



DD 98: Part 1: 1984

UDC 681.324.02

© British Standards Institution. No part of this publication may be photocopied or otherwise reproduced without the prior permission in writing of BSI

Draft for Development

CSMA/CD Local area networks

Part 1. Technical specification

Projet à développer

Réseaux locaux CSMA/CD (à accès multiple avec exploration de signaux porteurs
et à détection de collision)

Partie 1. Spécifications techniques

Vornorm

CSMA/CD-lokale Netzwerke

Teil 1. Technische Festlegungen



Documents

Document supplied by

British Standards Institution

Introduction

This Draft for Development has been prepared under the direction of the Information and Office Standards Committee, at the request of the Department of Trade and Industry. The primary objective of this Draft is to give advance guidance to product and system designers working in the field of local area networks.

The text of the Draft for Development is identical (with the exception of the clause on environmental conditions, which refers to electrical safety and interference) with Draft E of the standard finalized and approved by the Institute of Electrical and Electronics Engineers, Inc., IEEE 802.3 'CSMA/CD access method and physical layer specifications' and with International Standard 8802.3 which is currently at the draft proposal stage.

For ease of reproduction the IEEE text has been retained. As the environmental conditions, clause 7.8, refers directly to USA regulations, this text does not form part of this Draft for Development; it has not been deleted, however, as it gives useful guidance on USA practice, and margin marks adjacent to the text will refer the reader to the introduction. Users of the Draft for Development are advised to make reference to appropriate UK regulations and standards on electrical safety and interference. The UK member body of the International Organization for Standardization (ISO) TC 97 has proposed that clause 7.8 be replaced in the eventual International Standard by an internationally acceptable text prepared by the International Electrotechnical Commission (IEC).

The Department of Trade and Industry has prepared a Technical Guide that contains guidance on implementation and is complementary to this Draft for Development. This Technical Guide is published as DD 98 : Part 2.

Other Drafts for Development on local area networks which are in course of preparation include:

- DD 99 Logical link control for local area networks
Part 1 Technical specification
- DD 100 Token bus local area networks
Part 1 Technical specification

It is anticipated that a further Draft for Development will be issued in due course on the subject of token ring local area networks, which is likely to be based on IEEE 802.5, when finalized and approved by the IEEE. IEEE 802.1,

although currently at an early stage of development, provides an overview of local area network standardization. The necessary background information for each of the IEEE standards for networks is given in the foreword to each of these standards.

Relevant British Standard publications include:

- BS 6531 10 Mbps slotted ring local area network
- BS 6532 Data terminal equipment for the attachment to 10 Mbps slotted ring local area network

Terminology and conventions. The text of IEEE 802.3 has been approved as suitable for publication as a Draft for Development without deviation. Some terminology and certain conventions are not identical with those used in Drafts for Development; attention is drawn especially to the following.

Wherever the words 'IEEE Standard 802.3' appear they should be read as 'DD 98 : Part 1'.

This publication is not to be regarded as a British Standard.

It is being issued in the Draft for Development series of publications and is of a provisional nature because the application of standardization to LAN technology is rapidly advancing and not yet sufficiently stable, pending the outcome of deliberations in the International Standards Committee (ISO/TC 97, SC 6 Data Transmission).

It should be applied on this provisional basis, so that information and experience of its practical application may be obtained.

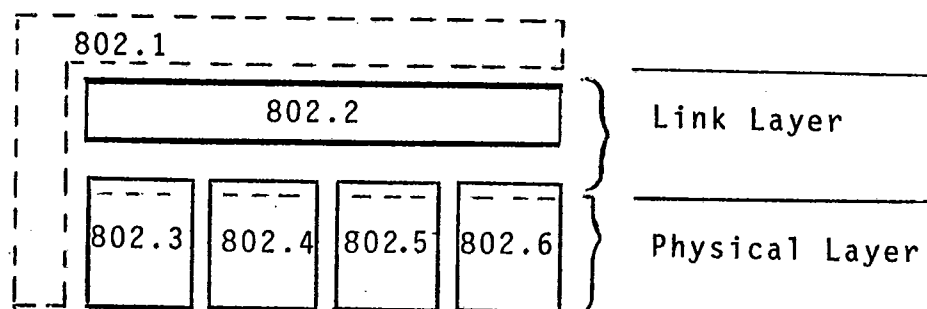
A review of this Draft for Development will be carried out not later than two years after its publication. Notification of the start of the review period, with a request for the submission of comments from users of this Draft for Development will be made in an announcement in the appropriate issue of *BSI News*. Observations which it is felt should receive attention before the official call for comments will be welcomed.

According to the replies received, the responsible BSI Committee will judge whether the Draft for Development can be converted into a British Standard or what other action should be taken.

FOREWORD

(This foreword is not part of the IEEE Standard for Local Area Networks)

IEEE Standard 802 is a family of standards for local area networks. The relationship among the IEEE 802 family is shown below:



The IEEE 802 family of standards specifies the physical and link layers as defined by the ISO Open System Interconnection Reference Model. The Standard defines a single logical link control protocol and four types of media and media access technologies each appropriate for particular applications. The standards defining the links protocol and media access technologies are:

- IEEE standard 802.2, a common logical link control protocol,
- *IEEE Standard 802.3, a bus utilizing CSMA/CD as the access method,*
- IEEE Standard 802.4, a bus utilizing token passing as the access method,
- IEEE Standard 802.5, a ring utilizing token passing as the access method,
- IEEE standard 802.6, a metropolitan area network.

The Logical Link Control protocol, IEEE Standard 802.2, is intended to be used in conjunction with each of the four media and media access standards.

A companion document, IEEE Standard 802.1, describes the relationship among these standards and the ISO Open Systems interconnection model in more detail. This companion document also explains the relation of the IEEE 802 Standards to higher layer protocols, and discusses internet-working and networking management issues.

The reader of this Standard is also urged to become familiar with the other IEEE 802 Standards.

The local area network access mechanism specified by this standard may include patented matter. The IEEE Standards Office calls attention to the fact that it is claimed that the process of the local area network access mechanism described throughout this standard is the subject of United States patent numbers 4,063,220 and 4,099,024 and corresponding patents of foreign countries owned by the Xerox Corporation. Although these patents appear to cover the access mechanism subject in this standard, the IEEE takes no position with respect to patent validity. The Xerox Corporation has assured the IEEE that it is willing to grant a license under these patents on reasonable and nondiscriminatory terms and conditions to anyone wishing to obtain such a license. The Xerox Corporation's undertakings in this respect are on file with the IEEE Standards Office and the license details may be obtained from the Office of General Counsel of Xerox Corporation, whose address is Post Office Box 1600, Stamford, Connecticut 06904.

TABLE OF CONTENTS

SECTION ONE: IEEE 802.3 CSMA/CD: INTRODUCTION

1.1. CSMA/CD Overview	1-1
1.1.1 Basic Concepts	1-1
1.1.2 Application Areas	1-3
1.2. Notation	1-3
1.2.1 State Diagram Conventions	1-3
1.2.2 Service Specification Method	1-5

SECTION TWO: MAC SERVICE SPECIFICATION

2.1. Scope and Field Of Application	2-1
2.2. Overview of the Service	2-1
2.2.1 General Description of Services	2-1
2.2.2 Model Used for the Service Specification	2-1
2.2.3 Overview of Interactions	2-2
2.2.4 Basic Services and Options	2-2
2.3. Detailed Service Specification	2-2
2.3.1 MA_DATA.request	2-2
2.3.2 MA_DATA.confirm	2-3
2.3.3 MA_DATA.indicate	2-4

SECTION THREE: MEDIA ACCESS CONTROL FRAME STRUCTURE

3.1. Media Access Control Frame Structure	3-1
3.1.1 MAC Frame Format	3-1
3.2. Elements of the MAC Frame	3-2
3.2.1 Preamble Field	3-2
3.2.2 Start Frame Delimiter Field	3-2
3.2.3 Address Fields	3-2
3.2.4 Destination Address Field	3-4
3.2.5 Source Address Field	3-4
3.2.6 Length Field	3-4
3.2.7 Data and Pad Fields	3-4
3.2.8 Frame Check Sequence Field	3-5
3.3. Order of Bit Transmission	3-5
3.4. Invalid MAC Frame	3-5

SECTION FOUR: MEDIA ACCESS METHOD

4.1. Functional Model of the IEEE 802.3 CSMA/CD Access Method	4-1
4.1.1. Overview	4-1
4.1.2. IEEE 802.3 CSMA/CD Operation	4-1
4.1.3. Relationships to LLC and Physical Layers	4-5
4.1.4. CSMA/CD Access Method Functional Capabilities	4-5
4.2. IEEE 802.3 CSMA/CD Access Method: Formal Specification	4-7
4.2.1. Introduction	4-7
4.2.2. Overview of the Procedural Model	4-7
4.2.3. Frame Transmission Model	4-13
4.2.4. Frame Reception Model	4-16
4.2.5. Preamble Generation	4-17
4.2.6. Start Frame Sequence	4-18
4.2.7. Global Declarations	4-18
4.2.8. Frame Transmission	4-21
4.2.9. Frame Reception	4-26
4.2.10. Common Procedures	4-28
4.3. IEEE 802.3 CSMA/CD: Interfaces to/from Adjacent Layers	4-30
4.3.1. Overview	4-30
4.3.2. Services Provided by the Media Access Sublayer	4-30
4.3.3. Services Required from the Physical Layer	4-31
4.4. Specific Implementations	4-34
4.4.1. Compatibility Overview	4-34
4.4.2. Recommended Implementation	4-34
Appendix 4.A CSMA/CD: STATE DIAGRAMS	4-35

SECTION FIVE: PLS SERVICE SPECIFICATIONS

5.1. Scope and Field of Application	5-1
5.2. Overview of the Service	5-1
5.2.1 General Description of Services	5-1
5.2.2 Model Used for Service Specification	5-1
5.2.3 Overview of Interactions	5-1
5.2.4 Basic Services and Options	5-2
5.3. Detailed Service Specification	5-3
5.3.1 Peer-to-Peer Service Primitives	5-3
5.3.2 Sublayer-to-sublayer Service Primitives	5-6

SECTION SIX: PLS AND AU INTERFACE SPECIFICATIONS

6.1. SCOPE	6-1
6.1.1 Definitions	6-1
6.1.2 Summary of Major Concepts	6-2
6.1.3 Application	6-3
6.1.4 Modes of Operation	6-4
6.1.5 Allocation of Function	6-4
6.2. REFERENCES	6-4
6.3. FUNCTIONAL SPECIFICATION	6-5
6.3.1 PLS-PMA (DTE-MAU) Interface Protocol	6-6
6.3.2 PLS Interface to MAC and NMT	6-10
6.3.3 Frame Structure	6-14
6.3.4 PLS Functions	6-17
6.4 SIGNAL CHARACTERISTICS	6-28
6.4.1 Signal Encoding	6-28
6.4.2 Signaling Rate	6-31
6.4.3 Signaling Levels	6-31
6.5. ELECTRICAL CHARACTERISTICS	6-32
6.5.1 Driver Characteristics	6-32
6.5.2 Receiver Characteristics	6-36
6.5.3 Attachment Unit Interface Cable Characteristics	6-41
6.6 FUNCTIONAL DESCRIPTION OF INTERCHANGE CIRCUITS	6-44
6.6.1 General	6-44
6.6.2 Definition of Interchange Circuits	6-44
6.7. MECHANICAL CHARACTERISTICS	6-48
6.7.1 Definition of Mechanical Interface	6-48
6.7.2 Line Interface Connector	6-48
6.7.3 Connector Pin Assignment	6-49

