

Lokale und Regionale Netze
 Bus mit Sendeberechtigungsmarke
 Identisch mit ISO/IEC 8802-4 : 1990

DIN
66 325
 Teil 4

Local and Metropolitan Area Networks; Token-passing bus access method; Identical with ISO/IEC 8802-4 : 1990
 Réseaux Locaux et Métropolitains; Méthode d'accès relative au bus à passage de jeton; Identique à ISO/IEC 8802-4 : 1990

Die Internationale Norm ISO/IEC 8802-4 : 1990, „Information processing systems; Local area networks; Part 4: Token-passing bus access method and physical layer specifications“, ist unverändert in diese Deutsche Norm übernommen worden.

Nationales Vorwort

Einführung

Die Internationale Norm ISO/IEC 8802-4 : 1990 wurde im Rahmen des Projektes IEEE 802.4¹⁾ erarbeitet und von ISO/IEC JTC 1/SC 6²⁾ unter Mitwirkung des Arbeitsausschusses Datenkommunikation im Normenausschuß Informationsverarbeitungssysteme (NI) überarbeitet und übernommen.

Das Präsidium des DIN hat mit Präsidialbeschluß 13/1983 festgelegt, daß deutsche Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Die Bedingungen sind für diese Norm erfüllt.

Um den Gebrauch der Norm zu erleichtern, enthält das nationale Vorwort eine deutsche Erläuterung und eine Fachwörterliste in Englisch-Deutsch und Deutsch-Englisch.

Anmerkung: Angaben über elektrische Sicherheit und elektromagnetische Verträglichkeit werden derzeit von den zuständigen nationalen Gremien bearbeitet. Aussagen zu den oben genannten Punkten sind nicht Teil der Norm und haben lediglich informativen Charakter.

Verweisung auf andere Normen für Lokale Netze

Diese Norm gehört zu einer Normen-Reihe für Lokale und Regionale Netze. Sie beschreibt die Bitübertragungsschicht und einen Teil der Sicherungsschicht nach dem Basis-Referenzmodell DIN ISO 7498 für die Kommunikation Offener Systeme. Es werden derzeit drei Typen von Steuerungsverfahren für den Mediumzugriff (MAC) mit entsprechenden physikalischen Medien festgelegt.

Die unterschiedlichen MAC-Verfahren sind beschrieben in DIN ISO 8802 Teil 3, DIN ISO 8802 Teil 5³⁾ und in dieser Norm.

Das Steuerungsverfahren im Übermittlungsabschnitt ist beschrieben in DIN 66 325 Teil 2.

¹⁾ IEEE — The Institute of Electrical and Electronics Engineers

²⁾ ISO — International Organization for Standardization

IEC — International Electrotechnical Commission

JTC 1 — Joint Technical Committee Information Technology

SC 6 — Telecommunications and information exchange between systems

³⁾ Z. Z. Entwurf

Fortsetzung Seite 2 bis 4
 und 281 Seiten ISO/IEC-Norm

Normenausschuß Informationsverarbeitungssysteme (NI) im DIN Deutsches Institut für Normung e.V.

Grundkonzept des Busses mit Sendeberechtigungsmarke

Beim Bus mit Sendeberechtigungsmarke kontrolliert die Sendeberechtigungsmarke (ein charakteristisches Bitmuster) den geordneten Zugriff auf das physikalische Medium, welches als Bus realisiert ist. Prinzipiell kann jede Station, die am Bus angeschlossen ist, alle gesendeten Signale empfangen.

Eine Station, die die Sendeberechtigungsmarke besitzt, übt die momentane Kontrolle über das Medium aus. Durch die Weitergabe der Sendeberechtigungsmarke wechselt diese Kontrolle von Station zu Station. Die Sendeberechtigungsmarke wird so weitergegeben, daß die Stationen in numerisch fallender Reihenfolge der Stationsadressen durchlaufen werden. Jede Station kennt außer ihrer eigenen Adresse auch die Adresse ihrer direkten Vorgänger- und Nachfolgestation. Die Station mit der kleinsten Stationsadresse gibt die Sendeberechtigungsmarke an die Station mit der größten Stationsadresse weiter, wodurch ein logischer Ring aufgebaut wird. Die die Sendeberechtigungsmarke weitergebende Station kontrolliert jeweils auch den Erfolg der Weitergabe, indem sie das Medium abhört und die nachfolgenden Aktivitäten registriert.

Das MAC-Verfahren definiert u. a. Prozeduren zur Überbrückung ausgefallener Stationen, zum Einbau neuer Stationen in den logischen Ring, zur Initialisierung des Systems und zur Auflösung von Konflikten.

Die wahlfreie Verwendung von Nachrichtenprioritäten erlaubt es, zeitkritische Kommunikationsdienste bevorzugt zu behandeln und ermöglicht somit eine bessere Nutzung der zur Verfügung stehenden Übertragungskapazität.

Bei der hier beschriebenen Methode handelt es sich um ein verteiltes (dezentrales) Zugriffsverfahren, d. h., alle Stationen sind gleichermaßen für den logischen Ring verantwortlich, und es besteht keine Notwendigkeit für eine zentrale Steuer- oder Kontrolleinheit. Dieses Zugriffsverfahren ist deterministisch in dem Sinne, daß es erlaubt, Aussagen über die maximalen Reaktionszeiten des Systems zu treffen.

Die vorliegende Norm enthält auch die Spezifikationen dreier verschiedener Übertragungstechniken, die Datenraten von 1 Mbit/s, 5 Mbit/s und 10 Mbit/s realisieren. Zukünftige Ausgaben dieser Norm können ähnliche Spezifikationen für andere Implementierungen (andere Medien und Datenraten) enthalten.

Zitierte Normen

— in der englischen Originalfassung:

Siehe Abschnitt 1.3

— in nationalen Zusätzen:

DIN ISO 7498 Informationsverarbeitung; Kommunikation Offener Systeme; Basis-Referenzmodell

DIN 66 325 Teil 2 Lokale und Regionale Netze, LLC; Steuerungsverfahren im Übermittlungsabschnitt

DIN ISO 8802 Teil 3 Lokale Netze; Lokales Netz für Vielfach-Zugriff mit Aktivitätsüberwachung und Kollisionserkennung (CSMA/CD)

DIN ISO 8802 Teil 5 (z. Z. Entwurf) Lokale Netze; Ring mit Sendeberechtigungsmarke (Token Ring)

Internationale Patentklassifikation

G 06 F 13/00

H 04 L 29/02

Fachwörterliste

Abkürzungen sind gegebenenfalls den Fachwörtern angefügt und von diesen durch ein Komma getrennt.

Englisch-Deutsch

baseband LAN	Basisband-LAN
bridge	MAC-Übergangseinheit
broadband LAN	Trägerfrequenz-LAN
bus network	Busnetz
carrier sense	Aktivitätsüberwachung
ending frame delimiter	Blockendebegrenzer
forward LAN channel	LAN-Hinkanal
headend	Kopfstation
LAN broadcast	LAN-Sammelauf Ruf
LAN gateway	LAN-Übergangseinheit
LAN multicast	LAN-Gruppenaufruf
LLC procedure	LLC-Verfahren
local area network, LAN	Lokales Netz, LAN
logical link control, LLC	LLC
medium access control, MAC procedure	Steuerungsverfahren für den Mediumzugriff, MAC
medium attachment unit, MAU	Medium-Anschlußeinheit, MAU
medium dependent interface	Medium-Schnittstelle
metropolitan area network, MAN	Regionales Netz
repeater	Regenerator
reverse LAN channel	LAN-Rückkanal
round trip propagation time	Hin- und Rücklaufzeit
slot-time	Schlitzzeit
splitter	Verzweiger
starting frame delimiter	Blockanfangsbegrenzer
token	Sendeberechtigungsmarke
token (passing) bus	Bus mit Sendeberechtigungsmarke
token passing procedure	Steuerungsverfahren mit Sendeberechtigungsmarke
transmission frame	Übertragungsblock
transmission medium	Übertragungsmedium

Deutsch-Englisch

Aktivitätsüberwachung

Basisband-LAN

Blockanfangsbegrenzer

Blockendebegrenzer

Bus mit Sendeberechtigungsmarke

Busnetz

Hin- und Rücklaufzeit

Kopfstation

LAN-Gruppenaufruf

LAN-Hinkanal

LAN-Rückkanal

LAN-Sammelaufruf

LAN-Übergangseinheit

LLC

LLC Verfahren

Lokales Netz, LAN

MAC-Übergangseinheit

Medium-Anschlußeinheit, MAU

Medium-Schnittstelle

Regenerator

Regionales Netz

Schlitzzeit

Sendeberechtigungsmarke

Steuerungsverfahren für den Mediumzugriff, MAC

Steuerungsverfahren mit Sendeberechtigungsmarke

Trägerfrequenz-LAN

Übertragungsblock

Übertragungsmedium

Verzweiger

carrier sense

baseband LAN

starting frame delimiter

ending frame delimiter

token (passing) bus

bus network

round trip propagation time

headend

LAN multicast

forward LAN channel

reverse LAN channel

LAN broadcast

LAN gateway

logical link control, LLC

LLC procedure

local area network, LAN

bridge

medium attachment unit, MAU

medium dependent interface

repeater

metropolitan area network, MAN

slot-time

token

medium access control, MAC procedure

token passing procedure

broadband LAN

transmission frame

transmission medium

splitter

INTERNATIONAL
STANDARD

**ISO/IEC
8802-4**

**ANSI/IEEE
Std 802.4**

First edition
1990-08-17

**Information processing systems —
Local area networks —**

**Part 4 :
Token-passing bus access method and physical layer
specifications**

Systèmes de traitement de l'information — Réseaux locaux —

*Partie 4 : Spécification pour la méthode d'accès et la couche physique relatives au
bus à passage de jeton*



Reference number
ISO/IEC 8802-4 : 1990 (E)
ANSI/IEEE
Std 802.4-1990

International Standard ISO/IEC 8802-4 : 1990
ANSI/IEEE Std 802.4-1990
(Revision of ANSI/IEEE 802.4-1985)

Information processing systems — Local area networks —

Part 4: Token-passing bus access method and physical layer specifications

Correction Sheet
August 17, 1990

Pages 243 and 252 (Footnote 16 and 17) should be corrected as follows: change "not" to "now" in the last sentence. The corrected footnote should read: "Although a strong attempt was made to align IEEE and ISO/IEC versions of this Standard, the version approved by the IEEE Standards Board included this Appendix as a normative Appendix. Therefore, this Appendix is now under consideration by IEEE as an informative Appendix."

International Standard ISO/IEC 8802-4 : 1990
ANSI/IEEE Std 802.4-1990
(Revision of ANSI/IEEE Std 802.4-1985)

Information processing systems — Local area networks —

Part 4: Token-passing bus access method and physical layer specifications

Sponsor
Technical Committee on Computer Communications
of the
IEEE Computer Society

Approved February 27, 1990
IEEE Standards Board

Approved July 19, 1990
American National Standards Institute

Approved 1990 by the
International Organization for Standardization
and by the
International Electrotechnical Commission



Adopted as an International Standard by the
International Organization for Standardization
and by the
International Electrotechnical Commission

Published by
The Institute of Electrical and Electronics Engineers, Inc.

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1990 by

*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.*



Documents
Document Sharing Hub

International Standard ISO/IEC 8802-4 : 1990

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) together form a system for worldwide standardization as a whole. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to the national bodies for approval before their acceptance as International Standards. They are approved in accordance with procedures requiring at least 75% approval by the national bodies voting.

In 1985, IEEE Std 802.4-1985 was adopted by ISO Technical Committee 97, *Information processing systems*, as draft International Standard ISO/DIS 8802-4. A further revision was subsequently approved by ISO/IEC JTC 1 in the form of this new edition, which is published as International Standard ISO/IEC 8802-4:1990.

For the purpose of assigning global addresses, the Institute of Electrical and Electronics Engineers, Inc., USA, has been designated by the ISO Council as the Registration Authority. Communications on this subject should be addressed to Registration Authority for ISO/IEC 8802-4.

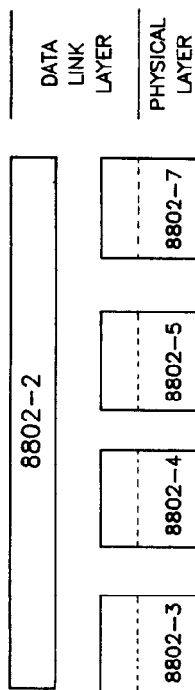
c/o The Institute of Electrical and Electronics Engineers, Inc.
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA



International Organization for Standardization/International Electrotechnical Commission
Case postale 56 • CH-1211 Genève 20 • Switzerland

Foreword to International Standard ISO/IEC 8802-4 : 1990

This standard is part of a family of standards for Local Area Networks (LANs). The relationship between this standard and the other members of the family is shown below. (The numbers in the figure refer to ISO Standard numbers.)



This family of standards deals with the physical and data link layers as defined by the ISO Open Systems Interconnection Reference Model (ISO 7498 : 1984). The access standards define four types of medium access technologies and associated physical media, each appropriate for particular applications or system objectives. The standards defining these technologies are

- (1) ISO 8802-3 [IEEE Std 802.3-1988], a bus utilizing CSMA/CD as the access method,
- (2) ISO/IEC 8802-4 [IEEE Std 802.4-1990], a bus utilizing token passing as the access method,
- (3) ISO 8802-7, a ring utilizing slotted ring as the access method.

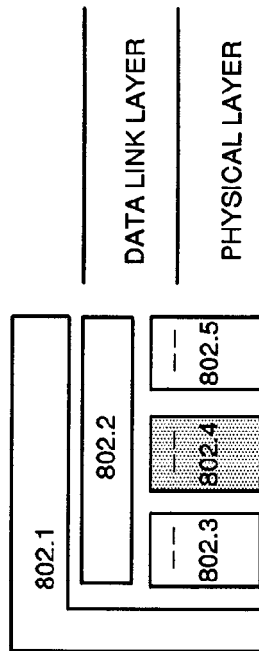
ISO 8802-2 [IEEE Std 802.2-1988], Logical Link Control protocol, is used in conjunction with the medium access standards.

The reader of this document is urged to become familiar with the complete family of standards.

The main body of this standard serves for both the ISO/IEC 8802-4 : 1990 and IEEE 802.4:1990 standards, except for portions that specifically state that they are not a part of the ISO/IEC standard. Such portions apply to the IEEE standard only. ISO/IEC and IEEE each have unique foreword sections. The Appendices serve as useful reference material to both standards.

(This Foreword is not a part of ISO/IEC 8802-4:1990 or of IEEE Std 802.4-1990.)

This standard is part of a family of standards for Local Area Networks (LANs). The relationship between this standard and other members of the family is shown below. (The numbers in the figure refer to IEEE Standard numbers.)



This family of standards deals with the physical and data link layers as defined by the ISO Open Systems Interconnection Basic Reference Model (ISO 7498:1984). The access standards define three types of medium access technologies and associated physical media, each appropriate for particular applications or system objectives. The standards defining these technologies are

- (1) IEEE Std 802.3-1988 [ISO 8802-3], a bus utilizing CSMA/CD as the access method,
- (2) IEEE Std 802.4-1990 [ISO/IEC 8802-4], a bus utilizing token passing as the access method,
- (3) IEEE Std 802.5-1989, a ring utilizing token passing as the access method.

Other access methods (for example, metropolitan area networks and integrated voice-data networks) are under investigation.

IEEE Std 802.2-1989 [ISO 8802-2], the Logical Link Control standard, is used in conjunction with the medium access standards.

IEEE 802.1¹ (a series of related standards) describes the relationship among the family of 802 standards and their relationship to the ISO Open Systems Interconnection Basic Reference Model in more detail. IEEE 802.1 will also contain networking management standards and information on interworking.

The reader of this standard is urged to become familiar with the complete family of standards.

¹ IEEE Std 802.1A-1990, Overview and Architecture of Network Standards; IEEE Std 802.1D-1990, MAC (Media Access Control) Bridges; and IEEE Std 802.1E-1990, System Load Protocol have been approved as IEEE Standards, but are not yet published. Other projects in the 802.1 series are currently under development.

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE which have expressed an interest in participating in the development of the standard.

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

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.

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 all 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 technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331
USA

IEEE Standards documents are adopted by the Institute of Electrical and Electronics Engineers without regard to whether their adoption may involve patents on articles, materials, or processes. Such adoptions do not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards documents.

Summary of Changes

This standard is a major revision of IEEE Std 802.4-1985 (ISO/DIS 8802/4). This revision incorporates the results of over four years of work by the IEEE 802.4 Working Group and by other organizations.

This section of the Foreword summarizes the changes for the reader's convenience.

The management sections, Sections 3 and 9, have been completely rewritten. These revised sections are now in conformance with the work of the IEEE 802 Committee and in alignment, to the extent possible, with other groups working on management standards.

The access control machine (ACM), the heart of the Medium Access Control (MAC) protocol, has been revised. Errors and ambiguities that existed in the previous version were corrected, and enhancements were made to improve error recovery. The intention in revising the ACM was to retain complete interoperability with the ACM described by the previous standard. We believe that intention has been realized.

The name of the request_with_response option has been changed from "immediate response" option to avoid conflicts with other uses of the phrase immediate response (see 6.1.1 and 6.6.2).

Additional MAC Capabilities (6.7) has been added to give implementors suggestions for providing extra MAC features and to give guidance to conformance testers.

The specification of an interface within the Physical Layer for a separate modem has been included as Section 10.

Section 11 has been reserved to retain compatible section numbers and for future additions.

The phase-coherent FSK Physical Layer and medium, described in Sections 12 and 13, underwent revision. Based on implementation experience, the working group found it necessary to make substantive technical changes to the standard. The transmit and receive levels were changed, the preamble pattern and length were changed, and a receiver blanking specification was added. The result of these changes is that the implementations of this revised standard will not interoperate with implementations of the previous standard.

The broadband Physical Layer and medium, described in Sections 14 and 15, underwent minor revisions to clarify the specification and to provide guidance for conformance testing.

A fiber optic Physical Layer and medium specification has been added as Sections 16 and 17.

The 1 Mb/s phase-continuous FSK Physical Layer and medium specification was moved from Sections 10 and 11 to Sections 18 and 19, to avoid renumbering the more commonly referenced Physical Layer and media sections. The coding of the end delimiter coding pattern was changed to increase the reliability (Hamming distance) of the protocol.

This standard was submitted to ISO/IEC JTC1 for consideration as a revision and addendum to the previous edition of the token bus LAN standard, IEEE Std 802.4-1985 (ISO/DIS 8802/4). To facilitate processing of that document, portions that were not appropriate for an international standard were prefaced with a note enclosed in braces {...}.

These same portions contained within the current edition are not a part of the International Standard and are stated as such. These portions are peculiar to the IEEE version of this standard and consist of areas relating to

- (1) References to national standards
- (2) Recommended frequency allocations for North American CATV systems (see 14.8.4)

- (3) Recommendations and guidelines related to safety concerns

This standard contains state-of-the-art material. The area covered by this standard is undergoing evolution. Revisions to this standard may occur either to clarify existing material, to correct possible errors, or to incorporate new, related material.

