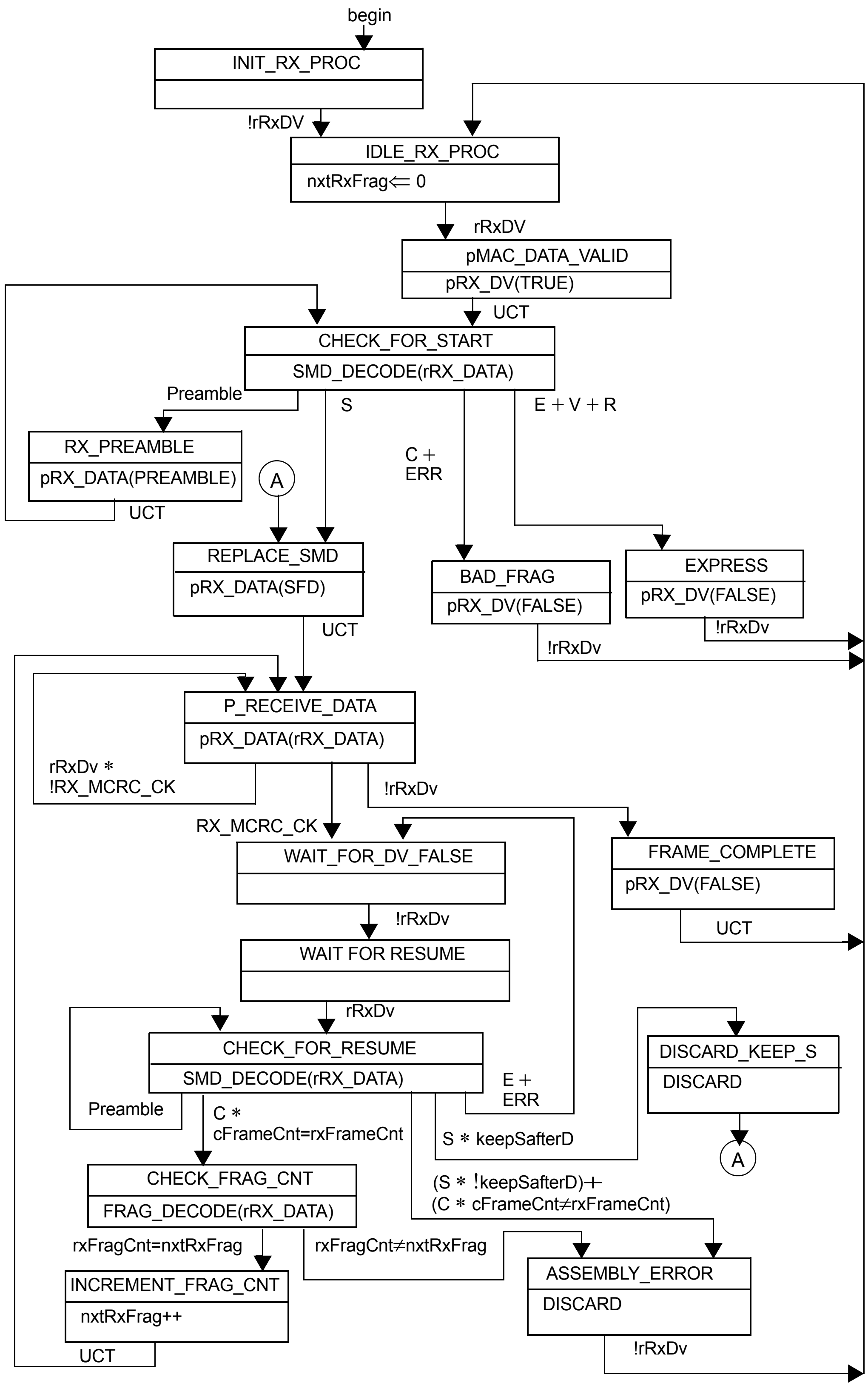


Figure 99–6—Receive Processing State Diagram

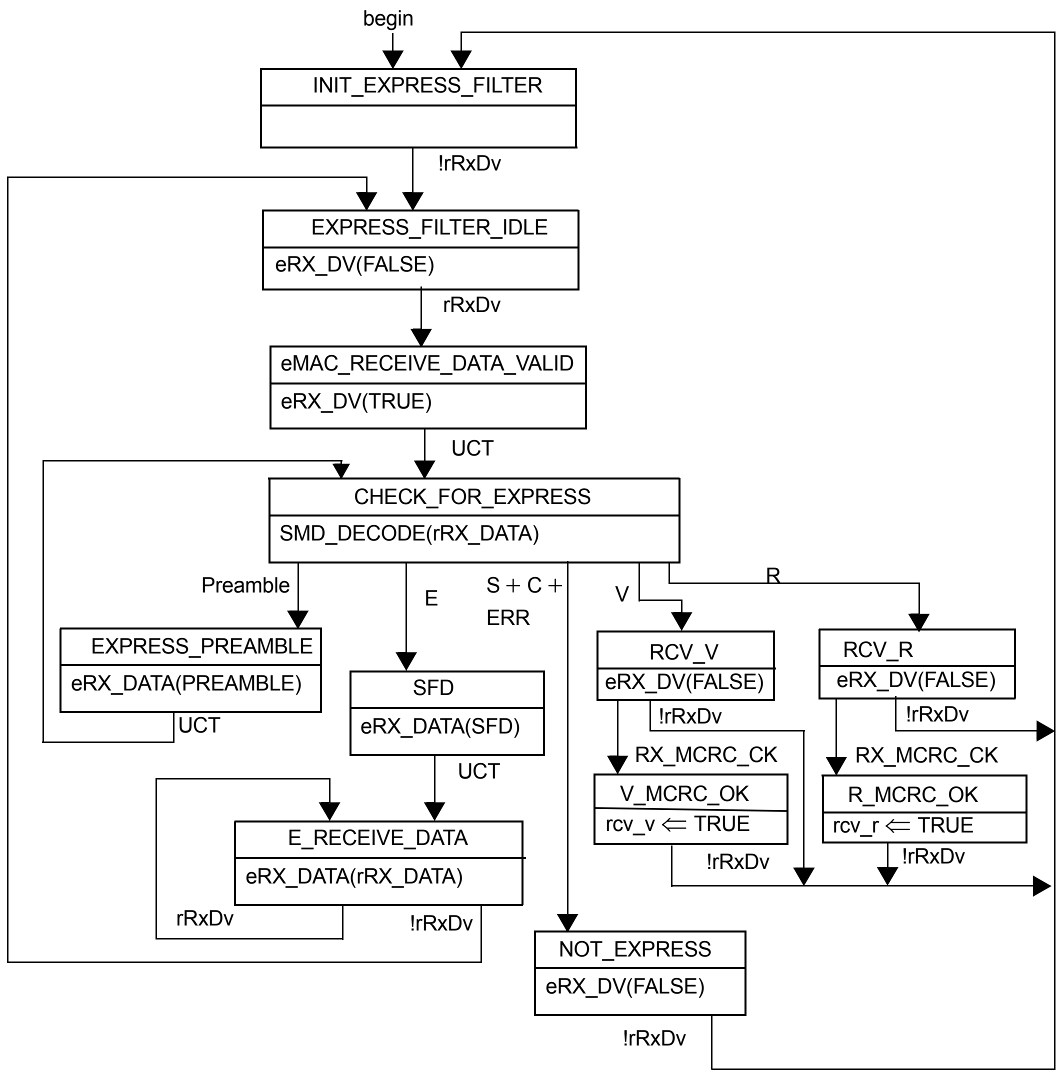


Figure 99–7—Express Filter State Diagram

99.4.8 Delay Constraints

Predictable operation for time sensitive traffic (e.g., IEEE Std 802.1Q, Clause 31, Annex 31D PFC operation) demands that there be an upper bound on the propagation delays through the network. This implies that MAC, MAC Control sublayer, and PHY implementors must conform to certain delay maxima, and that network planners and administrators conform to constraints regarding the cable topology and concatenation of devices.

