INTERNATIONAL STANDARD

ISO 8802-2
ANSI/IEEE Std 802.2

First edition
1989-12-31

Information processing systems — Local area networks —

Part 2 : Logical link control

Systèmes de traitement de l'information — Réseaux locaux — Partie 2 : Contrôle de liaison logique

Reference number
ISO 8802-2 : 1989 (E)
ANSI/IEEE
Std 802.2-1989

First Printing
December 1989

ISBN 1-55937-019-X
Library of Congress Catalog Card Number 88-46183

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

International Standard ISO 8802-2 : 1989
ANSI/IEEE Std 802.2-1989
(Revision of ANSI/IEEE Std 802.2-1985)

Information processing systems—Local area networks—

Part 2: Logical link control

Sponsor

Technical Committee on Computer Communications of the IEEE Computer Society

Approved August 17, 1989

IEEE Standards Board

Approved January 12, 1990

American National Standards Institute

Approved 1989 by the

International Organization for Standardization

Adopted as an International Standard by the International Organization for Standardization

Published by
The Institute of Electrical and Electronics Engineers, Inc

AMERICAN NATIONAL STANDARD

International Standard ISO 8802-2 : 1989

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

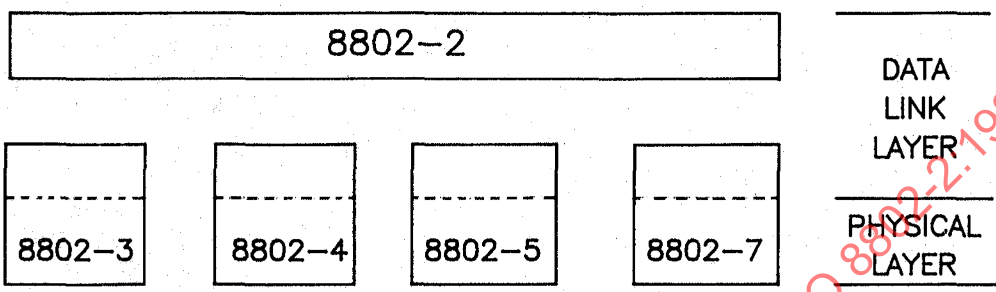
Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75% approval by the member bodies voting.

In 1985, ANSI/IEEE Std 802.2-1985 was adopted by ISO Technical Committee 97, *Information processing systems*, as draft International Standard ISO 8802-2. Following the procedures described above, the Standard was subsequently approved by ISO in the form of this new edition, which is published as International Standard ISO 8802-2 : 1989.

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

Foreword to International Standard ISO 8802-2 : 1989

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 [ANSI/IEEE Std 802.3-1988], a bus utilizing CSMA/CD as the access method,
(2) ISO 8802-4 [ANSI/IEEE Std 802.4-1985], a bus utilizing token passing as the access method,
(3) ISO 8802-5 [IEEE Std 802.5-1989], a ring utilizing token passing as the access method,
(4) ISO 8802-7, a ring utilizing slotted ring as the access method.

ISO 8802-2 [IEEE Std 802.2-1989], 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 8802-2 : 1989 and IEEE 802.2-1989 standards. ISO and IEEE each have unique foreword sections.

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:

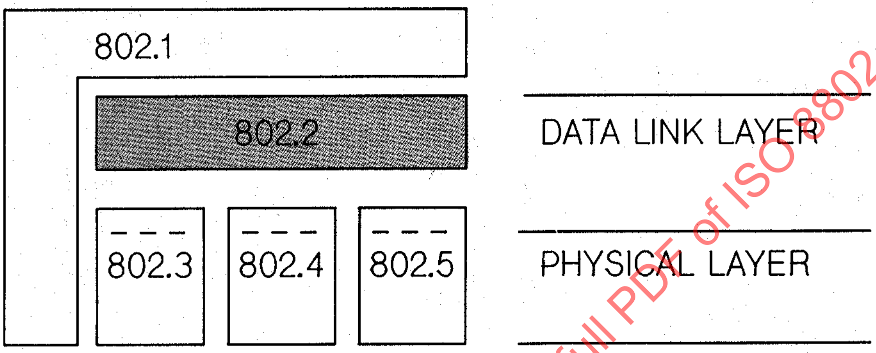
Secretary, IEEE Standards Board
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331
USA

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 does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards documents.

Foreword to IEEE Std 802.2-1989
(Revision of ANSI/IEEE Std 802.2-1985)

(This Foreword is not a part of ISO 8802-2 : 1989 or of IEEE Std 802.2-1989.)

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 Reference Model. 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) ANSI/IEEE Std 802.3-1988 [ISO 8802-3], a bus utilizing CSMA/CD as the access method,
(2) ANSI/IEEE Std 802.4-1985 [ISO 8802-4], a bus utilizing token passing as the access method,
(3) IEEE Std 802.5-1989 [ISO 8802-5], a ring utilizing token passing as the access method.

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

IEEE P802.1 describes the relationship among these standards and their relationship to the ISO Open Systems Interconnection Reference Model in more detail. This companion document also will contain networking management standards and information on internetworking.

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

At the time of approval of this standard in 1983, the following members were participants of IEEE Project 802 Working Group:

David E. Carlson, *Chairman*

Om Agrawal
Phil Arneth
Jeff Bobzin
Mark Bauer
Le Biu
Clyde Boenke
Bob Bowen
Bob Bridge*
Chuck Brill
Wayne Brodd*
Werner Bux
Jim Campbell
Tony Capel
Ron Cates
Rao Cherukuri
Po Chen*
Jade Chien
Mike Clader
Jerry Clancy*
Rich Collins
Steve Cooper
Bob Crowder*
Kirit Davé
John Davidson
Em Delahostria*
Jan Dolphin
Bob Donnan
Bob Douglas
Bill Durrenberger
Rich Fabbri
Eldon Feist*
Jim Field*
Larry Foltzer
Ron Floyd
Darrell Furlong
Mel Gable
Mike Garvey
Bud Glick
Arie Goldberg
Pat Gonia
Gordon Griffiths
Bob Grow
Maris Graube
Ed Harada
Lo Hsieh
Karen Hsing
Kevin Hughes
Marco Hurtado
Bob Husak
Dittmar Janetzky
Ross Jaibaji
George Jelatis
Gabor Kardos
Peggy Karp*
Kristin Kocan
Zak Kong*
Sy Korowitz
George Koshy
Don Kotas
Tony Kozlik
Mike Kryskow*
Dave Laffitte
Terry Lawell*
Ron Leuchs
Peter Lin
Jim Lindgren
Laurie Lindsey*
Bill Livingston*
Then Tang Liu
Don C. Loughry
Don J. Loughry
Bruce Loyer
Jerry Lurtz
Bill Miller
Ken Miller
Lou Mitta
Bob Moles
Jim Mollenauer
Ware Myers
Gene Nines
Bill Northup
Brian O'Neil*
Kul Padda
Mahendra Patel
Tom Phinney*
Juan Pimentel
Lavern Pope
Dave Potter
Dennis Quy
John Rance
Dan Ratner
Richard Read
Ted Rebenko
John Ricketson
Edouard Rocher
Rob Rosenthal*
Chip Schnarel
Walter Schreuer
Gerard Segarra
Dennis Sosnoski
Robert C. Smith
Mark Stahlman
Steve Stearns
Garry Stephens*
Mark Steiglitz*
Kathleen Sturgis
Bob Stover*
Bart Stuck
Dave Sweeton*
Dan Sze*
Vic Tarassov*
Angus Telfer*
Dave Thompson
Fouad Tobagi
Jean-Marie Tourret
Bo Viklund
Bruce Watson
Don Weir*
Dan Wendling
Walter Wheeler
Hugh White
Steve Whiteside
Earl Whitaker*
Ping Wu
Esin Ulug
Hiroshi Yoshida
Hank Zannini

* Principal contributors to Project 802.2

Additional individuals who made significant contributions were the following:

Don Andrews
Phil Arst
Ron Crane
Walt Elden
Ingrid Fromm
Atul Garg
Bryan Hoover
Andrew Huang
Hal Keen
Tony Lauck
Andy Luque
Dan Maltbie
Jane Munn
Wendell Nakamine
Liston Neely
Lee Neitzel
Dan Pitt
Robert Printis
Stephen Soto
Joshua Weiss

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

William B. Adams
Kit Athul
Chih-Tsai Chen
Michael H. Coden
Robert S. Crowder
George S. Curon
Mitchell Duncan
John E. Emrich
John W. Fendrich
Hal Folts
Harvey Freeman
D. G. Gan
Patrick Gonia
Ambuj Goyal
Maris Graube
J. Scott Haugdahl
Paul L. Hutton
Raj Jain
David M. Kollm
Anthony B. Lake
Mike Lawler
Jaiyong Lee
F. C. Lim
R. S. Little
William D. Livingston
Donald C. Loughry
Andy J. Luque
Richard Miller
Nirode C. Mohanty
John E. Montague
Kinji Mori
David J. Morris
M. Ravindranath Nayak
Arne A. Nilsson
Charles Oestereicher
Young Oh
Udo W. Pooch
John P. Riganati
Gary S. Robinson
Robert Rosenthal
Floyd Ross
S. I. Samoylenko
Julio Sanz Gonzalez
Norman Schneidewind
D. A. Sheppard
John Spragins
Carel M. Stillebroer
Fred Strauss
Peter Sugar
Efstathios D. Sykas
Daniel Sze
Nathan Tobol
L. David Umbaugh
Thomas A. Varetoni
James Vorhies
Don Weir
Earl J. Whitaker
George B. Wright
Oren Yuen

When the IEEE Standards Board approved this standard on August 17, 1989, it had the following membership:

Dennis Bodson, *Chairman* **Marco W. Migliaro,** *Vice Chairman*
Andrew G. Salem, *Secretary*

Arthur A. Blaisdell
Fletcher J. Buckley
Allen L. Clapp
James M. Daly
Stephen R. Dillon
Donald C. Fleckenstein
Eugene P. Fogarty
Jay Forster*
Thomas L. Hannan
Kenneth D. Hendrix
Theodore W. Hissey, Jr.
John W. Horch
David W. Hutchins
Frank D. Kirschner
Frank C. Kitzantides
Joseph L. Koepfinger*
Edward Lohse
John E. May, Jr.
Lawrence V. McCall
L. Bruce McClung
Donald T. Michael*
Richard E. Mosher
Stig Nilsson
L. John Rankine
Gary S. Robinson
Donald W. Zipse

*Member emeritus

Contents

FIGURES

TABLES

Information processing systems— Local area networks—

Part 2: Logical link control

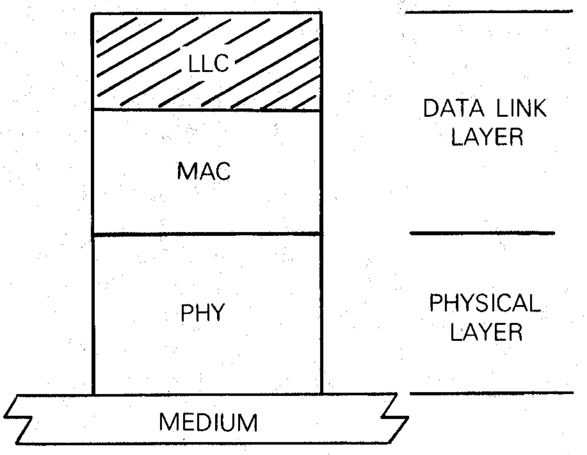
1. Introduction

1.1 Scope and Purpose. This International Standard is one of a set of international standards produced to facilitate the interconnection of computers and terminals on a Local Area Network (LAN). It is related to the other international standards by the Reference Model for Open Systems Interconnection.

NOTE: The exact relationship of the layers described in this International Standard to the layers defined by the OSI Reference Model is under study.

