



IEEE Standard for Local and metropolitan area networks— Link Aggregation

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997, USA
3 November 2008

IEEE Std 802.1AX™-2008

IEEE Standard for Local and metropolitan area networks— Link Aggregation

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 26 September 2008

IEEE-SA Standards Board

Abstract: Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a Media Access Control (MAC) Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of data terminal equipment (DTE) to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. This standard defines the MAC independent Link Aggregation capability, and general information relevant to specific MAC types that support Link Aggregation.

Keywords: Aggregated Link, Aggregator, Link Aggregation, Link Aggregation Group, local area network, management

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

Copyright © 2008 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 3 November 2008. Printed in the United States of America.

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

PDF: ISBN 973-07381-5794-8 STD95818
Print: ISBN 973-07381-5795-5 STDPD95818

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

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the 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.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS**.”

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. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, 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 publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgement 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.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received 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 interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the 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, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be submitted to the following address:

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

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, 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.

Introduction

This introduction is not part of IEEE Std 802.1AX-2008, IEEE Standard for Local and Metropolitan Area Networks—Link Aggregation.

The majority of the content of this standard is from IEEE Std 802.3™ where it was originally developed as the amendment project IEEE P802.3ad. Management information is from Clause 30 with the bulk of the functionality from Clause 43 and its annexes.

Notice to users

Laws and regulations

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

Copyrights

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

Updating of IEEE documents

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

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

Errata

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

Interpretations

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

Patents

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

Participants

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

Robert M. Grow, *IEEE 802.3 Working Group Chair, IEEE 802.3ax Task Force Editor*

David J. Law, *IEEE 802.3 Working Group Vice-Chair, IEEE 802.3ax Task Force Chair*

Wael William Diab, *IEEE 802.3 Working Group Secretary*

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

Bradley Booth, *IEEE 802.3 Working Group Treasurer*

John Abbott
Joe Abler
Taufique Ahmed
Youichi Akasaka
Arne Alping
Michael Altman
Pete Anslow
Joseph N. Babanezhad
Jim Barnette
Hugh Barrass
Scott Barrick
Howard Baumer
Denis Beaudoin
Michaël Beck
Christian Beia
Michael Bennett
Ralf-Peter Braun
Dave Brearley
Matt Brown
Phillip Brownlee
Robert Brunner
Robert Busse
Jeffrey Cain
J. Martin Carroll
Mandeep Chadha

David Chalupsky
Frank Chang
Luke Chang
George Claseman
Terry Cobb
James Collum
David Cunningham
Fumio Daido
John Dallesasse
Yair Darshan
Piers Dawe
Bill Delveaux
Joe DeNicholas
Claudio DeSanti
Suveer Dhamejani
Chris Di Minico
Thomas Dineen
Thuyen Dinh
Dan Dove
Mike Dudek
Marcus Duell
Joseph E. Dupuis
Frank J. Effenberger
Daniel Feldman
Thomas Fischer

Alan Flatman
Howard M. Frazier
Richard Frosch
Ilango S. Ganga
Ali Ghiasi
Sajol Ghoshal
Russ Gyurek
Marek Hajduczenia
Brian Holden
Keith Hopwood
Walter K. Hurwitz
John Jaeger
John Jetzt
Wenbin Jiang
Andrew C. Jimenez
Chad Jones
Thomas K. Jørgensen
Shinkyō Kaku
Boris Katzenberg
Tatsuya Kawashimo
Mitsunobu Kimura
David J. Koenen
Paul Kolesar
Derek Koonce
Glen Kramer
Jan Krellner

Hans Lackner
D. Matthew Landry
Daun Langston
Jeff Lapak
Ferdinando Lari
Dong-Soo Lee
Worayot Lertniphonphun
Robert Lingle, Jr.
Jeffrey Lynch
Eric R. Lynskey
Jeff Mandin
Arthur Marris
Thomas Mathey
Mike McConnell
Michael S. McCormack
Richard Y. Mei
Richard Mellitz
Tremont Miao
Charles Moore
Andy Moorwood
Wayne A. Mueller
Toshiaki Mukojima
Shimon Muller
Gary Nicholl
Shinji Nishimura
Ron Nordin
Bob Noseworthy
Mark Nowell
Steven O'Hara

Satoshi Obara
Gourgen Oganessyan
Bengt-Erik Olsson
Akihiro Otaka
George Oulundsen, III
Tom Palkert
Tim Parker
Glenn Parsons
Martin Patoka
Arkadiy Peker
Petar Pepeljugoski
John Picard
Petre Popescu
Carl R. Posthuma
Rick Rabinovich
Prakash Radhakrishnan
Duane Remein
Steve Robbins
Valerie Rybinski
Naoto Saeki
Ramesh Sastry
Olindo Savi
T. Shannon Sawyer
Fred Schindler
Peter Schoenmaker
Lee Sendelbach
Shoichiro Seno
Masayuki Shigematsu
Larry Shorthill

Andrew Smith
Fulvio Spagna
Clay Stanford
Ken-ichi Suzuki
Steve Swanson
Andre Szczepanek
Akira Takahashi
Noriyuki Takeda
Motoyuki Takizawa
Keiji Tanaka
Tsutomu Tatsuta
Vivek Telang
Geoffrey O. Thompson
Peter Tomaszewski
Hidehiro Toyoda
Stephen J. Trowbridge
Shinji Tsuji
Brad Turner
Kiyoshi Uematsu
Sterling A. Vaden
Magesh Valliappan
Schelto van Doorn
Anoop Vetteth
Chenxi Wang
Bill Woodruff
Hajime Yamashita
Yinglin (Frank) Yang
Bin Yeong Yoon
George Zimmerman

Historical participants

Included is a historical list of participants from the IEEE 802.3 and IEEE 802.1 Working Groups who have dedicated their valuable time, energy, and knowledge to the creation of IEEE Std 802.3ad-2000 Link Aggregation amendment from which this standard is derived.

Geoffrey O. Thompson, *IEEE 802.3 Working Group Chair*

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

Robert M. Grow, *IEEE 802.3 Working Group Secretary*

Steven R. Haddock, *Chair, IEEE P802.3ad Task Force*

Tony Jeffree, *Co-editor, IEEE P802.3ad Task Force*

Rich Seifert, *Co-editor, IEEE P802.3ad Task Force*

Oscar Agazzi
Don Alderrou
Abe Ali
Khaled Amer
Ralph Andersson
Ken-ichi Arai
Kameran Azadet
Keith Balmer
Denis Beaudoin
John L. Bestel
Michel Bohbot
Brad J. Booth
Paul Bottorff
Gary Bourque
Steve Brewer
Vince Bridgers

Benjamin Brown
Lisa Buckman
Robert Busse
Richard Cam
Bob Campbell
Edward G. Chang
Edward S. Chang
Hon Wan Chin
Chris Christ
Terry Cobb
Edward Cornejo
John Creigh
Kevin Daines
Tom Debiec
Chris Di Minico
Erik Dickens

Thomas J. Dineen
Dan Dove
Steve Dreyer
George Eisler
Michael Elswick
John F. Ewen
Mark Feuerstraeter
Jens Fiedler
Norival Figueira
Norman Finn
Alan Flatman
Howard M. Frazier
Scott Fritz
Darrell Furlong
Giorgio Giaretta
Joel Goergen

Bin Guo
Atikem Haile-Mariam
Sharam Hakimi
Del Hanson
Doug Harshbarger
Lloyd Hasley
Marwan Hassoun
Mehdi Hatamian
Gaby Hecht
Ken Herrity
John Hill
Henry Hinrichs
Clarence Joh
Thomas K. Jørgensen
Juan Jover
Paul Jury
Shinkyo Kaku
Mohan Kalkunte
Amrit Kalla
Hadriel Kaplan
Jaime Kardontchik
Toyoyuki Kato
Daniel Kelley
Yongbum Kim
Atsushi Kimoto
Keith Klamm
Christine Koenig
Paul F. Kolesar
Kishan Rao Konda
Josef Kosilek
Hans Lackner
Daun Langston
Loren Larsen
Brian E. Lemoff
Bill Lidinsky
George Lin
Jeffrey Lynch
Brian MacLeod
Kenneth MacLeod

Rabih Makarem
Robert A. Marsland
David W. Martin
Thomas Mathey
Tremont Miao
Colin Mick
Larry Miller
Cindy Montstream
Robert Mortonson
Simon Moseley
Shimon Muller
Denis Murphy
Yaron Nachman
Ken Naganuma
Hari Naidu
Paul Nikolich
Michael Nootbaar
Bob Noseworthy
Mark Nowell
Satoshi Obara
Toshio Ooka
Don Pannell
Luc Pariseau
John Payne
Robert Pieters
John Proffitt
Steve Pryor
William Quackenbush
Sailesh K. Rao
Peter Rautenberg
Anil Rijsinghani
Carlos Rodriguez
Shawn Rogers
Floyd E. Ross
Tam Ross
Larry Rubin
Peter Sallaway
Bill Sarles
J. David Schell

Ted Schroeder
Mick Seaman
Lee Sendelbach
Koichiro Seto
Som Sikdar
Alexander Smith
Andrew Smith
Michael A. Smith
David Sorensen
Michel Sørensen
Ronald Steudler
Stephen Strong
Steve Swanson
Tad Szostak
Rich Taborek
Pat Thaler
R. Jonathan Thatcher
Walter Thirion
Douglas Thomson
Bruce Tolley
Zbigniew Turlej
Edward Turner
Schelto van Doorn
Dono Van-Mierop
Nader Vijeh
Greg Wang
Peter Wang
Jim Welch
Willem Wery
Bob Williams
Joris Wils
John Wolcott
David Wong
Michael Wrigh
Chong Ho Yoon
Leonard Young
Ben Yu
Jing-fan Zhang

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

Thomas Alexander	Robert M. Grow	Satoshi Obara
Danilo Antonelli	C. Guy	George Oulundsen, III
Hugh Barrass	Joseph Gwinn	Maximilian Riegel
Michael Bennett	John Hawkins	Jessy V. Rouyer
Rahul Bhushan	Adam Healey	Randall Safier
Tomo Bogataj	Atsushi Ito	John Sauer
Ralf-Peter Braun	Raj Jain	Benson Schliesser
William Byrd	Tony Jeffree	Marco Scorrano
Jeffrey Cain	David Johnston	Michael Seaman
Edward Carley	Peter Jones	Thomas Starai
James Carlo	Piotr Karocki	Marian Stasney
Juan Carreon	Stuart J. Kerry	Walter Struppler
Keith Chow	Yongbum Kim	Alourdes Sully
Alexander Conta	Paul Kolesar	Robert Sultan
Charles Cook	Thomas Kurihara	Joseph Tardo
John Dallesasse	David J. Law	Michael Johas Teener
John Dambrosia	Juan L. Lazaro	Geoffrey O. Thompson
Wael William Diab	John Lemon	Oliver Thorp
Russell Dietz	William Lumpkins	Thomas Tullia
Thomas Dineen	Mark Maloney	Edward J. Turner
Marc Emmelmann	Peter Martini	Mark-Rene Uchida
Jose Dominic Espejo	W. Kyle Maus	Schelto van Doorn
Prince Francis	Brett McClellan	David Willow
Howard M. Frazier	Jonathon McLendon	Michael Wright
Yukihiro Fujimoto	John Messenger	Takahito Yoshizawa
Devon Gayle	Jose Morales	Oren Yuen
Michael Geipel	Joseph Moran	Paolo Zangheri
Sergiu Goma	Michael S. Newman	
Randall Groves	Nick S. A. Nikjoo	

When the IEEE-SA Standards Board approved this standard on 26 September 2008, it had the following membership:

Robert M. Grow, *Chair*
Thomas Prevost, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Victor Berman	Jim Hughes	Chuck Powers
Richard DeBlasio	Richard H. Hulett	Narayanan Ramachandran
Andy Drozd	Young Kyun Kim	Jon Walter Rosdahl
Mark Epstein	Joseph L. Koepfinger*	Robby Robson
Alexander Gelman	John Kulick	Anne-Marie Sahazizia
William R. Goldbach	David J. Law	Malcolm V. Thaden
Arnold M. Greenspan	Glenn Parsons	Howard L. Wolfman
Kenneth S. Hanus	Ronald C. Petersen	Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Michael H. Kelly, *NIST Representative*

Michelle D. Turner
IEEE Standards Program Manager, Document Development

Michael D. Kipness
IEEE Standards Program Manager, Technical Program Development

List of special symbols

For the benefit of those who have received this document by electronic means, what follows is a list of special symbols and operators. If any of these symbols or operators fail to print out correctly on your machine, the editors apologize, and hope that this table will at least help you to sort out the meaning of the resulting funny-shaped blobs and strokes.