SECTION SEVEN: BASEBAND MEDIUM ATTACHMENT UNIT

7.1. SCOPE.....	7-1
7.1.1. Overview.....	7-1
7.1.2 Definitions.....	7-3
7.1.3 Application Perspective: MAU Objectives..	7-5
 7.2 REFERENCES	 7-7
 7.3. MAU FUNCTIONAL SPECIFICATIONS.....	 7-7
7.3.1 MAU Physical Layer Functions..	7-8
7.3.2 MAU Interface Messages.....	7-11
7.3.3 MAU State Diagrams.....	7-14
 7.4. MAU-MEDIUM ELECTRICAL CHARACTERISTICS.....	 7-18
7.4.1 MAU-to-Coaxial Cable Interface.....	7-18
7.4.2 MAU Electrical Characteristics.....	7-22
7.4.3 MAU-DTE Electrical Characteristics.....	7-23
7.4.4 MAU-DTE Mechanical Connection.....	7-23
 7.5. COAXIAL CABLE CHARACTERISTICS.....	 7-24
7.5.1 Coaxial Cable Electrical Parameters.....	7-24
7.5.2 Coaxial Cable Mechanical Parameters.....	7-26
7.5.3 Total Segment D.C. Loop Resistance	7-28
 7.6. COAXIAL CABLE CONNECTORS.....	 7-29
7.6.1 Inline Extensions.....	7-29
7.6.2 Terminators.....	7-29
7.6.3 MAU-to-Coaxial Cable Connector.....	7-30
 7.7. SYSTEM CONSIDERATIONS.....	 7-34
7.7.1 Transmission system Model.....	7-34
7.7.2 Transmission System Requirements.....	7-39
7.7.3 Labeling	7-41
 7.8. ENVIRONMENTAL SPECIFICATIONS	 7-42
7.8.1 Safety.....	7-42
7.8.2 Electromagnetic	7-45
7.8.3 Temperature and Humidity.....	7-45
7.8.4 Regulatory.....	7-46
 7.9. REPEATER SPECIFICATIONS.....	 7-47
7.9.1 Carrier Presence and Transmit Repeat...	7-47
7.9.2 Data Propagation.....	7-48
7.9.3 Collision Detection and Jam Generation...	7-50
7.9.4 Test Functions.....	7-50

SECTION EIGHT: MEDIUM ATTACHMENT UNIT-BROADDBAND [to be added]

SECTION NINE: SYSTEM GUIDELINES

<i>9.1. Analog System Guidelines and concepts</i>	9-1
9.1.1 Baseband System	9-1
9.2 Network Management.....	9-8

APPENDIX A

CSMA/CD - References.....	A-1
---------------------------	-----

SECTION ONE

IEEE 802.3 CSMA/CD: INTRODUCTION

1.1 CSMA/CD OVERVIEW

1.1.1 Basic Concepts

There are two important ways to view local area network design, corresponding to:

Architecture, emphasizing the logical divisions of the system, and how they fit together.

Implementation, emphasizing the actual components, and their packaging and interconnection.

This section of the standard is organized along architectural lines, emphasizing the large-scale separation of the system into two parts: the Media Access sublayer of the Data Link Layer, and the Physical Layer. These layers are intended to correspond closely to the lowest layers of the ISO Model for Open Systems Interconnection. The LLC sublayer (as defined in IEEE 802.2 document) and Media Access sublayers together encompass the functions intended for the Data Link Layer as defined in the OSI model. An architectural organization of the standard has two main advantages:

Clarity: A clean overall division of the design along architectural lines makes the standard clearer.

Flexibility: Segregation of medium-dependent aspects in the Physical Layer allows the Logical Link Control and Media Access sublayers to apply to a family of transmission media. Partitioning the Data Link Layer allows various Media Access Methods within a single family of local network standards.

The architectural model is based on a set of interfaces that may be different from those emphasized in implementations. One crucial aspect of the design, however, must be addressed largely in terms of the implementation interfaces: compatibility. Two important compatibility interfaces are defined within what is architecturally the Physical Layer.

Physical Media interface: To communicate via the IEEE 802.3 Local Area Network, all stations must adhere rigidly to the exact specification of physical media signals defined in Section 7 of this standard to the procedures which define correct behavior of a station. The medium-independent aspects of the Logical Link Control sublayer and the Media Access sublayer should not be taken as detracting from this point: communication via the IEEE 802.3 Local Area Network requires complete compatibility at the Physical Media interface. (i.e., the

Previous page
is blank

coaxial cable interface).

Logical Link Control Sublayer Interface:

AU Interface: It is anticipated that most stations will be located some distance away from their connection to the coaxial cable. A small amount of circuitry will exist in the MAU directly adjacent to the coaxial cable, while the majority of the electronics will be placed within the station. The AU interface is defined as a second compatibility interface. While conformance with this interface is not strictly necessary to ensure communication, it is highly recommended, since it allows maximum flexibility in intermixing MAUs and stations.

The carrier sense, multiple access with collision detection (CSMA/CD) transmission system and access method is the means by which two or more stations communicate over a common bus transmission medium. To transmit, a station waits (defers) for a quiet period on the medium (i.e., no other station is transmitting) and then sends the intended message in bit-serial form. If, after initiating a transmission, the message collides with that of another station, then each transmitting station intentionally sends a few additional bytes to assure propagation of the collision throughout the system. The station remains silent for a random amount of time (backoff) before attempting to transmit again. Each aspect of this access method process is specified in detail in subsequent Sections of the Standard.

This is a comprehensive Standard for Local Area Networks employing CSMA/CD as the access method. This standard encompasses both baseband and broadband techniques for signaling rates of from 1 Mbps to 20 Mbps. This edition of the Standard provides the necessary specification and related parameter values for a 10 Mbps baseband implementation. Subsequent editions of this Standard may provide similar specifications for other implementations (e.g., data rates).

1.1.1.2 Layer Interfaces

In the architectural model used here, the layers interact via well defined interfaces, providing services as specified in Section 4.2. In general, the interface requirements are as follows:

The interface between the Media Access Sublayer and the Logical Link Control Sublayer includes facilities for transmitting and receiving frames, and provides per-operation status information for use by higher-level error recovery procedures.

The interface between the Media Access Sublayer and the Physical Layer includes signals for framing (carrier sense, transmit initiation) and contention resolution (collision detect), facilities for passing a pair of serial bit streams (transmit, receive) between the two layers, and a wait function for timing.

These interfaces are described more precisely in Section 4.3. Additional interfaces are necessary to allow a higher level network management facility to interact with these layers to perform operation, maintenance and planning functions (These functions are discussed in System Guidelines, Section 9).

1.1.2 Application Areas

The applications environment for the Local Area Network is intended to be commercial and light industrial. Use of CSMA/CD LAN's in home or heavy industrial environments, while not precluded, is not considered within the scope of this document.

Specific applications areas and the types of message traffic which would be suitable for CSMA/CD LAN's are to be added.

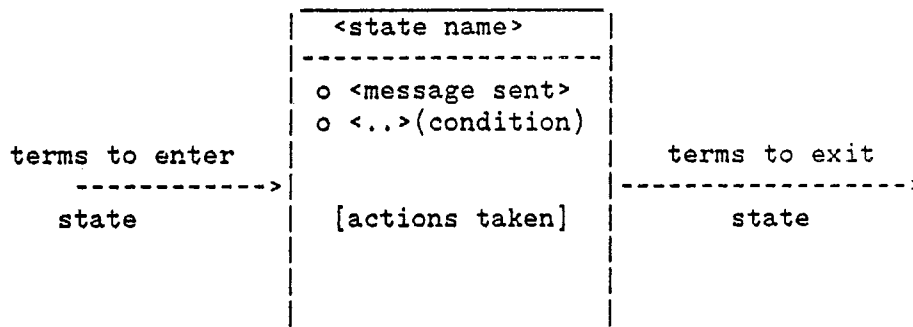
1.2 Notation

1.2.1 State Diagram Conventions

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

Each state that the function can assume is represented by a rectangle (or an ellipse). These are divided into two parts by a horizontal line. In the upper part the state is identified by a name. The lower part contains the name of any ON signal that is generated by the function.

All permissible transitions between the states of a function are represented graphically by arrows between them. A transition that is global in nature (i.e., exit condition from all states to the IDLE or RESET state) is indicated by an open arrow).



Key: () = condition, e.g., if NoCollision
 [] = action, e.g., reset PLS functions

 * = logical AND
 + = logical OR

Tw = Wait Time, implementation dependent
 Td = Delay Timeout
 Tb = Backoff Timeout
 UCT = unconditional transition

Fig. 1-1. State Diagram Example

1.2.2 Service Specification Method and Notation

The service of a layer or sublayer is the set of capabilities which it offers to a user in the next higher (sub)layer. Abstract services are specified here by describing the service primitives and parameters which characterize each service. This definition of service is independent of any particular implementation.

Specific implementations may also include provisions for interface interactions which have no direct end to end effects. Examples of such local interactions include interface flow control, status requests and indications, error notifications, and layer management. Specific implementation details are omitted from this service specification both because they will differ from implementation to implementation and because they do not impact the peer-to-peer protocols.

1.2.2.1 Classification of Service Primitives

Primitives are of three generic types.

REQUEST

The request primitive is passed from layer N to layer N-1 to request that a service be initiated.

INDICATION

The indication primitive is passed from layer N-1 to layer N to indicate an internal layer N-1 event which is significant to layer N. This event may be logically related to a remote service request, or may be caused by an event internal to layer N-1.

CONFIRM

The confirm primitive is passed from layer N-1 to layer N to convey the results of the associated previous service request.

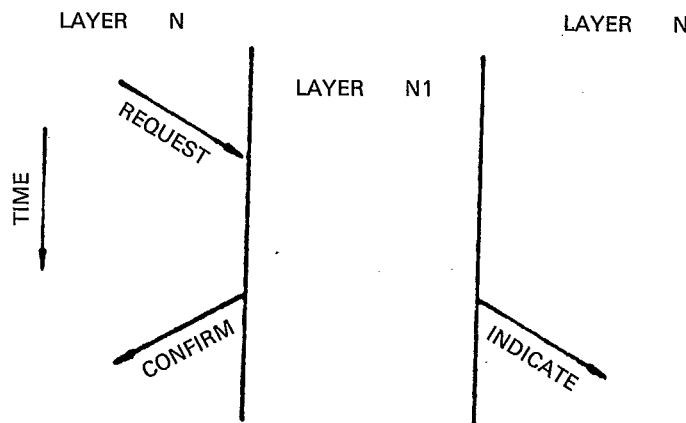


Figure 1.2. Service Primitives

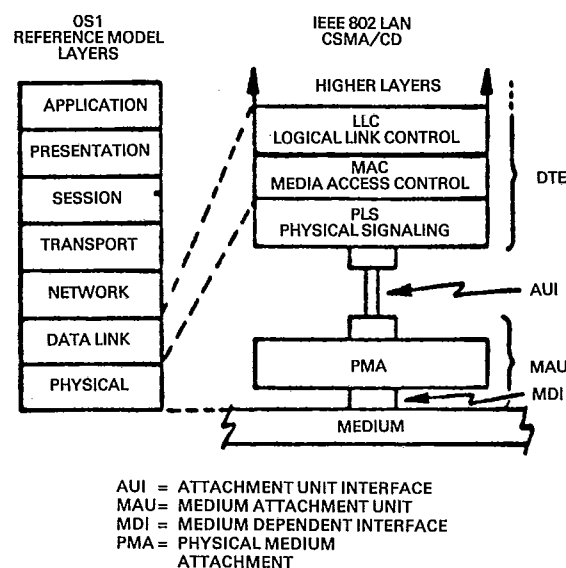
The service primitives are an abstraction of the functional specification and the user-layer interaction. The abstract definition does not contain any local detail of the user/provider interaction not involved in determining some aspect of interconnection of the users. For instance, it does not indicate the local mechanism that allows a user to indicate that it is awaiting an incoming call. Each primitive has a set of zero or more parameters, representing data elements which must be passed to qualify the functions invoked by the primitive. Parameters indicate information available in a user/provider interaction; in any particular interface, some parameters may be explicitly stated (even though not explicitly defined in the primitive) or implicitly associated with the service access point. Similarly, in any particular protocol specification, functions corresponding to a service primitive may be explicitly defined or implicitly available.