This International Standard describes the functions, features, protocol, and services of the Logical Link Control (LLC) sublayer in the ISO 8802 Local Area Network Protocol. The LLC sublayer constitutes the top sublayer in the data link layer (see Fig 1-1) and is common to the various medium access methods that are defined and supported by the ISO 8802 activity. Separate international standards describe each medium access method individually and indicate the additional features and functions that are provided by the Medium

Fig 1-1
Relationship to LAN Reference Model



Access Control (MAC) sublayer in each case to complete the functionality of the data link layer as defined in the LAN architectural reference model.

This International Standard describes the LLC sublayer service specifications to the network layer (Layer 3), to the MAC sublayer, and to the LLC sublayer management function. The service specification to the network layer provides a description of the various services that the LLC sublayer, plus underlying layers and sublayers, offer to the network layer, as viewed from the network layer. The service specification to the MAC sublayer provides a description of the services that the LLC sublayer requires of the MAC sublayer. These services are defined so as to be independent of the form of the medium access methodology, and of the nature of the medium itself. The service specification to the LLC sublayer management function provides a description of the management services that are provided to the LLC sublayer. All of the above service specifications are given in the form of primitives that represent in an abstract way the logical exchange of information and control between the LLC sublayer and the identified service function (network layer, MAC sublayer, or LLC sublayer management function). They do not specify or constrain the implementation of entities or interfaces.

This International Standard provides a description of the peer-to-peer protocol procedures that are defined for the transfer of information and control between any pair of data link layer service access points on a local area network. The LLC procedures are independent of the type of medium access method used in the particular local area network.

To satisfy a broad range of potential applications, two types of data link control operation are included (see Section 4). The first type of operation (see Section 6) provides a data-link-connectionless-mode service across a data link with minimum protocol complexity. This type of operation may be useful when higher layers provide any essential recovery and sequencing services so that these do not need replicating in the data link layer. In addition, this type of operation may prove useful in applications where it is not essential to guarantee the delivery of every data link layer data unit. This type of service is described in this International Standard in terms of "logical data links". The second type of operation (see Section 7) provides a data-link-connection-mode service across a data link comparable to existing data link control procedures provided in International Standards such as HDLC (see ISO 4335 : 1987 [1][1]). This service includes support of sequenced delivery of data link layer data units, and a comprehensive set of data link layer error recovery techniques. This second type of service is described in this international standard in terms of "data link connections".

This International Standard identifies two distinct "classes" of LLC operation. Class I provides data-link-connectionless-mode service only. Class II provides data-link-connection-mode service plus data-link-connectionless-mode service. Either class of operation may be supported.

The basic protocols described herein are peer protocols for use in multi-

[1] The numbers in brackets correspond to those of the references listed in 1.3.

station, multiaccess environments. Because of the multistation, multiaccess environment, it shall be possible for a station to be involved in a multiplicity of peer protocol data exchanges with a multiplicity of different stations over a multiplicity of different logical data links and/or data link connections that are carried by a single physical layer (PHY) over a single physical medium. Each unique to–from pairing at the data link layer shall define a separate logical data link or data link connection with separate logical parameters and variables. Except where noted, the procedures described in this chapter shall relate to each data link layer logical data link or data link connection separately and independently from any other logical data link or data link connection that might exist at the stations involved.

1.2 Standards Compatibility. The peer protocol procedures defined in Section 5 utilize some of the concepts and principles, as well as commands and responses, of the balanced data link control procedures known as Asynchronous Balanced Mode (ABM), as defined in ISO 7809 : 1984 [3]. (The ABM procedures provided the basis upon which the CCITT Recommendation X.25 Level 2 LAPB [4] procedures were defined.) The frame structure defined for the data link layer procedures as a whole is defined in part in Section 3 of this international standard and in part in those International Standards that define the various medium access control (MAC) procedures. The combination of a MAC sublayer address and an LLC sublayer address is unique to each data link layer service access point in the local area network.

NOTE: This division of data link layer addressing space into separate MAC and LLC address fields is not presently a part of any present ISO data link layer International Standard.

1.3 References

[1] ISO 4335 : 1987, Information processing systems, Data communications, High-level data link control elements of procedures.[2]

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

[3] ISO 7809 : 1984, Information processing systems, Data communications, High-level data link control procedures, Consolidation of classes of procedures.

[4] CCITT Recommendation X.25, Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit.[3]

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

[3] CCITT 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, Sales Section, Place des Nations, CH-1211, Genève 20, Switzerland/Suisse.

[5] CCITT Recommendation X.200, Reference model on open systems interconnection for CCITT applications.

1.4 Acronyms and Definitions

1.4.1 Acronyms

ABM	Asynchronous Balanced Mode
ACK	ACKnowledge
ADM	Asynchronous Disconnected Mode
C	Command
CCITT	International Telegraph and Telephone Consultative Committee
C/R	Command/Response
DA	Destination Address
DCE	Data Circuit-terminating Equipment
DIS	Draft International Standard
DISC	DISConnect
DM	Disconnected Mode
DSAP	Destination Service Access Point
DTE	Data Terminal Equipment
F	Final
FCS	Frame Check Sequence
FRMR	FRaMe Reject
HDLC	High-level Data Link Control
I	Information
I	Information transfer format
ISO	International Organization for Standardization
LAN	Local Area Network
LAPB	Link Access Procedure, Balanced
LLC	Logical Link Control
LSAP	Link layer Service Access Point
LSB	Least Significant Bit
LSDU	Link layer Service Data Unit
M	Modifier function bit
MAC	Medium Access Control
N(R)	Receive sequence Number
N(S)	Send sequence Number
OSI	Open Systems Interconnection
P	Poll
PDU	Protocol Data Unit
P/F	Poll/Final
PHY	PHYsical
R	Response
REJ	REJect
RNR	Receive Not Ready
RR	Receive Ready
S	Supervisory format