Special symbols and operators

Printed character	Meaning	Font
*	Boolean AND	Symbol
+	Boolean OR, arithmetic addition	Symbol
^	Boolean XOR	Times New Roman
!	Boolean NOT	Symbol
×	Multiplication	Symbol
<	Less than	Symbol
≤	Less than or equal to	Symbol
>	Greater than	Symbol
≥	Greater than or equal to	Symbol
=	Equal to	Symbol
≠	Not equal to	Symbol
←	Assignment operator	Symbol
∈	Indicates membership	Symbol
∉	Indicates nonmembership	Symbol
±	Plus or minus (a tolerance)	Symbol
°	Degrees	Symbol
Σ	Summation	Symbol
√	Square root	Symbol
—	Big dash (em dash)	Times New Roman
–	Little dash (en dash), subtraction	Times New Roman
	Vertical bar	Times New Roman
†	Dagger	Times New Roman
‡	Double dagger	Times New Roman
α	Lower case alpha	Symbol
β	Lower case beta	Symbol
γ	Lower case gamma	Symbol
δ	Lower case delta	Symbol
ε	Lower case epsilon	Symbol
λ	Lambda	Symbol
μ	Micro	Times New Roman
Ω	Omega	Symbol

Contents

1.	Overview.....	1
1.1	Scope.....	1
1.2	Purpose.....	1
2.	Normative references.....	2
3.	Definitions	3
4.	Acronyms and abbreviations	5
5.	Link Aggregation.....	7
5.1	Overview.....	7
5.1.1	State diagram conventions.....	7
5.1.1.1	Actions inside state blocks.....	8
5.1.1.2	State diagram variables.....	8
5.1.1.3	State transitions.....	8
5.1.1.4	Operators.....	9
5.1.2	Goals and objectives	9
5.1.3	Positioning of Link Aggregation within the IEEE 802.3 architecture.....	10
5.2	Link Aggregation operation.....	11
5.2.1	Principles of Link Aggregation.....	11
5.2.2	Service interfaces.....	13
5.2.3	Frame Collector	14
5.2.3.1	Frame Collector state diagram.....	14
5.2.3.1.1	Constants.....	14
5.2.3.1.2	Variables	14
5.2.3.1.3	Messages.....	14
5.2.3.1.4	State diagram	15
5.2.4	Frame Distributor.....	15
5.2.4.1	Frame Distributor state diagram	16
5.2.4.1.1	Variables	16
5.2.4.1.2	Messages.....	16
5.2.4.1.3	State diagram	16
5.2.5	Marker Generator/Receiver (optional).....	17
5.2.6	Marker Responder.....	17
5.2.7	Aggregator Parser/Multiplexer	17
5.2.7.1	Aggregator Parser state diagram.....	17
5.2.7.1.1	Constants.....	17
5.2.7.1.2	Variables	18
5.2.7.1.3	Messages.....	18
5.2.8	Aggregator	18
5.2.8.1	State diagram	20
5.2.9	Control Parser/Multiplexer	21
5.2.9.1	Control Parser state diagram.....	21
5.2.9.1.1	Constants.....	21
5.2.9.1.2	Variables	21
5.2.9.1.3	Messages.....	22
5.2.9.1.4	State diagram	22
5.2.10	Addressing	22

5.3	Link Aggregation Control.....	23
5.3.1	Characteristics of Link Aggregation Control.....	23
5.3.2	System identification	24
5.3.3	Aggregator identification.....	24
5.3.4	Port identification	25
5.3.5	Capability identification	25
5.3.6	Link Aggregation Group identification	26
5.3.6.1	Construction of the Link Aggregation Group Identifier.....	26
5.3.6.2	Representation of the Link Aggregation Group Identifier.....	27
5.3.7	Selecting a Link Aggregation Group	28
5.3.8	Agreeing on a Link Aggregation Group	28
5.3.9	Attaching a link to an Aggregator.....	28
5.3.10	Signaling readiness to transfer user data.....	29
5.3.11	Enabling Collection and Distribution	29
5.3.12	Monitoring the membership of a Link Aggregation Group.....	29
5.3.13	Detaching a link from an Aggregator	29
5.3.14	Configuration and administrative control of Link Aggregation	30
5.3.15	Link Aggregation Control state information	30
5.4	Link Aggregation Control Protocol (LACP)	30
5.4.1	LACP design elements.....	31
5.4.2	LACPDU structure and encoding	31
5.4.2.1	Transmission and representation of octets.....	31
5.4.2.2	LACPDU structure	32
5.4.3	LACP state machine overview	35
5.4.4	Constants.....	37
5.4.5	Variables associated with the System.....	37
5.4.6	Variables associated with each Aggregator	38
5.4.7	Variables associated with each port.....	39
5.4.8	Variables used for managing the operation of the state machines.....	42
5.4.9	Functions.....	43
5.4.10	Timers	46
5.4.11	Messages.....	46
5.4.12	Receive machine	46
5.4.13	Periodic Transmission machine	49
5.4.14	Selection Logic	50
5.4.14.1	Selection Logic—Requirements	50
5.4.14.2	Selection Logic—Recommended default operation	51
5.4.15	Mux machine	52
5.4.16	Transmit machine	56
5.4.17	Churn Detection machines.....	57
5.5	Marker protocol	58
5.5.1	Introduction.....	58
5.5.2	Sequence of operations	59
5.5.3	Marker and Marker Response PDU structure and encoding	59
5.5.3.1	Transmission and representation of octets.....	59
5.5.3.2	Marker and Marker Response PDU structure.....	59
5.5.4	Protocol definition	61
5.5.4.1	Operation of the marker protocol.....	61
5.5.4.2	Marker Responder state diagram	61
5.5.4.2.1	Constants.....	61
5.5.4.2.2	Variables	62
5.5.4.2.3	Messages.....	62
5.6	Configuration capabilities and restrictions	62
5.6.1	Use of system and port priorities	62

5.6.2	Dynamic allocation of operational Keys	63
5.6.3	Link Aggregation on shared-medium links	64
5.6.4	Selection Logic variants.....	64
5.6.4.1	Reduced reconfiguration.....	64
5.6.4.2	Limited Aggregator availability.....	64
5.7	Protocol implementation conformance statement (PICS) proforma for Clause 5, Aggregation of Multiple Link Segments.....	66
5.7.1	Introduction.....	66
5.7.2	Abbreviations and special symbols.....	66
5.7.3	Instructions for completing the PICS proforma.....	66
5.7.4	Additional information	67
5.7.5	Exceptional information	67
5.7.6	Conditional items.....	67
5.7.7	Identification.....	68
5.7.7.1	Implementation identification.....	68
5.7.7.2	Protocol summary.....	68
5.7.8	Major capabilities/options.....	69
5.7.9	Frame Collector	69
5.7.10	Frame Distributor.....	69
5.7.11	Marker protocol	70
5.7.12	Aggregator Parser/Multiplexer	70
5.7.13	Control Parser/Multiplexer	70
5.7.14	System identification	71
5.7.15	Aggregator identification.....	71
5.7.16	Port identification	71
5.7.17	Capability identification	71
5.7.18	Link Aggregation Group identification	72
5.7.19	Detaching a link from an Aggregator	72
5.7.20	LACPDU structure	72
5.7.21	State machine variables	72
5.7.22	Receive machine	73
5.7.23	Periodic Transmission machine	73
5.7.24	Selection Logic	73
5.7.25	Mux machine	74
5.7.26	Transmit machine	75
5.7.27	Churn Detection machines.....	75
5.7.28	Marker protocol	76
5.7.29	Configuration capabilities and restrictions	76
5.7.30	Link Aggregation on shared-medium links	77
6.	Management.....	79
6.1	Overview.....	79
6.1.1	Systems management overview	79
6.1.2	Management model.....	80
6.2	Managed objects	81
6.2.1	Introduction.....	81
6.2.2	Overview of managed objects.....	81
6.2.2.1	Text description of managed objects	81
6.2.3	Containment.....	82
6.2.4	Naming.....	82
6.2.5	Capabilities	82
6.3	Management for Link Aggregation	85
6.3.1	Aggregator managed object class	85

6.3.1.1	Aggregator attributes	86
6.3.1.1.1	aAggID.....	86
6.3.1.1.2	aAggDescription	86
6.3.1.1.3	aAggName	87
6.3.1.1.4	aAggActorSystemID	87
6.3.1.1.5	aAggActorSystemPriority	87
6.3.1.1.6	aAggAggregateOrIndividual	87
6.3.1.1.7	aAggActorAdminKey	87
6.3.1.1.8	aAggActorOperKey	88
6.3.1.1.9	aAggMACAddress.....	88
6.3.1.1.10	aAggPartnerSystemID	88
6.3.1.1.11	aAggPartnerSystemPriority	88
6.3.1.1.12	aAggPartnerOperKey	88
6.3.1.1.13	aAggAdminState.....	89
6.3.1.1.14	aAggOperState.....	89
6.3.1.1.15	aAggTimeOfLastOperChange	89
6.3.1.1.16	aAggDataRate	89
6.3.1.1.17	aAggOctetsTxOK	90
6.3.1.1.18	aAggOctetsRxOK	90
6.3.1.1.19	aAggFramesTxOK	90
6.3.1.1.20	aAggFramesRxOK.....	90
6.3.1.1.21	aAggMulticastFramesTxOK.....	91
6.3.1.1.22	aAggMulticastFramesRxOK	91
6.3.1.1.23	aAggBroadcastFramesTxOK.....	91
6.3.1.1.24	aAggBroadcastFramesRxOK.....	91
6.3.1.1.25	aAggFramesDiscardedOnTx	92
6.3.1.1.26	aAggFramesDiscardedOnRx	92
6.3.1.1.27	aAggFramesWithTxErrors.....	92
6.3.1.1.28	aAggFramesWithRxErrors	92
6.3.1.1.29	aAggUnknownProtocolFrames.....	93
6.3.1.1.30	aAggPortList.....	93
6.3.1.1.31	aAggLinkUpDownNotificationEnable	93
6.3.1.1.32	aAggCollectorMaxDelay	93
6.3.1.2	Aggregator Notifications	93
6.3.1.2.1	nAggLinkUpNotification	93
6.3.1.2.2	nAggLinkDownNotification	94
6.3.2	Aggregation Port managed object class.....	94
6.3.2.1	Aggregation Port Attributes.....	94
6.3.2.1.1	aAggPortID	94
6.3.2.1.2	aAggPortActorSystemPriority	94
6.3.2.1.3	aAggPortActorSystemID	94
6.3.2.1.4	aAggPortActorAdminKey	95
6.3.2.1.5	aAggPortActorOperKey	95
6.3.2.1.6	aAggPortPartnerAdminSystemPriority	95
6.3.2.1.7	aAggPortPartnerOperSystemPriority.....	95
6.3.2.1.8	aAggPortPartnerAdminSystemID	95
6.3.2.1.9	aAggPortPartnerOperSystemID.....	96
6.3.2.1.10	aAggPortPartnerAdminKey	96
6.3.2.1.11	aAggPortPartnerOperKey	96
6.3.2.1.12	aAggPortSelectedAggID	96
6.3.2.1.13	aAggPortAttachedAggID.....	97
6.3.2.1.14	aAggPortActorPort	97
6.3.2.1.15	aAggPortActorPortPriority	97
6.3.2.1.16	aAggPortPartnerAdminPort.....	97