SECTION TWO

IEEE 802.3 CSMA/CD: MAC SERVICE SPECIFICATION

2.1. SCOPE AND FIELD OF APPLICATION

This section specifies the services provided by the Medium Access Control (MAC) sublayer to the Logical Link Control (LLC) sublayer for the IEEE 802 Local Area Network standard (see Figure 2-1). The services are described in an abstract way and do not imply any particular implementation, or any exposed interface. There is not necessarily a one-to-one correspondence between the primitives and the formal procedures and interfaces described in sections 4.2 and 4.3.



Fi. 2-1. Service Specification Relation to the LAN Model.

2.2. OVERVIEW OF THE SERVICE

2.2.1 General Description of Services Provided by the Layer

The services provided by the MAC sublayer allow the local LLC sublayer entity to exchange LLC data units with peer LLC sublayer entities. Optional support may be provided for resetting the MAC sublayer entity to a known state.

2.2.2 Model Used for the Service Specification

The model used in this service specification is identical to that used in Section 1.2.

2.2.3 Overview of Interactions

MA_DATA.request
MA_DATA.confirm
MA_DATA.indicate

2.2.4 Basic Services and Options

The MA_DATA.request, MA_DATA.confirm, and MA_DATA.indicate service primitives described in this section are considered mandatory.

2.3. DETAILED SERVICE SPECIFICATION

2.3.1 MA_DATA.request

2.3.1.1 Function

This primitive defines the transfer of data from a local LLC sublayer entity to a single peer LLC entity or multiple peer LLC entities in the case of group addresses.

2.3.1.2 Semantics of the Service Primitive

The semantics of the primitive are as follows:

```
MA_DATA.request (
    destination__address,
    m__sdu,
    service__class
)
```

The destination address parameter may specify either an individual or a group MAC entity address. It must contain sufficient information to create the DA field that is appended to the frame by the local MAC sublayer entity as well as any lower level address information. The m__sdu parameter specifies the MAC service data unit to be transmitted by the MAC sublayer entity. There is sufficient information associated with m__sdu for the MAC sublayer entity to determine the length of the data unit. The service_class parameter indicates a quality of service requested by LLC or higher layer (see Additional Comments Subsection 2.3.1.5).

2.3.1.3 When Generated

This primitive is generated by the LLC sublayer entity whenever data must be transferred to a peer LLC entity or entities. This can be in response to a request from higher layers of protocol or from data generated internally to the LLC sublayer, such as required by Type 2 service.

2.3.1.4 Effect of Receipt

The receipt of this primitive will cause the MAC entity to append all MAC specific fields, including DA, SA, and any fields that are unique to the particular media access method, and pass the properly formed frame to the lower layers of protocol for transfer to the peer MAC sub-layer entity or entities.

2.3.1.5 Additional Comments

The CSMA/CD MAC protocol provides a single quality of service regardless of the service_class requested.

2.3.2 MA_DATA.confirm

2.3.2.1 Function

This primitive has local significance and shall provide an appropriate response to the LLC sublayer MA_DATA.request primitive signifying the success or failure of the request.

2.3.2.2 Semantics of the Service Primitive

The semantics of this primitive are as follows:

MA_DATA.confirm (transmission_status)

The transmission_status parameter is used to pass status information back to the local requesting LLC sublayer entity. It is used to indicate the success or failure (e.g. excessive collisions) of the previous associated MA_DATA.request.

2.3.2.3 When Generated

This primitive is generated in response to an MA_DATA.request from the local LLC sublayer entity.

2.3.2.4 Effect of Receipt

The effect of receipt of this primitive by the LLC sublayer is unspecified.

2.3.2.5 Additional Comments

It is assumed that sufficient information is available to the LLC sublayer to associate the response with the appropriate request. (e.g. the association may be implied by the order of the responses, since the MAC sublayer requires that the requests be serviced in a first-in-first-out manner)

2.3.3 MA_DATA.indicate

2.3.3.1 Function

This primitive defines the transfer of data from the MAC sublayer entity to the LLC sublayer entity or entities in the case of group addresses.

2.3.3.2 Semantics of the Service Primitive

The semantics of the primitive are as follows:

```
MA_DATA.indicate (
    destination__address,
    source__address,
    m_sdu,
    reception__status
)
```

The destination_address parameter may be either an individual or a group address as specified by the DA field of the incoming frame. The m_sdu parameter specifies the MAC service data unit as received by the local MAC entity. The reception_status parameter is used to pass status information to the peer LLC sublayer entity.

2.3.3.3 When Generated

The MA_DATA.indicate is passed from the MAC sublayer entity to the LLC sublayer entity or entities to indicate the arrival of a frame to the local MAC sublayer entity. Such frames are reported only if they are validly formed, received without error, and their destination address designates the local MAC entity.

2.3.3.4 Effect of Receipt

The effect of receipt of this primitive by the LLC sublayer is unspecified.

2.3.3.5 Additional Comments

If the local MAC sublayer entity is designated by the destination_address parameter of an MA_DATA.request, the indicate primitive will also be invoked by the MAC entity to the local LLC entity. This full duplex characteristic of the MAC sublayer may be due to unique functionality within the MAC sublayer or full duplex characteristics of the lower layers. (e.g. all frames transmitted to the broadcast address will invoke MA_DATA.indicates at all stations in the network including the station that generated the request.)

SECTION THREE

IEEE 802.3 CSMA/CD: MEDIA ACCESS CONTROL FRAME STRUCTURE

3.1 Media Access Control Frame Structure

This section defines in detail the frame structure for data communication systems using local area network media access control (MAC) procedures. It defines the relative positions of the various components of the MAC frame. It defines the method for representing station addresses. It defines a partition of the address space into individual (single station) and group (multicast or multi-station) addresses, and into user administered and globally administered addresses.

3.1.1 MAC Frame Format

Figure 3-1 shows the eight fields of a frame: the preamble, start frame delimiter, the addresses of the frame's source and destination, a length field to indicate the length of the following field, containing the LLC data to be transmitted, a field that contains padding, if re-

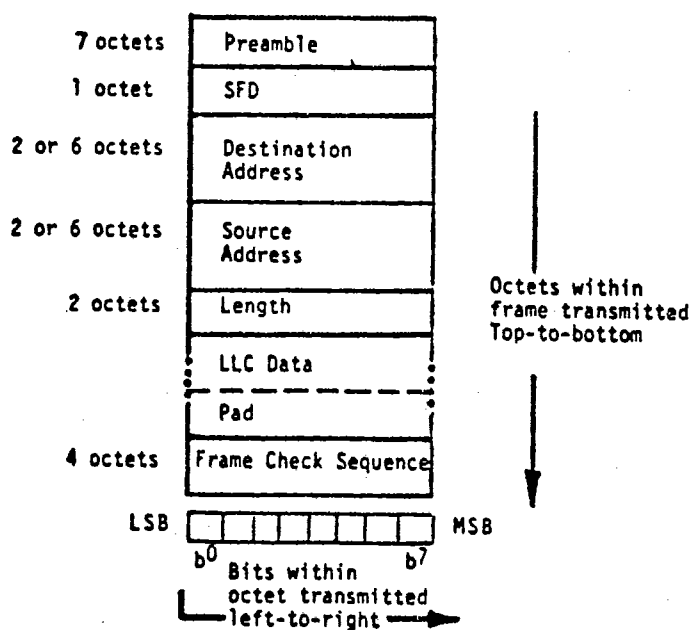


Fig. 3-1 MAC Frame Format.

quired, and the frame check sequence field containing a cyclic redundancy check value to detect transmission errors. Of these eight fields, all are of fixed size except the LLC data and PAD fields, which may contain any integral number of octets between the minimum and maximum values determined by the specific implementation of the CSMA/CD Media Access mechanism. See Section 4.4 for a particular implementation.

The minimum and maximum frame size limits in Sub-Section 4.4 refer to that portion of the frame from the destination address field through the frame check sequence field, inclusive.

Relative to Figure 3-1, the octets of a frame are transmitted from top to bottom, and the bits of each octet are transmitted from left to right.

3.2. Elements of the MAC Frame

3.2.1 Preamble Field

The preamble field is a 7 octet field which is used to allow the PLS circuitry to reach its steady-state (see sec.4.2.5 for preamble pattern and further details).

3.2.2 Start Frame Delimiter Field

The Start Frame Delimiter (SFD) field is the sequence 10101011. It immediately follows the preamble pattern and indicates the start of a valid frame.

3.2.3 Address Fields

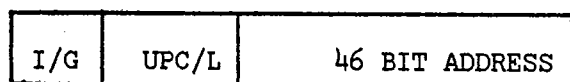
Each MAC frame shall contain two address fields: the Destination Address field and the Source Address field, in that order. The Destination Address field shall specify the destination addressee(s) for which the frame is intended. The Source Address field shall identify the station from which the frame was initiated. The representation of each address field shall be as follows (see Figure 3-2):

- a) Each address field shall contain either 16 or 48 bits. However, at any given time, the Source and Destination Address size must be the same for all stations on a particular local area network.
- b) The support of 16 or 48 bit address length for source and destination address shall be left to the manufacturer as an implementation decision. There is no requirement that manufacturers support both sizes.
- c) The first bit (LSB) shall be used in the Destination Address field as an address type designation bit to identify the destination address either as an individual or as a group address and shall not be used in the source Address field, but shall be reserved and set to

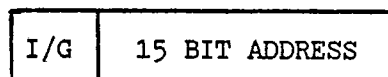
"0". If this bit is "0", it shall indicate that the address field contains an individual address. If this bit is "1", it shall indicate that the address field contains a group address that identifies none, one or more, or all of the stations connected to the local area network.

- d) For 48 bit addresses, the second bit shall be used to distinguish between locally or globally administered addresses. For globally administered (or UPC) addresses, the bit is set to "0". If an address is to be assigned locally, this bit shall be set to "1". Note that for the broadcast address, this bit is also a "1".
- e) Each octet of each address field shall be transmitted least significant bit first.