Readers wishing to know the state of revisions should contact the

Secretary, IEEE Standards Board

445 Hoes Lane, P.O. Box 1331

Piscataway, NJ 08855-1331, USA

The following were voting members of the IEEE 802.4 Token-Passing Bus Access Method Working Group at the time of approval of this standard:

Paul S. Eastman, Chair

Louis F. Wojnarowski, Vice Chair
Rhonda Alexis-Dirvin, Secretary

Ernest E. Bergmann
Clyde A. Boenke
Shalom Bresticker
Charles Brill
Mike Bukowski
Hsin-Hwal Chen
Jade Y. Chen
Robert S. Crowder
C. David Daly
Sanjay Dhawan
H. N. "Hank" Dorris
Robert H. Douglas
Daniel Dove
Richard B. Formester
Anatoly V. Fridland
Darrell Fulton
T. R. (Tom) Gandhi
Mavis Graube
David J. Greenspan

Gerhard Hanmer
Richard F. Hunter
Dittmar Janetzky
Lawrence R. Johnson
Walter Kammerer
Bradford S. Keller
Robert W. Kilgore
William H. Kind
Michael T. Klein
Simon Korowitz
Leonid Koshevoy
Detlev Leisegang
Daniel N. LeBlanc
Bruce A. Loyer
Wayne A. Mack
Charles H. March
Gunter J. Martin
Michael A. Maskid
Stephen G. McChrystal
Dale D. Murry
David L. Nicholson

Toshio Ogawa
Ilan Pardo
Michael J. Perkins
Thomas L. Pinney
Fred Rhine
David W. Riley
Ken Ross
Chandos A. Rypinski
Toshio Saito
Walter Schreier
Rich Seifert
Yehuda Shvager
Daniel P. Stokesberry
Orest Lev Storoshchuk
Catherine F. Summers
Chuck Thurwacher
Lourens Van Der Jagt
Earl J. Whitaker
Roger Wilmarth
Yong W. Yi

Special thanks to the following task group chairmen whose task groups contributed to this revision:

802.1A, Broadband	—	Donald J. Loughry
802.4B, Carrier Band	—	Michael T. Klein
802.4C, MAC Management	—	Robert H. Douglas
802.4D, Physical Management	—	Michael J. Perkins
802.4F, MAC Revision	—	Louis F. Weharaoshi
802.4G, DTE-DCE Interface and past Chair	—	Richard M. Collins
802.4H, Fiber Optic Media	—	Clyde A. Boenke
802.4I, Conformance Testing	—	Robert S. Crowder
802.4J, and past Chair	—	Paul S. Eastman
	—	Robert L. Husak (deceased)

Special thanks to our past chairman, 1983–87, under whose guidance we developed the current draft and reached committee consensus:

Robert H. Douglas, past Chair

The following persons were on the balloting committee that approved this document for submission to the IEEE Standards Board:

William B. Adams	G. Juanlee	Robert S. Primis
J. Ashok	S. L. Junker	Keith Alan Rhodes
Kit Athul	M. Kezunovic	John Riganati
William E. Ayen	Samuel Kho	G. Robinson
Ali Bahrobloomi	T. M. Kurihara	D. J. Rycka
Michael Benedek	Anthony B. Lake	S. I. Sanoylenko
Peter I. P. Boulton	Mike Lawler	George Sanderson
G. S. Carson	F. Y. Lee	Amberjandi R. Sastri
Chih-Tsai Chen	Sateesh B. Lele	N. F. Schneidewind
Lap Yan Chung	F. C. Lim	D. A. Sheppard
Khinam Chyon	R. S. Little	Steve Showalter
Michael H. Coden	William Livingston	L. Sintonen
A. F. Conrad	Donald C. Loughry	Ing Alex Soceanu
R. A. Conser	J. P. Luhnkey	H. P. Sobornon
Richard J. Corley	Wo-Shun Luk	Robert E. Southard
Robert S. Crowder	Andy J. Luque	John Spragins
Andrew I. Davidson	George Mattathil	T. Stack
R. T. Denton	P. S. McIntosh	Michael J. Stephenson
M. G. Duncan	David S. Millman	C. M. Stillebroer
Paul Eastman	Gilbert Melanson	F. J. Strauss
John E. Enrich	John E. Montague	P. Sugar
Philip H. Enslow	Luis F. M. de Moraes	E. D. Sykas
J. W. Fendrich	M. A. F. Morganti	A. N. Tantawi
Harold Fols	Kinji Mori	Nathan Tobol
Harvey A. Freeman	David Morris	Wei-Tek Tsai
Patrick S. Gonia	Steven Moustakas	L. D. Umbaugh
Julio Gonzalez-Sanz	Gerry Nadler	Thomas A. Varettoni
Marie Graube	R. R. Nelson	J. T. Vorhies
Stephen Harris	Arne A. Nilsson	Donald F. Weir
J. Scott Hauglahl	J. D. Northcutt	William J. Wenker
Anne B. Horton	C. Oestericher	Eard J. Whitaker
Chris Hsieh	Young Oh	Michael Willett
Richard J. Huff	Ari Ollikainen	David Wood
Raj Jani	Thomas L. Phinney	George E. Wright
Duncan P. Johnson	Rafal Pirzada	Oren Yuen
	Udo Pooch	

The final conditions for approval of this standard were met on February 27, 1990. This standard was conditionally approved by the IEEE Standards Board on February 15, 1990, with the following membership:

Marco W. Migliaro, Chair	James M. Daly, Vice Chair
Andrew G. Salen, Secretary	
Paul L. Borrell	Kenneth D. Hendrix
Fletcher J. Buckley	John W. Horch
Allen L. Clapp	Joseph L. Koepfinger*
Stephen R. Dillon	Michael A. Lawler
Donald C. Fleckenstein	Donald C. Loughry
Jay Fortner*	John E. May, Jr.
Thomas L. Hannan	Lawrence V. McCall

*Member Emeritus

Contents

SECTION	PAGE
1. Introduction and Overview	17
1.1 Scope	18
1.2 Definitions	18
1.3 References	18
1.4 Conformance	20
1.5 Overview of the Token Method	20
1.6 MAC Sublayer Internal Structure	22
1.7 PLE and Medium	23
1.8 Access Method Characteristics	28
1.9 Standard Organization	29
2. LLC-MAC Interface Service Specifications	31
2.1 Overview of the LLC-MAC Service	31
2.2 Detailed Interactions with the LLC Entity	32
3. MAC Sublayer Management	35
3.1 Overview	36
3.2 MAC Management Facilities	36
4. Frame Formats	43
4.1 Frame Components	44
4.2 Enumeration of Frame Types	50
4.3 Appendix — Recommendation for a Hierarchical Structure for Locally Administered Addresses	52
5. Elements of MAC Sublayer Operation	55
5.1 Basic Operation	56
5.2 Access Control Machine (ACM) States	64
5.3 Interface Machine (IFM) Description	70
5.4 Receive Machine (RxM) Description	71
5.5 Transmit Machine (TxM) Description	75
5.6 Regenerative Repeater Machine (RRM) Description	76
6. MAC Sublayer Definitions and Requirements	77
6.1 MAC Definitions	77
6.2 Transmission Order	78
6.3 Delay Labeling	80
6.4 Miscellaneous Requirements	80
6.5 Use of Address Bits in Contention Algorithms	82
6.6 Options within MAC Sublayer	83
6.7 Additional MAC Capabilities	84
6.8 Delegation of Right to Transmit	86
7. Access Control Machine (ACM) Description	87
7.1 Variables and Functions	87
7.2 Access Control Machine (ACM) Formal Description	97

SECTION	PAGE
8. MAC Sublayer-Physical Layer Interface Service Specification	133
8.1 Overview of the LAN Physical Layer Service	133
8.2 Detailed Specifications	134
9. Physical Layer Entity (PLE) Management	139
9.1 Overview	140
9.2 Physical Management Facilities	140
9.3 Additional Management	144
10. Exposed DTE-DCE Interface	145
10.1 Overview of DTE-DCE Interface	146
10.2 PHY-UNITDATA Request and Indication	146
10.3 DCE Management	149
10.4 Electrical Characteristics	156
10.5 Mechanical Characteristics	160
10.6 Appendix — Clarification of Management Mode	164
11. RESERVED	167
12. Single-Channel Phase-Coherent-FSK Bus Physical Layer Entity (PLE)	169
12.1 Nomenclature	170
12.2 Object	171
12.3 Compatibility Considerations	171
12.4 Medium Overview	171
12.5 PLE Overview	171
12.6 Application of Management	173
12.7 Functional, Electrical, and Mechanical Specifications	173
12.8 Environmental Specifications	181
12.9 Labeling	181
13. Single-Channel Phase-Coherent-FSK Bus Medium	183
13.1 Nomenclature	184
13.2 Object	185
13.3 Compatibility Considerations	185
13.4 Overview	185
13.5 Functional, Electrical, and Mechanical Specifications	186
13.6 Environmental Specifications	189
13.7 Transmission_path_delay Considerations	190
13.8 Documentation	190
13.9 Network Sizing	190
13.10 Appendix — Guidelines for Configuring the Medium	191
14. Broadband Bus Physical Layer Entity (PLE)	195
14.1 Nomenclature	196
14.2 Object	197
14.3 Compatibility Considerations	197
14.4 Operational Overview of Single-Cable Medium	198
14.5 Operational Overview of a Dual-Cable Medium	198
14.6 General Overview	199
14.7 Application of Management	201
14.8 Functional, Electrical, and Mechanical Specifications	201
14.9 Environmental Specifications	215
14.10 Labeling	216
14.11 Appendix — Provisions of Two MAC-Symbol/PHY-Symbol Signaling	216
14.12 Appendix — Detailed Scrambling and Descrambling Process	220
15. Broadband Bus Medium	221
15.1 Nomenclature	222
15.2 Object	224
15.3 Compatibility Considerations	224
15.4 Overview	224
15.5 Functional, Electrical, and Mechanical Specifications	226
15.6 Environmental Specifications	228
15.7 Transmission_path_delay Considerations	229
15.8 Documentation	229
15.9 Network Sizing	229
16. Fiber Optic Physical Layer Entity (PLE)	231
16.1 Nomenclature	232
16.2 Object	234
16.3 Compatibility Considerations	234
16.4 Operational Overview	235
16.5 Physical Layer Overview	235
16.6 Application of Management	236
16.7 Functional, Optical, Electrical, and Mechanical Specifications	236
16.8 Environmental Specifications	242
16.9 Labeling	242
16.10 Appendix — Alternative Fiber Optic Medium	243
16.11 Appendix — Comparison of Parameter Values of Standard and Alternative Fiber Optic Medium	244
17. Fiber Optic Medium	245
17.1 Nomenclature	246
17.2 Object	248
17.3 Compatibility Considerations	248
17.4 Overview	249
17.5 Functional, Optical, and Mechanical Specifications	249
17.6 Safety Requirements	251
17.7 Transmission_path_delay Considerations	251
17.8 Documentation	251
17.9 Appendix — Alternative Fiber Optic Medium	252
17.10 Appendix — Comparison of Parameter Values of Standard and Alternative Fiber Optic Medium	252

SECTION	PAGE
17.11 Appendix — Network Size and Configuration	253
18. Single-Channel Phase-Continuous-FSK Bus Physical Layer Entity (PLE) ..	255
18.1 Nomenclature	256
18.2 Object	257
18.3 Compatibility Considerations	257
18.4 Medium Overview	257
18.5 Physical Layer Overview	258
18.6 Application of Management	259
18.7 Functional, Electrical, and Mechanical Specifications	259
18.8 Environmental Specifications	263
18.9 Labeling	264
19. Single-Channel Phase-Continuous-FSK Bus Medium	265
19.1 Nomenclature	266
19.2 Object	267
19.3 Compatibility Considerations	267
19.4 General Overview	268
19.5 Functional, Electrical, and Mechanical Specifications	268
19.6 Environmental Specifications	270
19.7 Transmission_path_delay Considerations	270
19.8 Documentation	271
19.9 Network Sizing	271
19.10 Appendix — Guidelines for Configuring the Medium	271
FIGURES	
Fig 1-1 Relationship of Adjacent Protocol Layers	17
Fig 1-2 Logical Ring on Physical Bus	21
Fig 1-3 MAC Sublayer Functional Partitioning	22
Fig 2-1 Relation to LAN Model	31
Fig 3-1 Relation to LAN Model	35
Fig 5-1 Relation to LAN Model	55
Fig 5-2 Token Rotation Time "Priority" Example	63
Fig 5-3 MAC Finite State Machine Diagram	65
Fig 5-4 Receive Machine	72
Fig 5-5 sl/act Detector Finite State Machine	73
Fig 5-6 SD Detector Finite State Machine	73
Fig 5-7 ED Detector Finite State Machine	74
Fig 5-8 NB/BQ Finite State Machine	75
Fig 5-9 Bus Repeater	76
Fig 6-1 MAC Protocol_data_unit Transmission Order	79
Fig 6-2 Logical Token-Passing Ring	81
Fig 8-1 Relation to LAN Model	133
Fig 9-1 Relation to LAN Model	139
Fig 10-1 Relation to LAN Model	145
Fig 10-2 Single-ended Interface Connection	157
FIGURE	
Fig 10-3 Indication Channel Timing	158
Fig 10-4 Request Channel Timing Using TXCLK	158
Fig 10-5 Typical Driver — Differential Interface	159
Fig 10-6 Typical Receivers — Differential Interface	159
Fig 10-7 Request Channel Timing Using RETURNCLK	160
Fig 10-8 Typical Derivation of RETURNCLK	160
Fig 10-9 Cable Detection Circuits	161
Fig 10-10 Single-ended Interface Connector Receptacle	161
Fig 10-11 Typical Station Management	164
Fig 10-12 Writing to a Register	165
Fig 10-13 Typical Modem Failure Sequence	166
Fig 12-1 Relation to LAN Model	169
Fig 12-2 Physical Hardware Partitioning	170
Fig 12-3 MAC-Symbol Encodings	174
Fig 12-4 Eye Patterns	176
Fig 12-5 Receiver Blanking	179
Fig 13-1 Relation to LAN Model	183
Fig 13-2 Physical Hardware Partitioning	184
Fig 13-3 Limits for Percent PDD	188
Fig 14-1 Relation to LAN Model	195
Fig 14-2 Physical Hardware Partitioning	196
Fig 14-3 Limits for Amplitude Distortion	209
Fig 14-4 Transmit Spectrum Mask	210
Fig 14-5 $1 + X^{-6} + X^{-7}$ Scrambler	220
Fig 14-6 Exemplary Scrambler/Descrambler Circuitry	220
Fig 15-1 Relation to LAN Model	221
Fig 15-2 Physical Hardware Partitioning	222
Fig 16-1 Relation to LAN Model	231
Fig 16-2 Physical Hardware Partitioning	232
Fig 16-3 Optical Wave Shape Template	238
Fig 17-1 Relation to LAN Model	245
Fig 17-2 Physical Hardware Partitioning	246
Fig 17-3 n by m Splitter/Combiner Primitive	249
Fig 17-4 Star Topology	254
Fig 18-1 Relation to LAN Model	255
Fig 18-2 Physical Hardware Partitioning	256
Fig 19-1 Relation to LAN Model	265
Fig 19-2 Physical Hardware Partitioning	266
TABLES	
Table 10-1 Request Channel Encoding — MAC Mode	148
Table 10-2 Indication Channel Encoding — MAC Mode	149
Table 10-3 Request Channel Encoding — Management Mode	152
Table 10-4 Indication Channel Encoding — Management Mode	152

TABLE	PAGE
Table 10-5 Single-ended Connector Pinout	162
Table 10-6 Differential Connector Pinout	163
Table 12-1 Data Rate Versus Signaling Frequencies	175
Table 14-1 Relative Power Level Reporting	201
Table 14-2 Baseband Pulse Coding Rules	204
Table 14-3 Usual North American Mid-split Channels — Nomenclature and Pairing	206
Table 14-4 Required Transmit Level	207
Table 14-5 Breakpoints of Amplitude Distortion Limits	208
Table 14-6 Required Noise Floor and In-band Signal Power	212
Table 14-7 Minimum Length of <i>Pad_idle</i> Preamble	212
Table 16-1 Data Rate Versus Media Transmission Rate	237
Table 16-2 Alternate Test Fiber Parameter Values	243
Table 16-3 Parameter Values for Standard and Alternate Test Fiber	244
Table 17-1 Alternate Test Fiber Parameter Values	252
Table 17-2 Parameter Values for Standard and Alternate Test Fiber	253
Table 17-3 Passive Star Power Loss Budget	254

APPENDIX

A A Model Used for the Service Specification	275
A1. Service Hierarchy	275
A2. N-Layer Interface	275
A3. Specification of Services	276
A4. Classification of N-Layer Service Primitives	276
A5. Interaction Behavior	276

APPENDIX FIGURES

Fig A1 Service Hierarchy Relationships	275
Fig A2 Service Primitive Interactions	276

INDEX	277
-------------	-----

Information processing systems — Local area networks —

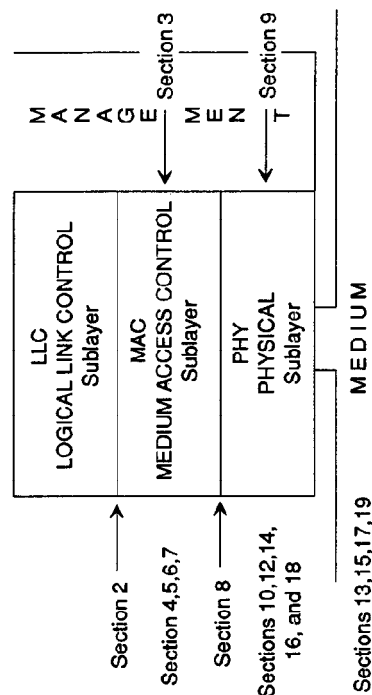
Part 4:

Token-passing bus access method and physical layer specifications

1. Introduction and Overview

This portion of the Local Area Network (LAN) standards deals with all elements of the token-passing bus access method and its associated physical signaling and media technologies. The access function coordinates the use of the shared medium among the attached stations and has the relationship to the other protocol functions given by Fig 1-1.

Fig 1-1
Relationship of Adjacent Protocol Layers



1.1 Scope. For the purpose of compatible interconnection of stations by way of a LAN using the token-passing bus access method this standard

- (1) Specifies the electrical and/or optical and physical characteristics of the transmission medium,
- (2) Specifies the electrical or optical signaling method used,
- (3) Specifies the frame formats transmitted,
- (4) Specifies the actions of a station upon receipt of a frame,
- (5) Specifies the services provided at the conceptual interface between the Medium Access Control (MAC) sublayer and the Logical Link Control (LLC) sublayer above it,
- (6) Specifies the actions, entities, and values used to manage the MAC sublayer and Physical Layer Entity (PLE).

Within this standard the operation of a station is specified in terms of the layered model shown in Fig 1-1 and described in ISO 7498 : 1984 [13].² Additional information on the 802 family of LAN standards can be found in IEEE Std 802.1A [9].

Figure 1-1 also shows which sections of the standard specify interfaces between layers and which sections specify the operation of the layers themselves.

Recommendations for frequency allocations for community antenna television (CATV) systems are subjects for national standardization and are outside the scope of this International Standard.

1.2 Definitions. The definitions used in this standard are consistent with ISO 2382-9 : 1984 [12]. A more specific information processing vocabulary pertaining to LAN systems is in development as WP 2382/25.

1.3 References.³ When the following standards referred to in this standard are superseded by an approved revision, the latest revision shall apply.

References to IEEE and ISA standards are not a part of this International Standard.

[1] ANSI/ISA-S72.01-1986, PROWAY-LAN Industrial Data Highway.⁴

[2] CCITT Recommendation X.150, Principles of Maintenance Testing for Public Data Networks Using Data Terminal Equipment (DTE) Data Circuit-Terminating (DCE) Test Loops. In Vol. VIII.3 of the *CCITT Blue Books* — Data communication network transmission, signaling and switching, network aspects, maintenance and administrative arrangements. Geneva: International Telecommunications Union, 1980.⁵

²The numbers in brackets correspond to those of the references listed in 1.3.

³See index to locate citations in the text of the following publications.

⁴ISA documents are available from the Instrument Society of America, Standards Dept., 67 Alexandria Drive, P.O. Box 12277, Research Triangle Park, NC 27709.

⁵CCITT documents are available in the US from the US Dept. of Commerce, National Technical Information Service, 5285 Port Royal Rd., Springfield VA 22161. CCITT documents are also available from CCITT General Secretariat, International Telecommunications Union (ITU), Sales Section, Place des Nations, CH-1211, Genève 20, Switzerland/Suisse.