S	Supervisory function bit
SA	Source Address
SABME	Set Asynchronous Balanced Mode Extended
SAP	Service Access Point
SSAP	Source Service Access Point
TEST	TEST
U	Unnumbered format
UA	Unnumbered Acknowledgment
UI	Unnumbered Information
V(R)	Receive state Variable
V(S)	Send state Variable
XID	eXchange IDentification

1.4.2 Definitions. For the purpose of this International Standard the following definitions shall apply:

accept. The condition assumed by an LLC upon accepting a correctly received PDU for processing.

address fields (DSAP and SSAP). The ordered pair of service access point addresses at the beginning of an LLC PDU which identifies the LLC(s) designated to receive the PDU and the LLC sending the PDU. Each address field is one octet in length.

basic status. An LLC's capability to send or receive a PDU containing an information field.

command. In data communications, an instruction represented in the control field of a PDU and transmitted by an LLC. It causes the addressed LLC(s) to execute a specific data link control function.

command PDU. All PDUs transmitted by an LLC in which the C/R bit is equal to "0".

control field (C). The field immediately following the DSAP and SSAP address fields of a PDU. The content of the control field is interpreted by the receiving destination LLC(s) designated by the DSAP address field:

(a) As a command, from the source LLC designated by the SSAP address field, instructing the performance of some specific function; or
(b) As a response, from the source LLC designated by the SSAP address field.

data link. An assembly of two or more terminal installations and the interconnecting commmunications channel operating according to a particular method that permits information to be exchanged; in this context the term *terminal installation* does not include the data source and the data sink.

data link layer. The conceptual layer of control or processing logic existing in the hierarchical structure of a station that is responsible for maintaining control of the data link. The data link layer functions provide an interface between the station higher layer logic and the data link. These functions include address/control field interpretation, channel access and command PDU/response PDU generation, transmission, and interpretation.

exception condition. The condition asssumed by an LLC upon receipt of a command PDU which it cannot execute due to either a transmission error or an internal processing malfunction.

global (broadcasting) DSAP address. The predefined LLC DSAP address (all ones) used as a broadcast (all parties) address. It can never be the address of a single LLC on the data link.

group (multicast) DSAP address. A destination address assigned to a collection of LLCs to facilitate their being addressed collectively. The least significant bit shall be set equal to "1".

higher layer. The conceptual layer of control or processing logic existing in the hierarchical structure of a station that is above the data link layer and upon which the performance of data link layer functions are dependent; for example, device control, buffer allocation, LLC station management, etc.

information field (I). The sequence of octets occurring between the control field and the end of the LLC PDU. The information field contents of I, TEST, and UI PDUs are not interpreted at the LLC sublayer.

invalid frame. A PDU that either

(a) Does not contain an integral number of octets,
(b) Does not contain at least two address octets and a control octet, or
(c) Is identified by the physical layer or MAC sublayer as containing data bit errors.

LLC. That part of a data station that supports the logical link control functions of one or more logical links. The LLC generates command PDUs and response PDUs for transmission and interprets received command PDUs and response PDUs. Specific responsibilities assigned to an LLC include

(a) Initiation of control signal interchange,
(b) Organization of data flow,
(c) Interpretation of received command PDUs and generation of appropriate response PDUs, and
(d) Actions regarding error control and error recovery functions in the LLC sublayer.

MAC. That part of a data station that supports the medium access control functions that reside just below the LLC sublayer. The MAC procedures include framing/deframing data units, performing error checking, and acquiring the right to use the underlying physical medium.

N-layer. A subdivision of the architecture, constituted by subsystems of the same rank (N).

N-user. An N+1 entity that uses the services of the N-layer, and below, to communicate with another N+1 entity.

octet. A bit-oriented element that consists of eight contiguous binary bits.

peer protocol. The sequence of message exchanges between two entities in the same layer that utilize the services of the underlying layers to effect the successful transfer of data and/or control information from one location to another location.

priority (use in primitives). A parameter used to convey the priority required or desired.

protocol data unit (PDU). The sequence of contiguous octets delivered as a unit from or to the MAC sublayer. A valid LLC PDU is at least 3 octets in length, and contains two address fields and a control field. A PDU may or may not include an information field in addition.

response. In data communications, a reply represented in the control field of a response PDU. It advises the addressed destination LLC of the action taken by the source LLC to one or more command PDUs.

response PDU. All PDUs sent by a LLC in which the C/R bit is equal to "1".

service. The capabilities and features provided by an N-layer to an N-user.

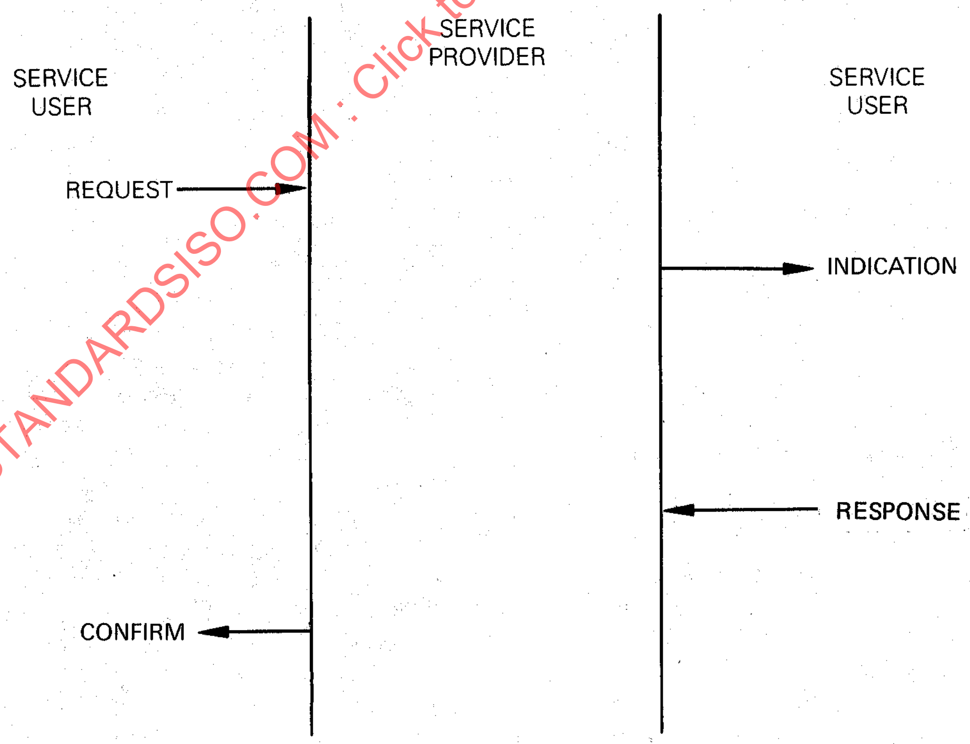
service class (use in primitives). A parameter used to convey the class of service required or desired.