48 BIT ADDRESS FORMAT



16 BIT ADDRESS FORMAT



I/G = 0 Individual Address
 I/G = 1 Group Address
 UPC/L = 0 Globally Administered Address
 UPC/L = 1 Locally Administered Address

Fig. 3-2. Address Field Format

3.2.3.1 Address Designation

A Media Access Control Sublayer address is of one of two types:

Individual address: The address associated with a particular station on the network.

Group address: A multi-destination address, associated with one or more stations on a given network. There are two kinds of multicast address:

- Multicast-group address: An address associated by higher-level convention with a group of logically related stations.

- Broadcast address: A distinguished, predefined multicast address which always denotes the set of all stations on a given local area network.

All "1"s in the destination address field (for 16 or 48 bit address size LAN's) shall be predefined to be the Broadcast address. This group shall be predefined for each communication medium to consist of all stations actively connected to that medium; it shall be used to broadcast to all the active stations on that medium. All stations must be able to recognize the Broadcast address. It is not necessary that a station be capable of generating the Broadcast address.

The address space shall also be partitioned into locally administered and globally administered addresses. The nature of a body and the procedures by which it administers these global addresses is beyond the scope of this standard.

3.2.4 *Destination Address Field*

The destination address field specifies the station(s) for which the frame is intended. It may be an individual or multicast (including broadcast) address.

3.2.5 *Source Address Field*

The source address field specifies the station sending the frame. The source address field is not interpreted by the CSMA/CD Media Access Sublayer.

3.2.6 *Length Field*

The length field is a 2 octet field whose value indicates the number of LLC data octets in the data field. If the value is less than the minimum required for proper operation of the protocol, a pad which is a sequence of octets will be added at the end of the data field but prior to the FCS field, specified below. The procedure which determines the size of the pad field is specified in Section 4.2.8. The length field is transmitted and received high order octet first.

3.2.7 *Data and PAD Fields*

The data field contains a sequence of n octets. Full data transparency is provided, in the sense that any arbitrary sequence of octet values may appear in the data field up to a maximum limit specified by the particular implementation of this standard. A minimum frame size is required for correct CSMA/CD protocol operation and is specified by the particular implementation of the standard. If necessary, the data field is extended by appending extra bits (i.e., a pad) in units of octets after the LLC data field but prior to calculating and appending the FCS. The size of the pad, if any, is determined by the size of the data field supplied by LLC and the minimum frame size and address size parameters of the particular implementation.

The maximum size of the data field is determined by the maximum frame size and address size parameters of the particular implementation.

The length of pad required for LLC data n octets long is $\max(0, \text{minFrameSize} - (8*n + 2*\text{addressSize} + 48))$ bits. The maximum possible size of the LLC data field is $\text{maxFrameSize} - (2*\text{addressSize} + 48)/8$ octets. (see Sub-Section 4.4 for a discussion of implementation parameters; see Sub-Section 4.2.3.3 for a discussion of the minFrameSize).

3.2.8 Frame Check Sequence Field

A Cyclic Redundancy Check (CRC) is used by both the transmit and receive algorithms to generate a CRC value for the FCS field. The frame check sequence (FCS) field contains a 4-octet (32-bit) cyclic redundancy check (CRC) value. This value is computed as a function of the contents of the source, destination, length, LLC data, and pad i.e., all fields except the preamble, SFD, and FCS.). The encoding is defined by the following generating polynomial [See Appendix, Reference 3-1], where the symbol x^N represents an exponent:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Mathematically, the CRC value corresponding to a given frame is defined by the following procedure:

1. The first 32 bits of the frame are complemented.
2. The n bits of the frame are then considered to be the coefficients of a polynomial $M(x)$ of degree $n-1$. (The first bit of the destination address field corresponds to the $x^{(n-1)}$ term and the last bit of the data field corresponds to the x^0 term.)
3. $M(x)$ is multiplied by x^{32} and divided by $G(x)$, producing a remainder $R(x)$ of degree < 31 .
4. The coefficients of $R(x)$ are considered to be a 32-bit sequence.
5. The bit sequence is complemented and the result is the CRC.

The 32 bits of the CRC value are placed in the frame check sequence field so that the x^{31} term is the leftmost bit of the first octet, and the x^0 term is the rightmost bit of the last octet. (The bits of the CRC are thus transmitted in the order $x^{31}, x^{30}, \dots, x^1, x^0$.)

3.3 Order of Bit Transmission

Each octet of the MAC frame, with the exception of the FCS, is transmitted low-order bit first.

3.4. Invalid MAC Frame

An invalid MAC frame shall be defined as one which meets at least one of the following conditions:

DD 98 : Part 1 : 1984

- a) The frame length is inconsistent with the length field.
- b) It is not an integral number of octets in length.
- c) The bits of the incoming frame (exclusive of the FCS field itself) do not generate a CRC value identical to the one received.

The contents of invalid MAC frames shall not be passed to LLC. The occurrence of invalid MAC frames may be communicated to network management.

SECTION FOUR

*IEEE 802.3 CSMA/CD: Media Access Method**4.1. FUNCTIONAL MODEL OF THE IEEE 802.3 CSMA/CD ACCESS METHOD**4.1.1. Overview*

The architectural model described in Section 1.1 is used in this section to provide a functional description of the IEEE 802 Local Network Standard's CSMA/CD Access Mechanism.

The Media Access Sublayer defines a medium-independent facility, built on the medium-dependent physical facility provided by the Physical Layer, and under the access-layer-independent local network Logical Link Control (LLC) Sublayer. It is applicable to a general class of local area broadcast media suitable for use with the medium access discipline known as carrier-sense multiple-access with collision-detection (CSMA/CD).

The LLC Sublayer and the Media Access Sublayer together are intended to have the same function as that described in the OSI model for the data link layer alone. In a broadcast network, the notion of a data link between two network entities does not correspond directly to a distinct physical connection. Nevertheless, the partitioning of functions presented in this standard requires two main functions generally associated with a data link control procedure to be performed in the Media Access Sublayer. They are:

Data encapsulation (Transmit-and Receive - Data Encapsulation)

- framing (frame boundary delimitation, frame synchronization)
- addressing (handling of source and destination addresses)
- error detection (detection of physical medium transmission errors)

Medium Access Management

- medium allocation (collision avoidance)
- contention resolution (collision handling)

The remainder of this subsection provides a functional model of the CSMA/CD Media Access method.

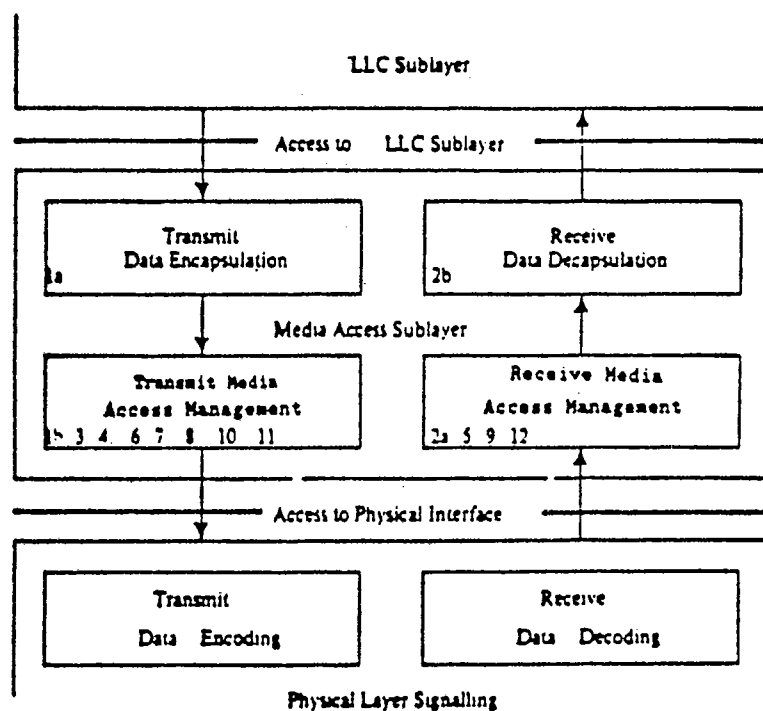
4.1.2 IEEE 802 CSMA/CD Operation

This section provides an overview of frame transmission and reception in terms of the functional model of the architecture. This overview is descriptive, rather than definitional; the formal specifications of the operations described here are given in Sections 4.2 and 4.3. Specific

implementations for CSMA/CD mechanisms that meet this standard are given in Section 4.4. Figure 4-1 provides the architectural model described functionally in the sections below.

The Physical Layer Signalling (PLS) component of the Physical Layer provides an interface to the Media Access Sublayers for the serial transmission of bits onto the physical media. For completeness, in the operational description below, some of these functions are included as descriptive material. The concise specification of these functions is given in Section 4.2 for the MAC functions and in Section 6 for PLS.

Transmit frame operations are independent from the receive frame operations. A transmitted frame addressed to the originating station will be received and passed to the LLC sublayer at that station. This characteristic of the MAC sublayer may be implemented by functionality with the MAC sublayer or full C duplex characteristics of portions of the lower layers.



Note: Numbers refer to functions listed in Sec. 4.1.4.

Fig. 4-1. IEEE 802.3 CSMA/CD Media Access Control Functions.

4.1.2.1 *Normal Operation*

4.1.2.1.1 *Transmission Without Contention*

When a LLC Sublayer requests the transmission of a frame, the Transmit Data Encapsulation component of the CSMA/CD Media Access Sublayer constructs the frame from the LLC-supplied data. It appends a preamble and a start of frame delimiter to the beginning of the frame. Using information passed by the LLC sublayer, the CSMA/CD Media Access Sublayer also appends a PAD at the end of the MAC information field of sufficient length to ensure that the transmitted frame length satisfies a minimum frame size requirement, see Section 4.2.3.3. It also appends destination and source addresses, a length count field and a frame check sequence to provide for error detection. The frame is then handed to the Transmit Media Access Management component in the Media Access Sublayer for transmission.

Transmit Media Access Management then attempts to avoid contention with other traffic on the medium by monitoring the carrier sense signal provided by the Physical Layer Signalling (PLS) component and deferring to passing traffic. When the medium is clear, frame transmission is initiated (after a brief interframe delay to provide recovery time for other CSMA/CD Media Access sublayers and for the physical medium). The Media Access Sublayer then provides a serial stream of bits to the PLS interface for transmission.

The PLS performs the task of actually generating the electrical signals on the medium which represent the bits of the frame. Simultaneously, it monitors the medium and generates the collision detect signal, which, in the contention-free case under discussion, remains off for the duration of the frame. A functional description of the Physical Layer is given in Sections 6, 7, and 8.

When transmission has completed without contention, the CSMA/CD Media Access Sublayer so informs the LLC Sublayer using the LLC-Media Access interface and awaits the next request for frame transmission.

4.1.2.1.2 *Reception Without Contention*

At each receiving station, the arrival of a frame is first detected by the PLS, which responds by synchronizing with the incoming preamble, and by turning on the carrier sense signal. As the encoded bits arrive from the medium, they are decoded and translated back into binary data. It then passes subsequent bits up to the Media Access Sublayer, where first, the leading bits are discarded, up to and including the end of the preamble and start frame delimiter.