[3] CISPR Publication 22 (1985), Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment.⁶

[4] IEC Publication 169-8 (1978), Radio Frequency Connectors, Part 8: RF Coaxial Connectors with Inner Diameter of Outer Conductor 5.5 mm (0.256 in) with Bayonet Lock and Characteristic Impedance of 50 Ω (type BNC).

[5] IEC Publication 807-2 (1985), Detailed Specification for a Range of Connectors and Round Contacts (Fixed Solder Contact Types).

[6] IEC Publication 825 (1984), Radiation Safety of Laser Products, Equipment Classification, Requirements and User's Guide.

[7] IEC Publication 950 (1986), Safety of Information Technology Equipment, Including Electrical Business Equipment (Amendment 1, 1988).

[8] IEEE C97.90.1-1989, IEEE Standard Surge Withstand Capability (SWC) Tests for Protective Relays and Relay Systems (ANSI).⁸

[9] IEEE Std 802.1A-1990, IEEE Standard for Local Area Networks and Metropolitan Area Networks: Overview and Architecture of Network Standards.

[10] Reserved for future use.⁷

[11] IEEE Std 802.7-1989, IEEE Recommended Practices for Broadband Local Area Networks (ANSI).

[12] ISO 2382-9 : 1984, Data processing — Vocabulary — Part 09: Data communication.⁹

[13] ISO 7498 : 1984, Information processing systems — Open systems interconnection — Basic reference model.

[14] ISO/IEC 7498-4 : 1989, Information processing systems — Open systems interconnection — Basic reference model — Part 4: Management framework.

[15] ISO 8022-2 : 1989 (IEEE Std 802.2-1989), Information processing systems — Local area networks — Part 2: Logical link control.

[16] ISO 8824 : 1987, Information processing systems — Open systems interconnection — Specification of abstract syntax notation one (ASN.1).

⁶CISPR and IEC documents are available in the US from the Sales Department, American National Standards Institute, 1430 Broadway, New York, NY 10018. These documents are also available from International Electrotechnical Commission, 3 rue de Varembe, Case postale 131, CH-1211, Genève 20, Switzerland/Suisse.

⁷When the following IEEE project is approved and published, it will become a part of this reference section: P802.1B, Local Area Networks and Metropolitan Networks Management.

⁸IEEE documents are available from the Service Center, Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331.

⁹ISO documents are available in the US from the Sales Department, American National Standards Institute, 1430 Broadway, New York, NY 10018. ISO documents are also available from the ISO Office, 1 rue de Varembe, Case postale 56, CH-1211, Genève 20, Switzerland/Suisse.

- [17] Reserved for future use.¹⁰
- [18] Reserved for future use.¹¹
- [19] Reserved for future use.¹²
- [20] Reserved for future use.¹³

1.4 Conformance. Implementations that claim conformance with this standard shall

- (1) Offer the mandatory LLC-MAC interface services specified in Section 2,
- (2) Support the mandatory layer management actions, entities, and values specified in Sections 3 and 9,
- (3) Generate, transmit, receive, and recognize the frames and sequences specified in Section 4,
- (4) Support the mandatory capabilities, constraints, and behavior of the medium access protocol specified in Sections 6 and 7,
- (5) Offer the mandatory MAC sublayer-Physical Layer interface services specified in Section 8,
- (6) Support the mandatory capabilities, features, and constraints of at least one of the specified PLEs and medium specifications defined in Sections 12 and 13, 14 and 15, 16 and 17, and 18 and 19,
- (7) Support at least one of the data rates specified for the chosen PLE, and
- (8) Support the capabilities, behavior, or values defined for all options claimed to be supported by the implementation.

Unless otherwise stated herein, all specifications shall apply over the entire operating environmental range for which the manufacturer claims conformance. A number of options are specified in this standard. An implementation shall indicate which, if any, of these options are supported.

1.5 Overview of the Token Method

1.5.1 The Essence of the Token Access Method

- (1) A token controls the right of access to the physical medium; the station which holds (possesses) the token has momentary control over the medium.

¹⁰When the following ISO/IEC project is completed, approved, and published, it will become a part of this reference section: JTC1 DIS 9014-3, Information processing systems — Fibre distributed data interface (FDDI) — Part 3: Physical layer medium dependent (PMD) requirements.

¹¹When the following ISO/IEC project is completed, approved, and published, it will become a part of this reference section: JTC1 DIS 9506, Information processing systems — Open systems interconnection — Management information service definition.

¹²When the following ISO/IEC project is completed, approved, and published, it will become a part of this reference section: JTC1 DIS 9586, Information processing systems — Open Systems Interconnection — Management information protocol specification.

¹³When the following ISO/IEC project is completed, approved, and published, it will become a part of this reference section: JTC1 DIS 10039, Information technology — Telecommunications and information exchange between systems — Medium access control service definition.

- (2) The token is passed by stations residing on the medium. As the token is passed from station to station a logical ring is formed.
- (3) Steady state operation consists of a data transfer phase and a token transfer phase.
- (4) Ring maintenance functions within the stations provide for ring initialization, lost token recovery, new station addition to the logical ring, and general housekeeping of the logical ring. The ring maintenance functions are replicated among all the token-using stations on the network.

Shared media generally can be categorized into two major types. These types are *broadcast* and *sequential*. This standard deals exclusively with the broadcast type. On a broadcast medium, every station may receive all signals transmitted. Media of the broadcast type are usually configured as a physical bus.

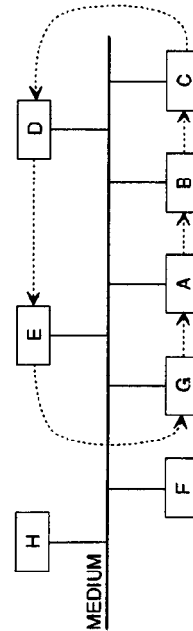
In Fig 1-2, note that the token medium access method is always sequential in a logical sense. That is, during normal, steady state operation, the right to access the medium passes from station to station. Furthermore, note that the physical connectivity has little impact on the order of the logical ring and that stations can respond to a query from the token holder even without being part of the logical ring. (For example, stations H and F can receive frames and could respond but cannot initiate a transmission since they will never be sent the token.)

The MAC sublayer provides sequential access to the shared bus medium by passing control of the medium from station to station in a logically circular fashion. The MAC sublayer determines when the station has the right to access the shared medium by recognizing and accepting the token from the predecessor station, and it determines when the token shall be passed to the successor station.

1.5.2 General Sublayer Functions

- (1) Lost token timer
- (2) Distributed initialization
- (3) Token holding timer
- (4) Limited data buffering
- (5) Node address recognition
- (6) Frame encapsulation (including token preparation)
- (7) Frame check sequence (FCS) generation and checking

Fig 1-2
Logical Ring on Physical Bus



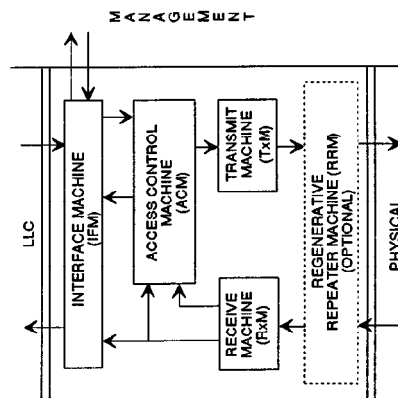
- (8) Valid token recognition
- (9) Ring member addition/deletion
- (10) Node failure error recovery

1.6 MAC Sublayer Internal Structure. The MAC sublayer performs several functions that are loosely coupled. The descriptions and specifications of the MAC sublayer in this standard are organized in terms of one of several possible partitions of these functions. The partitioning used here is illustrated in Fig 1-3, which shows five asynchronous logical "machines," each of which handles some of the MAC functions, as discussed in 1.6.1 through 1.6.5.

Of these five machines, the access control machine (ACM) is the most critical and the most complex; it is the key control mechanism for the token-passing bus access method, and it cooperates closely with ACMs in other stations given only limited information regarding the state of the network. Because of its importance, and because much of its operation is not easily inferred from its functional requirements, the explanation and specification of the ACM are the major concerns of Sections 5, 6, and 7.

The interface machine (IFM) and receive machine (RxM) participate heavily in the operation of the MAC sublayer protocol. They are, however, discussed only in sufficient detail so that the reader will understand their role in the MAC sublayer in support of the ACM. The level of discussion was chosen to avoid any ambiguities which might compromise the coexistence of conformant stations on a single bus.

Fig 1-3
MAC Sublayer Functional Partitioning



1.6.1 Interface Machine (IFM). This machine acts as an interface and buffer between the LLC and MAC sublayers and between network management and the MAC sublayer. It interprets all incoming MA-UNITDATA and other service primitives and generates appropriate outgoing service primitives. This machine handles the mapping of "quality of service" parameters from the LLC view to the MAC view, where this is necessary. It handles queuing of service requests, for example, requests to send an LLC protocol data unit (PDU). Finally, it performs the "address recognition" function on received data frames, accepting only those addressed to this station.

1.6.2 Access Control Machine (ACM). This machine cooperates with the ACMs of all other stations on the bus in handling the token to control transmission access to the shared bus. The ACM may (optionally) manage multiple MAC access classes so as to provide different levels of "quality of service" to the LLC sublayer. The ACM is also responsible for initialization and maintenance of the logical ring, including the admission of new stations. Finally, it has responsibility for the detection of and, where possible, recovery from faults and failures in the token-passing bus network.

1.6.3 Receive Machine (RxM). This machine accepts atomic symbol inputs from the Physical Layer and assembles them into frames which it validates and passes to the IFM and ACM. The RxM accomplishes this by recognizing the frame start and the frame end delimiters (SD and ED), checking the FCS and validating the frame's structure. The RxM also identifies and indicates the reception of noise bursts and the bus quiet condition.

1.6.4 Transmit Machine (TxM). This machine generally accepts a frame from the ACM and transmits it, as a sequence of atomic symbols in the proper format, to the PLE. The TxM builds a MAC protocol data unit by prefacing each frame with the required preamble and SD, and appending the FCS and ED. When operating with a Regenerative Repeater Machine (RRM), the TxM's operation may be somewhat different.

1.6.5 Regenerative Repeater Machine (RRM). This machine is an optional MAC component that is present only in special "repeater" stations, for example, a broadband remodulator. In such repeater stations, the RRM, when appropriate, repeats the incoming atomic symbol stream, from the PLE, back to the PLE for retransmission; in such cases the PLE is understood to be connected to at least two different segments of a single bus. The RxM and TxM may cooperate with the RRM in such repeating operations.

1.7 PLE and Medium. The broad outlines for the PLE sections of this standard and also an introduction to the various physical signaling techniques and media defined for LANs using the token-passing MAC protocol are given here.

Four bus media and corresponding PLEs are specified for use with the token-passing MAC protocol.

Each PLE and corresponding bus medium is described in a pair of consecutive sections (that is, Sections 12 and 13, Sections 14 and 15, Sections 16 and 17, and Sections 18 and 19), consisting of the following:

- (1) A section specifying a specific PLE, including how the generic management objects (Section 9) in that PLE are particularized, and
 - (2) A section specifying the medium appropriate to that PLE.
- Four different PLEs with corresponding media suitable for use with the token-passing bus MAC protocol are specified in this standard. They are distinguished primarily by the different forms of media and signaling specified for each type of PLE. The remainder of this subsection outlines the salient points of each type of PLE and corresponding medium.

1.7.1 Summary of Phase-Coherent Frequency Shift Keying (FSK)

Topology. Omnidirectional bus

Cable. 75 Ω coaxial cable, such as RG-11 type or semirigid

Connector at Station. 75 Ω female F-series, to be specified by IEC (see NOTES in 13.5.1)

Recommended Cable Configuration. CATV trunk of either RG-11 type or semirigid and flexible drop cables of up to 50 m in length

Trunk Connection Unit. 75 Ω nondirectional 20 dB passive impedance-matching tap

Repeaters. Active regenerative repeaters used for high-fanout branching and extension of the system beyond the limits of a single segment of the cable

Transmit Level. +63 to +66 dB (1 mV, 75 Ω) [dBmV]

Receiver Sensitivity. +10 dBmV to +66 dBmV

Data Rates. 5 Mb/s and 10 Mb/s

Signaling. Direct encoding of *data* and *non_data* symbols, each as an integral number of cycles of constant frequency, with frequency changes only at the zero crossings of the waveforms. Two frequencies are used:

- (1) The lower is 1 Hz/(b/s) (that is, 5 MHz at 5 Mb/s, 10 MHz at 10 Mb/s)
- (2) The higher is 2 Hz/(b/s) (that is, 10 MHz at 5 Mb/s, 20 MHz at 10 Mb/s)

The symbol representations are

zero — two full cycles of the higher frequency

one — one full cycle of the lower frequency

pairs of *non_data* — one full cycle of the higher frequency, one full cycle of the lower frequency, and another one full cycle of the higher frequency

Pad_idle. Alternating *one* and *zero* symbols, starting with *one*

Clock Recovery. From zero crossings in the received signal

Transmit Data Timing. Phase-locked to transmit frequencies. All ports of a regenerative repeater use the same transmit data timing and, hence, identical transmit frequencies.

1.7.2 Summary of Multilevel Duobinary Amplitude Modulation/Phase Shift Keying (AM/PSK)

Topology. Directional bus with active repeater

Cable. 75 Ω coaxial cable, such as RG-6 type or semirigid

Connector at Station. 75 Ω female F-series, to be specified by IEC (see NOTE in 15.5.1)

Recommended Cable Configuration. CATV-like semirigid trunk and flexible drop cable

Trunk Connection Unit. 75 Ω directional passive impedance-matching tap

Repeaters. Regenerative repeater used as system data-rate clock source, as central assessor of contention and noise, and to retransmit all signaling received on the directional medium

Amplifiers. Standard CATV bidirectional (or unidirectional for dual-cable configuration) broadband amplifiers used for extension of the system beyond the basic signal loss budget

Channel Bandwidths. 1.5 MHz, 6 MHz, and 12 MHz

Transmit Level

In 1.5 MHz bandwidth, +41 dB (1 mV, 75 Ω) [dBmV]

In 6 MHz bandwidth, +47 dB (1 mV, 75 Ω) [dBmV]

In 12 MHz bandwidth, +50 dB (1 mV, 75 Ω) [dBmV]

Receiver Sensitivity

In 1.5 MHz, -13 to +4 dB (1 mV, 75 Ω) [dBmV]

In 6 MHz, -7 to +10 dB (1 mV, 75 Ω) [dBmV]

In 12 MHz, -4 to +13 dB (1 mV, 75 Ω) [dBmV]

Data Rates. 1 Mb/s in 1.5 MHz, 5 Mb/s in 6 MHz, 10 Mb/s in 12 MHz

Channel Frequency Assignments. Channel frequency assignments are a subject for national standardization. The following North American recommended channel assignments are included for information only. They are not a part of this International Standard.

For 10 Mb/s, channels 3' and 4' (59.75–71.75 MHz) and P and Q (252–264 MHz)

For 5 Mb/s, channels 3' (59.75–65.75 MHz) and P (252–258 MHz), or 4' (65.75–71.75 MHz) and Q (258–264 MHz)

For 1 Mb/s, any of the eight equally spaced 1.5 MHz subchannels of reverse channels 3' and 4', paired with the corresponding forward channels P and Q

Scrambler. A self-sync scrambler with a generator polynomial of $1 + X^{-6} + X^{-7}$ is applied to all framed data prior to encoding for transmission, both to increase the average number of transitions within a transmission and to randomize spectral components of the transmitted modulation.

Signaling. *Data* and *non_data* symbols are encoded so as to specify the amplitude of the eventual modulation. A single form is defined in this standard: One MAC-symbol per PHY-symbol, together with compatibility considerations for an additional form described in 14.11.

In both forms of signaling, the middle level is used only to signal the *non_data* symbols found in frame delimiters and reported *silence*, and to break up long sequences of a single other signaling level (see 14.8.2.1(4)(C)). Such long sequences are unlikely since a scrambler is applied to framed data before it is encoded for transmission.

For one MAC-symbol per PHY-symbol signaling, the symbol representations at the receiver are

$\{0\}$ = zero — zero amplitude
 $\{4\}$ = one — "maximum" amplitude
 $\{2\}$ = *non_data* — "middle" amplitude of one half "maximum"

Modulation. Multilevel duobinary AM/PSK

Pad_idle. Alternating $\{4\}$ and $\{0\}$ symbols, starting with $\{4\}$

Reported Silence. The repeated sequence of symbols transmitted by the remodulator to report that it is not receiving any signaling. The sequence has a period of four symbols and the repetition may break off after any symbol of the sequence. Listening modems can set their automatic gain control (AGC), and can determine the signaling mode of the system by the sequence used.

For one MAC-symbol per PHY-symbol signaling, the sequence is:

$\{2\} \{2\} \{0\} \{4\}$

For a possible future two MAC-symbols per PHY-symbol signaling mode, the following sequence is reserved:

$\{2\} \{2\} \{4\} \{0\}$

Implementations that detect this reserved repeating sequence should either switch to the enhanced mode of operation or inhibit transmission.

Clock Recovery. From level transitions within the received signaling

Transmit Data Timing. Sourced by remodulator; frequency-locked to received data timing at all other stations

1.7.3 Summary of Fiber Optic Media

Topology. Directional bus using active or passive stars

Cable. The standard assumes a silica fiber optic waveguide with the following nominal characteristics: a core diameter of 62.5 μm , outside diameter of 125 μm , and an effective numerical aperture of 0.275.

NOTE: Operation using a 50 μm alternate test fiber is described in Appendices 16.10 and 17.9.

Connector at Station. The cable plant interface connector (CPIC) is a duplex connector as defined in ISO/IEC 9314-3 [16].

Repeaters. Active regenerative repeaters used for high-fanout star topologies

Transmit Characteristics. -7 to -11 dBm effective optical launch power with center wavelength between 800 and 910 nm

Receiver Sensitivity

Moderate sensitivity: -11 to -31 dBm effective power with silence level of -40 dBm

High sensitivity: -21 to -41 dBm effective power with silence level of -50 dBm

Data Rates. 5 Mb/s, 10 Mb/s, and 20 Mb/s

Signaling. Manchester encoding of *data* and *non_data* symbols

The symbol representations are

$\{H L\}$ = zero

$\{L H\}$ = one

$\{L L H H\}$ and $\{H H L L\}$ = pairs of *non_data* symbols

Pad_idle. Alternating *one* and *zero* symbols, starting with *one*

Clock Recovery. From transitions generated by the Manchester encoding

Transmit Data Timing. Phase-locked to transmit frequencies. All ports of a regenerative repeater use the same transmit data timing and, hence, identical transmit frequencies.

1.7.4 Summary of Phase-Continuous FSK

Topology. Omnidirectional bus

Trunk Cable. 75 Ω coaxial cable, such as types RG-6, RG-11, and semirigid

Drop Cable. 36-50 Ω coaxial cable stub, less than 350 mm long

Connector at Station. 50 Ω male BNC-series, see IEC 169-8 (1978) [4]

Recommended Cable Configuration. Long unbranched trunk cable with very short "stub" drop cables

Trunk Connection Unit. 75 Ω tee connector

Repeaters. Active regenerative repeaters used for branching and extension of the system beyond the basic signal loss budget

Transmit Level. +54 to +60 dB (1 mV, 37.5 Ω)

Receiver Sensitivity. +24 dB (1 mV, 37.5 Ω)

Data Rate. 1 Mb/s

Signaling. Manchester encoding of *data* and *non_data* symbols

The symbol representations are

$\{H L\}$ = zero — initial high, final low level

$\{L H\}$ = one — initial low, final high level

$\{L L H H\}$ and $\{H H L L\}$ = pairs of *non_data* symbols, initial pair of low level followed by a final pair of high level or vice versa

Modulation. Phase-continuous FSK (a form of frequency modulation), with Manchester representation:

- (1) Frequency of high level = 6.25 ± 0.08 MHz
- (2) Frequency of low level = 3.75 ± 0.08 MHz

Pad_idle. Alternating *one* and *zero* symbols, starting with *one*

Clock Recovery. From transitions generated by the Manchester encoding

1.7.5 Alternate Physical Layer and Medium for Industrial Control. Alternate Physical Layers and media for industrial control are specified in ANSI/ISA-S72.01-1986 [1]. These alternate Physical Layers and media are compatible at the MAC sublayer-Physical Layer interface.