If the preemption capability is active, transmission of preemptable traffic shall be preempted within the hold response time (HRT) after the MAC Client sends MM_CTL.request with a hold_req parameter value of HOLD. HRT is (1240 + 512 × addFragSize) bit times.

When preemptable traffic is held by an MM_CTL.request with a hold_req parameter value of HOLD sent by the MAC Client at least HRT before the MA_DATA.request, the maximum cumulative MAC Control, MAC, MAC Merge sublayer and RS delay for an express packet shall meet the delay specified for a MAC Control, MAC and RS delay based on the MAC operating speed.

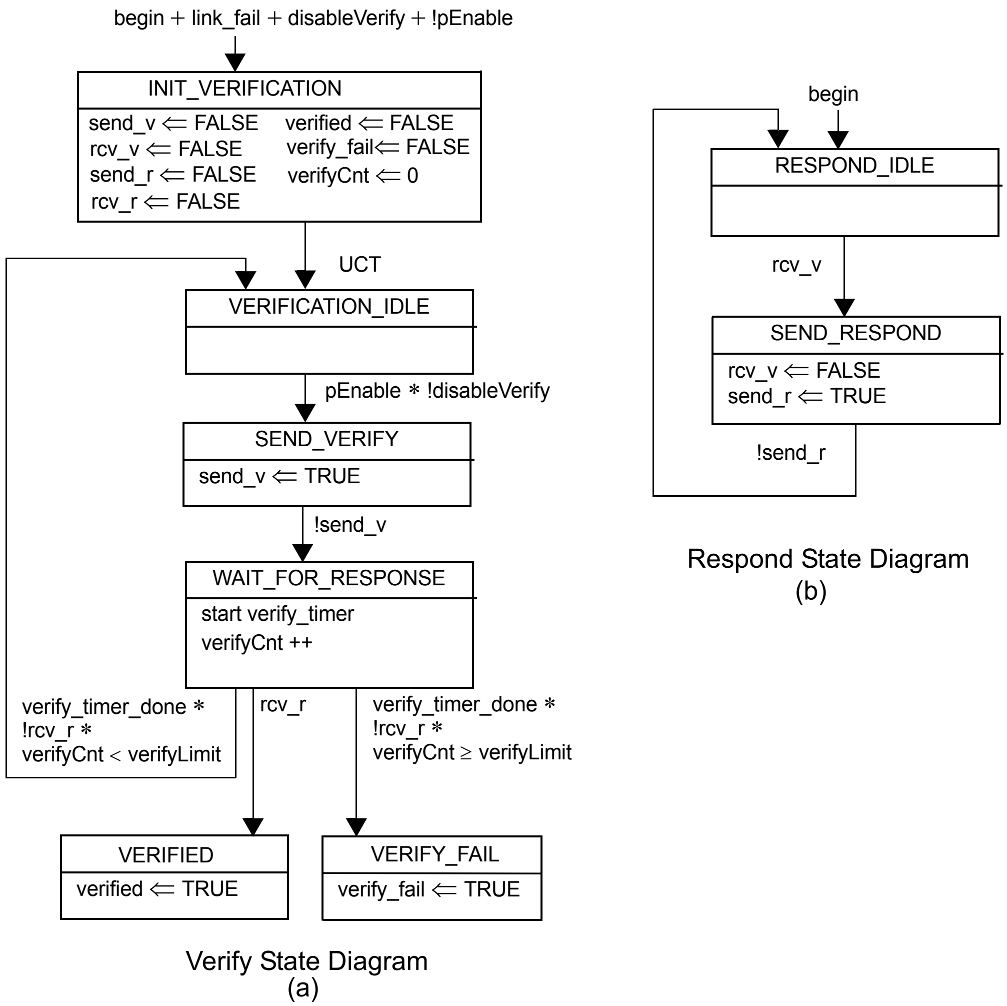


Figure 99–8—Verify State Diagram (a) and Respond State Diagram (b)

When preemption is active and preemptable traffic is not held by an MM_CTL.request with a hold_req parameter value of HOLD sent by the MAC Client at least HRT before the MA_DATA.request, the maximum cumulative MAC Control, MAC, MAC Merge sublayer, and RS delay for an express packet shall meet the delay specified for a MAC Control, MAC and RS delay based on the MAC operating speed delay plus HRT.