Meanwhile, the Receive Media Access Management component of the Media Access Sublayer, having seen carrier sense go on, has been waiting for the incoming bits to be delivered. Receive Media Access Management collects bits from the PLS as long as the carrier sense signal remains on. When the carrier sense signal goes off, the frame is truncated to

an octet boundary if necessary and passed to Receive Data Decapsulation for processing.

Receive Data Decapsulation checks the frame's destination address field to decide whether the frame should be received by this station. If so, it passes the destination address (DA), the source address (SA), and the LLC data unit (L_SDU) to the LLC Sublayer along with an appropriate status code indicating `reception_complete` or `reception_too_long`. It also checks for invalid MAC frames by inspecting the frame check sequence to detect any damage to the frame enroute, and by checking for proper octet-boundary alignment of the end of the frame. Frames with a valid FCS may also be checked for proper octet-boundary alignment.

4.1.2.2 Access Interference and Recovery

If multiple stations attempt to transmit at the same time, it is possible for their transmitting Media Access sublayers to interfere with each others' transmissions, in spite of their attempts to avoid this by deferring. When two stations' transmissions overlap, the resulting contention is called a collision. A given station can experience a collision during the initial part of its transmission (the "collision window"), before its transmitted signal has had time to propagate to all stations on the CSMA/CD medium and the effects of that signal to propagate back. Once the collision window has passed, the station is said to have acquired the medium; subsequent collisions are avoided, since all other (properly functioning) stations can be assumed to have noticed the signal (via carrier sense) and to be deferring to it. The time to acquire the medium is thus based on the round-trip propagation time of the physical layer whose elements include the PLS, PMA, and physical media.

In the event of a collision, the PLS of a transmitting station's Physical Layer first notices the interference on the medium and turns on the collision detect signal. This is noticed in turn by the Transmit Media Access Management component of the Media Access Sublayer, and collision handling begins. First, Transmit Media Access Management must enforce the collision by transmitting a bit sequence called the jam. In Section 4.4 an implementation which uses this enforcement procedure is provided. This insures that the duration of the collision is sufficient to be noticed by the other transmitting station(s) involved in the collision. After the jam is sent, Transmit Media Access Management terminates the transmission and schedules a retransmission attempt for a randomly selected time in the near future. Retransmission is attempted repeatedly in the face of repeated collisions. Since repeated collisions indicate a busy medium, however, Transmit Media Access Management attempts to adjust to the medium load by backing off (voluntarily delaying its own retransmissions to reduce its load on the medium). This is accomplished by expanding the interval from which the random retransmission time is selected on each retransmission attempt. Eventually, either the transmission succeeds, or the attempt is abandoned on the assumption that the medium has failed or has become overloaded.

At the receiving end, the bits resulting from a collision are received and decoded by the PLS just as are the bits of a valid frame. Fragmentary frames received during collisions are distinguished from valid transmissions by the Media Access Sublayer's Receive Media Access Management component.

4.1.3 Relationships to LLC Sublayer and Physical Layer

The CSMA/CD Media Access Sublayer provides services to the LLC Sublayer required for the transmission and reception of frames. Access to these services is specified in Section 4.3. The CSMA/CD Media Access Sublayer makes a best effort to acquire the medium and transfer a serial stream of bits to the PLS. Although certain errors are reported to the LLC, error recovery is not provided by MAC. Error recovery may be provided by the LLC or higher Sublayers.

4.1.4. CSMA/CD Access Method Functional Capabilities

The following is a summary of the functional capabilities of the IEEE 802 CSMA/CD Media Access sublayer. It is intended as a quick reference guide to the capabilities of the standard.

1. For Frame Transmission

- a) Accepts data from the LLC Sublayer and constructs a frame.
- b) Presents a bit-serial data stream to the physical layer for transmission on the medium.

NOTE: Assumes data passed from the LLC Sublayer are octet multiples.

2. For Frame Reception

- a) Receives a bit-serial data stream from the physical layer.
- b) Presents to the LLC Sublayer frames that are either broadcast frames or directly addressed to the local station.
- c) All frames not addressed to the receiving station are either discarded or passed to Network Management.

3. Defers transmission of a bit-serial stream whenever the physical medium is busy.

4. Appends proper FCS value to outgoing frames and verifies full octet boundary alignment.

5. Checks incoming frames for transmission errors via FCS and verifies full octet boundary alignment.

6. Delays transmission of frame bit stream for specified interframe gap period.

7. Aborts transmission when collision is detected.
8. Schedules retransmission after a collision until a specified retry limit is exceeded.
9. Enforces existing collision to ensure propagation throughout network by sending "jam" message to medium via physical layer.
10. Discards received transmissions that are less than a minimum length.
11. Appends preamble, start frame delimiter, Destination Address, Source Address, length count, and FCS to all frames, and appropriate padding for frames whose LLC data length is less than a minimum value.
12. Removes preamble, start frame delimiter, DA, SA, length, FCS, and padding (if necessary) from received frames.

4.2 IEEE 802.3 CSMA/CD MEDIA ACCESS METHOD: FORMAL SPECIFICATION

4.2.1 Introduction

A precise algorithmic definition is given in this section, providing a procedural model for the IEEE 802.3 CSMA/CD Media Access Method in form of a program in the language Pascal. Note that whenever there is any apparent ambiguity concerning the definition of some aspect of the IEEE 802 CSMA/CD Media Access Method, it is the Pascal procedural specification in Sections 4.2.7 through 4.2.10 which should be consulted for the definitive statement. Subsections 4.2.2 through 4.2.8 provide, in prose, a description of the access mechanism, along with the formal terminology to be used in the remaining subsections.

4.2.2 Overview of the Procedural Model

The functions of the IEEE 802.3 CSMA/CD Media Access Method are pre-se below, modeled as a program written in the language Pascal [See Appendix, Reference 4-1]. This procedural model is intended as the primary specification of the functions to be provided in any IEEE 802 CSMA/CD Media Access Sublayer implementation. It is important to distinguish, however, between the model and a real implementation. The model is optimized for simplicity and clarity of presentation, while any realistic implementation must place heavier emphasis on such constraints as efficiency and suitability to a particular implementation technology or computer architecture. In this context, several important properties of the procedural model must be considered.

4.2.2.1 Ground Rules for the Procedural Model

- a) First, it must be emphasized that *the description of the Media Access Sublayer in a programming language is in no way intended to imply that procedures must be implemented as a program executed by a computer*. The implementation may consist of any appropriate technology including hardware, firmware, software, or any combination.
- b) Similarly, it must be emphasized that it is the behavior of Media Access Sublayer implementations that must match the standard, not their internal structure. The internal details of the procedural model are useful only to the extent that they help specify that behavior clearly and precisely.
- c) The handling of incoming and outgoing frames is rather stylized in the procedural model, in the sense that frames are handled as single entities by most of the Media Access Sublayer and are only serialized for presentation to the Physical Layer. In reality, many implementations will instead handle frames serially on a bit, octet or word basis. This approach has not been reflected in the procedural model, since this would only complicate the description of the functions without changing them in any way.

d) The model consists of algorithms designed to be executed by a number of concurrent processes; these algorithms collectively implement the IEEE 802.3 CSMA/CD procedure. The timing dependencies introduced by the need for concurrent activity are resolved in two ways:

- Processes vs. External events: It is assumed that the algorithms are executed "very fast" relative to external events, in the sense that a process never falls behind in its work and fails to respond to an external event in a timely manner. For example, when a frame is to be received, it is assumed that the Media Access procedure ReceiveFrame is always called well before the frame in question has started to arrive.
- Processes vs. Processes: Among processes, no assumptions are made about relative speeds of execution. This means that each interaction between two processes must be structured to work correctly independent of their respective speeds. Note, however, that the timing of interactions among processes is often, in part, an indirect reflection of the timing of external events, in which case appropriate timing assumptions may still be made.

It is intended that the concurrency in the model reflect the parallelism intrinsic to the task of implementing the IEEE 802 LLC and Media Access procedures, although the actual parallel structure of the implementations is likely to vary.

4.2.2.2 Use of Pascal in the Procedural Model

Several observations need to be made about the way in which Pascal is used for the model, including:

a) Some limitations of the language have been circumvented in order to simplify the specification:

- 1) The elements of the program (variables, procedures, etc) are presented in logical groupings, in top-down order. Certain Pascal ordering restrictions have thus been circumvented to improve readability.
- 2) The *process* and *cycle* constructs of the Pascal derivative Concurrent Pascal [See Appendix, Reference 4-1] have been introduced to indicate the sites of autonomous concurrent activity. As used here, a process is simply a parameterless procedure that begins execution at "the beginning of time" rather than being invoked by a procedure call. A cycle statement represents the main body of a process and is executed repeatedly forever.
- 3) The lack of variable array bounds in the language has been circumvented by treating frames as if they are always of a single fixed size (which is never actually specified). In fact, of course, the size of a frame depends on the size of

its data field, hence the value of the "pseudo-constant" `frameSize` should be thought of as varying in the long-term, even though it is fixed for any given frame.

- 4) The use of a variant record to represent a frame (both as fields and as bits) follows the letter but not the spirit of the Pascal Report, since it allows the underlying representation to be viewed as two different data types.

b). The model makes no use of any explicit interprocess synchronization primitives. Instead, all interprocess interaction is done via carefully stylized manipulation of shared variables. For example, some variables are set by only one process and inspected by another process in such a manner that the net result is independent of their execution speeds. While such techniques are not generally suitable for the construction of large concurrent programs, they simplify the model and more nearly resemble the methods appropriate to the most likely implementation technologies (e.g. microcode, hardware state-machines, etc.)

4.2.2.3 Organization of the Procedural Model

The procedural model used here is based on five cooperating concurrent processes. Three are actually defined in the Media Access Sublayer. The remaining two processes are provided by the clients of the Media Access Sublayer (which may include the LLC Sublayer) and utilize the interface operations provided by the Media Access Sublayer. The five processes are thus:

Frame Transmitter Process Frame Receiver Process

Media Access Sublayer:

Bit Transmitter Process Bit Receiver Process

Deference Process

This organization of the model is illustrated in Figure 4-3 and reflects the fact that the communication of entire frames is initiated by the client of the Media Access Sublayer, while the timing of collision backoff and of individual bit transfers is based on interactions between the Media Access Sublayer and the Physical-Layer-dependent bit-time.

Figure 4-2 depicts the static structure of the procedural model, showing how the various processes and procedures interact by invoking each other. Figures 4-3 and 4-4 summarize the dynamic behavior of the model during transmission and reception, focusing on the steps that must be performed, rather than the procedural structure which performs them. The usage of the shared state variables is not depicted in the figures, but is described in the comments and prose in the following sections.