1.8 Access Method Characteristics. An understanding of the basic characteristics of the token-passing access method is useful, so as to better understand where and when a token-passing bus is an appropriate LAN technology.

Some of the important features of this medium access method are as follows:

- (1) The method is efficient in the sense that, under high offered load, the coordination of the stations requires only a small percentage of the medium's capacity.
- (2) The method is fair in the sense that it offers each station an equal share of the medium's capacity. It does not, however, require any station to use its full share.
- (3) The method permits multiple classes of service.
- (4) The method coordinates the stations' transmissions so that they minimize and control their interference with each other.
- (5) The method imposes no additional requirements on the medium or the PLE's capabilities over those necessary for transmission and reception of multibit, multiframe sequences at the specified bit error rate.
- (6) In the absence of system noise, the method provides computable, deterministic, worst-case bounds on access delay for the highest priority class of service for any given network and loading configuration.
- (7) Periods of controlled interference are distinguishable; system noise measurements are possible during the remaining periods.
- (8) The method places minimal constraints on how a station that momentarily controls the medium may use its share of the medium's capacity.
- (9) The method efficiently supports the proposed LLC Type 3 Service by allowing the token holding station to wait for the receiving station to respond to the token holder's transmission.
- (10) Though not specified by this standard, the method permits the presence of large numbers of low-cost, reduced-function stations in the network, together with one or more full-function stations. (At least one full-function station is needed to make the system operational.) An example of a reduced function station is one that does not contain access control logic.

1.9 Standard Organization. This standard is organized in 19 sections, which are summarized below.

Section 1 (this section) begins with a general discussion of the token-passing bus access method. The MAC sublayer functional partitioning used in subsequent discussions is introduced here. The PLE and media options are also surveyed. Finally, features of the token-passing bus access method are reviewed.

Section 2 details the logical interfaces between the LLC sublayer and the MAC sublayer, and the services and command interfaces (such as sending a frame) provided to the LLC sublayer.

Section 3 details the management parameters, actions, and events within the MAC sublayer.

Section 4 details the general MAC frame structure, including delimiters, addressing, and the FCS. All of the frame formats that the MAC handles, including MAC control frames, are enumerated.

Section 5 discusses the basic concepts of the access protocol and provides an informal description of the actions in each state of the ACM. The other state machines of the MAC sublayer are also described in Section 5.

Section 6 contains definitions of essential MAC terms and components.

Section 7 specifies the MAC ACM by means of a state machine model. This is the definitive specification of the token-passing bus MAC operation. Section 7 also describes the MAC sublayer variables, functions, and procedures used in the state machine.

Section 8 details the logical interface between the MAC sublayer and the Physical Layer. Included here are descriptions of the interface symbols, requests, and responses.

Section 9 details the management parameters, actions, and events within the PLE.

Section 10 defines the logical, electrical, and mechanical interface within the PLE between the station and a separated modem.

Section 11 is reserved.

Sections 12 and 13 detail the PLE and medium, respectively, for a single-channel (that is, omnidirectional) phase-coherent-FSK coaxial cable bus at 5 Mb/s or 10 Mb/s.

Sections 14 and 15 detail the PLE and medium, respectively, for a dual-channel (that is, head-ended) broadband duobinary AM/PSK coaxial cable bus at 1 Mb/s, 5 Mb/s, or 10 Mb/s.

Sections 16 and 17 detail the PLE and medium, respectively, for a fiber optic bus at 5 Mb/s, 10 Mb/s, or 20 Mb/s.

Sections 18 and 19 detail the PLE and medium, respectively, for a single-channel (that is, omnidirectional) phase-continuous-FSK coaxial cable bus at 1 Mb/s.

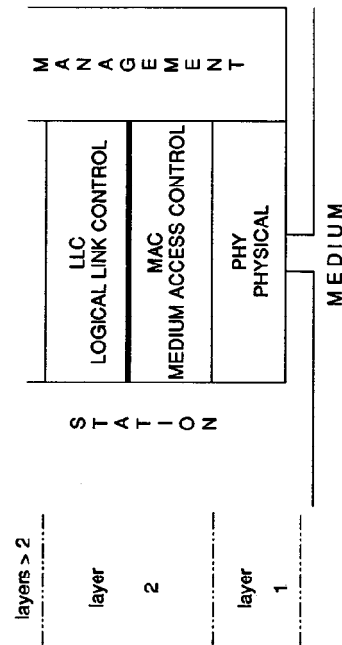
This section specifies the services provided to the Logical Link Control (LLC) sublayer, and to the alternative sublayer specified in ANSI/ISA-S72.01-1986 [1], at the boundary between the LLC functions and the Medium Access Control (MAC) sublayer of the Data Link Layer of the reference model. This standard specifies these services in an abstract way. It does not specify or constrain the implementation entities and interfaces within a computer system. The relationship of this section to other sections of this standard and to LAN specifications is illustrated in Fig 2-1.

NOTES: (1) The exact relationship of the layers described in this standard to the layers defined by the Open Systems Interconnection (OSI) Basic Reference Model (ISO 7498 : 1984 [13]) is for future study. (2) Work is in process to produce a single service specification that is common to all the MAC sublayers; see ISO/IEC 10039 [20]. When that standard is completed, approved, and published, Section 2 will be deleted.

2.1 Overview of the LLC-MAC Service

2.1.1 General Description of Services Provided. This section informally describes the services provided to the LLC sublayer by the token-passing MAC sublayer, both of the Data Link Layer. These services provide only connectionless data

Fig 2-1
Relation to LAN Model



transfer services between peer LLC entities. They provide the means by which LLC entities can exchange MAC service data units (M_SDU) without the establishment of an underlying point-to-point connection. The data transfer can be point-to-point or multipoint, unacknowledged or acknowledged.

2.1.2 Model Used for the Service Specification. The model and descriptive method are detailed in Appendix A.

2.1.3 Overview of Interactions. The primitives associated with this connectionless data transfer service are

- (1) MA-UNITDATA request
- (2) MA-UNITDATA indication
- (3) MA-UNITDATA confirm

An MA-UNITDATA request primitive is passed to the MAC sublayer to request that an M_SDU be sent. (All M_SDUs are sent using connectionless procedures.) An MA-UNITDATA indication primitive is passed from the MAC sublayer to indicate the arrival of an M_SDU. An MA-UNITDATA confirm primitive is passed from the MAC sublayer to indicate the status of the previous associated MA-UNITDATA request primitive.

2.1.4 Basic Services and Options. All services are mandatory and are required in all implementations.

2.2 Detailed Interactions with the LLC Entity. This subsection describes in detail the primitives and parameters associated with the connectionless data transfer services provided to LLC by MAC. Note that the parameters are specified in an abstract sense. The parameters specify the information that shall be available to the receiving entity. A specific implementation is not constrained in the method of making this information available. For example, the M_SDU parameter associated with some of the data transfer service primitives may be provided by actually passing the MAC service data unit, by passing a descriptor, or by other means. The values of some selection parameters may be implied by an implementation.

2.2.1 MA-UNITDATA request

2.2.1.1 Function. This primitive is the service request primitive for the connectionless data transfer service.

2.2.1.2 Semantics. The primitive shall provide parameters as follows:

MA-UNITDATA request (destination_address,
source_address,
M_SDU,
desired_quality)

The destination_address parameter specifies either an individual or a group MAC-entity address. The source_address parameter specifies the originating MAC-entity address, typically the local station. The M_SDU parameter specifies the MAC service data unit to be transmitted by the MAC sublayer entity for the requesting LLC sublayer entity.

The desired_quality parameter specifies the desired quality of service. The semantics of this parameter include both MAC-level priority, with a range of 0 (lowest) to 7 (highest) (see 6.6.1.2), and MAC-level delivery confirmation service, with values of request_with_no_response, request_with_response, and response.

2.2.1.3 When Generated. This primitive is passed from the LLC sublayer entity to the MAC sublayer entity to request that the MAC sublayer entity compose and transmit the specified frame at the desired quality of service on the LAN.

2.2.1.4 Effect on Receipt. Receipt of this primitive causes the MAC sublayer entity to attempt to compose and transmit the specified frame.

2.2.1.5 Additional Comments. A value of request_with_response for the delivery-confirmation component of the quality parameter indicates that the next MA-UNITDATA indication should itself have a quality parameter specifying response, in which case that next MA-UNITDATA indication shall be associated with this MA-UNITDATA request.

A value of response for the delivery-confirmation component of the quality parameter indicates that the immediately prior MA-UNITDATA indication shall itself have had a quality parameter specifying request_with_response.

A group destination_address should not be used when specifying request_with_response.

2.2.2 MA-UNITDATA indication

2.2.2.1 Function. This primitive is the service indication primitive for the connectionless data transfer service.

2.2.2.2 Semantics. The primitive shall provide parameters as follows:

MA-UNITDATA indication (destination_address,
source_address,
M_SDU,
quality)

The destination_address and source_address parameters specify the DA and SA fields of a frame (see Section 4) as received by the local MAC entity, and thus the MAC entities purportedly involved in the communication. The M_SDU parameter specifies the MAC service data unit as received by the local MAC sublayer entity.

The quality parameter specifies the delivered quality of service. The semantics of this parameter include both MAC-level priority, with a range of 0 (lowest) to 7 (highest) (see 6.6.1.2), and MAC-level delivery-confirmation service, with values of request_with_no_response, request_with_response, and response.

2.2.2.3 When Generated. This primitive is passed from the MAC sublayer entity to the LLC sublayer entity to indicate the arrival of a data frame from the Physical Layer entity (PLE). Such frames are reported only when they are free of detected errors and their (individual or group) destination address designates the local MAC entity.

2.2.2.4 Effect on Receipt. The effect of receipt of this primitive by the LLC entity is specified in ISO 8802-2 : 1989 (IEEE Std 802.2-1989) [15].

2.2.2.5 Additional Comments. In the absence of undetected errors, the con-

tents of the M_SDU parameter are logically complete and unchanged relative to the M_SDU parameter in the associated MA-UNITDATA request at the sending station.

NOTE: This is a guarantee of transparency.

A value of request_with_response for the delivery-confirmation component of the quality parameter indicates that the receiving LLC sublayer entity should immediately respond with an MA-UNITDATA request which itself has a quality parameter specifying response.

A value of response for the delivery-confirmation component of the quality parameter indicates that this MA-UNITDATA indication may be associated with a prior MA-UNITDATA request which itself had a quality parameter specifying request_with_response, and which was issued by the same LLC sublayer entity.

2.2.3 MA-UNITDATA confirm

2.2.3.1 Function. This primitive provides the LLC sublayer with status information for the previous associated MA-UNITDATA request primitive.

2.2.3.2 Semantics. The primitive shall provide parameters as follows:

```
MA-UNITDATA confirm (
    destination_address,
    source_address,
    status,
    provided_quality
)
```

The destination_address and source_address parameters specify the destination address and source address fields of the associated MA-UNITDATA request primitive. The status parameter indicates the status of the service provided for a previous associated MA-UNITDATA request primitive.

The provided_quality parameter specifies the quality of service actually provided for the previous request. The semantics of this parameter include both MAC-level priority, with a range of 0 (lowest) to 7 (highest) (see 6.6.1.2), and MAC-level delivery confirmation service.

2.2.3.3 When Generated. This primitive is passed from the MAC sublayer entity to the LLC sublayer entity to indicate the status of the service provided for the previous associated LLC data transfer request.

A success indication is passed when the request has been completed successfully, either because no response was requested or because the requested response was received. A failure indication is passed upon local failure or when a requested response is not received, even after the specified number of retries. When the quality parameter of that request specifies request_with_response, a failure indication is also passed if the allowed number of retries has occurred with no response.

2.2.3.4 Effect on Receipt. The effect of receipt of this primitive by the LLC sublayer entity is specified in ISO 8802-2 : 1989 (IEEE Std 802.2-1989) [15].

2.2.3.5 Additional Comments. It is assumed that sufficient information is available to the LLC sublayer to associate the status with the appropriate request.

3. MAC Sublayer Management

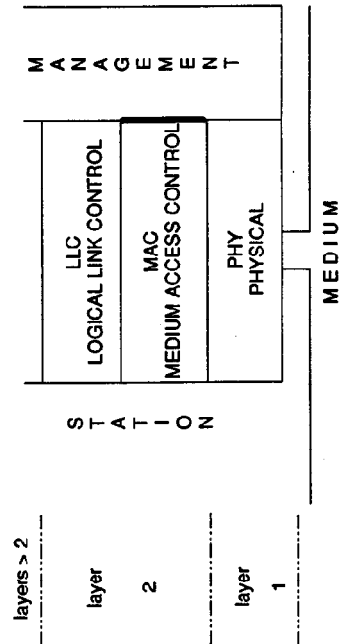
This section specifies the abstract entities (parameters, events, and actions) used in managing the Medium Access Control (MAC) sublayer. The relationship of this section to other sections of this standard and to Local Area Network (LAN) specifications is illustrated in Fig 3-1.

There are two ways to manage a MAC sublayer: locally or remotely.

Local management is required to initially configure a MAC and to operate it when remote management is not available. Since the operation of local MAC management, other than the need to manage the entities described in this section, does not affect station interoperability, the specification of local management is outside the scope of this standard.

This standard uses the concept of remote management as defined in the Open Systems Interconnection (OSI) Management Framework described in ISO/IEC 7488-4 : 1989 [14]. Remote management uses a communications protocol to manipulate and observe the managed entities. This section is designed to be used with the companion remote management standard described by IEEE P802.1B [9].

Fig 3-1
Relation to LAN Model



Other protocols, such as described by ISO/IEC 9595 [18] and ISO/IEC 9596 [19], may also be used.

To implement a remote management protocol, the identification and value representation of the managed entities must be defined. Such a definition is given by a layer management interface (LMI) protocol.

NOTE: The definition of a MAC sublayer LMI protocol is under study. An addendum to this standard is planned that will contain this protocol.

Within this document the general term "management" is used to refer to either local or remote management, unless otherwise indicated.

3.1 Overview. This section describes the entities communicable via the LMI, shown as a thick line in Fig 3-1. The MAC sublayer entities described in this section consist of

- (1) Parameters within the MAC sublayer written and read by the management entity,
- (2) Actions initiated by the management entity that cause changes within the MAC sublayer,
- (3) Events within the MAC sublayer entity that are passed to the management entity.

The notation used to specify the entities defined in this section is the Abstract Transfer Syntax and Notation (ASN.1), defined in ISO 8824 : 1987 [16]. This syntax differs from that used in Sections 4 through 7 to describe the token bus MAC. The difference is in the spelling of the entity names. The ASN.1 notation does not allow the underscore character, "-" and uses a combination of uppercase and lowercase characters to improve readability.

For example, this section defines a parameter *inRing*. Section 7 defines a parameter *inRing*. While the syntax of the names is different, the parameters are equivalent.

3.2 MAC Management Facilities. This subsection contains a description of use of the management parameters, actions, and events within the token-passing bus MAC sublayer.

3.2.1 Organization. There are three types of management entities:

- (1) Parameters
- (2) Actions
- (3) Events

The MAC sublayer parameters accessible by the management entity are described first. These parameters are organized into groups. The groups not only collect the variables, they also define collections of parameters that can be accessed in a common manner.

NOTE: In implementing control of access to parameters, the intent of this standard is to implement the granularity of access control only to the group level. Thus, all parameters within a group share a common access level.

Any or all of the parameters in a group may be passed in a single remote management request. The semantics and grouping of the parameters so communicated are specified in the following subsections.

3.2.2 ResourceTypeID. This subsection discusses the parameters of the MAC sublayer LMI that are common to any layer. These parameters create a "name-plate" that identifies the basic attributes of the MAC sublayer.

The ResourceTypeID is common to all MAC sublayers. Implementation of this group shall be required. The ResourceTypeID group is defined in IEEE P802.1B [9].

3.2.2.1 resourceType. This parameter identifies the resource as a MAC sublayer.

3.2.2.2 standardRevision. This parameter identifies the revision of this standard implemented by the MAC sublayer.

3.2.2.3 lmeOptions. This parameter identifies the options supported by the implementation. The options defined by this standard are priority and request_ with _response.

3.2.3 MAC Characteristics Group. The MAC characteristics group defines attributes of the MAC sublayer that are specific to a token-passing bus MAC sublayer. The implementation of the parameters in the MAC characteristics group shall be optional except for maxSDUSize, which is required. All parameters shall be read-only.

3.2.3.1 maxSDUSize. The maximum size service data unit that the sublayer implementation is capable of accepting.

The value of maxSDUSize is between 516 (to allow 512 octet frames) and 8191 octets.

3.2.3.2 transmitTimeDelay. The estimated maximum delay incurred by the MAC in sending a data frame, including the time in the MAC transmit queue. The value of this parameter is for use by higher layers in computing a time-out delay to use in detecting lost frames. The type is an array of time values indexed by MAC service class (see 6.6.1.2).

The units of this parameter and methods of computing the values are for future study.

3.2.4 MAC Management Group. The MAC management parameters describe the current state of the MAC sublayer. The implementation of access by the management entity to these parameters shall be optional. Access by the management entity to these parameters shall be read-only to prevent unpredictable changes to MAC operation. In order to change the value of one of these parameters, the management entity must change the corresponding value in the initialization group (see 3.2.7) and then perform the reset action to force the station to reinitialize.

3.2.4.1 state. MAC sublayer state. The current state of the MAC entity. The MAC state is changed by management actions (see 3.2.1.1) and events within the MAC access control machine (ACM) (see Section 7).

The range of values of the state parameter is offline, tokenBus, tokenBusBridge, and tokenBusRepeater.

NOTE: The operation of a station in the token-passing bus bridge mode is for future study. Bridge operation is envisioned to include acceptance of frames addressed to stations on networks other than the one to which the station is connected.

3.2.4.2 testStatus. The status of a previously initiated test. Negative, implementation-dependent values shall be treated as "failure" by entities that lack further information on the implementation-dependent meaning.
See 7.1.1 for a detailed description of the following parameters of the MAC management group.

3.2.4.3 ts. This station's address.

3.2.4.4 slotTime. The network slot_time.

3.2.4.5 minPostSilencePreambleLength. The minimum number of octets of preamble to prefix to a frame after silence. The value is obtained from the PLE and used by the transmit machine (TxM).

3.2.5 Station Performance Group. These parameters affect the operation of the station and the station's performance in the network. Management may read and write the station performance parameters in order to alter network operation without having to re-initialize the station. The implementation of access by the management entity to these parameters shall be optional.
See 7.1.1 for a detailed description of the parameters of the station performance group.

3.2.5.1 maxInterSolicitCount

3.2.5.2 maxRetryLimit

3.2.5.3 hiPrTokenHoldTime

3.2.5.4 targetRotationTime

3.2.5.5 ringMaintenanceTime:InitialValue

3.2.5.6 inRingDesired

3.2.6 Station Monitoring Group. These parameters reflect the current status of the network and the station's status with respect to the network.

The implementation of access by the management entity to these parameters shall be optional. The station monitoring parameters shall be read-only with the exception of maxTokenRotationTime, which shall be read-write.

See 7.1.6 for a description of the parameters of the station monitoring group not discussed in the following paragraphs.