When preemptable traffic is released by an MM_CTL.request with a hold_req parameter value of RELEASE sent by the MAC Client at least 96 bit times before the MA_DATA.request and no express packets are being sent, the maximum cumulative MAC Control, MAC, MAC Merge sublayer, and RS delay for a preemptable packets shall meet the delay specified for a MAC Control, MAC and RS delay based on the MAC operating speed.

99.5 Protocol implementation conformance statement (PICS) proforma for Clause 99, MAC Merge sublayer[5]

99.5.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 99, MAC Merge sublayer, shall complete the following protocol implementation conformance statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

99.5.2 Identification

99.5.2.1 Implementation identification

Supplier[1]	
Contact point for enquiries about the PICS[1]	
Implementation Name(s) and Version(s)[1, 3]	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)[2]	
NOTE 1—Required for all implementations. NOTE 2—May be completed as appropriate in meeting the requirements for the identification. NOTE 3—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

99.5.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3br-2016, Clause 99, MAC Merge sublayer
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No [ ] Yes [ ] (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3br-2016.)	

Date of Statement	

[5]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

99.5.3 PICS proforma tables for MAC Merge sublayer

99.5.3.1 Functional specifications

Item	Feature	Subclause	Value/Comment	Status	Support
MM2	mCRC computation	99.3.6	Calculated as described	M	Yes [ ]
MM3	Enabling preemption capability	99.4.2	Enabled only if the link partner announces support for the pre-emption capability as described	M	Yes [ ]
MM5	Preemption capability disabled on link failure	99.4.2	Disabled on link failure indication	M	Yes [ ] No [ ]
MM6	Activating preemption capability	99.4.3	When verification is enabled, capability is active only after verification has completed successfully	M	Yes [ ]
MM7	Verify	99.4.3	Performed as specified in part (a) of Figure 99–8 if enabled	M	Yes [ ]
MM8	Respond	99.4.3	Performed as specified in part (b) of Figure 99–8	M	Yes [ ]
MM9	Transmit processing	99.4.4	Performed as specified in Figure 99–5	M	Yes [ ]
MM10	PLS.CARRIER.indication	99.4.4	If received, sent to the pMAC and eMAC	M	Yes [ ]
MM11	Receive processing	99.4.5	Performed as specified in Figure 99–6	M	Yes [ ]
MM12	Frame error handling	99.4.5	Ensure that the MAC detects FrameCheckError	M	Yes [ ]
MM13	Express filter	99.4.6	Performed as specified in Figure 99–7	M	Yes [ ]
MM14	No PAUSE	99.1	MAC Control sublayer does not generate PAUSE	M	Yes [ ]

99.5.3.2 Delay constraints

Item	Feature	Subclause	Value/Comment	Status	Support
DC1	Time to preempt after MAC Client sends MM_CTL.request	99.4.8	No more than HRT	M	Yes [ ]
DC2	Delay to transmit express packet when preemptable traffic is held by MM_CTL.request	99.4.8	Meets the maximum cumulative MAC Control, MAC and RS delay	M	Yes [ ]
DC3	Delay to transmit express frame when preemptable traffic is not held by MM_CTL.request	99.4.8	Meets the maximum cumulative MAC Control, MAC and RS delay plus HRT	M	Yes [ ]
DC4	Delay to transmit preemptable packet when traffic is released by MM_CTL.request and no express packets are being sent.	99.4.8	Meets the maximum cumulative MAC Control, MAC and RS delay	M	Yes [ ]

Consensus

WE BUILD IT.

IEEE
standards.ieee.org
Phone: +1 732 981 0060 Fax: +1 732 562 1571