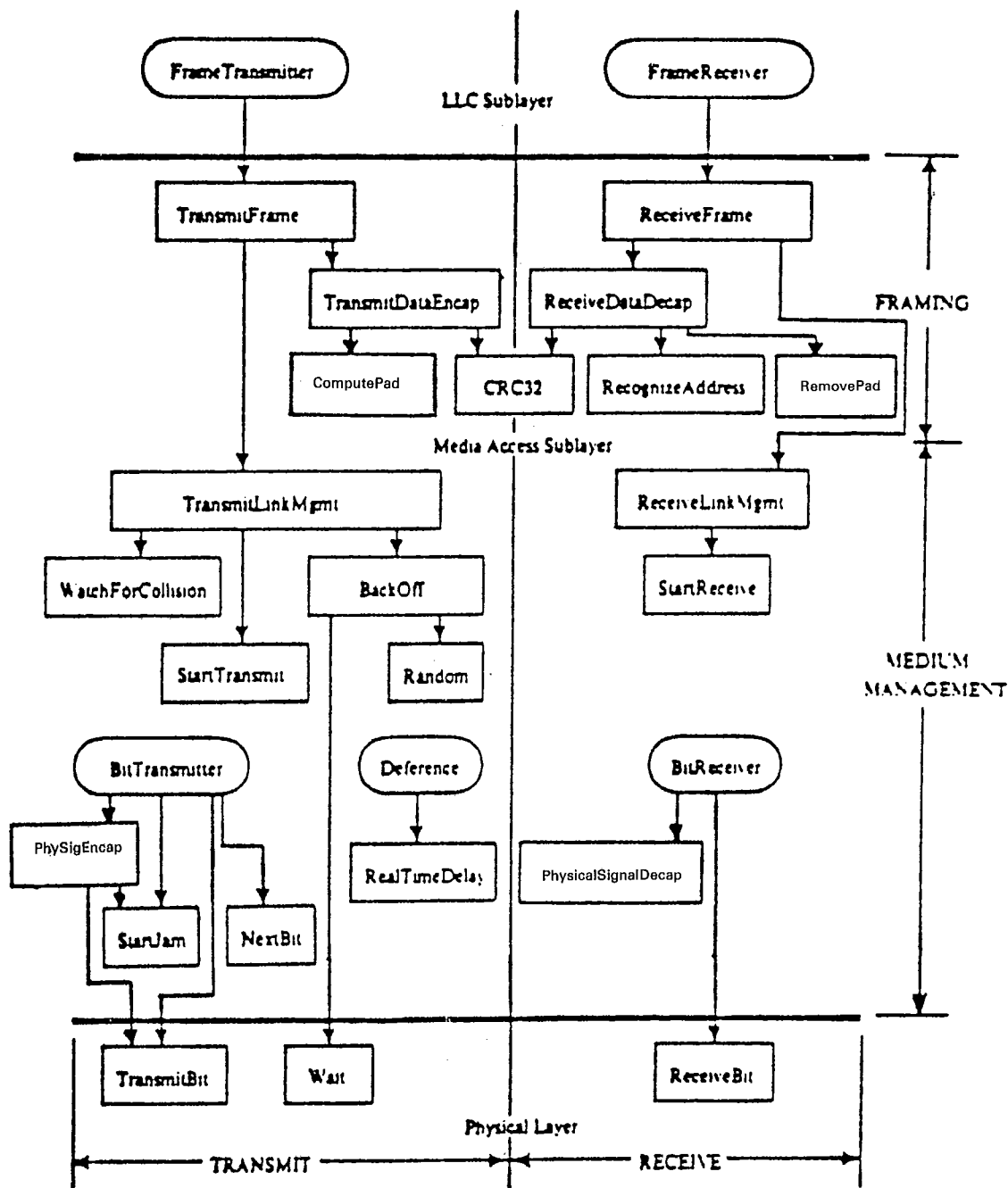


Fig. 4-2. Relationship among CSMA/CD procedures

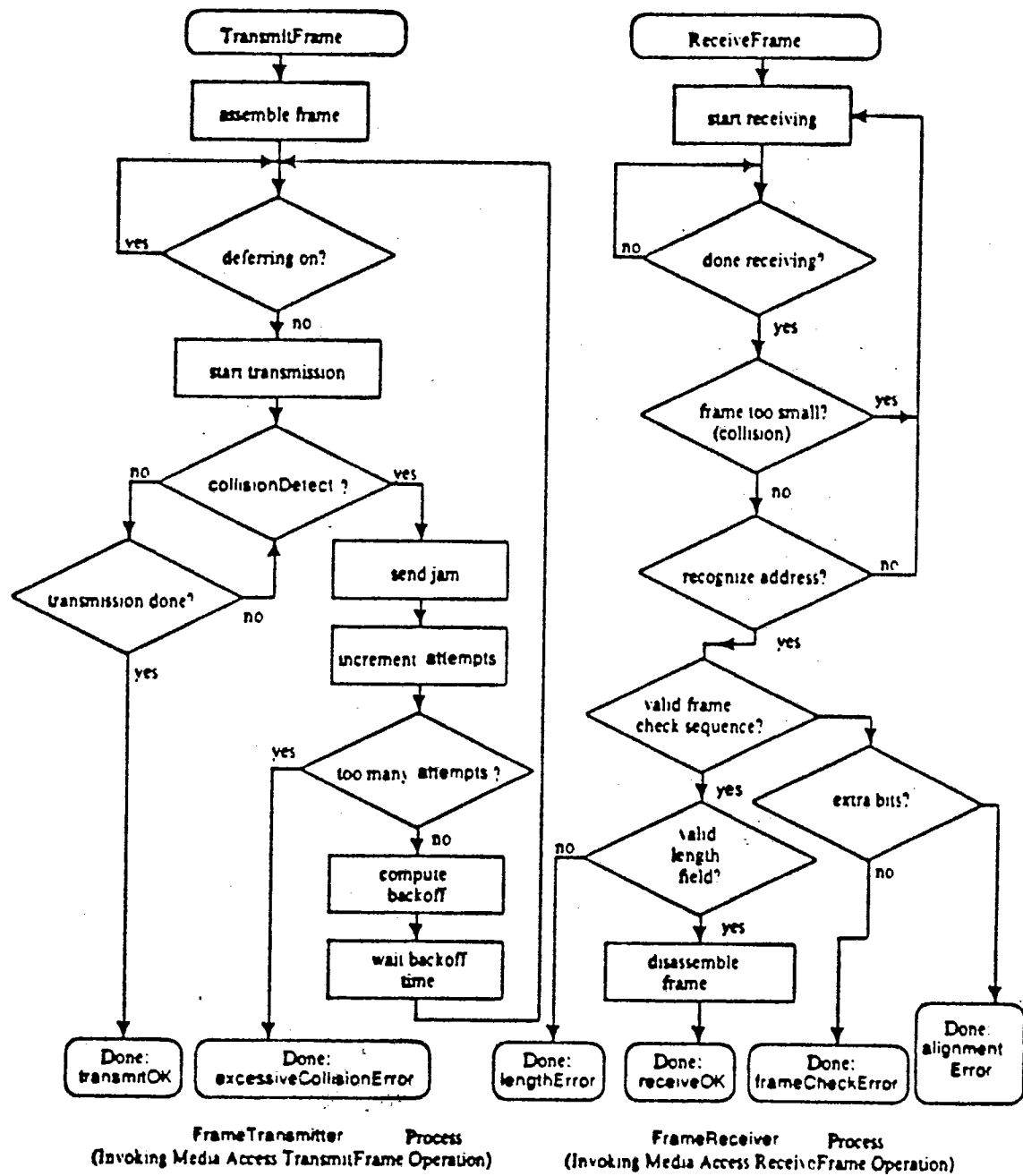


Fig. 4-3. Control Flow Summary

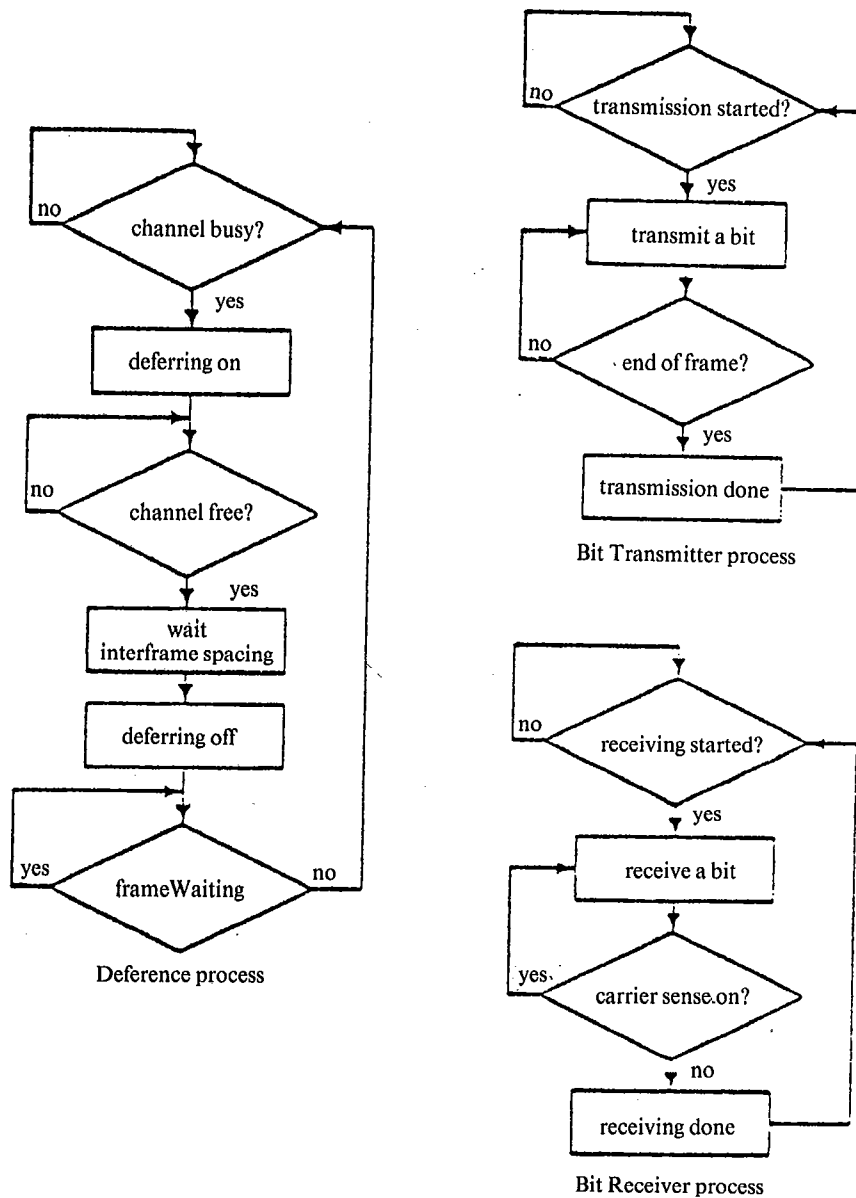


Fig. 4-4. Control Flow -- Media Access Sublayer

4.2.3. *Frame Transmission Model*

Frame transmission includes data encapsulation and Media Access management aspects:

Transmit Data Encapsulation includes the assembly of the outgoing frame (from the values provided by the LLC Sublayer) and frame check sequence generation.

Transmit Media Access Management includes carrier deference, inter-frame spacing, collision detection and enforcement, and collision backoff and retransmission.

4.2.3.1 *Transmit Data Encapsulation*

4.2.3.1.1 *Frame Assembly*

The fields of the CSMA/CD Media Access frame are set to the values provided by the LLC Sublayer as arguments to the TransmitFrame operation (see Section 4.3), with the exception of the padding necessary to enforce the minimum frame size, and the frame check sequence, which is set to the CRC value generated by the Media Access sublayer.

4.2.3.1.2 *Frame Check Sequence Generation*

The CRC value defined in Section 3.8 is generated and inserted in the frame check sequence field, following the fields supplied by the LLC Sublayer.