	6.3.2.1.17	aAggPortPartnerOperPort	97
	6.3.2.1.18	aAggPortPartnerAdminPortPriority	98
	6.3.2.1.19	aAggPortPartnerOperPortPriority	98
	6.3.2.1.20	aAggPortActorAdminState	98
	6.3.2.1.21	aAggPortActorOperState	98
	6.3.2.1.22	aAggPortPartnerAdminState	98
	6.3.2.1.23	aAggPortPartnerOperState	99
	6.3.2.1.24	aAggPortAggregateOrIndividual	99
6.3.3		Aggregation Port Statistics managed object class	99
6.3.3.1		Aggregation Port Statistics attributes	99
	6.3.3.1.1	aAggPortStatsID	99
	6.3.3.1.2	aAggPortStatsLACPDUsRx	99
	6.3.3.1.3	aAggPortStatsMarkerPDUsRx	100
	6.3.3.1.4	aAggPortStatsMarkerResponsePDUsRx	100
	6.3.3.1.5	aAggPortStatsUnknownRx	100
	6.3.3.1.6	aAggPortStatsIllegalRx	100
	6.3.3.1.7	aAggPortStatsLACPDUsTx	100
	6.3.3.1.8	aAggPortStatsMarkerPDUsTx	101
	6.3.3.1.9	aAggPortStatsMarkerResponsePDUsTx	101
6.3.4		Aggregation Port Debug Information managed object class	101
6.3.4.1		Aggregation Port Debug Information attributes	101
	6.3.4.1.1	aAggPortDebugInformationID	101
	6.3.4.1.2	aAggPortDebugRxState	101
	6.3.4.1.3	aAggPortDebugLastRxTime	102
	6.3.4.1.4	aAggPortDebugMuxState	102
	6.3.4.1.5	aAggPortDebugMuxReason	102
	6.3.4.1.6	aAggPortDebugActorChurnState	102
	6.3.4.1.7	aAggPortDebugPartnerChurnState	103
	6.3.4.1.8	aAggPortDebugActorChurnCount	103
	6.3.4.1.9	aAggPortDebugPartnerChurnCount	103
	6.3.4.1.10	aAggPortDebugActorSyncTransitionCount	103
	6.3.4.1.11	aAggPortDebugPartnerSyncTransitionCount	104
	6.3.4.1.12	aAggPortDebugActorChangeCount	104
	6.3.4.1.13	aAggPortDebugPartnerChangeCount	104
		Annex A (informative) Collection and Distribution functions	105
A.1		Introduction	105
A.2		Port selection	106
A.3		Dynamic reallocation of conversations to different ports	106
A.4		Topology considerations in the choice of distribution algorithm	107
		Annex B (informative) LACP standby link selection and dynamic Key management	109
B.1		Introduction	109
B.2		Goals	109
B.3		Standby link selection	109
B.4		Dynamic Key management	110
B.5		A dynamic Key management algorithm	110
B.6		Example 1	112
B.7		Example 2	112
		Annex C (normative) SNMP MIB definitions for Link Aggregation	115

C.1	Introduction.....	115
C.2	The SNMP Management Framework	115
C.3	Security considerations	115
C.4	Structure of the MIB	116
C.4.1	Relationship to the managed objects defined in Clause 6	116
C.4.2	The Aggregator Group	119
C.4.3	The Aggregator Port List Group	119
C.4.4	The Aggregation Port Group	120
C.4.5	The Aggregation Port Statistics Group	120
C.4.6	The Aggregation Port Debug Group	120
C.5	Relationship to other MIBs	120
C.5.1	Relationship to the Interfaces MIB	120
C.5.2	Layering model	121
C.5.3	ifStackTable	121
C.5.4	ifRcvAddressTable	121
C.6	Definitions for Link Aggregation MIB	121

IEEE Standard for Local and metropolitan area networks— Link Aggregation

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

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

1. Overview

1.1 Scope

Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of data terminal equipment (DTE) to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. This standard defines the MAC independent Link Aggregation capability, and general information relevant to specific MAC types that support Link Aggregation.

1.2 Purpose

Link Aggregation allows the establishment of full duplex point-to-point links that have a higher aggregate bandwidth than the individual links that form the aggregation. This allows improved utilization of available links in bridged LAN environments, along with improved resilience in the face of failure of individual links

2. Normative references

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

IEEE Std 802[®]-2001, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.^{1, 2}

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

IEEE Std 802.3[™], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

IETF RFC 1155, *Structure and Identification of Management Information for TCP/IP-based Internets*, Rose, M., and K. McCloghrie, May 1990.³

IETF RFC 1157, *A Simple Network Management Protocol (SNMP)*, Case, J., Fedor, M., Schoffstall, M., and J. Davin, May 1990.

IETF RFC 1212, *Concise MIB Definitions*, Rose, M., and K. McCloghrie, March 1991.

IETF RFC 1213 (IETF STD 17), *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, McCloghrie K., and M. Rose, Editors, March 1991.

IETF RFC 1215, *A Convention for Defining Traps for use with the SNMP*, M. Rose, March 1991.

IETF RFC 1901, *Introduction to Community-based SNMPv2*, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, January 1996.

ISO/IEC 8824: 1990, Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1).⁴

ISO/IEC 10165-2: 1992, Information technology—Open Systems Interconnection—Structure of management information: Definition of management information.

ISO/IEC 10165-4: 1992, Information technology—Open Systems Interconnection—Management information services—Structure of management information—Part 4: Guidelines for the definition of managed objects.

ISO/IEC 15802-1: 1995, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.

¹IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ, 08854, USA (<http://standards.ieee.org/>).

²The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

³IETF RFCs are available from the Internet Engineering Task Force website at <http://www.ietf.org/rfc.html>.

⁴ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (<http://www.iso.ch/>). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).

3. Definitions

For the purposes of this standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standard Terms* (IEEE 100) should be referenced for terms not defined in this standard.

3.1 Actor: The local entity in a Link Aggregation Control Protocol exchange. (See IEEE Std 802.1AX, Clause 5.)

3.2 agent: A term used to refer to the managed nodes in a network. Managed nodes are those nodes that contain a network management entity (NME), which can be used to configure the node and/or collect data describing operation of that node. The agent is controlled by a network control host or manager that contains both an NME and network management application (NMA) software to control the operations of agents. Agents include systems that support user applications as well as nodes that provide communications services such as front-end processors, bridges, and routers. (See IEEE Std 802.3, Clause 30.⁵)

3.3 Aggregation Key: A parameter associated with each port and with each Aggregator of an Aggregation System identifying those ports that can be aggregated together. Ports in an Aggregation System that share the same Aggregation Key value are potentially able to aggregate together. (See IEEE 802.1AX, Clause 5.)

3.4 Aggregation Link: An instance of a MAC-Physical Layer-Medium Physical Layer-MAC entity between a pair of Aggregation Systems. (See IEEE Std 802.1AX, Clause 5.)

3.5 Aggregation Port: An instance of a MAC-Physical Layer entity within an Aggregation System. (See IEEE Std 802.1AX, Clause 5.)

3.6 Aggregation System: A uniquely identifiable entity comprising (among other things) an arbitrary grouping of one or more ports for the purpose of aggregation. An instance of an aggregated link always occurs between exactly two Aggregation Systems. A physical device may comprise a single Aggregation System or more than one Aggregation System. (See IEEE Std 802.1AX, Clause 5.)

3.7 bridge: A layer 2 interconnection device that does not form part of a CSMA/CD collision domain but conforms to IEEE Std 802.1D. A bridge does not form part of a CSMA/CD collision domain but, rather appears as a Media Access Control (MAC) to the collision domain.

3.8 conversation: A set of frames transmitted from one end station to another, where all of the frames form an ordered sequence, and where the communicating end stations require the ordering to be maintained among the set of frames exchanged. (See IEEE Std 802.1AX, Clause 5.)

3.9 data terminal equipment (DTE): Any source or destination of data connected to the local area network.

3.10 end station: A system attached to a LAN that is an initial source or a final destination of frames transmitted across that LAN. A Network layer router is, from the perspective of the LAN, an end station; a bridge, in its role of forwarding frames from one LAN to another, is not an end station. (See IEEE Std 802.1AX, Clause 5.)

3.11 frame: Consists of the Destination Address, Source Address, Length/Type field, MAC Client Data, Pad (if required), and Frame Check Sequence.

3.12 full duplex: A mode of operation of a network, DTE, or Medium Attachment Unit (MAU) that supports duplex transmission as defined in IEEE 100. Within the scope of this standard, this mode of

⁵Information on references can be found in Clause 2.

operation allows for simultaneous communication between a pair of stations, provided that the Physical Layer is capable of supporting simultaneous transmission and reception without interference. (See IEEE Std 802.3.)

3.13 half duplex: A mode of operation of a CSMA/CD local area network (LAN) in which DTEs contend for access to a shared medium. Multiple, simultaneous transmissions in a half duplex mode CSMA/CD LAN result in interference, requiring resolution by the CSMA/CD access control protocol. (See IEEE Std 802.3.)

3.14 Key: *See: Aggregation Key.*

3.15 link: *See: Aggregation Link.*

3.16 Link Aggregation Group: A group of links that appear to a MAC Client as if they were a single link. All links in a Link Aggregation Group connect between the same pair of Aggregation Systems. One or more conversations may be associated with each link that is part of a Link Aggregation Group. (See IEEE Std 802.1AX, Clause 5.)

3.17 Management Information Base (MIB): A repository of information to describe the operation of a specific network device.

3.18 Media Access Control (MAC): The data link sublayer that is responsible for transferring data to and from the Physical Layer.

3.19 Partner: The remote entity in a Link Aggregation Control Protocol exchange. (See IEEE Std 802.1AX, Clause 5.)

3.20 port: *See: Aggregation Port.*

3.21 switch: A layer 2 interconnection device that conforms to ISO/IEC 10038 [ANSI/IEEE 802.1D]. *Syn: bridge.*

3.22 System: *See: Aggregation System.*

3.23 Type/Length/Value (TLV): A short, variable length encoding of an information element consisting of sequential type, length, and value fields where the type field identifies the type of information, the length field indicates the length of the information field in octets, and the value field contains the information, itself. The type value is locally defined and needs to be unique within the protocol defined in this standard.

4. Acronyms and abbreviations

This standard contains the following abbreviations:

ANSI	American National Standards Institute
ASN.1	abstract syntax notation one as defined in ISO/IEC 8824: 1990
CMIP	common management information protocol as defined in ISO/IEC 9596-1: 1991
DA	destination address
DTE	data terminal equipment
FCS	frame check sequence
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
LACP	Link Aggregation Control Protocol
LACPDU	Link Aggregation Control Protocol Data Unit
LAG	Link Aggregation Group
LAG ID	Link Aggregation Group Identifier
LAN	local area network
LLC	logical link control
LLID	logical link identifier
MAC	medium access control
MAN	metropolitan area network
MIB	management information base
NTT	Need To Transmit
PDU	Protocol Data Unit
PHY	Physical Layer entity sublayer
PICS	protocol implementation conformance statement
SA	source address
TLV	Type/Length/Value
UCT	unconditional transition

5. Link Aggregation

5.1 Overview

This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, Link Aggregation specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate.

5.1.1 State diagram conventions

The following state diagram conventions are used throughout this standard.

The operation of a protocol can be described by subdividing the protocol into a number of interrelated functions. The operation of the functions can be described by state diagrams. Each diagram represents the domain of a function and consists of a group of connected, mutually exclusive states. Only one state of a function is active at any given time (see Figure 5–1).