3.2.6.1 ns. The address of the next station in the logical ring.

3.2.6.2 ps. The address of the previous station in the logical ring.

3.2.6.3 nsKnown

3.2.6.4 inRing

3.2.6.5 soleActiveStation

3.2.6.6 lastTokenRotationTime. The observed token rotation time for the last token rotation, timed from token arrival to token arrival. Units are octet times.

A value of zero indicates that the value is not currently available; for example, possibly when the station is not a member of the logical ring. A negative value indicates a range overflow.

3.2.6.7 maxTokenRotationTime. The maximum token rotation time. The value is created by comparing the old value of maxTokenRotationTime with the value of lastTokenRotationTime whenever the token is received. If the old maxTokenRotationTime value is less than the lastTokenRotationTime, the value of maxTokenRotationTime is set equal to the value of lastTokenRotationTime. Units are octet times.

The value of this parameter may be set to zero by the management entity to begin a time interval over which the maximum token rotation time is to be measured.
A value of zero indicates that the value is not currently available. A negative value indicates a range overflow.

The creation of other statistics from the maximum and instantaneous token rotation time variables is for future study.

3.2.6.8 approximateRingSize. The approximate number of stations in the logical ring.

A possible method of obtaining the value of approximateRingSize is for the station to count the number of valid token frames that it hears between sending and receiving the token. Such a simplistic approach is prone to error when tokens are repeated or heard in error. Implementors can improve the approximation with more sophisticated techniques if desired.

A value of zero means the value is currently unavailable, for example, possibly when the station is not a member of the logical ring. A negative value indicates a range overflow.

3.2.7 Initialization Group. The initialization parameters are copied to the MAC sublayer operational parameters when the station is initialized. The initialization parameters shall be required in implementations where the corresponding operational parameters exist. The initialization parameters shall be readable and writable by the management entity.

See 7.1.7 for a detailed description of the parameters of the initialization group.

3.2.7.1 initTS

3.2.7.2 initSlotTime

3.2.7.3 initMaxInterSolicitCount

3.2.7.4 initMaxRetryLimit

3.2.7.5 initHiPrTokenHoldTime

3.2.7.6 initTargetRotationTime

3.2.7.7 initRingMaintenanceTime:InitialValue

3.2.7.8 initInRingDesired

3.2.8 Interface Machine (IFM) Counters Group. These parameters monitor the operation of the IFM. In counting frame length, octets in a frame are counted between the start delimiter and the end delimiter, exclusively. Service class to access class correspondence is specified in 6.6.1.2.

The implementation of the IFM counters and access by the management entity shall be optional. Access by the management entity to these parameters shall be read-only.

3.2.8.1 messagesSent. The number of data frames sent at the corresponding access class.

3.2.8.2 octetsSent. The number of octets in data frames sent at the corresponding access class, including control fields and excluding delimiters.

3.2.8.3 messagesReceived. The number of valid data frames received and accepted at the corresponding access class.

3.2.8.4 octetsReceived. The number of octets of valid data frames received and accepted at the corresponding access class, including control fields and excluding delimiters.

NOTES: (1) This standard does not include a count of frames discarded at the MAC-LLC interface due to lack of buffer space. The presumption is that the LLC sublayer is responsible for buffer management and discarding frames for lack of space.

(2) It is possible to have underruns while transmitting frames and/or overruns while receiving frames. Implementations may have counters that record the occurrence of these events. It is recommended that these counters be included in private management parameters of the IFM counters group.

3.2.9 Access Control Machine (ACM) Counters Group. These parameters monitor the operation of the ACM. The implementation of the ACM counters and access by the management entity shall be optional.

Implementation of a threshold facility associated with each counter shall be treated as a separate option. Thus, one of three conditions exist: No implementation, 32-bit counters without threshold, or 32-bit counters with threshold. If implemented, access by the management entity to these parameters shall be read-only, except for the portions associated with the threshold mechanism. (The threshold facility can be used to generate an event to the management entity when the value of a counter exceeds a specified value.)

NOTE: The condition described in 6.7.4 can cause the token_pass_failure, who_follows_query, and solicit_any counters to increment during correct operation of the network.

3.2.9.1 whoFollowsQuery. The number of occurrences of the ACM who_follows_query state transition. (See 7.2.3.8.)

3.2.9.2 tokenPassFailed. The number of occurrences of the ACM token_pass_failed state transition when pass_state equals pass_token.

3.2.9.3 solicitAny. The number of occurrences of the ACM solicit_any state transition not due to the ACM do_solicit_any state transition.

3.2.9.4 noSuccessor. The number of occurrences of the ACM no_successor_8 state transition.

3.2.9.5 unexpectedFrame. The number of occurrences of the ACM unexpected_frame_6 and unexpected_frame_10 state transitions.

3.2.9.6 claimToken. The number of occurrences of the ACM no_token state transition.

3.2.10 Receiver Machine (RxM) Counters Group. These parameters monitor the operation of the RxM. The implementation of the RxM counters and access by the management entity shall be optional. Thus the implementation of the MAC variable noise_expected discussed in the next paragraph, shall also be optional. Access to the RxM counters shall be read-only, except for the portions associated with the optional threshold mechanism.

The MAC variable noise_expected is used only in the definition of the RxM counters and is not described elsewhere in this standard. The noise_expected variable attempts to track the state of the network so that errors reported by the modem can be classified.

Errors reported by the modem may actually be induced by "collisions" when multiple stations transmit during times when contention can occur. By tracking the state of the network, the ACM attempts to predict the times of possible contention. The variable noise_expected conveys this information.

To simplify the explanation, the ACM state tables in 7.2 do not include the variable noise_expected. If required to implement the counters defined in this subsection, noise_expected shall be reset whenever the ACM

- (1) Receives or sends a valid token frame
- (2) Receives or sends a valid data frame

The noise_expected variable shall be set whenever the ACM

- (a) Traverses the initialize arc
- (b) Receives or sends a solicit_successor_1 frame
- (c) Receives or sends a solicit_successor_2 frame
- (d) Receives or sends a resolve_contention frame
- (e) Receives or sends a who_follows frame
- (f) Receives or sends a claim_token frame

Implementation of a threshold facility associated with each counter except validFrames shall be treated as a separate option. Thus, one of three conditions exist: No implementation, simple 32-bit counters, or thresholded 32-bit counters. If implemented, access by the management entity to these parameters shall be read-only, except for the portions associated with the threshold mechanism.

3.2.10.1 validFrames. The number of valid frames received of all types, including MAC control frames. (See 4.1.)

3.2.10.2 modemErrors. The number of noise bursts that occur when noise_expected is reset. modemErrors is a count of noise bursts that are probably caused by errors in the PLE or as the result of noise introduced on the medium.

The following counters are defined in order to aid in diagnosing the cause of errors reported in modemErrors. The value of modemErrors is the sum of the values of the following counters:

- (1) **fecErrors.** The number of frames received with incorrect frame check sequence (FCS) and E-bit reset when noise_expected is reset. (This value represents a count of FCS errors not previously detected by a remodulator or other repeater.)
- (2) **cBitErrors.** The number of frames received with the E-bit set and the FCS invalid when noise_expected is reset. (This value represents a count of FCS errors previously detected by a remodulator or other repeater.)
- (3) **nonSilence.** The number of occurrences of non-silence followed by silence in which a start delimiter was not detected when noise_expected is reset. (This error is possibly caused by a noise "hit" on the start delimiter.)
- (4) **frameFragments.** The number of occurrences of a start delimiter followed by a start delimiter, invalid symbol sequence, or silence without an intervening end delimiter. frameFragments is only incremented when noise_expected is reset. (This error is probably caused by premature termination of a frame by an abort sequence.)

The treatment of frames that are received with the correct FCS and the E-bit set is not specified by this standard. However, if such frames are discarded by the MAC as invalid, the count of such discarded frames shall be included in cBitErrors.

3.2.11 Action Definitions. Actions are requests by the management entity to perform some activity. The reset and initialize with a value of initializeTokenBus actions shall be required locally. All other actions shall be optional.

3.2.11.1 reset Action. This action causes the MAC sublayer to go to the OFFLINE state. The MAC sublayer shall remain in this state until a local initialize action occurs. A remote initialize action shall be ignored following a remote reset action.

The action may be used to disable the MAC sublayer if it is suspected of malfunctioning.

3.2.11.2 initialize Action. This action resets the MAC sublayer and then causes the station's ACM to traverse the initialize arc and enter the IDLE state in accordance with the action request parameter. A remote initialize action shall be ignored if the MAC sublayer is already in the OFFLINE state. The action request parameter may assume the values tokenBus, tokenBusBridge, or tokenBusRepeater.

Following an initialize action, the values of the counters specified in 3.2.8 through 3.2.10 are unspecified.

3.2.11.3 initiateTest Action. The initiateTest action commences an internal test of the MAC sublayer if the station is in the OFFLINE state. The result of the most recently initiated test is reported in the parameter testStatus. (See 3.2.4.2.)

The effect of the initiateTest action when the station is not in the OFFLINE state is for future study.

3.2.12 Event Definitions. Events are spontaneously reported by the sublayer to the management entity. Remote reporting of events shall be optional.

Local support of the thresholdReached event is required when thresholded counters are implemented. Support of the newSuccessor event shall be optional. Passing of the new successor's address shall be optional. Support of the noSuccessor event shall be optional. Local support of the duplicateAddressFault and faultyTransmitterFault events shall be required.

NOTE: In implementing routing of events, the intent of this standard is to provide a unique routing path for each event.

3.2.12.1 thresholdReached. A thresholded counter has passed the threshold value. The event value shall be the counter's parameter identifier.

3.2.12.2 newSuccessor. The station's successor has changed. The event value shall be the new successor address.

3.2.12.3 noSuccessor. The station now considers itself the sole member or not a member of the ring.

3.2.12.4 duplicateAddressFault. The station has detected another station with the same MAC address. The state transition causing this event is duplicate_address_1. (See 6.7.1 for possible additional detection methods.)

3.2.12.5 faultyTransmitterFault. The station has decided that it may have a faulty transmitter. The state transitions associated with such a decision are

- (1) end_all_contention
- (2) no_future

4. Frame Formats

This section defines the required Medium Access Control (MAC) frame formats. This includes all allowed frame formats and the arrangement of all frame subfields. The term frame as used here refers to the protocol data units (PDUs) exchanged by MAC sublayer entities. The MAC service data units received from the Logical Link Control (LLC) sublayer are contained within some of these MAC frames. The frame components and formats used by medium access control are also described in this section.

The MAC frames and abort sequences are described in the following subsections. First, the components of the frames are discussed, followed by the definition of the valid frame formats. All frames sent or received by the MAC sublayer shall conform to the following general format:

PREAMELE	SD	FC	DA	SA	DATA_UNIT . . .	FCS	ED
----------	----	----	----	----	-----------------	-----	----

where

PREAMELE = pattern sent to set receiver's modem clock and level
(1 or more octets)

SD = start delimiter (1 octet)

FC = frame control (1 octet)

DA = destination address (2 or 6 octets)

SA = source address (2 or 6 octets)

DATA_UNIT = information (0 or more octets)

FCS = frame check sequence (4 octets)

ED = end delimiter (1 octet)

The number of octets between SD and ED, exclusive, of a data frame (see 4.1.3.2) shall be 8191 octets or fewer.

The abort sequence shall conform to the following format:

SD	ED
----	----

where

SD = start delimiter (1 octet)

ED = end delimiter (1 octet)

Within this section the following acronyms are used for the addresses of the station under discussion, its successor and its predecessor in the logical ring:

- TS — this station's address
- NS — next station's address
- PS — previous station's address

4.1 Frame Components. This subsection describes the frame components that are shown in the previous illustrations in greater detail.

A valid frame consists of preamble, a start delimiter, a frame control octet, destination and source addresses, possibly a data_unit, a valid FCS, and an end delimiter, and does not meet any of the criteria of an invalid frame as described in 4.2.3. This definition excludes abort sequences from the set of valid frames.

4.1.1 Preamble. The preamble pattern precedes every transmitted frame. Preamble is sent by the MAC sublayer entity as an appropriate number of *pad_idle* symbols. Preamble may be decoded by the receiver as arbitrary data symbols that occur outside frame delimiters. Preamble is primarily used by the receiving modem to acquire signal level and phase lock by using a known pattern. The preamble pattern is chosen for each modulation scheme and data rate for this purpose. The parameter *min_post_silence_preamble_length* specifies the minimum amount of preamble on the first frame transmitted after a period of "transmitted" *silence*. See 7.1.1 for use of the parameter within the MAC sublayer. See 12.6, 14.7, 16.6, and 18.6 for the specification of the different physical layers.

A secondary purpose for the preamble is to guarantee a minimum ED to SD time period to allow stations to process the frame previously received. The minimum amount of preamble transmitted is a function of both the data rate and the modulation scheme. The standard requires that the duration of the preamble shall be at least 2 μ s, regardless of data rate, and that an integral number of octets shall be sent. Thus, at a data rate of 1 Mb/s, one octet of preamble is required to meet the integral number of octet requirements, and at a data rate of 10 Mb/s, three octets are required to meet the minimum time requirement.

The maximum amount of preamble is constrained by the "jabber" control in the PLE. Additionally, for claim_token frames, all stations shall use the minimum number of preamble octets to ensure that all frames are of uniform specified length.

4.1.2 Start Delimiter (SD). The frame structure requires an SD, which begins the frame. The SD consists of signaling patterns that are always distinguishable from data.

The SD is coded as follows (see 12.7, 14.8, 14.11, 16.7, and 18.7 for representations of the symbol coding as present on the medium):

start delimiter (SD):

first MAC-symbol transmitted

N

N

N

N

O

O

O

O

1 2 3 4 5 6 7 8 ← bit positions

where

- N = non_data MAC-symbol
- 0 = zero MAC-symbol
- 1 = one MAC-symbol

4.1.3 Frame Control (FC) Field. The FC octet determines what class of frame is being sent from among the following general categories:

- (1) MAC control
- (2) LLC data

The FC format for each of these categories is illustrated by the following:

4.1.3.1 MAC Control Frame

first MAC-symbol transmitted

0	0	C	C	C	C	C	C
---	---	---	---	---	---	---	---

1 2 3 4 5 6 7 8 ← bit positions

where

CCCCC = type of MAC_control frame as follows:

C C C C C

3 4 5 6 7 8 ← bit positions

- 0 0 0 0 0 claim_token
- 0 0 0 0 1 solicit_successor_1 (has 1 response window)
- 0 0 0 1 0 solicit_successor_2 (has 2 response windows)
- 0 0 0 1 1 who_follows (has 3 response windows)
- 0 0 1 0 0 resolve_contention (has 4 response windows)
- 0 0 1 0 0 token
- 0 0 1 1 0 set_successor

4.1.3.2 Data frames

first MAC-symbol transmitted

F	F	M	M	M	P	P	P
---	---	---	---	---	---	---	---

1 2 3 4 5 6 7 8 ← bit positions

where

F F = frame type:

(1 2 ← bit positions)

- 0 1 = LLC_data_frame
- 1 0 = reserved (former management)
- 1 1 = reserved (further study)

M M M = MAC-action:
(3 4 5 ← bit positions)

0 0 0 = request with no response
0 0 1 = request with response (see 6.6.2)
0 1 0 = response (see 6.6.2)

P P P = priority:
(6 7 8 ← bit positions)

1 1 1 = highest priority
1 1 0
1 0 1
1 0 0
0 1 1
0 1 0
0 0 1
0 0 0 = lowest priority

NOTE: The use of the "10" in the frame type field to indicate frames destined for the management entity is not recommended and is included only to retain backwards compatibility with previous versions of this standard.

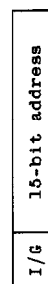
Other bit patterns in the FC octet are reserved for future study. The action of a station upon receiving an FC value not defined in this standard is not specified.

4.1.4 Address Fields. Each frame shall contain two address fields: The destination address field and the source address field, in that order. Addresses shall be either 16 bits or 48 bits in length. All addresses on a given Local Area Network (LAN) shall be of the same length.

4.1.4.1 Destination Address Field. The following illustration shows the possible representations of destination addresses:

(1) 16-bit address form

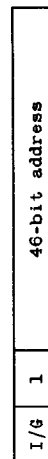
— first MAC-symbol transmitted



— most significant bit of address

(2) 48-bit locally administered form

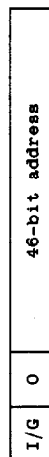
— first MAC-symbol transmitted



— most significant bit of address

(3) 48-bit universally administered form

— first MAC-symbol transmitted



— most significant bit of address

where

I/G = individual/group address bit

The first MAC-symbol transmitted of the destination address (the I/G bit) distinguishes individual addresses from group addresses:

0 = individual address
1 = group address

For 48-bit addresses, the second MAC-symbol transmitted of source or destination addresses (the local/universal or L/U bit) distinguishes locally administered addresses from universally administered, unique addresses.

Individual Addresses. An individual address identifies a particular station on the LAN and shall be distinct from all other individual station addresses on the same LAN.

Group Addresses. A group address is used to address a frame to multiple destination stations. Group addresses may be associated with zero, one, or more stations on a given LAN. In particular, a group address is an address associated by convention with a group of logically related stations.

Broadcast Addresses. The group address consisting of all ones (that is, 16 or 48 ones for two- or six-octet addressing, respectively) shall constitute a broadcast address, denoting the set of all stations on the given LAN.

NOTE: For some of the frame types used by the token-passing bus MAC procedures, the contents of the destination address field are irrelevant. In such cases, the originating station's own address or any other individual addresses may be sent in this field.

Address Administration (48-bit addresses only). There are two methods of administering the set of 48-bit station addresses: locally or through a universal authority. The second bit transmitted of the destination address indicates whether the address has been assigned by a universal or local administrator.

0 = universally administered
1 = locally administered

Universal Administration. With this method, all individual addresses are distinct from the individual addresses of all other LAN stations on a universal basis. The procedure for administration of these addresses is not specified in this standard.

NOTE: Information concerning the Registration Authority and its procedures may be obtained on request to the Secretary General, ISO Central Secretariat, 1 rue de Varembe, Case postale 56, CH1211, Geneva, Switzerland/Suisse, quoting the number of this International Standard.

For information on universal address administration contact the Secretary, IEEE Standards Board, Institute of Electrical and Electronics Engineers, Inc., P.O. Box 1331, 445 Hoes Lane, Piscataway, NJ 08855-1331, USA.

Local Administration. Individual station addresses are administered by a local (to the LAN) authority. (This is the only method allowed for 16-bit addresses.)

NOTE: Appendix 4.3 contains a suggested method for hierarchical structuring of locally administered addresses.

4.1.4.2 Source Address Field. The source address identifies the station originating the frame and has the same format and length as the destination address in a given frame, except that the I/G bit shall be set to 0; the significance of it being set to 1 is a subject for future study.

4.1.4.3 Numerical Interpretation of Addresses. Strictly speaking, addresses are bit strings which serve as unique station identifiers or group identifiers. For the purpose of the MAC address comparison within the token bus MAC sublayer, as used in ordering the logical ring and as expressed in the formal access control machine of 7.2.3, each MAC-address bit string is interpreted as if it were an unsigned integer value sent least significant bit first, and thus as if the last bit transmitted had the highest numeric significance.

NOTE: This interpretation does not extend beyond the logical ring ordering operations of the token-bus MAC sublayer.

Additionally, the address bits are used in determining delays in the contention process and transmission lengths in the token claiming process. These processes start with the most significant address bits, under the above interpretation, using two bits at a time. Thus the internal processing order is reversed from the serial transmission order on the medium.

4.1.5 MAC Data Unit Field. Depending on the bit pattern specified in the frame's FC octet, the MAC data unit field can contain either an LLC protocol data unit as specified in ISO 8802-2 : 1989 (IEEE Std 802.2-1989) [15], or a value specific to one of the MAC control frames.