4.2.3.2. *Transmit Media Access Management*

4.2.3.2.1. *Carrier Deference*

Even when it has nothing to transmit, the CSMA/CD Media Access sublayer monitors the physical medium for traffic by watching the carrierSense signal provided by the PLS. Whenever the medium is busy, the CSMA/CD Media Access sublayer defers to the passing frame by delaying any pending transmission of its own. After the last bit of the passing frame (i.e., when carrierSense changes from true to false), the CSMA/CD Media Access sublayer continues to defer for a proper interframe spacing, interFrameSpacing (see Section 4.2.3.2.2).

If, at the end of the interframe spacing, a frame is waiting to be transmitted, transmission is initiated independent of the value of carrierSense. When transmission has completed (or immediately, if there was nothing to transmit) the CSMA/CD Media Access sublayer resumes its original monitoring of carrierSense.

When a frame is submitted by the LLC Sublayer for transmission, the transmission is initiated as soon as possible, but in conformance with the rules of deference stated above.

Note: It is possible for the PLS carrier sense indication to fail to be asserted briefly during a collision on the media. If the Deference process simply times the interFrame gap based on this indication it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in Sub-Section 4.2.8, are recommended when interFrame SpacingPart1 is other than zero:

1) Upon completing a transmission, start timing the interpacket gap as soon as transmitting and carrierSense are both false, 2) when timing an interFrame gap following reception, reset the interFrame gap timing if carrierSense becomes true during the first $\frac{2}{3}$ of the interFrame gap timing interval. During the final $\frac{1}{3}$ of the interval the timer must not be reset to ensure fair access to the medium. An initial period shorter than $\frac{2}{3}$ of the interval is permissible including zero.

4.2.3.2.2. Interframe Spacing

As defined in Section 4.2.3.2.1, the rules for deferring to passing frames insure a minimum interframe spacing of interFrameSpacing seconds. This is intended to provide interframe recovery time for other CSMA/CD sublayers and for the physical medium.

Note that interFrameSpacing is the minimum value of the interframe spacing. If necessary for implementation reasons, a transmitting sublayer may use a larger value with a resulting decrease in its throughput. The larger value is determined by the parameters of the implementation, see Section 4.4.

4.2.3.2.3. Collision Handling

Once a CSMA/CD sublayer has finished deferring and has started transmission, it is still possible for it to experience contention for the medium. Collisions can occur until acquisition of the network has been accomplished through the deference of all other stations' CSMA/CD sublayers.

The dynamics of collision handling are largely determined by a single parameter called the slot time. This single parameter describes three important aspects of collision handling:

- It is an upper bound on the acquisition time of the medium.
- It is an upper bound on the length of a frame fragment generated by a collision.
- It is the scheduling quantum for retransmission.

In order to fulfill all three functions, the slot time must be larger than the sum of the Physical Layer round-trip propagation time and the Media Access Layer maximum jam time. The slot time is determined by the parameters of the implementation, see Section 4.4.

4.2.3.2.4. Collision Detection and Enforcement

Collisions are detected by monitoring the collisionDetect signal provided by the Physical Layer. When a collision is detected during a frame transmission, the transmission is not terminated immediately. Instead, the transmission continues until additional bits specified by JamSize have been transmitted (counting from the time collisionDetect went on). This collision enforcement or "jam" guarantees that the duration of the collision is sufficient to insure its detection by all transmitting stations on the network. The content of the jam is unspecified; it may be any fixed or variable pattern convenient to the Media Access implementation, however, the implementation shall not be intentionally designed to be the 32-bit CRC value corresponding to the (partial) frame transmitted prior to the jam.

4.2.3.2.5. Collision Backoff and Retransmission

When a transmission attempt has terminated due to a collision, it is retried by the transmitting CSMA/CD sublayer until either it is successful, or a maximum number of attempts, attemptLimit, have been made and all have terminated due to collisions. Note that all attempts to transmit a given frame are completed before any subsequent outgoing frames are transmitted. The scheduling of the retransmissions is determined by a controlled randomization process called "truncated binary exponential backoff". At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to retransmit the frame. The delay is an integral multiple of slotTime. The number of slot times to delay before the nth retransmission attempt is chosen as a uniformly distributed random integer r in the range $0 \leq r < 2^{*k}$ where $k = \min(n, 10)$. If all attemptLimit attempts fail, this event is reported as an error. Algorithms used to generate the integer r should be designed to minimize the correlation between the numbers generated by any 2 stations at any given time.

Note that the values given above define the most aggressive behavior that a station may exhibit in attempting to retransmit after a collision. In the course of implementing the retransmission scheduling procedure, a station may introduce extra delays which will degrade its own throughput, but in no case may a station's retransmission scheduling result in a lower average delay between retransmission attempts than the procedure defined above.

4.2.3.3 Minimum Frame Size

The CSMA/CD Media Access mechanism requires that a minimum frame length of minFrameSize bits be transmitted. If frameSize is less than min-

FrameSize, then the CSMA/CD Medium Access Sublayer must append extra bits, or a pad, in units of octets, after the end of the LLC data field but prior to calculating, and appending, the FCS. The number of pad units must be sufficient to ensure that the frame, from the destination address field through the frame check sequence field inclusive, is at least minFrameSize bits. The content of the pad is unspecified.

4.2.4. *Frame Reception Model*

CSMA/CD Media Access Sublayer frame reception includes both data decapsulation and Media Access management aspects:

Receive Data Decapsulation comprises address recognition, frame check sequence validation, and frame disassembly to pass the fields of the received frame to the LLC Sublayer.

Receive Media Access Management comprises recognition of collision fragments from incoming frames and truncation of frames to octet boundaries.

4.2.4.1. *Receive Data Decapsulation*

4.2.4.1.1. *Address Recognition*

The CSMA/CD sublayer is capable of recognizing individual and group addresses.

Individual Addresses: The CSMA/CD sublayer recognizes and accepts any frame whose destination field contains the individual address of the station.

Group Addresses: The CSMA/CD Media Access Sublayer recognizes and accepts any frame whose destination field contains the broadcast address.

The CSMA/CD sublayer is capable of activating some number of group addresses as specified by higher layers. The CSMA/CD sublayer recognizes and accepts any frame whose destination field contains an active group address. An active group address may be deactivated.

4.2.4.1.2. *Frame Check Sequence Validation*

FCS validation is essentially identical to FCS generation. If the bits of the incoming frame (exclusive of the FCS field itself) do not generate a CRC value identical to the one received, an error has occurred and the frame is identified as invalid.

4.2.4.1.3. *Frame Disassembly*

Upon recognition of the start frame delimiter at the end of the preamble sequence, the CSMA/CD Media Access Sublayer accepts the frame. If there are no errors, the frame is disassembled and the fields are pass-

ed to the LLC Sublayer via the output parameters of the ReceiveFrame operation.

4.2.4.2. Receive Media Access Management

4.2.4.2.1. Framing

The CSMA/CD sublayer recognizes the boundaries of an incoming frame by monitoring the carrierSense signal provided by the PLS. There are two possible length errors that can occur, which indicate ill-framed data: the frame may be too long, or its length may not be an integral number of octets.

Maximum Frame Size: The receiving CSMA/CD sublayer is not required to enforce the frame size limit, but it is allowed to truncate frames longer than MaxFrameSize octets and report this event as an (implementation-dependent) error.

Integral Number of Octets in Frame: Since the format of a valid frame specifies an integral number of octets, only a collision or an error can produce a frame with a length that is not an integral multiple of 8. Complete frames (i.e., not rejected as collision fragments; see Section 4.2.4.2.2) that do not contain an integral number of octets are truncated to the nearest octet boundary. If frame check sequence validation detects an error in such a frame, the status code alignmentError is reported.

4.2.4.2.2. Collision Filtering

The smallest valid frame must be at least one slotTime in length. This determines the minFrameSize. Any frame containing less than minFrameSize bits is presumed to be a fragment resulting from a collision. Since occasional collisions are a normal part of the Media Access management procedure, the discarding of such a fragment is not reported as an error to the LLC Sublayer.

4.2.5 Preamble Generation

In an IEEE 802.3 Local Area Network implementation, most of the Physical Layer components are allowed to provide valid output some number of bit times after being presented valid input. Thus it is necessary for a preamble to be sent before the start of data, to allow the PLS circuitry to reach its steady-state. Upon request by TransmitLinkMgmt to transmit the first bit of a new frame, PhysicalSignalEncap shall first transmit the preamble, a bit sequence used for physical medium stabilization and synchronization, followed by the start frame delimiter. If, while transmitting this stream, the PLS asserts the collision detect signal, the preamble pattern is:
The preamble pattern is:

10101010 10101010 10101010 10101010 10101010 10101010 10101010

The bits are transmitted in order, from left to right. The nature of the pattern is such that when encoded using Manchester it appears as a periodic waveform on the medium that enables bit synchronization. It should be noted that the preamble ends with a '0'.

4.2.6 Start Frame Sequence

The PLS recognizes the presence of activity on the medium through the carrier sense signal. This is the first indication that the frame reception process should begin. Upon reception of the sequence 10101011 immediately following a latter part of the preamble pattern, PhysicalSignalDecap shall begin passing successive bits to ReceiveLink-Mgmt for passing to the LLC Sublayer.

4.2.7 Global Declarations

This section provides detailed formal specifications for the IEEE 802 CSMA/CD Media Access Mechanism. It is a specification of generic features and parameters to be used in systems implementing this IEEE 802 media access method. Section 4.4 provides values for these sets of parameters for recommended implementations of this media access mechanism.

4.2.7.1 Common Constants and Types

The following declarations of constants and types are used by the frame transmission and reception sections of each CSMA/CD sublayer:

const

```
addressSize = ... ; {16 or 48 bits in compliance with Sec. 4.3.3}
lengthSize = 16; {in bits}
LLCdataSize = ...; {LLC Data, see section 4.2.2.2, note 3}
padSize = ...; {in bits, = max (0, minFrameSize - (2*addressSize
    + lengthSize + LLCdataSize + crcSize))}.
dataSize = ...; {=LLCdataSize + padSize }
crcSize = 32; {32 bit CRC = 4 octets}
frameSize = ...; {=2* addressSize + lengthSize + dataSize +
    crcSize, see section 4.2.2.2, note 3}
minFrameSize = ... ; {in bits, implementation dependent,
    see Section 5}
slotTime = ... ; {unit of time for collision handling,
    implementation dependent, see section 5}
preambleSize = ... ; {in bits, physical medium dependent}
sfdSize = 8; {8 bit start frame delimiter}
headerSize = ...; {sum of preambleSize and sfdSize}
```

type

```
Bit = 0..1;
AddressValue = array [1..addressSize] of Bit;
LengthValue = array [1..lengthSize] of Bit;
DataValue = array [1..dataSize] of Bit;
```

```

CRCValue = array [1..crcSize] of Bit;
PreambleValue = array [1..preambleSize] of Bit;
SfdValue = array [1..sfdSize] of Bit;
ViewPoint = (fields, bits); {Two ways to view the contents
                             of a frame}
HeaderViewPoint = (headerFields, headerBits);
Frame = record {Format of Media Access frame}
  case view: ViewPoint of
    fields: (
      destinationField: AddressValue;
      sourceField: AddressValue;
      lengthField: LengthValue;
      dataField: DataValue;
      fcsField: CRCValue);
    bits: (contents: array [1..frameSize] of Bit)
  end; {Frame}
Header = record {Format of preamble and start frame delimiter}
  case headerView: HeaderViewPoint of
    headerFields: (
      preamble: PreambleValue;
      sfd: SfdValue);
    headerBits: (
      headerContents: array [1..headerSize] of Bit)
  end; {defines header for MAC frame}