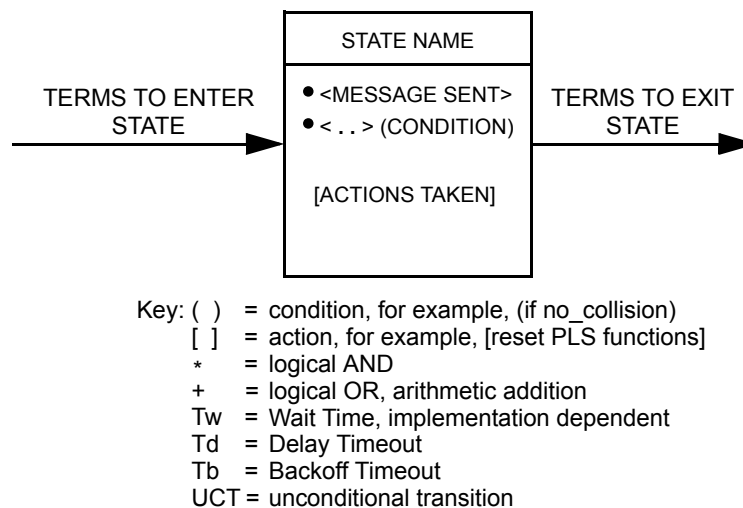


Figure 5–1—State diagram notation example

Each state that the function can assume is represented by a rectangle. These are divided into two parts by a horizontal line. In the upper part the state is identified by a name in capital letters. The lower part contains the name of any ON signal that is generated by the function. Actions are described by short phrases and enclosed in brackets.

All permissible transitions between the states of a function are represented graphically by arrows between them. A transition that is global in nature (for example, an exit condition from all states to the IDLE or RESET state) is indicated by an open arrow. Labels on transitions are qualifiers that must be fulfilled before the transition will be taken. The label UCT designates an unconditional transition. Qualifiers described by short phrases are enclosed in parentheses.

State transitions and sending and receiving of messages occur instantaneously. When a state is entered and the condition to leave that state is not immediately fulfilled, the state executes continuously, sending the messages in the state in a continuous manner.

The following conventions are used to describe a term-assignment statement that is associated with a transition:

- a) The character “:” (colon) is a delimiter used to denote that a term assignment statement follows.
- b) The character “←” (left arrow) denotes assignment of the value following the arrow to the term preceding the arrow.

The state diagrams contain the authoritative statement of the functions they depict; when apparent conflicts between descriptive text and state diagrams arise, the state diagrams are to take precedence. This does not override, however, any explicit description in the text that has no parallel in the state diagrams.

The models presented by state diagrams are intended as the primary specifications of the functions to be provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology. It is the functional behavior of any unit that must match the standard, not its internal structure. The internal details of the model are useful only to the extent that they specify the external behavior clearly and precisely.

5.1.1.1 Actions inside state blocks

The actions inside a state block execute instantaneously. Actions inside state blocks are atomic (i.e., uninterruptible).

After performing all the actions listed in a state block one time, the state block then continuously evaluates its exit conditions until one is satisfied, at which point control passes through a transition arrow to the next block. While the state awaits fulfillment of one of its exit conditions, the actions inside do not implicitly repeat.

The characters • and [bracket] are *not* used to denote any special meaning.

Valid state actions may include .indication and .request messages.

No actions are taken outside of any state block.

5.1.1.2 State diagram variables

Once set, variables retain their values as long as succeeding blocks contain no references to them.

Setting the parameter of a formal interface message assures that, on the next transmission of that message, the last parameter value set will be transmitted.

Testing the parameter of a formal interface messages tests the value of that message parameter that was received on the last transmission of said message. Message parameters may be assigned default values that persist until the first reception of the relevant message.

5.1.1.3 State transitions

The following terms are valid transition qualifiers:

- a) Boolean expressions
- b) An event such as the expiration of a timer: timer_done
- c) An event such as the reception of a message: PMA_UNITDATA.indication
- d) An unconditional transition: UCT
- e) A branch taken when other exit conditions are not satisfied: ELSE

Any open arrow (an arrow with no source block) represents a global transition. Global transitions are evaluated continuously whenever any state is evaluating its exit conditions. When a global transition becomes true, it supersedes all other transitions, including UCT, returning control to the block pointed to by the open arrow.

5.1.1.4 Operators

The state machine operators are shown in Table 5–1.

Table 5–1—State machine operators

Character	Meaning
*	Boolean AND
+	Boolean OR
$\wedge$	Boolean XOR
!	Boolean NOT
<	Less than
$\leq$	Less than or equal to
=	Equals (a test of equality)
$\neq$	Not equals
$\geq$	Greater than or equal to
>	Greater than
()	Indicates precedence
$\leftarrow$	Assignment operator
$\in$	Indicates membership
$\notin$	Indicates nonmembership
	Catenate
ELSE	No other state condition is satisfied

5.1.2 Goals and objectives

Link Aggregation, as specified in this clause, provides the following:

- Increased bandwidth**—The capacity of multiple links is combined into one logical link.
- Linearly incremental bandwidth**—Bandwidth can be increased in unit multiples as opposed to the order-of-magnitude increase available through Physical Layer technology options (10 Mb/s, 100 Mb/s, 1000 Mb/s, etc.).
- Increased availability**—The failure or replacement of a single link within a Link Aggregation Group need not cause failure from the perspective of a MAC Client.
- Load sharing**—MAC Client traffic may be distributed across multiple links.
- Automatic configuration**—In the absence of manual overrides, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they will aggregate.
- Rapid configuration and reconfiguration**—In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration, typically on the order of 1 second or less.
- Deterministic behavior**—Depending on the selection algorithm chosen, the configuration can be made to resolve deterministically; i.e., the resulting aggregation can be made independent of the order in which events occur, and be completely determined by the capabilities of the individual links and their physical connectivity.

- h) **Low risk of duplication or misordering**—During both steady-state operation and link (re)configuration, there is a low probability that frames are duplicated or misordered.
- i) **Support of existing IEEE 802.3 MAC Clients**—No change is required to existing higher-layer protocols or applications to use Link Aggregation.
- j) **Backwards compatibility with aggregation-unaware devices**—Links that cannot take part in Link Aggregation—either because of their inherent capabilities, management configuration, or the capabilities of the devices to which they attach—operate as normal, individual IEEE 802.3 links.
- k) **Accommodation of differing capabilities and constraints**—Devices with differing hardware and software constraints on Link Aggregation are, to the extent possible, accommodated.
- l) **No change to the IEEE 802.3 frame format**—Link aggregation neither adds to, nor changes the contents of frames exchanged between MAC Clients.
- m) **Network management support**—The standard specifies appropriate management objects for configuration, monitoring, and control of Link Aggregation.

Link Aggregation, as specified in this clause, does not support the following:

- n) **Multipoint Aggregations**—The mechanisms specified in this clause do not support aggregations among more than two Systems.
- o) **Dissimilar MACs**—Link Aggregation is supported only on links using the IEEE 802.3 MAC.
- p) **Half-duplex operation**—Link Aggregation is supported only on point-to-point links with MACs operating in full duplex mode.
- q) **Operation across multiple data rates**—All links in a Link Aggregation Group operate at the same data rate (e.g., 10 Mb/s, 100 Mb/s, or 1000 Mb/s).

5.1.3 Positioning of Link Aggregation within the IEEE 802.3 architecture

Link Aggregation comprises an optional sublayer between a MAC Client and the MAC (or optional MAC Control sublayer). Figure 5–2 depicts the positioning of the Link Aggregation sublayer in the CSMA/CD layer architecture, and the relationship of that architecture to the Data Link and Physical Layers of the OSI Reference Model. The figure also shows the ability of the Link Aggregation sublayer to combine a number of individual links in order to present a single MAC interface to the MAC Client.

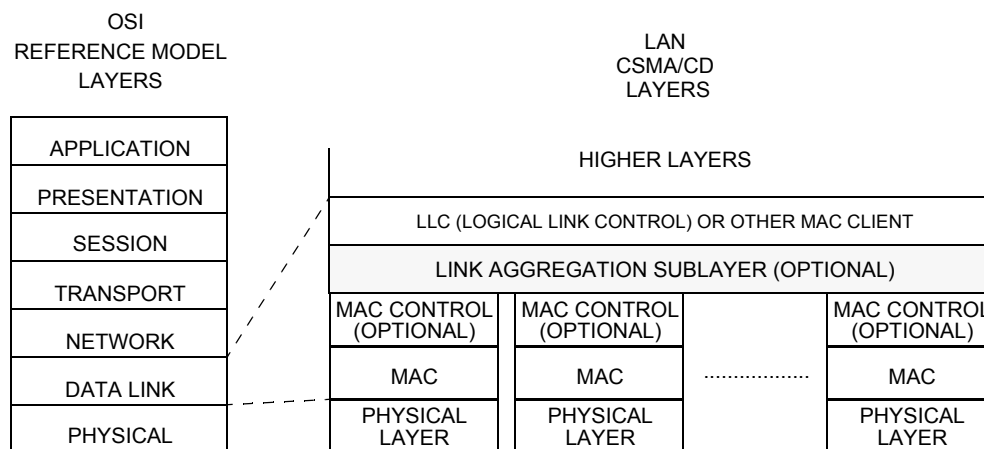


Figure 5–2—Architectural positioning of Link Aggregation sublayer

Figure 5–3 depicts the major blocks that form the Link Aggregation sublayer, and their interrelationships.

It is possible to implement the optional Link Aggregation sublayer for some ports within a System while not implementing it for other ports; i.e., it is not necessary for all ports in a System to be subject to Link Aggregation. A conformant implementation is not required to be able to apply the Link Aggregation sublayer to every port.

5.2 Link Aggregation operation

As depicted in Figure 5–3, the Link Aggregation sublayer comprises the following functions:

- a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 5.2.4, 5.2.5, and 5.5.)
- b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 5.2.3 and 5.5.)
- c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively).
- d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator.
- e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation.
- f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator.

5.2.1 Principles of Link Aggregation

Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts:

- a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3 service interface to the MAC Client. The Aggregator binds to one or more ports within a System.
- b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently.
- c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time.
- d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed.

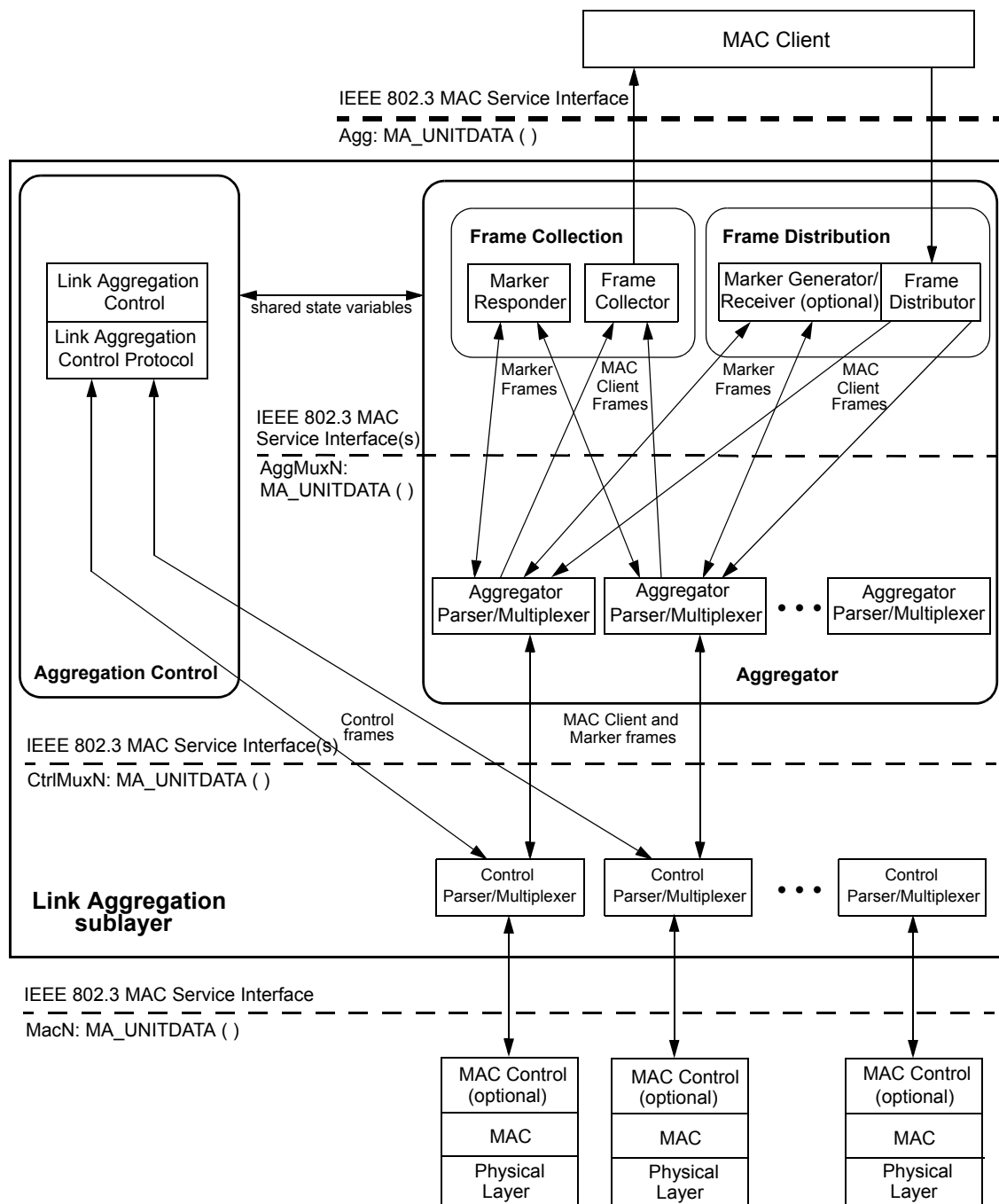


Figure 5-3—Link Aggregation sublayer block diagram

- e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

- f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see Clause 3). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation.
- g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures.
- h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported.
- i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges).
NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames.⁶
- j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 5.2.10).