2. LLC Sublayer Service Specifications

This section covers the services required of, or by, the LLC sublayer at the logical interfaces with the network layer, the MAC sublayer, and the LLC sublayer management function.

In general, the services of a layer (or sublayer) are the capabilities which it offers to a user in the next higher layer (or sublayer). In order to provide its service, a layer (or sublayer) builds its functions on the services which it requires from the next lower layer (or sublayer). Fig 2-1 illustrates this notion of service hierarchy and shows the relationship of the two correspondent N-users and their associated N-layer (or sublayer) peer protocol entities.

Fig 2-1
Service Primitives



Services are specified by describing the information flow between the N-user and the N-layer (or sublayer). This information flow is modeled by discrete, instantaneous events, which characterize the provision of a service. Each event consists of passing a service primitive from one layer (or sublayer) to the other through an N-layer (or sublayer) service access point associated with an N-user. Service primitives convey the information required in providing a particular service. These service primitives are an abstraction in that they specify only the service provided rather than the means by which the service is provided. This definition of service is independent of any particular interface implementation.

Services are specified by describing the service primitives and parameters that characterize each service. A service may have one or more related primitives that constitute the activity that is related to the particular service. Each service primitive may have zero or more parameters that convey the information required to provide the service.

Primitives are of four generic types:

REQUEST The request primitive is passed from the N-user to the N-layer (or sublayer) to request that a service be initiated.

INDICATION The indication primitive is passed from the N-layer (or sublayer) to the N-user to indicate an internal N-layer (or sublayer) event which is significant to the N-user. This event may be logically related to a remote service request, or may be caused by an event internal to the N-layer (or sub layer).

RESPONSE The response primitive is passed from the N-user to the N-layer (or sublayer) to complete a procedure previously invoked by an indication primitive.

CONFIRM The confirm primitive is passed from the N-layer (or sublayer) to the N-user to convey the results of one or more associated previous service request(s).

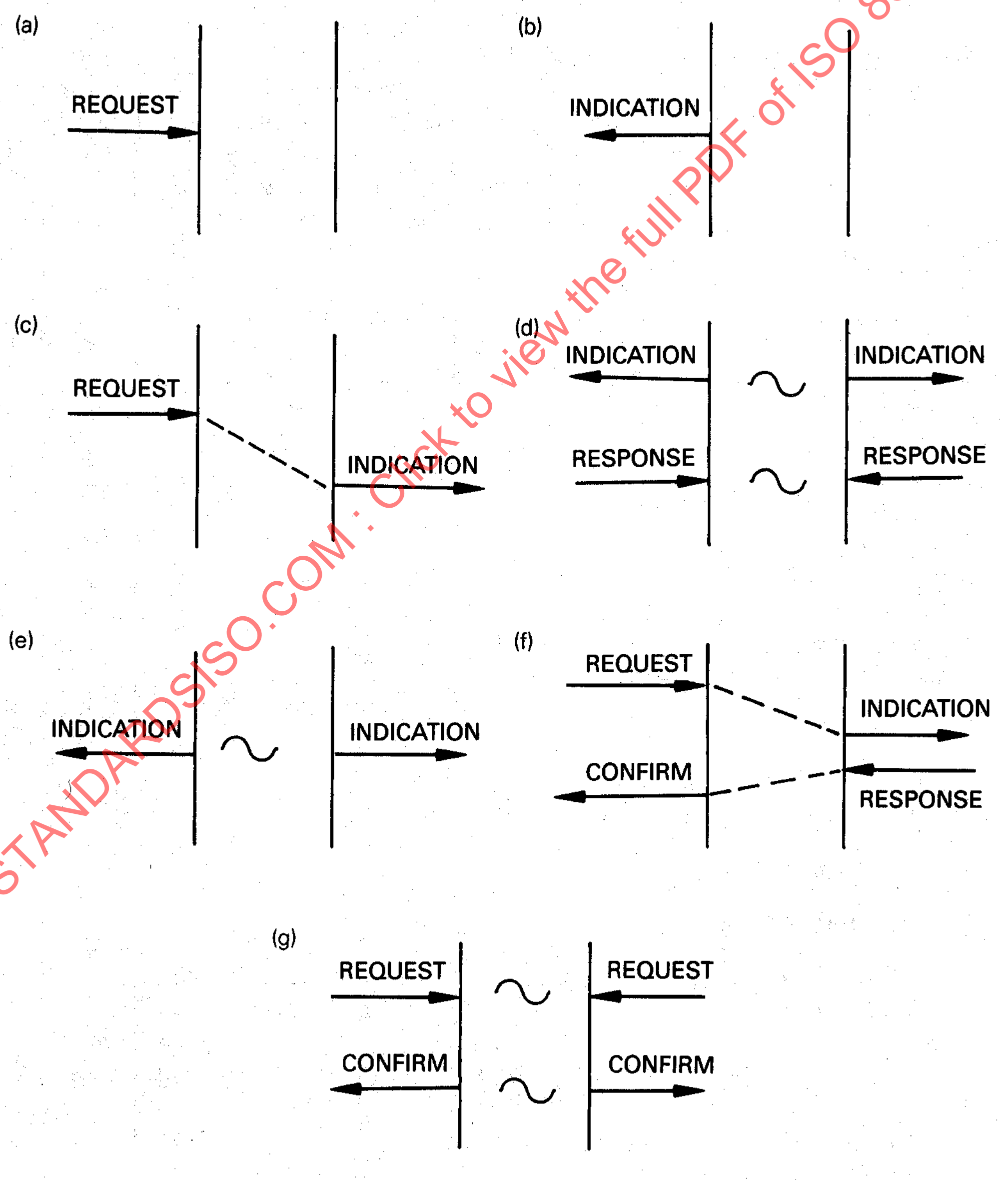
Possible relationships among primitive types are illustrated by the time-sequence diagrams shown in Fig 2-2. The figure also indicates the logical relationship of the primitive types. Primitive types that occur earlier in time and are connected by dotted lines in the diagrams are the logical antecedents of subsequent primitive types.