```

4.2.7.2 Transmit State Variables

The following items are specific to frame transmission. (See also Section 4.4)

```

const
  interFrameSpacing = ... ; {minimum time between frames}
  interFrameSpacingPart1 = ... ; {duration of first portion of
                                   interFrame timing. In range
                                   0 up to 2/3 interFrameSpacing}
  interFrameSpacingPart2 = ... ; {duration of remainder of inter-
                                   Frame timing. Equal to inter-
                                   FrameSpacing - interFrameSpacing-
                                   Part1}
  attemptLimit = ... ; {Max number of times to attempt trans-
                        mission}
  backOffLimit = ... ; {Limit on number of times to back off}
  jamSize = ... ; {in bits: the value depends upon medium and
                  collision detect implementation}

var
  outgoingFrame: Frame; {The frame to be transmitted}
  outgoingHeader: Header;
  currentTransmitBit, lastTransmitBit: 1..frameSize;
  {Positions of current and last outgoing bits in outgoingFrame}
  lastHeaderBit: 1..headerSize;
  deferring: Boolean; {Implies any pending transmission}

```

```

        must wait for the medium to clear}
frameWaiting: Boolean; {Indicates that outgoingFrame is deferring
attempts: 0..attemptLimit; {Number of transmission attempts on
                           outgoingFrame}
newCollision: Boolean; {Indicates that a collision has occurred
                        but has not yet been jammed}
transmitSucceeding: Boolean; {Running indicator of whether
                             Transmission is succeeding}

```

4.2.7.3 Receive State Variables

The following items are specific to frame reception. (See also Section 4.4)

```

var
incomingFrame: Frame; {The frame being received}
currentReceiveBit: 1..frameSize; {Position of current bit in
                                   incomingFrame}
receiving: Boolean; {Indicates that a frame reception is in
                    progress}
excessBits: 0..7; {Count of excess trailing bits beyond octet
                  boundary}
receiveSucceeding: Boolean; {Running indicator of whether
                             reception is succeeding}
validLength: Boolean; {Indicator of whether received frame
                      has a length error}

```

4.2.7.4 Summary of Interlayer Interfaces

The interface to the LLC Sublayer, defined in Section 4.4.2, is summarized below:

```

type
TransmitStatus = (transmitOK, excessiveCollisionError);
                 {Result of TransmitFrame operation}
ReceiveStatus = (receiveOK, lengthError, frameCheckError,
                 alignmentError);
                 {Result of ReceiveLFrame operation}

function TransmitFrame (
destinationParam: AddressValue;
sourceParam: AddressValue;
lengthParam: LengthValue;
dataParam: DataValue): TransmitStatus; {Transmits one frame}

function ReceiveFrame (
var destinationParam: AddressValue;
var sourceParam: AddressValue;
var lengthParam: LengthValue;
var dataParam: DataValue): ReceiveStatus; {Receives one frame}

```

The interface to the Physical Layer, defined in Section 4.4.3, is summarized below:

```

var
  carrierSense: Boolean; {Indicates incoming bits}
  transmitting: Boolean; {Indicates outgoing bits}
  was Transmitting: Boolean; {Indicates transmission in
                              progress or just completed}
  collisionDetect: Boolean; {Indicates medium contention}
  procedure TransmitBit (bitParam: Bit); {Transmits one bit}
  function ReceiveBit: Bit; {Receives one bit}
  procedure Wait (bitTimes: integer); {Waits for indicated
                                       number of bit-times}

```

4.2.7.5 State Variable Initialization

The procedure Initialize must be run when the Media Access Sublayer begins operation, before any of the processes begin execution. Initialize sets certain crucial shared state variables to their initial values. (All other global variables are appropriately reinitialized before each use.) Initialize then waits for the the medium to be idle, and starts operation of the various processes.

```

procedure Initialize;
begin
  frameWaiting := false;
  deferring := false;
  newCollision := false;
  transmitting := false; {In interface to Physical Layer;
                          see below}
  receiving := false;
  while carrierSense do nothing;
  {Start execution of all processes}
end; {Initialize}

```

4.2.8 Frame Transmission

The algorithms in this section define Media Access Sublayer frame transmission.

The function TransmitFrame implements the frame transmission operation provided to the LLC Sublayer:

```

function TransmitFrame (
  destinationParam: AddressValue;
  sourceParam: AddressValue;
  lengthParam: LengthValue;
  dataParam: DataValue): TransmitStatus;
  procedure TransmitDataEncap; ... {nested procedure; see
  body below}
begin
  TransmitDataEncap;

```

```

    TransmitFrame := TransmitLinkMgmt
end; {TransmitFrame}

```

First, TransmitFrame calls the internal procedure TransmitDataEncap to construct the frame. Next, TransmitLinkMgmt is called to perform the actual transmission. The TransmitStatus returned indicates the success or failure of the transmission attempt.

TransmitDataEncap builds the frame and places the 32-bit CRC in the frame check sequence field:

```

procedure TransmitDataEncap;
begin
    with outgoingFrame do
        begin {assemble frame}
            view := fields;
            destinationField := destinationParam;
            sourceField := sourceParam;
            lengthField := lengthParam;
            dataField := ComputePad (lengthParam, dataParam);
            fcsField := CRC32(outgoingFrame);
            view := bits
        end {assemble frame}
    With outgoingHeader do
        begin
            headerView := headerFields;
            preamble := ...; {* '1010...10', LSB to MSB*}
            sfd := ...; {* '10101011', LSB to MSB*}
            headerView := headerBits
        end
    end; {TransmitDataEncap}

```

ComputePad appends an array of arbitrary bits to the LLCdataField to pad the frame to the minimum frame size.

```

function ComputePad(
    var lengthParam:LengthValue
    var dataParam:DataValue) :DataValue;
begin
    ComputePad := {Append an array of size PadSize of
                    arbitrary bits to the LLCdataField}
end; {ComputePadParam}

```

TransmitLinkMgmt attempts to transmit the frame, deferring first to any passing traffic. If a collision occurs, transmission is terminated properly and retransmission is scheduled following a suitable backoff interval:

```

function TransmitLinkMgmt: TransmitStatus;
begin
    attempts := 0; transmitSucceeding := false;
    while attempts < attemptLimit and not transmitSucceeding do

```

```

begin {loop}
  if attempts > 0 then BackOff;
  frameWaiting := true;
  while deferring do nothing; {defer to passing frame, if any}
  frameWaiting := false;
  StartTransmit;
  while transmitting do WatchForCollision;
  attempts := attempts+1
end; {loop}
if transmitSucceeding then TransmitLinkMgmt := transmitOK
else TransmitLinkMgmt := excessiveCollisionError
end; {TransmitLinkMgmt}

```

Each time a frame transmission attempt is initiated, StartTransmit is called to alert the BitTransmitter process that bit transmission should begin:

```

procedure StartTransmit;
begin
  currentTransmitBit := 1;
  lastTransmitBit := frameSize;
  transmitSucceeding := true;
  transmitting := true;
  lastHeaderBit := headerSize
end; {StartTransmit}

```

Once frame transmission has been initiated, TransmitLinkMgmt monitors the medium for contention by repeatedly calling WatchForCollision:

```

procedure WatchForCollision;
begin
  if transmitSucceeding and collisionDetect then
  begin
    newCollision := true;
    transmitSucceeding := false
  end
end; {WatchForCollision}

```

WatchForCollision, upon detecting a collision, updates newCollision to insure proper jamming by the BitTransmitter process.

After transmission of the jam has completed, if TransmitLinkMgmt determines that another attempt should be made, BackOff is called to schedule the next attempt to retransmit the frame.

```

var maxBackOff: 2..1024; {Working variable of BackOff}

```

```

procedure BackOff;
begin
  if attempts = 1 then maxBackOff := 2
  else if attempts <= backoffLimit
  then maxBackOff := maxBackOff*2;

```

```

    Wait(slotTime*Random(0, maxBackOff))
end; {BackOff}

```

```

function Random (low, high: integer): integer;

```

```

begin

```

```

    Random := ...{uniformly distributed random integer r such
                    that low ≤ r < high}

```

```

end; {Random}

```

BackOff performs the truncated binary exponential backoff computation and then waits for the selected multiple of the slot time.

The Deference process runs asynchronously to continuously compute the proper value for the variable deferring.

```

process Deference;

```

```

begin

```

```

    cycle{main loop}

```

```

        while not carrierSense do nothing; {watch for
                                            carrier to appear}

```

```

        deferring := true; {delay start of new transmissions}
        wasTransmitting:=transmitting;

```

```

        while carrierSense or transmitting do

```

```

            wasTransmitting: = wasTransmitting or transmitting;

```

```

        if wasTransmitting do

```

```

            begin

```

```

                StartRealTimeDelay; {time out first part
                                    interframe gap}

```

```

                while RealTimeDelay(interFrameSpacingPart1)
                    do nothing

```

```

            end

```

```

        else

```

```

            begin

```

```

                StartRealTimeDelay;

```

```

                repeat

```

```

                    while carrierSense do StartRealTimeDelay

```

```

                    until not RealTimeDelay(interFrameSpacingPart1)

```

```

                end;

```

```

                StartRealTimeDelay; {time out second part interframe
                                    gap}

```

```

                while RealTimeDelay(interFrameSpacingPart2) do nothing;

```

```

                deferring: = false; {allow new transmissions to proceed}

```

```

                while frameWaiting do nothing; {allow waiting trans-
                                                mission if any}

```

```

            end {main loop}

```

```

end; {Deference}

```

```

procedure StartRealTimeDelay
begin
  {reset the realtimer and start it timing}
end; {StartRealTimeDelay}

function RealTimeDelay (usec:real): Boolean;
begin
  {return the value true if the specified number of
  microseconds have not elapsed since the most recent
  invocation of StartRealTimeDelay, otherwise return
  the value false}
end; {RealTimeDelay}

```

The BitTransmitter process runs asynchronously, transmitting bits at a rate determined by the Physical Layer's TransmitBit operation:

```

process BitTransmitter;
begin
  cycle {outer loop}
  If transmitting then
    begin {inner loop}
      PhysicalSignalEncap; {Send preamble and start of frame
                           delimiter}
      while transmitting do
        begin
          TransmitBit(outgoingFrame[currentTransmitBit]); {send
                                                             next bit to Physical Layer}
          if newCollision then StartJam else NextBit
        end;
      end; {inner loop}
    end; {outer loop}
end; {BitTransmitter}

procedure PhysicalSignalEncap;
begin
  while currentTransmitBit <= lastHeaderBit do
    begin
      TransmitBit(outgoingHeader[currentTransmitBit]); {transmit
                                                         header one bit at a time}
      currentTransmitBit := currentTransmitBit + 1;
    end
    if newCollision then StartJam else
      currentTransmitBit := 1
  end; {PhysicalSignalEncap}

procedure NextBit;
begin
  currentTransmitBit := currentTransmitBit+1;
  transmitting := (currentTransmitBit <= lastTransmitBit)
end; {NextBit}

```