5.2.2 Service interfaces

The MAC Client communicates with the Aggregator using the standard service interface specified in IEEE Std 802.3 Clause 2. Similarly, Link Aggregation communicates internally (between Frame Collection/Distribution, the Aggregator Parser/Multiplexers, the Control Parser/Multiplexers, and Link Aggregation Control) and with its bound ports using the same, standard service interface. No new interlayer service interfaces are defined for Link Aggregation.

Since Link Aggregation uses four instances of the MAC Service Interface, it is necessary to introduce a notation convention so that the reader can be clear as to which interface is being referred to at any given time. A prefix is therefore assigned to each service primitive, indicating which of the four interfaces is being invoked, as depicted in Figure 5–3. The prefixes are as follows:

- a) *Agg:*, for primitives issued on the interface between the MAC Client and the Link Aggregation sublayer.
- b) *AggMuxN:*, for primitives issued on the interface between Aggregator Parser/Multiplexer N and its internal clients (where N is the port number associated with the Aggregator Parser/Multiplexer).
- c) *CtrlMuxN:*, for primitives issued on the interface between Control Parser/Multiplexer N and its internal clients (where N is the port number associated with the Control Parser/Multiplexer).
- d) *MacN:*, for primitives issued on the interface between underlying MAC N and its Control Parser/Multiplexer (where N is the port number associated with the underlying MAC).

MAC Clients may generate *Agg:MA_DATA.request* primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. *MacN:MA_DATA.indication* primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector.

⁶Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

MAC Clients that generate MA_CONTROL.request primitives (and which expect MA_CONTROL.indication primitives in response) cannot communicate through a Link Aggregation sublayer. They must communicate directly with the MAC Control entity through which these control primitives are to be sent and received. The multiplexing of such MAC Control clients with a Link Aggregation sublayer for simultaneous use of a single port is outside the scope of this standard.

The multiplexing of MAC Clients with a Link Aggregation sublayer for simultaneous use of an individual MAC that is also part of a Link Aggregation Group is outside the scope of this standard.

5.2.3 Frame Collector

A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link's associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links.

The Frame Collector shall implement the function specified in the state diagram shown in Figure 5-4 and the associated definitions contained in 5.2.3.1.

5.2.3.1 Frame Collector state diagram

5.2.3.1.1 Constants

CollectorMaxDelay

In tens of microseconds, the maximum time that the Frame Collector may delay the delivery of a frame received from an Aggregator Parser to its MAC Client. Value is assigned by management or administration policy.

Value: Integer

5.2.3.1.2 Variables

DA

SA

mac_service_data_unit

status

The parameters of the MA_DATA.indication primitive, as defined in IEEE Std 802.3 Clause 2.

BEGIN

A Boolean variable that is set to TRUE when the System is initialized or reinitialized, and is set to FALSE when (re-)initialization has completed.

Value: Boolean

5.2.3.1.3 Messages

Agg:MA_DATA.indication

AggMuxN:MA_DATA.indication

The service primitives used to pass a received frame to a client with the specified parameters.

5.2.3.1.4 State diagram

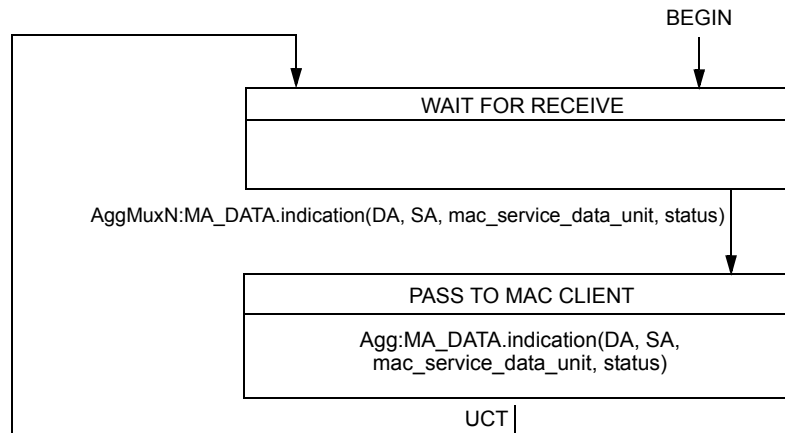


Figure 5-4—Frame Collector state diagram

The architectural models of the IEEE 802.3 MAC and Link Aggregation do not make any provision for queueing of frames between the link and the MAC Client. Furthermore, the state machine conventions used in this standard assume that actions within a state occur instantaneously (see 5.1.1). However, practical implementations of Link Aggregation will typically incur both queueing and delay in the Frame Collector. In order to ensure that frame delivery is not delayed indefinitely (which could cause a frame ordering problem when moving conversations from one link to another), the Frame Collector shall, upon receiving a frame from an Aggregator Parser, either deliver the frame to its MAC Client, or discard the frame within a CollectorMaxDelay time. The Frame Distributor (within the Partner System at the other end of the link) can assume that all frames transmitted on a given link have been either received by its Partner's MAC Client or discarded after a CollectorMaxDelay plus the propagation delay of the link. The use of CollectorMaxDelay is further discussed in A.3.

5.2.4 Frame Distributor

The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames.

This standard does not mandate any particular distribution algorithm(s); however, any distribution algorithm shall ensure that, when frames are received by a Frame Collector as specified in 5.2.3, the algorithm shall not cause

- a) Misordering of frames that are part of any given conversation, or
- b) Duplication of frames.

The above requirement to maintain frame ordering is met by ensuring that all frames that compose a given conversation are transmitted on a single link in the order that they are generated by the MAC Client; hence, this requirement does not involve the addition (or modification) of any information to the MAC frame, nor any buffering or processing on the part of the corresponding Frame Collector in order to reorder frames. This approach to the operation of the distribution function permits a wide variety of distribution and load balancing algorithms to be used, while also ensuring interoperability between devices that adopt differing algorithms.

NOTE—The subject of distribution algorithms and maintenance of frame ordering is discussed in Annex A.

The Frame Distributor shall implement the function specified in the state diagram shown in Figure 5–5 and the associated definitions contained in 5.2.4.1.

5.2.4.1 Frame Distributor state diagram

5.2.4.1.1 Variables

DA

SA

mac_service_data_unit

The parameters of the MA_DATA.request primitive, as defined in IEEE Std 802.3 Clause 2.

BEGIN

A Boolean variable that is set to TRUE when the System is initialized or reinitialized, and is set to FALSE when (re-)initialization has completed.

Value: Boolean

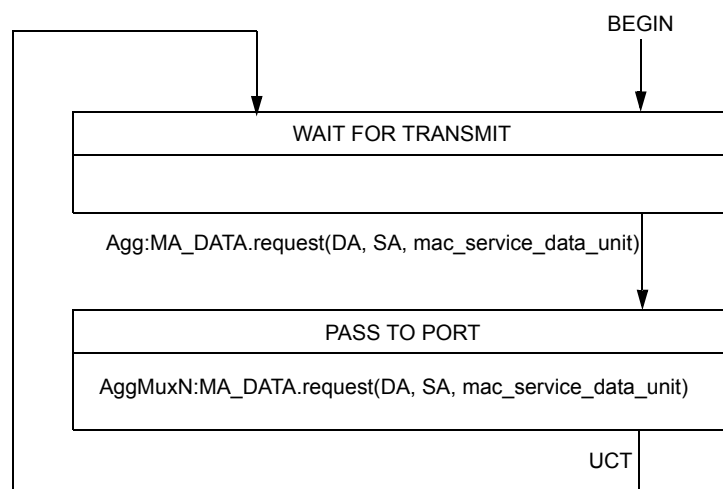
5.2.4.1.2 Messages

Agg:MA_DATA.request

AggMuxN:MA_DATA.request

The service primitives used to transmit a frame with the specified parameters.

5.2.4.1.3 State diagram



If a client issues an Agg:MA_DATA.request primitive that contains no SA parameter, the AggMuxN:MA_DATA.request primitive generated shall use the Aggregator's MAC address for the SA.

NOTE—The algorithm that the Frame Distributor uses to select the value of N in AggMuxN:MA_DATA.request for a given frame is unspecified.

Figure 5–5—Frame Distributor state diagram

5.2.5 Marker Generator/Receiver (optional)

The optional Marker Generator is used by the Marker protocol, as specified in 5.5. When implemented and so requested by the Distribution algorithm, the Marker Generator shall issue an AggMuxN:MA_DATA.request primitive, with a mac_service_data_unit containing a Marker PDU as defined in 5.5.3, to the port associated with the conversation being marked, subject to the timing restrictions for Slow Protocols specified in IEEE Std 802.3 Annex 57A.

The optional Marker Receiver is used by the Marker protocol, as specified in 5.5. It receives Marker Response PDUs from the Aggregator Parser.

5.2.6 Marker Responder

The Marker Responder is used by the Marker protocol, as specified in 5.5. The Marker Responder receives Marker PDUs (generated by a Partner System's Marker Generator), and transmits a Marker Response PDU through the same port from which the Marker PDU was received. While implementation of the Marker Generator/Receiver is optional, the ability to respond to a Marker PDU (the Marker Responder) is mandatory. An implementation conformant to this clause shall implement the Marker Responder as specified in 5.5.4.2, thus ensuring that implementations that need to make use of the protocol can do so.

5.2.7 Aggregator Parser/Multiplexer

On transmission, the Aggregator Multiplexer shall provide transparent pass-through of frames submitted by the Marker Responder and optional Marker Generator to the port specified in the transmission request. The Aggregator Multiplexer shall provide transparent pass-through of frames submitted by the Frame Distributor to the port specified in the transmission request only when the port state is Distributing (see 5.4.15); otherwise, such frames shall be discarded.

On receipt, the Aggregator Parser decodes frames received from the Control Parser, passes those frames destined for the Marker Responder or Marker Receiver to the selected entity, and discards frames with invalid Slow Protocol subtype values (see IEEE Std 802.3 Table 57A–2). The Aggregator Parser shall pass all other frames to the Frame Collector for passage to the MAC Client only when the port state is Collecting (see 5.4.15); otherwise, such frames shall be discarded. The Aggregator Parser shall implement the function specified in the state diagram shown in Figure 5–6 and the associated definitions contained in 5.2.7.1.

5.2.7.1 Aggregator Parser state diagram

5.2.7.1.1 Constants

Slow_Protocols_Multicast

The value of the Slow Protocols Multicast address. (See IEEE Std 802.3 Table 57A–1.)