2.1 Network Layer/LLC Sublayer Interface Service Specification. This section specifies the services required of the LLC sublayer by the network layer, as viewed from the network layer, to allow a local network layer entity to exchange packets with remote peer network layer entities. The services are described in an abstract way and do not imply any particular implementation or any exposed interface.

Two forms of service are provided—*unacknowledged connectionless-mode* and *connection-mode:*

Unacknowledged Connectionless-Mode Service. The data transfer service that provides the means by which network entities can exchange link service data units (LSDUs) without the establishment of a data link level connection. The data transfer can be point-to-point, multicast, or broadcast.

Fig 2-2
Time-Sequence Diagram



Connection-Mode Services. This set of services provides the means for establishing, using, resetting, and terminating data link layer connections. These connections are point-to-point connections between LSAPs.

The connection establishment service provides the means by which a network entity can request, or be notified of, the establishment of data link layer connections.

The connection-oriented data transfer service provides the means by which a network entity can send or receive LSDUs over a data link layer connection. This service also provides data link layer sequencing, flow control, and error recovery.

The connection reset service provides the means by which established connections can be returned to the initial state.

The connection termination service provides the means by which a network entity can request, or be notified of, the termination of data link layer connections.

The connection flow control service provides the means to control the flow of data associated with a specified connection, across the network layer/data link layer interface.

2.1.1 Overview of Interactions

2.1.1.1 Unacknowledged Connectionless-Mode Services

2.1.1.1.1 Unacknowledged Connectionless-Mode Data Transfer. The primitives associated with unacknowledged connectionless-mode data transfer are

DL-UNITDATA request

DL-UNITDATA indication

The DL-UNITDATA request primitive is passed to the LLC sublayer to request that an LSDU be sent using unacknowledged connectionless-mode procedures. The DL-UNITDATA indication primitive is passed from the LLC sublayer to indicate the arrival of an LSDU.

2.1.1.2 Connection-Mode Services

2.1.1.2.1 Connection Establishment. The service primitives associated with connection establishment are

DL-CONNECT request

DL-CONNECT indication

DL-CONNECT response

DL-CONNECT confirm

The DL-CONNECT request primitive is passed to the LLC sublayer to request that a data link connection be established between a local LSAP and a remote LSAP. The DL-CONNECT indication primitive is passed from the LLC sublayer to indicate the request by a remote entity to establish a connection to a local LSAP. The DL-CONNECT response primitive is passed to the LLC sublayer to signal acceptance of a connection. The DL-CONNECT confirm primitive is passed from the LLC sublayer to convey the results of the previous associated DL-CONNECT request primitive.

2.1.1.2.2 Connection-Mode Data Transfer. The primitives associated with connection-mode data transfer are

DL-DATA request

DL-DATA indication

The DL-DATA request primitive is passed to the LLC sublayer to request that an LSDU be sent using connection-mode procedures. The DL-DATA indication primitive is passed from the LLC sublayer to indicate the arrival of an LSDU.

2.1.1.2.3 Connection Termination. The primitives associated with connection termination are

DL-DISCONNECT request

DL-DISCONNECT indication

The DL-DISCONNECT request primitive is passed to the LLC sublayer to request the immediate termination of a data link connection. The DL-DISCONNECT indication primitive is passed from the LLC sublayer to indicate to the network layer that a connection has been terminated.

2.1.1.2.4 Connection Reset. The primitives associated with connection resetting are

DL-RESET request

DL-RESET indication

DL-RESET response

DL-RESET confirm

The DL-RESET request primitive is passed to the LLC sublayer to request that a connection be immediately reset to the initial state. The DL-RESET indication primitive is passed from the LLC sublayer to indicate a connection reset attempt by either a remote entity or the local LLC sublayer. The DL-RESET response primitive is passed to the LLC sublayer to signal acceptance of the reset condition. The DL-RESET confirm primitive is passed from the LLC sublayer to convey the results of the previous associated DL-RESET request primitive.

2.1.1.2.5 Connection Flow Control. The primitives associated with connection flow control are

DL-CONNECTION-FLOWCONTROL request

DL-CONNECTION-FLOWCONTROL indication

The DL-CONNECTION-FLOWCONTROL request primitive is passed to the LLC sublayer to control the flow from the LLC sublayer of DL-DATA indication primitives related to a connection. The DL-CONNECTION-FLOW-

CONTROL indication primitive is passed from the LLC sublayer to control the flow from the network layer of DL-DATA request primitives related to a connection.

2.1.2 Detailed Service Specifications. This section describes in detail the primitives and parameters associated with the identified services. Note that the parameters are specified in an abstract sense. The parameters specify the information that must be available to the receiving entity. A specific implementation is not constrained in the method of making this information available.

The "source_address" and "destination_address" parameters provide at a minimum the logical concatenation of the MAC address field (SA and/or DA) and the LLC address field (SSAP and/or DSAP). An implementation of connection-mode services may make use of a locally significant connection identifier to imply source and destination address parameters. The "data" parameter may be provided by actually passing the link service data unit, by passing a pointer, or by other means. The "priority" parameter provides the priority associated with the data unit transfer. The "priority" parameter is passed transparently to the underlying MAC sublayer via the appropriate LLC/MAC primitives, see 2.2. The "reason" parameter provides an explanation of the disconnection, including a request by the remote entity, or an error internal to the LLC sublayer. The "amount" parameter provides information regarding the amount of data that the LLC entity is allowed to pass.