In the case where the MAC data unit field contains a LLC PDU, the MAC data unit field shall be transmitted to the Physical Layer in the same bit order as received from the LLC sublayer. Likewise, the MAC data unit field shall be delivered to the LLC sublayer in the same bit order as received from the Physical Layer.

4.1.6 Frame Check Sequence (FCS) Field. The FCS is a 32-bit frame checking sequence, based upon the following standard generator polynomial of degree 32:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^6 + X^4 + X^2 + X + 1$$

The FCS is the one's complement of the sum (modulo 2) of

$$(1) \text{ The remainder of } X^K \cdot (X^{31} + X^{30} + X^{29} + \dots + X^2 + X + 1)$$

divided (modulo 2) by the standard 32-bit generating polynomial, where K is the number of bits in the FC, address (SA and DA), and MAC data unit fields, and

- (2) The remainder of the division (modulo 2) by the standard generator polynomial of the product of X^{32} by the content of the FC, address (SA and DA), and MAC data unit fields.

The FCS is transmitted commencing with the coefficient of the highest degree term.

As a typical implementation, at the transmitter, the initial content of the register of the device computing the remainder of the division is preset to all ones and is then modified by the generator polynomial (as described above) on the FC, address, and data unit fields. The one's complement of the resulting remainder is transmitted as the 32-bit FCS.

At the receiver, the initial content of the register of the device computing the remainder is preset to all ones. The incoming serial data bits and the FCS, when divided by the generator polynomial, result, in the absence of transmission errors, in a unique non-zero remainder value. The unique remainder value for the 32-bit FCS is the polynomial:

$$X^{31} + X^{30} + X^{26} + X^{25} + X^{24} + X^{19} + X^{15} + X^{14} + X^{12} + X^{11} + X^{10} + X^8 + X^6 + X^5 + X^4 + X^3 + X + 1$$

NOTES: (1) To test the FCS generation and checking logic in a station, an implementation should provide a means of bypassing the FCS generation circuitry and providing an FCS from an external source. The ability to pass frames that have FCS errors along with the received FCS value and an error indication, to higher levels of the protocol, is another desirable testability feature.

(2) The FCS polynomial provides a Hamming distance of four in the frame transmission and frame detection process, provided that the total frame length, between SD and ED exclusive, is less than 11454 octets. The maximum data frame length, 8191, is 11454 rounded down to the nearest power of two (minus one).

4.1.7 End Delimiter (ED). The frame structure requires an ED, which ends the frame and determines the position of the FCS. The data between the SD and the ED shall be an integral number of octets. All bits between the SDs and EDs are covered by the FCS.

The ED consists of signaling patterns that are always distinguishable from data. The ED also contains bits of information that are not error checked. The ED is coded as follows:

— first MAC-symbol transmitted

end delimiter (ED): N N 1 N N 1 I E

1 2 3 4 5 6 7 8 ← bit positions

where

N = non_data MAC-symbol

I = one MAC-symbol

I = intermediate bit

(1 = more to transmit,

0 = end of transmission)

E = error bit

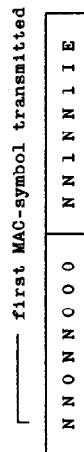
(0 = no error, 1 = error)

The seventh ED MAC-symbol is called the intermediate bit (or "I" bit). If *one*, it indicates that more transmissions from the station follow. If *zero*, it indicates that this is the last frame transmitted by the station and silence follows the ED. The I bit assists a repeater and possibly the Physical Layer in determining what follows the ED.

The eighth ED MAC-symbol is called the error bit (or "E" bit). When set to *one* by a repeater, the error bit indicates that this frame had an FCS error within the frame and the error did not occur on the communication path between the immediately preceding sending station/repeater and this receiving station. When the error bit is set to *one*, the receiving station may treat the frame as an invalid frame. The originating station shall always transmit the error bit as *zero*.

4.1.8 Abort Sequence. The abort sequence is used to prematurely terminate the transmission of a frame. The abort sequence shall be transmitted on an octet boundary relative to that part of the aborted frame already transmitted.

An abort sequence consists of the following pattern:

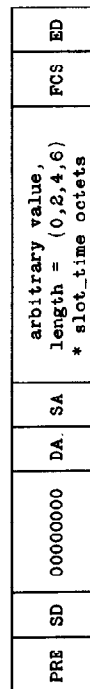


An abort sequence shall also be transmitted by a repeater upon receiving an invalid coding sequence.

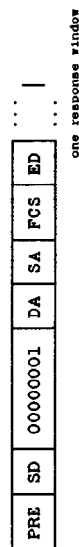
4.2 Enumeration of Frame Types. This subsection shows how the components of the frames are arranged in the various frame types transmitted by the MAC sub-layer. Section 5 discusses the frames and terminology used here.

4.2.1 MAC Control Frame Formats. The following frames are sent and received by the MAC sublayer and are not passed to higher layers.

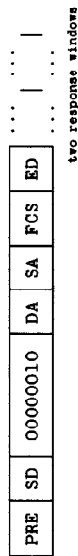
4.2.1.1 Claim_token. The frame has a data_unit whose value is arbitrary and whose length in octets is 0, 2, 4, or 6 times the system's slot_time also measured in octets.



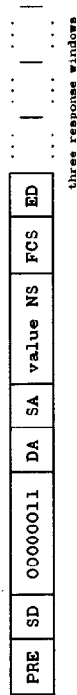
4.2.1.2 Solicit_successor_1. The frame has a DA equal to the value of the station's NS and a null data_unit. One response window always follows this frame.



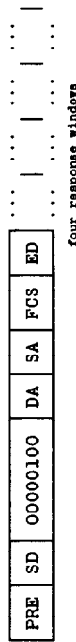
4.2.1.3 Solicit_successor_2. The frame has DA equal to the value of the station's NS or TS and a null data_unit. Two response windows always follow this frame.



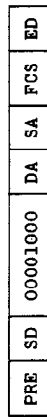
4.2.1.4 Who_follows. The frame has a data_unit equal to the value of the station's NS. The format and length of the data_unit is the same as a source address. Three response windows always follow this frame. (This gives receivers two extra slot_times to make a comparison with an address other than TS.)



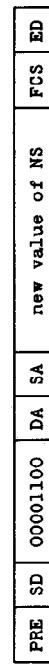
4.2.1.5 Resolve_contention. The frame has a null data_unit. Four response windows always follow this frame.



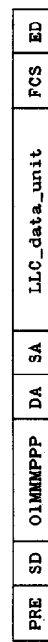
4.2.1.6 Token. The frame has DA equal to the value of the station's NS, and has a null data_unit.



4.2.1.7 Set_successor. The frame has data_unit equal to the value of the station's NS or TS. The format and length of the data_unit are the same as that of a source address.



4.2.2 LLC Data Frame Format. LLC data frames have a DA and data_unit specified by a station's LLC sublayer. A frame of this type with a non-null data_unit shall be passed to the receiving station's LLC sublayer (see 4.1.3.2).



4.2.3 Invalid Frames. An invalid frame is defined as one which meets at least one of the following conditions:

- (1) It is identified as such by the PLE. (For example, it contains *bad_signal* symbols.)
 - (2) It is not an integral number of octets in length.
 - (3) It does not consist of an SD, one FC field, two properly formed address fields, one data_unit field of appropriate length (dependent on the bit pattern specified in the FC field), one FCS field, and an ED, in that order.
 - (4) The FCS computation, when applied to all octets between the SD and the ED, fails to yield the unique remainder specified in 4.1.6.
- An implementation may also include any of the following additional conditions for an invalid frame:
- (5) The FC field contains an undefined bit pattern.
 - (6) The error bit within the ED is set to *one*.
- Invalid frames shall be treated as noise. Their existence, as noise bursts, is relevant at some points in the token bus elements of procedure.

Appendix

(This Appendix is not a part of this International Standard or of IEEE Std 802.4-1990, but is included for information only.)

4.3 Appendix — Recommendation for a Hierarchical Structure for Locally Administered Addresses. The concepts introduced in this Appendix are under study for inclusion in a future revision of this standard.

This Appendix describes an addressing structure for a bus network divided into multiple logical segments, with one or more MAC bridge stations interconnecting the segments. Structuring MAC addresses in a hierarchical fashion can facilitate the operation of these bridge stations.

A logical segment is defined as the collection of all stations of a LAN that have the same segment address and that can exchange frames without any intermediary MAC bridge entity. Stations of a segment can communicate with stations with different segment addresses only through a MAC bridge or some other intermediary. For example, for a token bus implementation, the stations of a logical segment share a single token forming a single logical ring distinct from other logical segments.

A hierarchical address permits a MAC bridge station to recognize frames that require forwarding to other logical segments.

The source and destination address partitioning recommended for this purpose is

(1) 16-bit hierarchical form

— first MAC-symbol transmitted

0	7-bit segment address	8-bit station subaddress
individual station address		

(2) 48-bit locally administered hierarchical form

— first MAC-symbol transmitted

0	1	6-bit segment address	8-bit station subaddress	32-bit station subaddress
region address				
individual segment address				
individual station address				

The 8-bit or 32-bit station subaddress distinguishes stations on a single logical segment. The 8-bit segment subaddress distinguishes logical segments within a single region defined by the 6-bit region address.

5. Elements of MAC Sublayer Operation

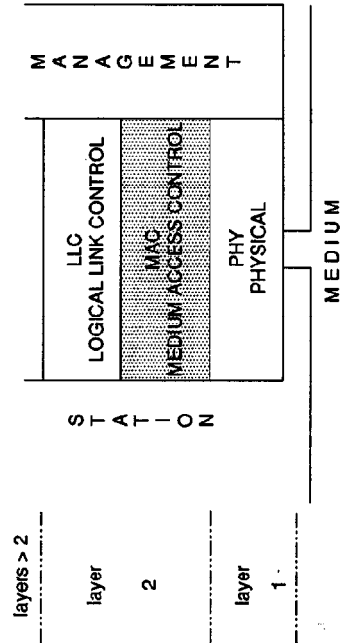
A description of the token-passing bus MAC mechanism is provided in this section. This section is intended to assist the reader in understanding the MAC sublayer and its operation.

Section 6 contains precise definitions of MAC-specific terms and mandatory aspects of the mechanism. Where statements included in this section conflict with those in Section 6, or are incomplete, those in Section 6 shall take precedence.

Section 7 describes the required behavior of the access control machine (ACM) of the MAC sublayer. Where statements in this section or Section 6 conflict with the formal description in Section 7, or are incomplete, the formal description in Section 7 shall take precedence.

This section describes the token-passing bus MAC sublayer's operational and exception recovery functions. The relationship of this section to other sections of this standard and to LAN specifications is illustrated in Fig 5-1.

Fig 5-1
Relation to LAN Model



Specific responsibilities of the MAC sublayer for a broadcast medium involve managing ordered access to the medium, providing a means for admission and deletion of stations (adjustment of logical ring membership), and handling fault recovery.

The faults considered here are those caused by communications errors or station failures. These faults include

- (1) Multiple tokens
- (2) Lost token
- (3) Token pass failure
- (4) "Deaf" station (that is, a station with an inoperative receiver)
- (5) Duplicate station addresses
- (6) Faulty transmitter

This MAC protocol is intended to be robust, in the sense that it should tolerate and survive multiple concurrent errors.

Some basic observations are useful in understanding the operation of token passing on a broadcast medium:

- (a) Stations are connected in parallel to the medium. Thus, when a station transmits, its signal is received (or "heard") by all stations on the medium. Other stations can interfere with the first station's transmission but cannot predictably alter its contents.
- (b) When a station transmits, it may assume that all other stations hear something (though not necessarily what was transmitted).
- (c) When a station receives a valid frame (properly formed and delimited and containing a correct FCS), it may infer that some station transmitted the frame, and therefore, that all stations heard something.
- (d) When a station receives something other than a valid frame (that is, noise), it may make no inference about what the other stations on the medium might have heard.
- (e) Not all stations need be involved in token passing (only those which desire to initiate transmissions).
- (f) Multiple tokens and lost tokens may be detected by any station. There are no special "monitor" stations required to perform token recovery functions.
- (g) Due to spatial separation, stations cannot be guaranteed to have a common perception of the system state at any instant. (The MAC protocol described herein accounts for this.)

5.1 Basic Operation. Steady-state operation (the network condition where a logical ring has been established and no error conditions are present) simply requires the sending of the token to a specific successor station as each station is finished transmitting (see Fig 1-2).

Other essential and more difficult tasks are establishment of the logical ring (at either initialization or reestablishment in the case of a catastrophic error), and maintenance of the logical ring (allowing stations to enter and leave the logical ring without disrupting the other stations in the network).

The token passes the right to transmit among all stations in the logical ring. Each participating station knows the address of its predecessor (the station from which it received the token), referred to as Previous Station or PS. It knows its successor (which station the token should be sent to next), referred to as Next Station or NS. It knows its own address, referred to as This Station or TS. These predecessor and successor addresses are dynamically determined and maintained by the algorithms described. Whenever a station changes its successor (NS), an indication of this change is passed to the management entity (if enabled).

The following subsections introduce major elements and features of the token bus access protocol.

5.1.1 Slot_time. In describing the access operations, the term slot_time is used to refer to the maximum time any station need wait for an immediate MAC level response from another station. Slot_time is precisely defined in 6.1.9.

The slot_time (along with the station's address and several other management parameters) shall be known to the station before it attempts to transmit on the network. If all stations in a network are not using the same value for slot_time, the MAC protocol may not operate properly. The method of setting these parameters in each station is outside the scope of this standard.

5.1.2 Right to Transmit. The token (right to transmit) is passed from station to station in descending numerical order of station address. When a station which belongs to the logical ring hears a token frame addressed to itself, it "has the token" and may transmit data frames. When a station has completed transmitting data frames it passes the token to the next station in the logical ring, as discussed in 5.1.3. When a station has the token it may temporarily delegate to another station its right to transmit by sending a request_with_response data frame. When a station hears a request_with_response data frame addressed to itself it shall respond with a response data frame, if the request_with_response option is implemented. The response data frame causes the right to transmit to revert back to the station which sent the request_with_response data frame.

5.1.3 Token Passing. After each station has completed transmitting any data frames it may have and has completed other maintenance functions (described in 5.1.4), the station passes the token to its successor by sending a "token" MAC_control frame.

After sending the token frame the station listens for evidence that its successor has heard the token frame and is active. If the sender hears a valid frame following the token within one slot_time, it assumes that its successor has the token and is transmitting. Otherwise the token sending station attempts to assess the state of the network.

If the token sending station hears a noise_burst (e.g., an unidentifiable sequence or a frame with an incorrect FCS), it cannot be sure which station sent the transmission. The MAC protocol treats this condition in a way which minimizes the chance of the station causing a serious error.

Since a station on a broadband network should always hear its own frames, if the token sending station hears a single noise_burst without hearing its own token frame, the station assumes that it heard its own token that had been garbled and continues to listen.

If a second noise_burst is heard or if a noise_burst is heard after the station hears its own token frame, the token sending station continues to listen in the CHECK_TOKEN_PASS state for up to four more slot_times. If nothing more is heard, the station assumes that the noise_burst it heard was not a garbled frame from the successor station and so it repeats the token transmission. If anything is heard during the four slot_time delay, the station assumes its successor successfully received the token.

If the token holder does not hear a valid frame after sending the token the first time, it repeats the token pass operation once, performing the same monitoring as during the first attempt.

If the successor does not transmit after a second token frame, the sender assumes that its successor has failed. The sender then sends a who_follows frame with its successor's address in the data_unit field of the frame. All stations compare the value of the data_unit field of a who_follows frame with the address of their predecessor (the station that normally sends them the token). The station whose predecessor is the successor of the sending station responds to the who_follows frame by sending its address in a set_successor frame. The station holding the token thus establishes a new successor, bridging the failed station out of the logical ring.

If the sending station hears no response to a who_follows frame, it sends the frame a second time. If there is still no response, the station tries another strategy to reestablish the logical ring. The station now sends a solicit_successor_2 frame with its own address as both DA and SA, asking any station in the system to respond to it. Any operational station that hears the request and needs to be part of the logical ring responds, and the logical ring is reestablished using the response window process discussed in 5.1.4.

If all attempts at soliciting a successor fail, the station assumes that a fault may have occurred; either all other stations have failed, all stations have left the logical ring, the medium has broken, the station's own transmitter is malfunctioning, or the station's own receiver has failed so that it cannot hear other stations who have been responding to its requests. Under such conditions the station quits attempting to maintain the logical ring. If the station has no frames to send, it listens for some indication of activity from other stations. If the station has data frames to send, it sends its remaining data frames and then repeats the token pass process. Once the station has sent its frames and still cannot locate a successor, it becomes silent and listens for another station's transmissions.

In summary, the token is normally passed from station to station using a short token frame. If a station fails to pick up the token, the sending station uses a series of recovery procedures that grow increasingly more drastic as the station repeatedly fails to find a successor station.

5.1.4 Response Windows. New stations are added to the logical ring through a controlled contention process using "response windows." A response window is a controlled interval of time (equal to one slot_time) after transmission of a MAC_control frame in which the station sending the frame pauses and listens for a response. If the station hears a transmission start during the response window, the station continues to listen to the transmission, even after the response window

time expires, until the transmission is complete. Thus, the response windows define the time interval during which a station will hear the beginning of a response from another station.

The two frame types, solicit_successor_1 and solicit_successor_2, indicate the opening of response windows for stations wishing to enter the logical ring. The solicit_successor frame specifies a range of station addresses between the frame source and destination addresses. Stations whose addresses fall within this range and who wish to enter the logical ring respond to the frame.

The sender of a solicit_successor frame transmits the frame and then waits, listening for a response in the response window(s) following the frame. Responding stations send the frame sender their requests to become the next station in the logical ring. If the frame sender hears a valid request, it allows the new station to enter the logical ring by changing the address of its successor to the new station and passing its new successor the token.

In any response window there exists the possibility that more than one station simultaneously will desire logical ring entry. To minimize contention when this happens, the ring entry sequence is limited by requiring that a station request admission only when a window is opened for an address range that spans its address.

There are two solicit_successor frames. Solicit_successor_1 has one response window. Solicit_successor_2 has two response windows. Solicit_successor_1 is sent when the station's successor's address is less than the station's address. This is the normal case when the token is being passed from higher to lower addressed stations. Solicit_successor_1 allows only stations whose address is in the range between the token sender and its expected next token holder to respond, thus limiting the possible contenders and preserving the descending order of the logical ring.

Exactly one station in the logical ring has its station's address below that of its successor (that is, the unique station having the lowest address which sends the token to the "top" of the address ordered logical ring). When soliciting successors, this station shall open two response windows, one for those stations with addresses below its own, and one for stations with addresses above its successor. The station with the lowest address sends a solicit_successor_2 frame when opening response windows. Stations having an address below the sender respond in the first response window, while stations having an address higher than the sender's successor respond in the second response window.

In any response window, when the soliciting station hears a valid set_successor frame, it has found a new successor. When multiple stations simultaneously respond, only unrecognizable noise may be heard during the response period. The soliciting station then sequences through an arbitration algorithm to identify a single responder by sending a resolve_contention frame. The stations that had responded to the earlier solicit_successor frame and that have not yet been eliminated by the algorithm, choose a two-bit *listen_delay* value (usually from the station's address) and listen for 0, 1, 2, or 3 slot_times as determined by that listen delay value. (This listen delay value is further described later.) If these contending stations hear anything (that is, non-silence) while listening, they eliminate them-

selves from the arbitration. If they hear only silence, they continue to respond to further resolve contention requests from the soliciting station. This process normally leads to the establishment of a single new successor.

5.1.5 Bound on Token Rotation Time. The worst-case token rotation time determines the maximum delay a station experiences in gaining access to the network. A very long token rotation occurs if many stations attempt to add new stations to the logical ring by performing the solicit successor procedure on the same rotation of the token.