Slow_Protocols_Type

The value of the Slow Protocols Length/Type field. (See IEEE Std 802.3 Annex 57A.)

Marker_subtype

The value of the Subtype field for the Marker protocol. (See 5.5.3.)

Value: Integer

2

Marker_Information

The encoding of the Marker Information TLV_type field. (See 5.5.3.)

Value: Integer

1

Marker_Response_Information

The encoding of the Marker Response Information TLV_type field. (See 5.5.3.)

Value: Integer

2

5.2.7.1.2 Variables

DA

SA

mac_service_data_unit

status

The parameters of the MA_DATA.indication primitive as defined in IEEE Std 802.3 Clause 2.

Length/Type

The value of the Length/Type field in a received frame.

Value: Integer

Subtype

The value of the octet following the Length/Type field in a Slow Protocol frame.
(See IEEE Std 802.3 Annex 57A.)

Value: Integer

TLV_type

The value contained in the octet following the Version Number in a received Marker or Marker Response frame. This identifies the “type” for the Type/Length/Value (TLV) tuple. (See 5.5.3.)

Value: Integer

BEGIN

A Boolean variable that is set to TRUE when the System is initialized or reinitialized, and is set to FALSE when (re-)initialization has completed.

Value: Boolean

5.2.7.1.3 Messages

CtrlMuxN:MA_DATA.indication

AggMuxN:MA_DATA.indication

The service primitives used to pass a received frame to a client with the specified parameters.

5.2.8 Aggregator

An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3 MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a

logical MAC, bound to one or more ports, through which the MAC client is provided access to the MAC service.

A single, individual MAC address is associated with each Aggregator (see 5.2.10).

An Aggregator is available for use by the MAC Client if the following are all true:

- a) It has one or more attached ports.
- b) The Aggregator has not been set to a disabled state by administrative action (see 6.3.1.1.13).
- c) The collection and/or distribution function associated with one or more of the attached ports is enabled (see 6.3.1.1.14).

NOTE—To simplify the modeling and description of the operation of Link Aggregation, it is assumed that there are as many Aggregators as there are ports in a given System; however, this is not a requirement of this standard. Aggregation of two or more ports consists of changing the bindings between ports and Aggregators such that more than one port is bound to a single Aggregator. The creation of any aggregations of two or more links will therefore result in one or more Aggregators that are bound to more than one port, and one or more Aggregators that are not bound to any port. An Aggregator that is not bound to any port appears to a MAC Client as a MAC interface to an inactive port. During times when the bindings between ports and Aggregators are changing, or as a consequence of particular configuration choices, there may be occasions when one or more ports are not bound to any Aggregator.

5.2.8.1 State diagram

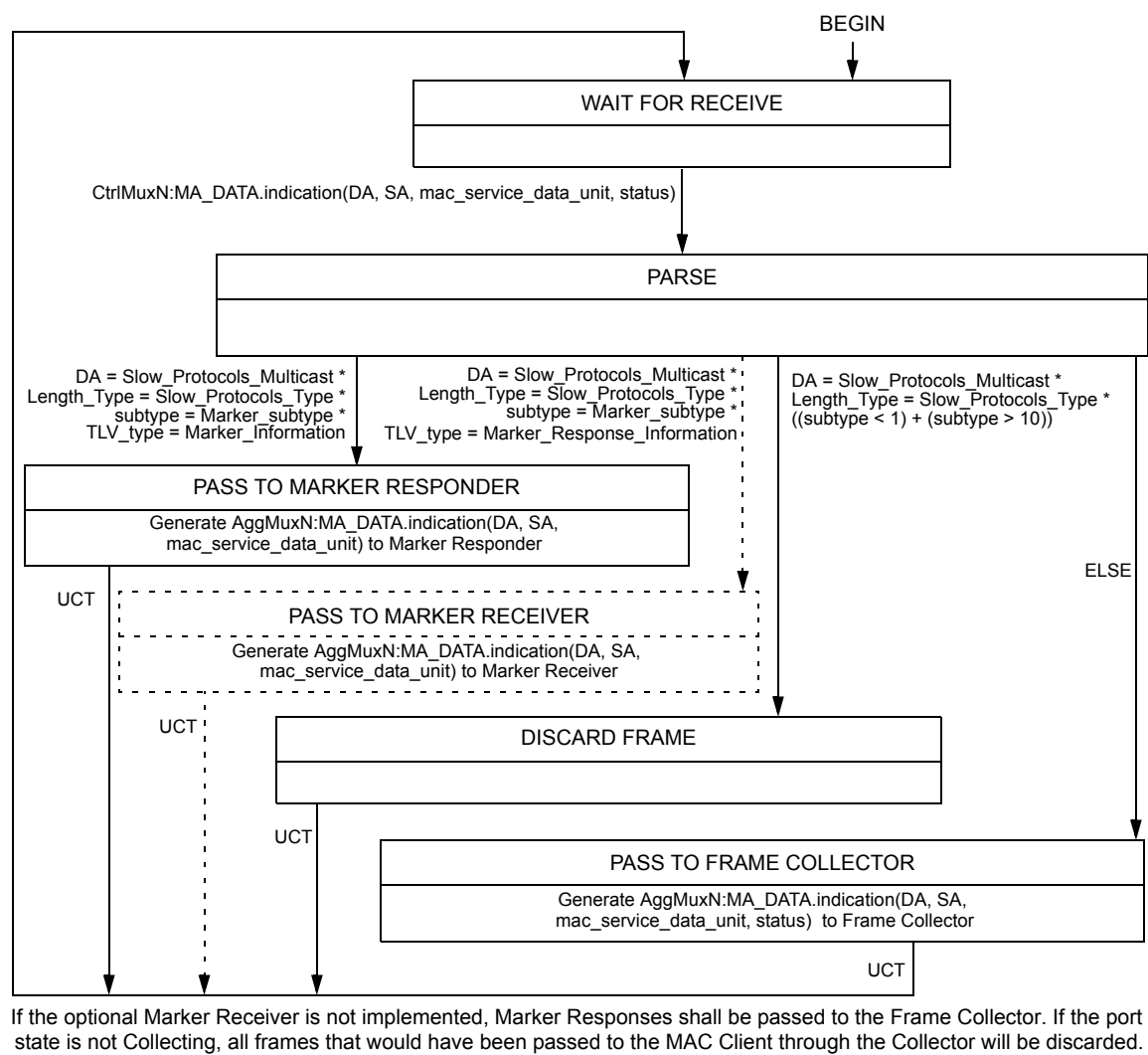


Figure 5–6—Aggregator Parser state diagram

5.2.9 Control Parser/Multiplexer

On transmission, the Control Multiplexer shall provide transparent pass-through of frames submitted by the Aggregator and Link Aggregation Control Protocol to the port specified in the transmission request.

On receipt, the Control Parser decodes frames received from the various ports in the Link Aggregation Group, passes those frames destined for the Link Aggregation Control Protocol to the appropriate entity, and passes all other frames to the Aggregator Parser. The Control Parser shall implement the function specified by the state diagram shown in Figure 5–7 and the associated definitions contained in 5.2.9.1.

5.2.9.1 Control Parser state diagram

5.2.9.1.1 Constants

Slow_Protocols_Multicast

The value of the Slow Protocols Multicast address. (See IEEE Std 802.3 Table 57A–1.)

Slow_Protocols_Type

The value of the Slow Protocols Length/Type field. (See IEEE Std 802.3 Table 57A–2.)

LACP_subtype

The value of the Subtype field for the Link Aggregation Control Protocol. (See IEEE Std 802.3 Table 57A–3.)

Value: Integer

1

5.2.9.1.2 Variables

DA

SA

mac_service_data_unit

status

The parameters of the MA_DATA.indication primitive, as defined in IEEE Std 802.3 Clause 2.

Length/Type

The value of the Length/Type field in a received frame.

Value: Integer

Subtype

The value of the octet following the Length/Type field in a Slow Protocol frame.
(See IEEE Std 802.3 Annex 57A.)

Value: Integer

BEGIN

A Boolean variable that is set to TRUE when the System is initialized or reinitialized, and is set to FALSE when (re-)initialization has completed.

Value: Boolean

5.2.9.1.3 Messages

MacN:MA_DATA.indication

CtrlMuxN:MA_DATA.indication

The service primitives used to pass a received frame to a client with the specified parameters.

5.2.9.1.4 State diagram

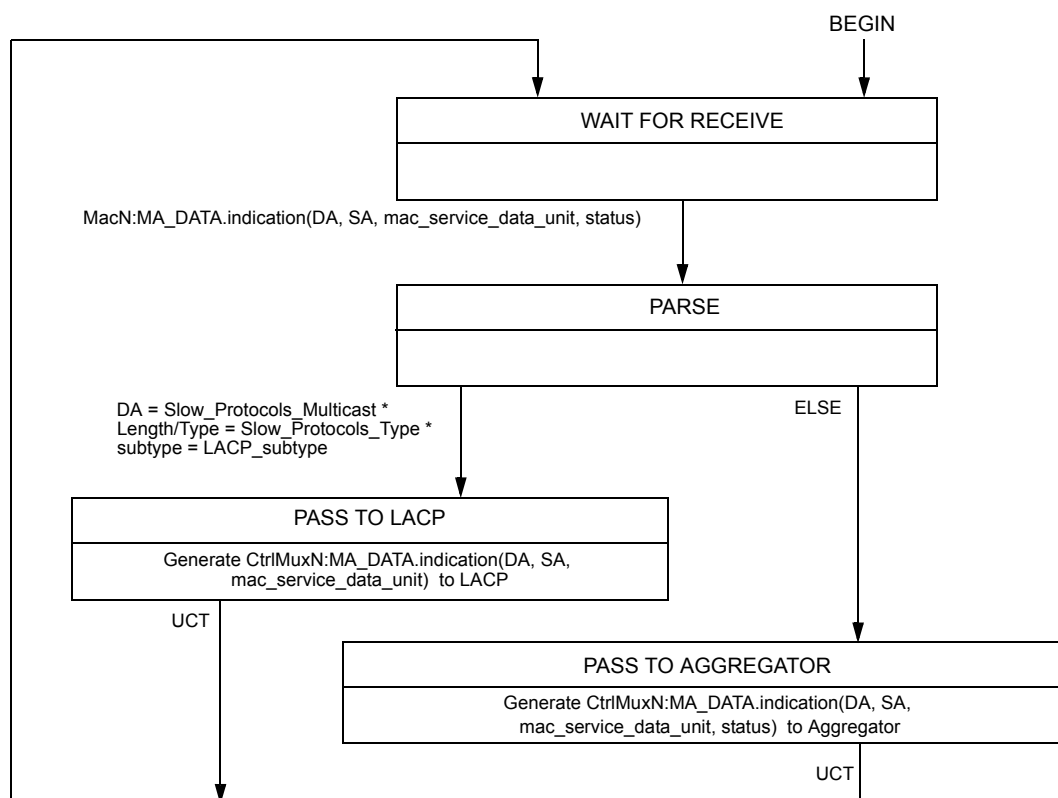


Figure 5–7—Control Parser state diagram

5.2.10 Addressing

Each IEEE 802.3 MAC has an associated individual MAC address, whether that MAC is used for Link Aggregation or not (see IEEE Std 802.3 Clause 4).

Each Aggregator to which one or more ports are attached has an associated globally-unique individual MAC address (see 5.3.3). The MAC address of the Aggregator may be the globally-unique individual MAC addresses of one of the MACs in the associated Link Aggregation Group, or it may be a distinct MAC address. The manner in which such addresses are chosen is not otherwise constrained by this standard.

Protocol entities sourcing frames from within the Link Aggregation sublayer (e.g., LACP and the Marker protocol) use the MAC address of the MAC within an underlying port as the source address in frames transmitted through that port. The MAC Client sees only the Aggregator and not the underlying MACs, and therefore uses the Aggregator's MAC address as the source address in transmitted frames. If a MAC Client

submits a frame to the Aggregator for transmission without specifying a source address, the Aggregator inserts its own MAC address as the source address for transmitted frames.