2.1.2.1 DL-UNITDATA request

2.1.2.1.1 Function. This primitive is the service request primitive for the unacknowledged connectionless-mode data transfer service.

2.1.2.1.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-UNITDATA request(
                     source_address,
                     destination_address,
                     data,
                     priority
                     )
```

The source_address and destination_address parameters specify the local and remote LSAPs involved in the data unit transfer. The destination_address may specify either an individual or group address. The data parameter specifies the link service data unit to be transferred by the data link layer entity. The priority parameter specifies the priority desired for the data unit transfer.

2.1.2.1.3 When Generated. This primitive is passed from the network layer to the LLC sublayer to request that a LSDU be sent to one or more remote LSAP(s) using unacknowledged connectionless-mode procedures.

2.1.2.1.4 Effect on Receipt. Receipt of this primitive causes the LLC sublayer to attempt to send the LSDU using unacknowledged connectionless-mode procedures.

2.1.2.1.5 Additional Comments. This primitive is independent of any connection with the remote LSAP.

A possible logical sequence of primitives associated with successful unacknowledged connectionless-mode data unit transfer is illustrated in Fig 2-2 (c).

2.1.2.2 DL-UNITDATA indication

2.1.2.2.1 Function. This primitive is the service indication primitive for the unacknowledged connectionless-mode data unit transfer service.

2.1.2.2.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-UNITDATA  indication(
                        source_address,
                        destination_address,
                        data,
                        priority
                        )
```

The source_address and destination_address parameters specify the local and remote LSAPs involved in the data unit transfer. The source address may be the address of a local LSAP, or may be a group address specifying multiple LSAPs, including a local LSAP. The data parameter specifies the link service data unit which has been received by the LLC sublayer entity. The priority parameter specifies the priority desired for the data unit transfer.

2.1.2.2.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to indicate the arrival of an LSDU from the specified remote entity.

2.1.2.2.4 Effect on Receipt. The effect of receipt of this primitive by the network layer is unspecified.

2.1.2.2.5 Additional Comments. This primitive is independent of any connection with the remote LSAP.

In the absence of errors, the contents of the data parameter are logically complete and unchanged relative to the data parameter in the associated DL-UNITDATA request primitive.

2.1.2.3 DL-CONNECT request

2.1.2.3.1 Function. This primitive is the service request primitive for the connection establishment service.

2.1.2.3.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-CONNECT request(
                   source_address,
                   destination_address,
                   priority
                   )
```

The source_address and destination_address parameters specify the local and remote LSAPs that are to be connected. The priority parameter specifies the priority desired for the connection.

2.1.2.3.3 When Generated. This primitive is passed from the network layer to the LLC sublayer when the network layer entity wishes to establish a logical link connection, of a given priority, to a remote LSAP.

2.1.2.3.4 Effect on Receipt. The receipt of this primitive by the LLC sublayer causes the local LLC entity to initiate the establishment of a connection with the remote LLC entity.

2.1.2.3.5 Additional Comments. A possible logical sequence of primitives associated with successful connection establishment is illustrated in Fig 2-2 (f).

2.1.2.4 DL-CONNECT indication

2.1.2.4.1 Function. This primitive is the service indication primitive for the connection establishment service.

2.1.2.4.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-CONNECT indication(
                        source_address,
                        destination_address,
                        priority
                        )
```

The source_address and destination_address parameters specify the local and remote LSAPs that are to be connected. The priority parameter indicates the priority desired for the connection.

2.1.2.4.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to indicate that a connection of a certain priority is being requested.

2.1.2.4.4 Effect on Receipt. The network layer entity shall issue either a DL-CONNECT response primitive to accept the connection, or a DL-DISCONNECT request primitive to refuse the connection.

2.1.2.4.5 Additional Comments. None.

2.1.2.5 DL-CONNECT response

2.1.2.5.1 Function. This primitive is the service response primitive for the connection establishment service.

2.1.2.5.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-CONNECT response(
                        source_address,
                        destination_address,
                        priority
                        )
```

The source_address and destination_address parameters specify the local and remote LSAPs that are to be connected. The priority parameter indicates the priority provided for the connection.

2.1.2.5.3 When Generated. This primitive is passed from the network layer to the LLC sublayer to indicate acceptance of the requested connection.

2.1.2.5.4 Effect on Receipt. The receipt of this primitive by the LLC sublayer causes the local LLC entity to accept a connection with the remote LLC entity.

2.1.2.5.5 Additional Comments. The network layer entity can return the same priority as given in the DL-CONNECT indication primitive or it may select a lower priority. After returning a DL-CONNECT response primitive, the network layer entity assumes that the connection is established.

2.1.2.6 DL-CONNECT confirm

2.1.2.6.1 Function. This primitive is the service confirm primitive for the connection establishment service.

2.1.2.6.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-CONNECT confirm(
                        source_address,
                        destination_address,
                        priority
                        )
```

The source_address and destination_address parameters specify the local and remote LSAPs that are to be connected. The priority parameter indicates the priority provided for the connection.

2.1.2.6.3 When Generated. This primitive is passed by the LLC sublayer to the network layer to convey the results of the previous associated DL-CONNECT request primitive. The results indicate that the connection attempt was successful and specify the priority obtained.

2.1.2.6.4 Effect on Receipt. The network layer entity may use this connection for data unit transfer.

2.1.2.6.5 Additional Comments. This primitive indicates that the remote network layer entity received and accepted the connection attempt.

2.1.2.7 DL-DATA request

2.1.2.7.1 Function. This primitive is the service request primitive for the connection-mode data unit transfer service.

2.1.2.7.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-DATA request(
                  source_address,
                  destination_address,
                  data
                  )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection. The data parameter specifies the link service data unit to be transferred by the LLC sublayer entity.