The maximum token rotation time is bounded by deferring the solicit successor procedure whenever the token rotation time is longer than a defined maximum. The timer `token_rotation_timer( ring_maintenance )` in each station measures the rotation time of the token. If the time exceeds a value set by the management entity, `target_rotation_time( ring_maintenance )`, the station does not perform the solicit successor procedure. On the next token rotation, the station determines if the token now is rotating fast enough to perform the solicit successor procedure.

In conjunction with the priority option (see 5.1.8), the value of `target_rotation_time( ring_maintenance )` determines the maximum token rotation time.

5.1.6 Initialization. Initialization is primarily a special case of adding new stations; it is triggered by the exhaustion of an inactivity timer (`bus_idle_timer`) in one station. If the inactivity timer expires, the station sends a `claim_token` frame. As in the response window algorithm, the initialization algorithm assumes that more than one station can try to initialize the network at a given instant. This is resolved by address sorting the initializers.

Each potential initializer sends a `claim_token` frame having an `data_unit` field length that is a multiple of the network slot time (the multiple being 0, 2, 4, or 6 based on selected bits of the station address). Each initializing station then waits one slot time for its own transmission, and those of other stations that chose the same frame length, to pass. The station then samples the state of the medium.

If a station senses non-silence, it knows that some other station(s) sent a longer length transmission. The station defers to those stations with the longer transmission and re-enters the IDLE state.

If silence was detected, and unused bits remain in the address string, the station attempting initialization repeats the process using the next two bits of its address to derive the length of the next transmitted frame. If all bits have been used and silence is still sensed, the station has "won" the initialization contest and now holds the token.

Once there is a unique token in the network, the logical ring builds by way of the response window process previously described.

NOTE: A random pair of bits is used at the end of the address sort algorithm to ensure that two stations with the same address (which is a fault condition) will not permanently bring down an entire system. If the two stations do not separate (random choices identical), they both attempt to form a logical ring and, at most, one of them should succeed. If they do separate (random choices are different), one will defer; in the latter case, the station that defers will hear a transmission from a station with an identical address and so will discover the error condition.

5.1.7 Exiting Ring. A station can remove itself from the logical ring at any time by simply choosing not to respond to a token passed to it, allowing the fault

recovery mechanisms in the MAC protocol to patch it out. A more efficient method is the following: when the station has the token and desires to exit the logical ring, the station sends a `set_successor` frame to its predecessor (the station that transmitted the token to it) containing the address of its successor. The exiting station then simply sends the token as usual to its successor. Readmission to the logical ring requires one of the sequences described in 5.1.4 and 5.1.6.

5.1.8 Priority Option. The token-passing bus access method provides an optional priority mechanism by which higher layer data frames awaiting transmission are assigned to different "service classes," ranked or ordered by their desired transmission priority. This priority mechanism allows the MAC sublayer to provide eight service classes to the LLC sublayer, and higher level protocols. The priority of each frame is determined by the service class requested in the MA-UNITDATA request command (see 2.2.1).

The token-passing bus access method distinguishes only four levels of priority, called "access classes." Thus there are four request queues to store frames pending transmission. The access classes are named 0, 2, 4, and 6, with 6 being the highest priority and 0 the lowest.

The MAC sublayer maps the service class requested by the LLC sublayer into a three bit priority value, which is included in the FC field. The priority value is then mapped into the MAC access class by ignoring the least significant bit of the priority field. Thus service classes 0 and 1 correspond to access class 0, service classes 2 and 3 to access class 2, service classes 4 and 5 to access class 4, and service classes 6 and 7 to access class 6.

Any station not using the optional priority feature shall transmit every data frame with an access class of 6 (the highest priority). The service class value in the MA-UNITDATA request shall still be carried in the FC octet. For all stations, the rule governing the transmission of these highest priority frames is that a station shall not transmit consecutive frames for more than some maximum time set by the management entity. This time, called the `hi_pri_token_hold_time`, prevents any single station from monopolizing the network. If a station has more data frames to send at access class 6 than it can transmit in one `hi_pri_token_hold_time` period, it is prohibited from sending additional frames after that time has expired and shall pass the token. However, when a station that is using the optional priority feature has lower access class frames to send and has time available, it shall send these frames subject to the priority system rules described in these paragraphs.

The object of the priority system is to allocate network bandwidth to the higher priority frames and only send lower priority frames when there is sufficient bandwidth. The network bandwidth is allocated by timing the rotation of the token around the logical ring. Each access class is assigned a "target" token rotation time. For each access class the station measures the time it takes the token to circulate around the logical ring. If the token returns to a station in less than the target rotation time, the station can send frames of that access class until the target rotation time has expired. If the token returns after the target rotation time has been reached, the station cannot send frames of that priority on this pass of the token.

Each station using the optional priority scheme shall have three token rotation timers for the three lower access classes. Each access class has a queue of frames to be transmitted. When a station receives the token it first services the highest access class queue, which uses the `hi_pri_token_hold` time to control its operation. After having sent any frames of the highest priority, the station begins to service the token rotation timers and queues working from higher to lower access class.

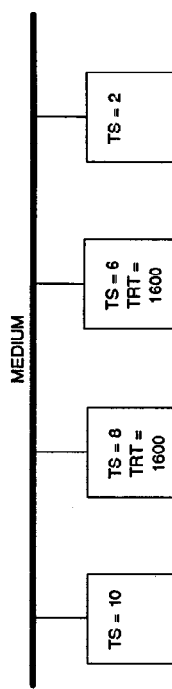
Each access class acts as a virtual substation in that the right to transmit is passed internally from the highest access class downward, through all access classes, before being passed to the station's successor.

The access class service algorithm consists of loading the residual value from a token rotation timer into the token hold timer and reloading the same token rotation timer with the `target_rotation_time` for that access class. (Thus frames sent by a station, for this access class, are accounted for in the access class's next token rotation time computation.) If the token hold timer has a remaining positive value, the station can transmit frames at this access class until either the token hold timer times out or this access class's queue is empty. When either event occurs the station begins to service the next lower access class. When the lowest level is serviced the station performs any required logical ring maintenance and passes the token to its successor.

5.1.9 Token Rotation Timer Example. The following example describes a simplified system with only two active access classes, the highest and one other lower level. Assume there are four stations in the logical ring, with addresses 10, 8, 6, and 2. Assume stations 10 and 2 send high priority frames every token revolution; while 8 and 6 send as many lower priority frames as possible. (The example assumes that propagation delays are negligible compared with the time to send data frames.)

The example begins after a period during which no data frames have been sent so that the token has been rotating as rapidly as possible. In Fig 5-2, each station is represented by a box that contains the station's address and target token rotation time value for the station's active frame queue. The values in the left-hand column under each station, labeled TRTC, are the token rotation times observed at that station for the previous rotation of the token. The values in the right-hand column, labeled XMIT, are the number of frames transmitted by the station each time it holds the token. Each row of this table represents one rotation of the token. In this example, station 10 is assumed to have three high priority frames of 128 octets, stations 8 and 6 have only low priority frames of 400 and 366 octets, respectively, and station 2 has two high priority frames of 305 octets. The token rotation time is given in octet_times and assumes 19 octet_times per token pass per station.

Looking at the example we see that on the first token rotation, station 10 measures an effective rotation time of 76 (4*19) and sends three frames of high priority data. Station 8 thus measures an effective rotation time of 460 (3*128+76), leaving 1140 units of "time" to send data before having to pass the token to station 6. Station 6 measures an effective token rotation time of 1660 (since 1660 octets of data have been sent since it last received the token). Station 6 cannot send low priority data and immediately passes the token. (If station 8 had less data to send, it could have passed the token to station 6 sooner, giving station 6 an opportunity



Token Rotation	TRTC	XMIT	TRTC	XMIT	TRTC	XMIT	TRTC	XMIT
1	76	3	460	3	1660	0	1680	2
2	2270	3	2270	0	1070	2	1782	2
3	1782	3	1782	0	1782	0	1070	2
4	1070	3	1070	2	1870	0	1870	2
5	1870	3	1870	0	1070	2	1782	2
6	1782	3	1782	0	1782	0	1070	2
7	1070	3	1070	2	1870	0	1870	2
8	1870	3	1870	0	1070	2	1782	1
9	1477	3	1477	1	1877	0	1165	1
10	1165	3	1165	2	1565	1	1921	1
11	1921	3	1921	0	1181	2	1477	1
12	1477	3	1477	1	1877	0	1165	1

Fig 5-2
Token Rotation Time "Priority" Example

to send on this rotation.) Station 2 sends two frames of high priority data, unconstrained by the lower priority rotation timer.

On the second rotation of the token station 8 cannot send, but the token arrives early enough for station 6 to send two frames of low priority data.

On the third rotation of the token in the example, stations 10 and 2 continue to send high priority data frames. Neither stations 8 nor 6 can send low priority data since the token is rotating too slowly on this circuit.

On the fourth and fifth rotations, stations 10 and 2 continue to use most of the network bandwidth for high priority frames. Stations 8 and 6 share the remaining bandwidth by each sending two frames on alternate token rotations.

Rotations 5 through 7 repeat the usage pattern shown in rotations 2 through 4, showing a stable cyclic bandwidth allocation pattern: over any three rotations, stations 10 and 2 use 66% of the bandwidth and stations 6 and 8 share the remaining 34%. If all stations always offer the same traffic, this sharing continues indefinitely.

On the eighth rotation of the token, station 2 begins to send only one frame of high priority data, allowing the lower priority stations more of the available network bandwidth. That a stable, cyclic usage pattern is again established can be seen by comparing token rotations 9 and 12. This usage pattern again repeats over

three rotations. Stations 10 and 2 together use 48% of the band-width, and stations 6 and 8 share the remaining 52%.

This example demonstrates how the token rotation timer method allows lower access class traffic to fill in bandwidth that is not used by higher access class traffic. Note how the network bandwidth not used by stations with higher access class traffic is more or less equally shared among stations with lower access class traffic. Although in this example each station only sends traffic of one access class, this standard allows any station to send at any access class.

5.1.10 MAC-Acknowledged Connectionless Class of Service. The request_with_response mechanism, coupled with appropriate MAC-user functions, provides a MAC-acknowledged connectionless class of service. When a higher layer entity requests a data transmission using the MAC-acknowledged connectionless class of service, the MAC-user entity issues to the MAC entity a MA-UNITDATA request with a quality-of-service parameter specifying request_with_response.

When the local MAC entity obtains the token, it sends the request_with_response frame and waits for another valid frame. If a timer expires without receiving a valid frame, the local MAC entity retransmits the original frame. This sequence is repeated until a valid frame is received or the allowed number of retries is exhausted.

When the remote (responding) MAC entity receives the frame specifying request_with_response, it passes the received frame to the remote MAC-user entity. That remote MAC-user entity generates an appropriate response and directs the remote MAC entity to send this response frame at once to the source of the request_with_response frame (that is, the originating station).

When the local (originating) MAC entity receives the response frame, or when the retry count is exhausted, it associates the frame (if available) with the request_with_response frame now being processed and notifies the originating MAC-user entity of the completion of its original request.

The retry mechanism of the request_with_response procedure prevents loss of frames at any arbitrary level of confidence determined by the max_retry_limit parameter (see 7.1.1). However, since the local station repeats the request_with_response frame when a response frame is not received, it is possible for the remote station to receive duplicates of the original request_with_response frame. The remote MAC-user entity should eliminate duplicate frames.

The local (originating) MAC entity engages in one request_with_response activity at a time. All retries and timeouts for that request are completed before the local MAC-entity processes another request or initiates the ring-maintenance/token-passing procedures.

5.1.11 Randomized Variables. Several of the variables used by the MAC protocol have two bit "random" values. Some of these randomized variables are used to improve error recovery probabilities under certain conditions. The randomization of max_inter_solicit_count causes stations to operate "out of step" when opening response windows.

5.2 Access Control Machine (ACM) States. The MAC logic in a station is described here as a computation machine that sequences through a number of dis-

tinct phases, called states. These states are introduced in the following subsections. The states and the transitions between them are illustrated in Fig 5-3. (The dashed lines group states into functional areas.) Section 7 contains the complete state transition table which provides a formal description of the token-passing bus ACM.

5.2.0 Offline. OFFLINE is the state the ACM is in immediately following power-up or following the detection, by the MAC sublayer, of certain fault conditions. After powering up, a station tests itself and its connection to the medium without transmitting on the medium. This "internal" self-testing is station-implementation-dependent and does not affect other stations on the network. Thus, the self-test procedure is beyond the scope of this standard.

After completing any power-up procedures, the station remains in the OFFLINE state until it has had all necessary internal parameters initialized and been instructed to go online.

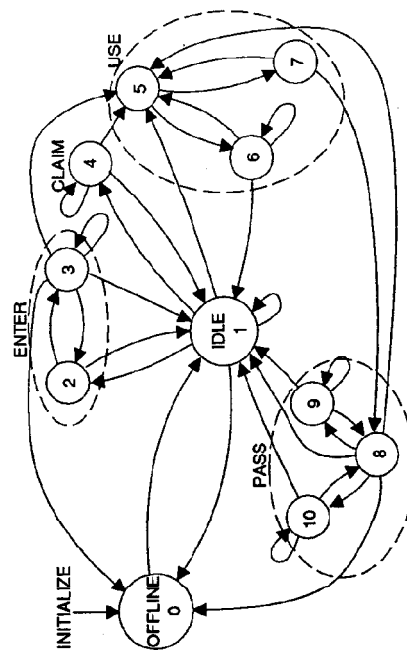
5.2.1 Idle. IDLE is the normal "quiescent" ACM state.

If a MAC-control frame is received for which the station shall take action, the appropriate state is entered. For example, if a token frame addressed to the station is received, the station enters the USE_TOKEN state.

If the station goes for a long period of time (a defined multiple of the slot_time) without hearing any activity on the medium, it may infer that recovery of the logical ring is necessary. The station attempts to claim the token (enters the CLAIM_TOKEN state) and (re)initialize the logical ring.

5.2.2 Demand In. The DEMAND_IN state is entered from the IDLE state if a solicit successor frame that spans the station's address is received by a station

Fig 5-3
MAC Finite State Machine Diagram



desiring logical ring entry. (The DEMAND_IN state is also entered from the DEMAND_DELAY state during the contention resolution process discussed in 5.2.3.) In the DEMAND_IN state the contending station sends the token holder a set_successor frame and goes to the DEMAND_DELAY state to await a response.

If a station intends to respond to a solicit_successor or a who_follows frame in the first response window, the station enters the DEMAND_IN state from the IDLE state with a zero delay and so immediately transmits a set_successor response and goes to the DEMAND_DELAY state. If the station intends to respond in the second or later response window after a frame, the station delays in the DEMAND_IN state before transmitting a set_successor frame.

While delaying in the DEMAND_IN state, if the station hears any transmissions, it assumes that another station with a higher numbered address is requesting the token and so it shall return to the IDLE state.

5.2.3 Demand Delay. DEMAND_DELAY is the state the station enters after having sent a set_successor frame in the DEMAND_IN state. In the DEMAND_DELAY state a station can expect to hear

- (1) A token from the token holder addressed to the station, indicating that the station's set_successor frame was heard,
 - (2) A token from the token holder addressed to another station, indicating that another station's set_successor frame was heard,
 - (3) A resolve_contention frame from the token holder indicating that all stations that are still demanding to be added to the logical ring should perform another step of the contention resolution process, or
 - (4) Set_successor frames from other stations, which the station ignores.
- If the station either hears nothing or hears a frame other than one of (1), (3), or (4), the station shall leave the DEMAND_DELAY state. The station then abandons soliciting the token and returns to the IDLE state.

In (1), the token holder has heard the soliciting station and sent the token. The contention resolution process is over. The soliciting station, upon receiving the token, goes to the USE_TOKEN state and begins transmitting.

In (2), the token holder has heard another soliciting station and sent it the token. The contention resolution process is over. The station abandons soliciting the token and returns to the IDLE state.

In (3), the token holder has heard responses from multiple stations soliciting the token and sent a resolve_contention frame. All stations currently in the DEMAND_DELAY state respond to this frame. The responding stations first set a delay and return to the DEMAND_IN state where they listen for other demanders. If no other demanders have been heard by the time the delay period has expired, each station then sends another set_successor frame to the token holder.

The contention for the token by multiple stations is resolved by having each station delay a period of time in the DEMAND_IN state before transmitting another set_successor frame. The delay interval is chosen by taking the station's (unique) address and using two bits from that address to determine the delay interval. The first pass of the resolution process uses the most significant two address bits; the next pass the next two address bits etc. Thus stations delay 0, 1, 2, or 3 slot_times when entering the DEMAND_IN state before transmitting.

When multiple stations request to enter the logical ring, the desired result is for the token holding station to pass the token to the highest addressed station. To select the highest addressed contending station from multiple contenders, the one's complement of the station's address is used to determine the delay in the DEMAND_IN state. Thus, stations with numerically higher addresses delay shorter intervals and send their set_successor messages sooner than stations with lower addresses. Stations with numerically lower addresses hear the transmissions of the higher addressed stations and drop out of the contention process.

If two contending stations have the same value for the selected two address bits, they delay the same amount of time and transmit more or less simultaneously. If the token holder hears multiple responses and does not hear a valid set_successor from any station, it sends another resolve_contention frame, starting another step of the contention resolution process.

The contention resolution process can take at most: 25 passes $(48/2+1)$, for 48-bit addresses) of the following cycle:

- (a) All remaining demanding stations send set_successor frames to the token holder.
 - (b) They all listen for a response from the token holder and ignore other set_successor frames.
 - (c) They all hear a resolve_contention frame from the token holder.
 - (d) They all delay a number of slot_times based on the next two bits of their own addresses.
 - (e) If they hear another frame during the delay, they drop from contention.
- The contention resolution process should resolve so that the contending station with the highest address is heard by the token holder and receives the token. However, if two stations are erroneously assigned the same station address, they will both sequence through the contention process using the same delays and may not resolve.

To permit eventual resolution in this error condition, a final resolution pass is taken, using a two-bit random number, after the station's address bits have all been used and the contention is still unresolved. If both stations choose the same random value or another error prevents resolution, then the token holder and the contending stations abandon the resolution process until the next response window is opened. (Thus, two stations with the same address that consistently choose the same random number value may never be able to enter the logical ring.)

5.2.4 Claim Token. CLAIM_TOKEN is entered from the IDLE state after the inactivity timer expires (and the station desires to be included in the logical ring). In this state, the station attempts to initialize or reinitialize the logical ring by sending claim_token frames.

To resolve multiple simultaneous stations sending claim_token frames, each station delays for one slot_time after sending the claim_token frame, and then monitors the medium as previously described. If after this delay the bus is quiet, the station sends another claim_token frame.

If the station sends max_pass_count (where max_pass_count is half the number of bits in the station's address plus one, since the address bits are used in

pairs) claim token frames without hearing other transmissions, the station has successfully "claimed" the token, and goes to the **USE_TOKEN** state.

5.2.5 Use Token. **USE_TOKEN** is entered after just receiving or claiming a token. This is the state in which a station can send data frames.

As the station enters this state, it starts the token hold timer, which limits how long the station may transmit before proceeding to the next access class or passing the token. The value initially loaded in the token holding timer, **hi_pri_token_hold_time**, is a system imposed parameter.

When a station's token holding timer has expired and any transmission in progress is complete, or when the station has no more to transmit, it enters the **CHECK_ACCESS_CLASS** state.

When a station sends a data frame it sets the response window timer to 3 slot times and enters the **AWAIT_IFM_RESPONSE** state.

5.2.6 Await IFM Response. **AWAIT_IFM_RESPONSE** is entered when a data frame has been sent. The ACM may wait for the interface machine (IFM) to signal the reception of a response.