NOTE—This behavior causes the Aggregator to behave the same way as a standard MAC with regard to frames submitted by its client.

5.3 Link Aggregation Control

Link Aggregation Control configures and controls the Link Aggregation sublayer using static information local to the control function and dynamic information exchanged by means of the Link Aggregation Control Protocol.

For each aggregateable port in the System, Link Aggregation Control

- a) Maintains configuration information (reflecting the inherent properties of the individual links as well as those established by management) to control aggregation.
- b) Exchanges configuration information with other Systems to allocate the link to a Link Aggregation Group.

NOTE—A given link is allocated to, at most, one Link Aggregation Group at a time. The allocation mechanism attempts to maximize aggregation, subject to management controls.
- c) Attaches the port to the Aggregator used by the Link Aggregation Group, and detaches the port from the Aggregator when it is no longer used by the Group.
- d) Uses information from the Partner System's Link Aggregation Control entity to enable or disable the Aggregator's Collector and Distributor.

The operation of Link Aggregation Control involves the following activities, which are described in detail in 5.3.1 through 5.3.15:

- e) Checking that candidate links can actually be aggregated.
- f) Controlling the addition of a link to a Link Aggregation Group, and the creation of the group if necessary.
- g) Monitoring the status of aggregated links to ensure that the aggregation is still valid.
- h) Removing a link from a Link Aggregation Group if its membership is no longer valid, and removing the group if it no longer has any member links.

In order to allow Link Aggregation Control to determine whether a set of links connect to the same System, and to determine whether those links are compatible from the point of view of aggregation, it is necessary to be able to establish

- i) A globally unique identifier for each System that participates in Link Aggregation (see 5.3.2).
- j) A means of identifying the set of capabilities associated with each port and with each Aggregator, as understood by a given System.
- k) A means of identifying a Link Aggregation Group and its associated Aggregator.

System identification allows the detection of links that are connected in a loopback configuration (i.e., both ends of the same link are connected to the same System).

5.3.1 Characteristics of Link Aggregation Control

Link Aggregation Control provides a configuration capability that is

- a) **Automatic.** In the absence of manual override controls, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they do aggregate.

- b) **Continuous.** Manual intervention or initialization events are not a requirement for correct operation. The configuration mechanism continuously monitors for changes in state that require reconfiguration. The configuration functions detect and correct misconfigurations by performing reconfiguration and/or by taking misconfigured links out of service.
- c) **Deterministic.** The configuration can resolve deterministically; i.e., the configuration achieved can be made independent of the order in which events occur, and be completely determined by the combination of the capabilities of the individual links and their physical connectivity.
- d) **Controllable.** The configuration capabilities accommodate devices with differing hardware and software constraints on Link Aggregation.
- e) **Compatible.** Links that cannot take part in Link Aggregation, either because of their inherent capabilities or of the capabilities of the devices to which they attach, operate as normal IEEE 802.3 links. The introduction of Link Aggregation capability at one or both ends of a link should not result in a degradation of the perceived performance of the link.
- f) **Rapid.** The configuration resolves rapidly to a stable configuration. Convergence can be achieved by the exchange of three LACPDUs, without dependence on timer values.

with

- g) **Low risk of misdelivery.** The operation of the (re-)configuration functions minimizes the risk of frames being delivered to the wrong Aggregator.
- h) **Low risk of duplication or misordering.** The operation of the (re-)configuration functions minimizes the risk of frame duplication and frame misordering.
- i) **Low protocol overhead.** The overhead involved in external communication of configuration information between devices is small.

5.3.2 System identification

The globally unique identifier used to identify a System shall be the concatenation of a globally administered individual MAC address and the System Priority. The MAC address chosen may be the individual MAC address associated with one of the ports of the System.

Where it is necessary to perform numerical comparisons between System Identifiers, each System Identifier is considered to be an eight octet unsigned binary number, constructed as follows:

- a) The two most significant octets of the System Identifier comprise the System Priority. The System Priority value is taken to be an unsigned binary number; the most significant octet of the System Priority forms the most significant octet of the System Identifier.
- b) The third most significant octet of the System Identifier is derived from the initial octet of the MAC address; the least significant bit of the octet is assigned the value of the first bit of the MAC address, the next most significant bit of the octet is assigned the value of the next bit of the MAC address, and so on. The fourth through eighth octets are similarly assigned the second through sixth octets of the MAC address.

5.3.3 Aggregator identification

Each Aggregator to which one or more ports are attached shall be assigned a unique, globally administered individual MAC address. The MAC address assigned to the Aggregator may be the same as the MAC address assigned to one of its bound ports. No Aggregator shall be assigned a MAC address that is the same as that of a port bound to a different Aggregator within the System. When receiving frames, a port is never required to recognize more than one unicast address, i.e., the Aggregator's MAC address.

NOTE—This ensures that Aggregators can be uniquely addressed, and allows (but does not require) the unique address to be allocated from the same set of addresses as are assigned to the ports. It also acknowledges the fact that locally administered addresses may be used in particular implementations or environments. The stated restriction on the allocation of MAC addresses to Aggregators may have implications with regard to the choice of selection algorithm.

An Aggregator also shall be assigned an integer identifier that is used by Link Aggregation Control to uniquely identify the Aggregator within the System. This value will typically be the same as the interface identifier (ifIndex) used for management purposes.

5.3.4 Port identification

Link Aggregation Control uses a Port Identifier, comprising the concatenation of a Port Priority and a Port Number, to identify the port. Port Numbers (and hence, Port Identifiers) shall be uniquely assigned within a System. Port Number 0 shall not be assigned to any port.

When it is necessary to perform numerical comparisons between Port Identifiers, each Port Identifier is considered to be a four octet unsigned binary number constructed as follows:

- a) The most significant and second most significant octets are the first and second most significant octets of the Port Priority, respectively.
- b) The third and fourth most significant octets are the first and second most significant octets of the Port Number, respectively.

5.3.5 Capability identification

The ability of one port to aggregate with another is summarized by a simple integer parameter, known as a Key. This facilitates communication and comparison of aggregation capabilities, which may be determined by a number of factors, including

- a) The port's physical characteristics, such as data rate, duplexity, and point-to-point or shared medium.
- b) Configuration constraints established by the network administrator.
- c) Use of the port by higher layer protocols (e.g., assignment of Network Layer addresses).
- d) Characteristics or limitations of the port implementation itself.

Two Keys shall be associated with each port—an operational Key and an administrative Key. The operational Key is the Key that is currently in active use for the purposes of forming aggregations. The administrative Key allows manipulation of Key values by management. The administrative and operational Keys assigned to a port may differ

- e) If the operation of the implementation is such that an administrative change to a Key value cannot be immediately reflected in the operational state of the port.
- f) If the System supports the dynamic manipulation of Keys, as discussed in 5.6.2, either to accurately reflect changes in operational capabilities of the port (for example, as a result of Auto-Negotiation), or to provide a means of handling constraints on aggregation capability.

A given Key value is meaningful only in the context of the System that allocates it; there is no global significance to Key values. Similarly, the relationship between administrative and operational Key values is meaningful only in the context of the System that allocates it. When a System assigns an administrative Key value to a set of ports, it signifies that the set of ports have the potential to aggregate together, subject to the considerations discussed in 5.6.2. When a System assigns an operational Key value to a set of ports, it signifies that, in the absence of other constraints, the current operational state of the set of ports allows any subset of that set of ports (including the entire set) to be aggregated together from the perspective of the System making the assignment. The set of such ports that will actually be aggregated will be those that terminate at a common Partner System, and for which that Partner System has assigned a common operational Key value, local to that Partner. The set of ports in a given System that share the same operational Key value are said to be members of the same Key Group.

A System may determine that a given link is not able to be aggregated with other links. Such links are referred to as Individual links (as opposed to Aggregateable links). A System may declare a link to be

Individual if the inherent properties of the link allow its use as part of an aggregation, but the system is aware of no other links that are capable of aggregating with this link (e.g., the System has allocated a unique operational Key value to the link).

The capability information communicated between Systems, therefore, includes this local knowledge of the aggregation capability of the link in addition to the operational Key value; i.e., whether the System considers the link to be Aggregateable or Individual.

An administrative Key value and an operational Key value shall also be associated with each Aggregator. The operational Key is the Key that is currently in active use for the purposes of forming aggregations. The administrative Key allows manipulation of Key values by management. The values of administrative and operational Key for an Aggregator may differ in the same manner as that of port Keys, per item e) and item f) in this subclause. Ports that are members of a given Key Group can only be bound to Aggregators that share the same operational Key value.

All Keys are 16-bit identifiers. All values except the null value (all zeros) are available for local use.

NOTE—This model allows for two convenient initial configurations. The first is achieved by assigning each port an initial administrative and operational Key value identical to its port number, and assigning the same port numbers as Keys to the corresponding Aggregators for each port. A device with this initial configuration will bring up all links as individual, non-aggregated links. The second is achieved by assigning the same administrative and operational Key values to all ports with a common set of capabilities, and also to all Aggregators. A device with this initial configuration will attempt to aggregate together any set of links that have the same Partner System ID and operational Key, and for which both Systems are prepared to allow aggregation.

5.3.6 Link Aggregation Group identification

A Link Aggregation Group consists of either

- a) One or more Aggregateable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or
- b) An Individual link.

5.3.6.1 Construction of the Link Aggregation Group Identifier

A unique Link Aggregation Group Identifier (LAG ID) is constructed from the following parameters for each of the communicating Systems:

- a) The System Identifier
- b) The operational Key assigned to the ports in the LAG
- c) The Port Identifier, if the link is identified as an Individual link

The local System's values for these parameters shall be non-zero. In cases where the local System is unable to determine the remote System's values for these parameters by exchange of protocol information, administrative values are used in the construction of the LAG ID. The value of these administrative parameters for the remote System may be configured as zero, provided that the port(s) concerned are also configured to be Individual.

A compound identifier formed from the System Identifiers and Key values alone is sufficient to identify a LAG comprising Aggregateable links. However, such an identifier is not sufficient for a LAG comprising a single Individual link where the Partner System Identifier and operational Key may be zero. Even if these are non-zero there may be multiple Individual Links with the same System Identifier and operational Key combinations, and it becomes necessary to include Port Identifiers to provide unique LAG IDs.

Given that

- d) S and T are System Identifiers,

- e) K and L are the operational Keys assigned to a LAG by S and T respectively, and
- f) P and Q are the Port Identifiers of the ports being attached if the LAG comprises a single Individual Link and zero if the LAG comprises one or more Aggregateable links,

then the general form of the unique LAG ID is [(SKP), (TLQ)].

To simplify comparison of LAG IDs it is conventional to order these such that S is the numerically smaller of S and T.

5.3.6.2 Representation of the Link Aggregation Group Identifier

In order to allow for convenient transcription and interpretation by human network personnel, this standard provides a convention for representing compound LAG IDs. Using this format

- a) All fields are written as hexadecimal numbers, two digits per octet, in canonical format.
- b) Octets are presented in order, from left to right. Within fields carrying numerical significance (e.g., priority values), the most significant octet is presented first, and the least significant octet last.
- c) Within fields that carry MAC addresses, successive octets are separated by dashes (-), in accordance with the hexadecimal representation for MAC addresses defined in IEEE Std 802-1990.
- d) Parameters of the LAG ID are separated by commas.

For example, consider the parameters for the two Partners in a Link Aggregation Group shown in Table 5–1.

Table 5–1—Example Partner parameters

	Partner SKP	Partner TLQ
System Parameters (S, T)	System Priority = 0x8000 (see 5.4.2.2) System Identifier = AC-DE-48-03-67-80	System Priority = 0x8000 (see 5.4.2.2) System Identifier = AC-DE-48-03-FF-FF
Key Parameter (K, L)	Key = 0x0001	Key = 0x00AA
Port Parameters (P, Q)	Port Priority = 0x80 (see 5.4.2.2) Port Number = 0x0002	Port Priority = 0x80 (see 5.4.2.2) Port Number = 0x0002

The complete LAG ID derived from this information is represented as follows, for an Individual link:

[(SKP), (TLQ)] = [(8000,AC-DE-48-03-67-80,0001,80,0002), (8000,AC-DE-48-03-FF-FF,00AA,80,0002)]

The corresponding LAG ID for a set of Aggregateable links is represented as follows:

[(SKP), (TLQ)] = [(8000,AC-DE-48-03-67-80,0001,00,0000), (8000,AC-DE-48-03-FF-FF,00AA,00,0000)]

NOTE—The difference between the two representations is that, for an Aggregateable link, the port identifier components are zero.

It is recommended that this format be used whenever displaying LAG ID information for use by network personnel.

5.3.7 Selecting a Link Aggregation Group

Each port is selected for membership in the Link Aggregation Group uniquely identified by the LAG ID (composed of operational information, both derived from local administrative parameters and received through the Link Aggregation Control Protocol). Initial determination of the LAG ID is delayed to allow receipt of such information from a peer Link Aggregation Control entity; in the event such information is not received, locally configured administrative defaults are assumed for the remote port's operational parameters.

Where a particular link is known to be Individual, the complete LAG ID is not required to select the Link Aggregation Group since the link will not be aggregated with any other.

5.3.8 Agreeing on a Link Aggregation Group

Before frames are distributed and collected from a link, both the local Link Aggregation Control entity and its remote peer (if present) need to agree on the Link Aggregation Group. The Link Aggregation Control Protocol allows each of the communicating entities to check their peer's current understanding of the LAG ID, and facilitates rapid exchange of operational parameters while that understanding differs from their own. The protocol entities monitor their operation and, if agreement is not reached (perhaps due to an implementation failure), management is alerted.

The ability of LACP to signal that a particular link is Individual can accelerate the use of the link since, if both Link Aggregation Control entities know that the link is Individual, full agreement on the LAG ID is not necessary.

5.3.9 Attaching a link to an Aggregator

Once a link has selected a Link Aggregation Group, Link Aggregation Control can attach that link to a compatible Aggregator. An Aggregator is compatible if

- a) The Aggregator's operational Key matches the port's operational Key, and
- b) All other links currently attached to the Aggregator have selected the same Link Aggregation Group.

If several compatible Aggregators exist, Link Aggregation Control may employ a locally determined algorithm, either to ensure deterministic behavior (i.e., independence from the order in which Aggregators become available) or to maximize availability of the aggregation to a MAC Client. If no compatible Aggregator exists, then it is not possible to enable the link until such a time as a compatible Aggregator becomes available.

NOTE—In a properly configured System, there should always be a suitable Aggregator available with the proper Key assigned to serve a newly created Link Aggregation Group, so the unavailability of a compatible Aggregator is normally a temporary state encountered while links are moved between Aggregators. However, given the flexibility of the Key scheme, and given that in some implementations there may not be enough Aggregators to service a given configuration of links, it is possible to create configurations in which there is no Aggregator available to serve a newly identified LAG, in which case the links that are members of that LAG cannot become active until such a time as the configuration is changed to free up an appropriate Aggregator.

Links that are not successful candidates for aggregation (e.g., links that are attached to other devices that cannot perform aggregation or links that have been manually configured to be non-aggregateable) are enabled to operate as individual IEEE 802.3 links. For consistency of modeling, such a link is regarded as being attached to a compatible Aggregator that can only be associated with a single link. That is, from the perspective of Link Aggregation, non-aggregated links are not a special case; they compose an aggregation with a maximum membership of one link.

More than one link can select the same Link Aggregation Group within a short period of time and, as these links detach from their prior Aggregators, additional compatible Aggregators can become available. In order

to avoid such events causing repeated configuration changes, Link Aggregation Control applies hysteresis to the attachment process and allows multiple links to be attached to an Aggregator at the same time.

5.3.10 Signaling readiness to transfer user data

Once a link has been attached to an Aggregator (5.3.9) compatible with the agreed-upon Link Aggregation Group (5.3.8), each Link Aggregation Control entity signals to its peer its readiness to transfer user data to and from the Aggregator's MAC Client. In addition to allowing time for the organization of local Aggregator resources, including the possibility that a compatible Aggregator may not exist, explicit signaling of readiness to transfer user data can be delayed to ensure preservation of frame ordering and prevention of frame duplication. Link Aggregation Control will not signal readiness until it is certain that there are no frames in transit on the link that were transmitted while the link was a member of a previous Link Aggregation Group. This may involve the use of an explicit Marker protocol that ensures that no frames remain to be received at either end of the link before reconfiguration takes place. The operation of the Marker protocol is described in 5.5. The decision as to when, or if, the Marker protocol is used is entirely dependent upon the nature of the distribution algorithm that is employed.

5.3.11 Enabling Collection and Distribution

Every Aggregator can enable or disable Collection and Distribution of frames for each port that is attached to the Aggregator. Initially, both Collection and Distribution are disabled. Once the Link Aggregation Control entity is ready to transfer user data using the link and its peer entity has also signaled readiness, the process of enabling the link can proceed. The Collector is enabled (thus preparing it to receive frames sent over the link by the remote Aggregator's Distributor) and that fact is communicated to the Partner. Once the received information indicates that the remote Aggregator's Collector is enabled, the Distributor is also enabled.

NOTE—This description assumes that the implementation is capable of controlling the state of the transmit and receive functions of the MAC independently. In an implementation where this is not possible, the transmit and receive functions are enabled or disabled together. The manner in which this is achieved is detailed in the description of the Mux machine (see 5.4.15).

If at least one port's Mux in the Link Aggregation Group is Collecting, then the Receive state of the corresponding Aggregator will be Enabled. If at least one port's Mux in the Link Aggregation Group is Distributing, then the Transmit state of the corresponding Aggregator will be Enabled.

5.3.12 Monitoring the membership of a Link Aggregation Group

Each link is monitored in order to confirm that the Link Aggregation Control functions at each end of the link still agree on the configuration information for that link. If the monitoring process detects a change in configuration that materially affects the link's membership in its current LAG, then it may be necessary to remove the link from its current LAG and to move it to a new LAG.

5.3.13 Detaching a link from an Aggregator

A port may be detached from the Aggregator used by its Link Aggregation Group as a result of protocol (e.g., Key) changes, or because of System constraints (e.g., exceeding a maximum allowable number of aggregated links, or device failures) at either end of the link. Both classes of events will cause the LAG ID information for the link to change, and it will be necessary for Link Aggregation Control to detach the link from its current Aggregator and move it to a new LAG (if possible). At the point where the change is detected, the Collecting and Distributing states for the port are set to FALSE. The Frame Distribution function is informed that the link is no longer part of the group, the changed configuration information is communicated to the corresponding Link Aggregation Partner, then the Frame Collection function is informed that the link is no longer part of the group.

Once a link has been removed from its Aggregator, the link can select its new Link Aggregation Group and then attach to a compatible Aggregator, as described in 5.3.7 and 5.3.9.

Any conversation that is reallocated to a different link as a result of detaching a link from an Aggregator shall have its frame ordering preserved. This may involve the use of the Marker protocol to ensure that no frames that form part of that conversation remain to be received at either end of the old link before the conversation can proceed on the new link.

5.3.14 Configuration and administrative control of Link Aggregation

Administrative configuration facilities allow a degree of control to be exerted over the way that links may be aggregated. In particular, administrative configuration allows

- a) Key values associated with a port to be identified or modified.
- b) Key values associated with an Aggregator to be identified or modified.
- c) Links to be identified as being incapable of aggregation.
- d) Link Aggregation Control Protocol parameters to be identified or modified.

5.3.15 Link Aggregation Control state information

The Link Aggregation Control function maintains the following information with respect to each link:

- a) The identifier of the Link Aggregation Group to which it currently belongs.
- b) The identifier of the Aggregator associated with that Link Aggregation Group.
- c) The status of interaction between the Frame Collection function of the Aggregator and the link (Collecting TRUE or Collecting FALSE). Collecting TRUE indicates that the receive function of this link is enabled with respect to its participation in an aggregation; i.e., received frames will be passed up to the Aggregator for collection.
- d) The status of interaction between the Frame Distribution function of the Aggregator and the link (Distributing TRUE or Distributing FALSE). Distributing TRUE indicates that the transmit function of this link is enabled with respect to its participation in an aggregation; i.e., frames may be passed down from the Aggregator's distribution function for transmission.

This state information is communicated directly between Link Aggregation Control and the Aggregator through shared state variables without the use of a formal service interface.

The Link Aggregation Control function maintains the following information with respect to each Aggregator:

- e) The status of the Frame Collection function (Receive Enabled or Receive Disabled).
- f) The status of the Frame Distribution function (Transmit Enabled or Transmit Disabled).

These status values are exactly the logical OR of the Collection and Distribution status of the individual links associated with that Aggregator; i.e., if one or more links in the Link Aggregation Group are Collecting, then the Aggregator is Receive Enabled, and if one or more links are Distributing, then the Aggregator is Transmit Enabled.

The Transmit and Receive status of the Aggregator effectively govern the point at which the Aggregator becomes available for use by the MAC Client, or conversely, the point at which it ceases to be available.

5.4 Link Aggregation Control Protocol (LACP)

The Link Aggregation Control Protocol (LACP) provides a standardized means for exchanging information between Partner Systems on a link to allow their Link Aggregation Control instances to reach agreement on

the identity of the Link Aggregation Group to which the link belongs, move the link to that Link Aggregation Group, and enable its transmission and reception functions in an orderly manner.

5.4.1 LACP design elements

The following considerations were taken into account during the development of the protocol described in this subclause:

- a) The protocol depends upon the transmission of information and state, rather than the transmission of commands. LACPDUs sent by the first party (the Actor) convey to the second party (the Actor's protocol Partner) what the Actor knows, both about its own state and that of the Partner.
- b) The information conveyed in the protocol is sufficient to allow the Partner to determine what action to take next.
- c) Active or passive participation in LACP is controlled by LACP_Activity, an administrative control associated with each port, that can take the value Active LACP or Passive LACP. Passive LACP indicates the port's preference for not transmitting LACPDUs unless its Partner's control value is Active LACP (i.e., a preference not to speak unless spoken to). Active LACP indicates the port's preference to participate in the protocol regardless of the Partner's control value (i.e., a preference to speak regardless).
- d) Periodic transmission of LACPDUs occurs if the LACP_Activity control of either the Actor or the Partner is Active LACP. These periodic transmissions will occur at either a slow or fast transmission rate depending upon the expressed LACP_Timeout preference (Long Timeout or Short Timeout) of the Partner System.
- e) In addition to periodic LACPDU transmissions, the protocol transmits LACPDUs when there is a Need To Transmit (NTT) something to the Partner; i.e., when the Actor's state changes or when it is apparent from the Partner's LACPDUs that the Partner does not know the Actor's current state.
- f) The protocol assumes that the rate of LACPDU loss is very low.

There is no explicit frame loss detection/retry mechanism employed by the LACP; however, if information is received from the Partner indicating that it does not have up-to-date information on the Actor's state, or if the next periodic transmission is due, then the Actor will transmit a LACPDU that will correctly update the Partner.

5.4.2 LACPDU structure and encoding

5.4.2.1 Transmission and representation of octets

All LACPDUs comprise an integral number of octets. The bits in each octet are numbered from 0 to 7, where 0 is the low-order bit. When consecutive octets are used to represent a numerical value, the most significant octet is transmitted first, followed by successively less significant octets.

When the encoding of (an element of) a LACPDU is depicted in a diagram

- a) Octets are transmitted from top to bottom.
- b) Within an octet, bits are shown with bit 0 to the left and bit 7 to the right, and are transmitted from left to right.
- c) When consecutive octets are used to represent a binary number, the octet transmitted first has the more significant value.
- d) When consecutive octets are used to represent a MAC address, the least significant bit of the first octet is assigned the value of the first bit of the MAC address, the next most significant bit the value of the second bit of the MAC address, and so on through the eighth bit. Similarly, the least significant through most significant bits of the second octet are assigned the value of the ninth through seventeenth bits of the MAC address, and so on for all the octets of the MAC address.