If the frame sent in the **USE_TOKEN** state was a request, with **_no_response** frame, no response is expected. The **USE_TOKEN** state is again entered to check for another frame or token holding timer timeout. If the frame sent was a request, with **_response** frame, the station waits in the **AWAIT_IFM_RESPONSE** state for one of the following

- (1) A response frame addressed to the requestor
- (2) Any other valid frame
- (3) A timeout

If a response frame addressed to the requestor is heard, the **USE_TOKEN** state is again entered to check for another frame or token holding timer timeout. (The IFM passes the response frame to the specified MAC-user entity as it does all other data frames addressed to the station. The IFM also associates the response frame with the request, with **_response** frame just previously transmitted.)

If any other valid frame is heard, an error has occurred. The station returns to the **IDLE** state and processes the received frame.

If a timeout occurs before a valid frame is heard, the station repeats sending the request, with **_response** data frame. If the station repeats sending that frame the number of times specified by the **max_retry_limit** parameter, then the request is abandoned. The IFM notifies the MAC-user entity that no response to the frame was received. The **USE_TOKEN** state is entered to check for another frame or token holding timer timeout.

5.2.7 Check Access Class. **CHECK_ACCESS_CLASS** controls the transmission of frames for different access classes. If the priority option is not implemented all frames are considered to be high priority and the **CHECK_ACCESS_CLASS** state serves to control entry to the **PASS_TOKEN** state.

If the priority option is implemented, a station may send frames for lower priority access classes before passing the token. Each access class other than the highest has a target token rotation time. At the time that the station has the token and begins to consider transmitting frames for that access class the residual time left in the target rotation timer is loaded into the token holding timer and the station

returns to the **USE_TOKEN** state. At this time the target rotation timer is also reloaded to its initial value.

Thus, the station will alternate between **USE_TOKEN** and **CHECK_ACCESS_CLASS** states for each access class. If time is available, data frames will be sent in the **USE_TOKEN** state. When the lowest priority access class has been checked, the station will proceed to the pass token process, described next.

When a station has completed sending data frames, it shall enter the **PASS_TOKEN** state. Three conditions can occur:

- (1) The station knows its successor, so it simply passes the token and enters the **CHECK_TOKEN_PASS** state.
- (2) The station knows its successor but must first check if new stations desire to enter the logical ring. The station sends a **solicit_successor** frame and enters the **AWAIT_RESPONSE** state.
- (3) The station does not know its successor. (This condition occurs after initialization and under error conditions.) The station sends a **solicit_successor_2** frame opening response windows for all stations in the system and enters the **AWAIT_RESPONSE** state.

5.2.8 Pass Token. **PASS_TOKEN** is the state in which a station attempts to pass the token to its successor or solicit a new successor.

When its **inter_solicit_count** value is zero and time remains on the ring-maintenance timer, the station allows new stations to enter the logical ring before passing the token. The token holding station does this by sending a **solicit_successor_1** or **solicit_successor_2** frame, as appropriate, and enters the **AWAIT_RESPONSE** state. (See 5.1.4 for the details of this operation.)

If the address of the successor, **NS**, is known, the station performs a simple token pass following any new successor solicitation. (See 5.1.3 for details of this operation.) If the successor responds and the station hears a valid frame, the station has completed its token passing obligations.

If **NS** is not known, the station sends a **solicit_successor_2** frame to itself. Since this frame has two response windows and identical source and destination addresses, it forces all stations on the network that desire to be in the logical ring (whether or not they were previously) to respond. Those stations whose addresses are lower than that of the sender of the token frame transmit in the first window; those with addresses higher transmit in the second window.

During the response window(s), the token holding station listens for **set_successor** frames from potential successors. If no such frame is heard, the station stops trying to maintain the logical ring and listens for transmissions from any other station. (See 5.1.4 for details of this operation.)

5.2.9 Check Token Pass. **CHECK_TOKEN_PASS** is the state in which the station waits for a reaction from the station to which it just passed the token.

The station sending the token waits one slot time for the station receiving the token to transmit. The one slot time delay accounts for the time delay between sending the token frame and the arrival back at the sender of the corresponding response.

If a valid frame is heard that started during the response window, the station assumes the token pass is successful. The frame is processed as if it were received in the IDLE state.

If nothing is heard in one slot time, the station sending the token assumes the token pass was unsuccessful and returns to the PASS_TOKEN state to either repeat the pass or try another strategy.

If noise or an invalid frame is heard, the station continues to listen for additional transmissions, as described in detail in 5.1.3.

5.2.10 Await Response. AWAIT_RESPONSE is the state in which the station attempts to sequence candidate successors through a distributed contention resolution algorithm until one of those successors' set_successor frames is correctly received or until no successors appear. The state is entered from the PASS_TOKEN state whenever the station determines it is time to open a response window or if the station does not know its successor (as in initialization or when a token pass fails).

The station waits in the AWAIT_RESPONSE state for a number of response window times. If nothing is heard for the entire duration of the response window(s), the station goes to the PASS_TOKEN state, either to pass the token to its known successor or to try a different token pass strategy.

If a set_successor frame is received, the station waits for the remainder of the response window time to expire. The station then enters the PASS_TOKEN state and sends the token to the new successor.

If the received frame is other than a set_successor frame, the station drops the token (since some other station is acting as if also has a token, thus creating a multiple token situation) and reenters the IDLE state.

If noise is heard during the response windows, the station cycles through a procedure of sending resolve_contention frames that each open four response windows, and waits for a distinguishable response that began in a response window. The loop repeats a maximum of max_pass_count times, each time instructing contending stations to select a different two bits of their address to determine in which of the four opened windows to transmit.

5.3 Interface Machine (IFM) Description. The IFM acts as an intermediary between the other functional machines of the MAC sublayer and the MAC-user entities with which the MAC entity communicates. Its internal operation is largely unspecified, because the MAC sublayer will function correctly, no matter how the IFM functional requirements are met.

The IFM has eight primary functions, three of which are optional.

- (1) Accept and generate the service primitives specified for the LLC-MAC interface.
- (2) Queue pending service requests.
- (3) Recognize addresses of data frames destined for this station.
- (4) Map LLC quality of service requests from LLC terms (service_class) into MAC terms (access_class).
- (5) Optionally, maintain multiple queues of send requests, separated by access_class (or by service_class).

- (6) Maintain first-in, first-out ordering within each queue of pending service requests.
- (7) Optionally, generate a response_received indication to the ACM when a data frame is received whose quality parameter specifies response.
- (8) Optionally, accept a response from the responding MAC-user entity following receipt of a frame specifying request_with_response, and send a response frame.

Since the service primitives are treated in detail elsewhere in this standard, they are not discussed further here.

The IFM should also handle address recognition for data frames. While some address recognition might be done in the RxM, the need to check potentially large numbers of group addresses for relevance could place unreasonable demands on the RxM.

NOTE: The definition of the group address recognition mechanism is under study.

The notion of quality of service is similar to that of priority, though it is slightly less specific, and deliberately so. Higher protocol layers may rank their messages in some way that may be only approximately realizable by the lower layers. In the LAN case, the LLC sublayer may assign any one of eight service_class rankings to each frame it requests to be sent. The token bus MAC sublayer may not be able to provide different qualities of service, or MAC access_classes, so it shall be able to map all service_classes into one access_class. The MAC sublayer may, optionally, provide four distinct access_classes. When it does so, the IFM shall maintain multiple request queues, so that the requests may be handled according to both access_class (or service_class) and arrival order.

5.4 Receive Machine (RxM) Description. The RxM (see Figs 1-3 and 5-4) accepts MAC-symbols (see 6.1.2) from the Physical Layer and generates high-level data structures and signals for the MAC ACM and the MAC IFM. The interface between the Physical Layer and the MAC sublayer is the PHY-UNITDATA indication primitive specified in Section 8. This description of the RxM embodies the PHY-UNITDATA indication primitive as an encoded MAC-symbol and an associated clock. The other RxM interfaces are internal to MAC and are composed of the signals and data structures shown in Fig 5-4.

bus_quiet. This signal is asserted when the medium is inactive. It is set and cleared by the RxM, and it is read by the ACM.

Rx_protocol_frame. A Boolean variable that is true when a valid MAC_control frame has been received. This variable is set by the RxM, and it is read and cleared by the ACM.

Rx_data_frame. A Boolean variable which is true when a valid data frame has been received. This variable is set by the RxM, read by both the ACM and the IFM, but cleared only by the IFM.

noise_burst. This signal indicates that there was activity on the medium that did not result in a valid frame (see 4.2.3 for the specification of invalid frames). This signal is set by the RxM, and it is read and cleared by the ACM.

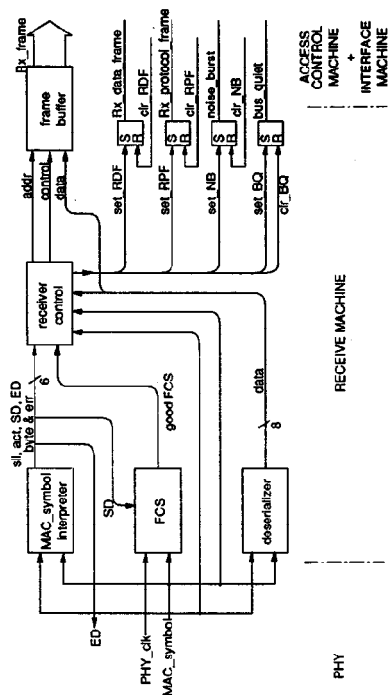


Fig 5-4
Receive Machine

frame_buffer. When a valid frame has been received, this data structure contains the frame control octet, destination_address, source_address, data, unit, and FCS fields of the received frame. The contents of this structure are written by the RxM. When the Rx_protocol_frame signal is set, the contents of this structure are used by the ACM. When the Rx_data_frame signal is set, the contents of this structure are used by the IFM and ACM.

There are four major functional blocks in the RxM: the MAC-symbol interpreter, the FCS block, the deserializer, and the receiver control block. (See Fig 5-4.) These blocks are described as synchronous machines clocked by PHY_clk. Where practical, these blocks are decomposed into finite state machines. See Figs 5-5, 5-6, and 5-7. In these figures, an unbracketed label on an arc is the condition for the arc; a label in brackets is an output of that arc.

The MAC-symbol interpreter accepts MAC-symbols and generates the following signals:

sil, act. These signals represent silence (*sil*) and activity (*act*) on the bus, respectively. The signals *sil* and *act* control set_BQ and clr_BQ (set and clear bus_quiet, respectively). The only reason they are not exactly equivalent is that the receiver control block shall maintain certain timing relationships among the bus_quiet and noise_burst variables. The state machine shown generates *sil* only after eight MAC-symbols of *silence* (S), and requires three non-*silence* MAC-symbols to generate *act*. In part, this is merely to provide a degree of hysteresis in the receiver and the exact counts may be varied by the implementer; however, when a valid frame is received,

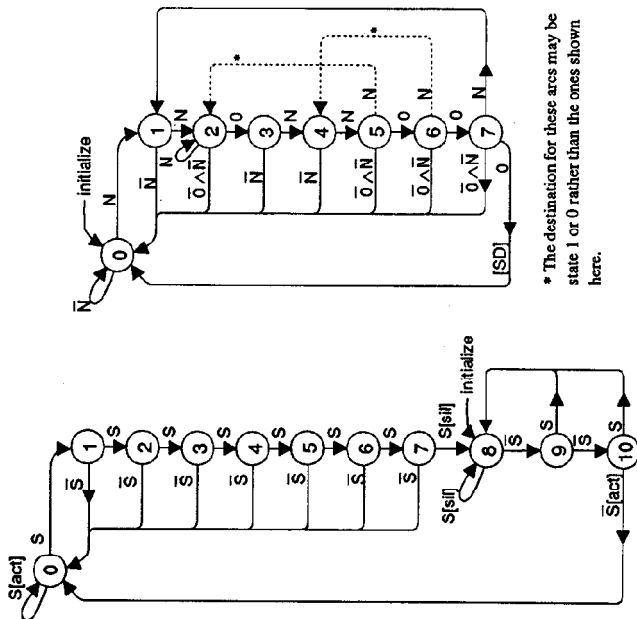


Fig 5-5

sil/act Detector Finite State Machine

Fig 5-6

SD Detector Finite State Machine

bus_quiet shall be set simultaneously or after the appropriate Rx_frame signal is set. Failure to do so can cause the ACM to malfunction.

SD, ED. Any valid frame delimiter in the received MAC-symbol sequence is detected, regardless of context. The ED signal is returned to PHY for use by conditioning and synchronizing circuits.

byte, err. (State machine not shown.) The byte/err detector is unconditionally initialized by SD, establishing byte alignment. While only 0 and 1 MAC-symbols are subsequently received, byte strobes true every eight PHY_clk periods. The reception of any MAC-symbol other than 0 or 1 starts an exit sequence. (See 6.1.2 for definitions of MAC-symbols.) Possible exit sequences are as follows:

- (1) Reception of a P, B, or S MAC-symbol: report the error and terminate (wait for initialization).

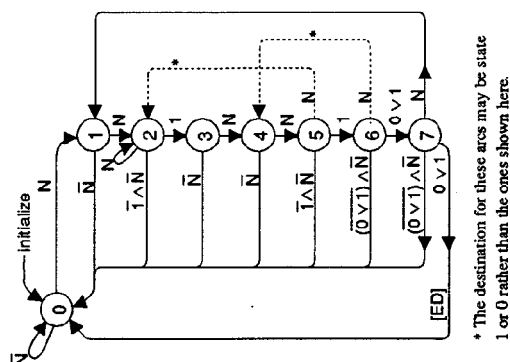


Fig 5-7
ED Detector Finite State Machine

- (2) Reception of an N not on an octet boundary: report the error and terminate.
- (3) Reception of an N on an octet boundary that does not prove to be a frame delimiter: report the error and terminate.
- (4) Reception of an N on an octet boundary that introduces SD: report an error if currently receiving a frame and initialize.
- (5) Reception of an N on an octet boundary that introduces ED: terminate. The FCS block implements the FCS calculation specified in 4.1.6. It is initialized by any SD, and shifts in data until a nondata symbol is received. The output of the deserializer is defined only when SD, ED, or a data byte is reported.

NOTE: Reporting of the value of the data bits in delimiters to higher layers is not required by this specification.

The receiver control block has three components. The noise_burst/bus_quiet (NB/BQ) detector sets and clears bus_quiet, and it sets noise_burst on the rising edge of bus_quiet if a valid frame has not been received (see Fig 5-8). The FC block transfers the deserializer output to the frame buffer, checks the FC octet for data frames, and determines whether the frame contained a legal number of octets. The valid frame detector further verifies that the FCS was correct and that the frame was composed of valid octets.

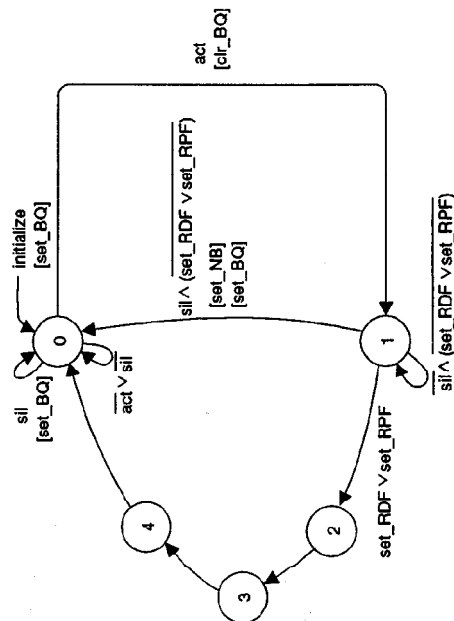


Fig 5-8
NB/BQ Finite State Machine

NOTE: The presentation of this section is for use by the ACM. The statistics recorded for the management entity in the TxM counters group (see 3.2.10) may use additional information from the received data to better report what is heard.

5.5 Transmit Machine (TxM) Description. The TxM has a fairly straightforward and simple operation. The ACM forwards frames for transmission to the TxM as a unit (at least for the purposes of this description). The TxM then passes the frame, along with appropriate delimitation, to the PLE, one MAC-symbol at a time, for transmission on the medium. The TxM is responsible for sending the proper amount of preamble, computing the FCS and including it within the transmitted frame, and delimiting the frame with SD and ED.

5.6 Regenerative Repeater Machine (RRM) Description. Bus repeaters are used to connect electrically, optically, or frequency separate bus segments together into an extended logical bus network. A repeater has a separate receiver for each segment to which it is connected. Each receiver separates clock and MAC-symbols, and has a receive switch that controls which receiver is the MAC-symbol source for the stuffer. A receiver is selected as the stuffer source by reporting non-silence when there is no other receiver acting as stuffer source. A stuffer source is not deselected until it reports silence.

Figure 5-9 shows a generalized repeater. Included in the figure are n distinct physical segments of a single bus. The simplest form of a repeater (a broadband remodulator) has only one transmitter/receiver pair. The stuffer shall

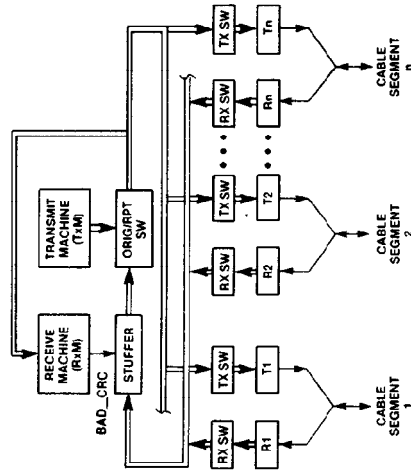
- (1) Re-insert preamble symbols that are lost in the receiver [except as noted in 8.2.1.5(3)],
- (2) Transmit the abort sequence when *bad_signal* is reported to it. The abort sequence always begins on an octet boundary relative to that part of the repeated data already transmitted, and
- (3) Set the E bit in the ED if the repeated frame has an FCS error.

The output of the stuffer goes to one input of the originate/repeat switch; the other input to this switch comes from the TxM. When the TxM reports non-silence, the originate/repeat switch unconditionally selects that input. When the TxM reports *silence*, the switch selects the stuffer, after a short delay, to guarantee that symbols that originated in the local TxM will not be repeated.

While the originate/repeat switch is enabling the local TxM, all transmit switches are on, so the symbols are transmitted onto all segments. While the originate/repeat switch is selecting the stuffer, all transmit switches are on, except the one corresponding to the segment whose receiver is the current symbol source for the stuffer.

A repeater may or may not be a station within the network. If the repeater also functions as a station, the repeater's RxM and TxM perform all the functions described in this standard. If the repeater does not function as a station, the RxM need only check the FCS of received frames, so as to compute the E-bit value for the retransmitted ED. The TxM of a nonstation repeater continuously sends *silence*.

Fig 5-9
Bus Repeater



6. MAC Sublayer Definitions and Requirements

All of the aspects of the MAC sublayer operation and mechanism that are required for conformance to this standard and that are not specified in Sections 4 or 7 are specified in this section.

6.1 MAC Definitions. Critical MAC parameters that are constrained by this specification are defined as follows:

6.1.1 Immediate_response. The immediate transmission of a MAC-level response to a received frame. This assumes that no other transmissions or actions were intervening.

6.1.2 MAC-Symbols. The smallest unit of information exchanged between MAC sublayer entities. The six MAC-symbols are as follows:

Name	Abbreviation
zero	0
one	1
non_data	N
pad_idle	P
silence	S
bad_signal	B

Where binary 0 and 1 data bits are discussed, they are sent and received as *zero* and *one* MAC-symbols, respectively.

6.1.3 MAC-Symbol_time. The time required to send a single MAC-symbol. This is the inverse of the LAN data rate.

Nominal Data Rate	Nominal MAC-Symbol_time
1 Mb/s	1.0 μ s
5 Mb/s	0.2 μ s
10 Mb/s	0.1 μ s
20 Mb/s	0.05 μ s

6.1.4 Octet_time. Corresponds to the time interval required to transmit eight MAC-symbols.