2.1.2.7.3 When Generated. This primitive is passed from the network layer to the LLC sublayer to request that a LSDU be transferred to a remote LSAP over an existing connection.

2.1.2.7.4 Effect on Receipt. The receipt of this primitive by the LLC sublayer causes the LLC sublayer to transfer the LSDU over the specified connection using connection-mode procedures.

2.1.2.7.5 Additional Comments. The DL-DATA request primitive does not contain a priority parameter because priority must be uniform for all DL-DATA requests in a particular connection.

A possible logical sequence of primitives associated with successful connection-mode data unit transfer is illustrated in Fig 2-2(c).

2.1.2.8 DL-DATA indication

2.1.2.8.1 Function. This primitive is the service indication primitive for the connection-mode data unit transfer service.

2.1.2.8.2 Semantics of the Service Primitive. The primitive shall provide parameters, as follows:

```
DL-DATA  indication(
                    source_address,
                    destination_address,
                    data
                    )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection. The data parameter specifies the link service data unit which has been received by the LLC sublayer entity.

2.1.2.8.3 When Generated. This primitive is passed by the LLC sublayer to the network layer to indicate the arrival of an LSDU from the specified remote network layer entity over a particular connection.

2.1.2.8.4 Effect on Receipt. The effect of receipt of this primitive by the network layer is unspecified.

2.1.2.8.5 Additional Comments. In the absence of errors, the contents of the data parameter are logically complete and unchanged relative to the data parameter in the associated DL-DATA request primitive.

2.1.2.9 DL-DISCONNECT request

2.1.2.9.1 Function. This primitive is the service request primitive for the connection termination service.

2.1.2.9.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-DISCONNECT request(
                      source_address,
                      destination_address
                      )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection to be terminated.

2.1.2.9.3 When Generated. This primitive is passed from the network layer to the LLC sublayer when the network layer entity wishes to terminate a connection.

2.1.2.9.4 Effect on Receipt. Receipt of this primitive causes the LLC sublayer to immediately terminate the connection.

2.1.2.9.5 Additional Comments. All unacknowledged LSDUs are discarded. The connection termination service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful disconnection (i.e., without loss of data) is the responsibility of a higher layer protocol.

A possible logical sequence of primitives associated with successful connection termination is illustrated in Fig 2-2(c).

2.1.2.10 DL-DISCONNECT indication

2.1.2.10.1 Function. This primitive is the service indication primitive for the connection termination service.

2.1.2.10.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-DISCONNECT indication(
                          source_address,
                          destination_address,
                          reason
                          )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the terminated connection. The reason parameter specifies the reason for the disconnection. The reasons for disconnection may include a request by the remote entity, or an error internal to the LLC sublayer.

2.1.2.10.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to inform the network layer that a connection has been terminated.

2.1.2.10.4 Effect on Receipt. The network entity may no longer use this connection for data unit transfer.

2.1.2.10.5 Additional Comments. All unacknowledged LSDUs are discarded. The connection termination service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful disconnection (i.e., without loss of data) is the responsibility of a higher layer protocol.

2.1.2.11 DL-RESET request

2.1.2.11.1 Function. This primitive is the service request primitive for the connection reset service.

2.1.2.11.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-RESET request(
                    source_address,
                    destination_address
                    )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection to be reset.

2.1.2.11.3 When Generated. This primitive is passed from the network layer to the LLC sublayer to request that a connection be reset to the initial state.

2.1.2.11.4 Effect on Receipt. Receipt of this primitive causes immediate resetting of the connection.

2.1.2.11.5 Additional Comments. All unacknowledged LSDUs are discarded. The connection reset service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful reset (i.e., without loss of data) is the responsibility of a higher layer protocol.

A possible logical sequence of primitives associated with successful connection reset is illustrated in Fig 2-2(f).

2.1.2.12 DL-RESET indication

2.1.2.12.1 Function. This primitive is the service indication primitive for the connection reset service.

2.1.2.12.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-RESET indication(
                    source_address,
                    destination_address,
                    reason
                    )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection affected. The reason parameter specifies the cause for the reset indication primitive. One of the reason codes indicates that a reset was requested by a remote network layer entity or a remote LLC sublayer peer (as shown in Fig 2-2, parts (f) and (d), respectively. All other codes indicate that the local LLC sublayer has detected the need for a connection reset (as shown in Fig 2-2(b)).

2.1.2.12.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to indicate either that a connection reset has been requested by the remote network layer entity or remote LLC sublayer peer, or that the local LLC sublayer has determined that the data link connection is in need of reinitialization (as shown in Fig 2-2 (d)).

2.1.2.12.4 Effect on Receipt. For remote reset request, the network layer entity shall issue either a DL-RESET response primitive to signal acceptance of the connection reset, or a DL-DISCONNECT request primitive to terminate the connection. For local reset condition indication, the network layer entity shall issue either a DL-RESET request primitive to reinitialize the connection, or a DL-DISCONNECT request primitive to terminate the connection.

2.1.2.12.5 Additional Comments. The reasons for the reset may include a request by the remote entity, or an error condition detected by the local LLC sublayer. All unacknowledged LSDUs are discarded. The connection reset service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful reset (i.e., without loss of data) is the responsibility of a higher layer protocol.

2.1.2.13 DL-RESET response

2.1.2.13.1 Function. This primitive is the service response primitive for the connection reset service.

2.1.2.13.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-RESET response(
                    source_address,
                    destination_address
                    )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection affected.

2.1.2.13.3 When Generated. This primitive is passed by the network layer to the LLC sublayer to indicate acceptance of the connection reset.

2.1.2.13.4 Effect on Receipt. The receipt of this primitive by the LLC sublayer causes the local LLC sublayer entity to complete the connection reset.

2.1.2.13.5 Additional Comments. All unacknowledged LSDUs are discarded. The connection reset service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful reset (i.e., without loss of data) is the responsibility of a higher layer protocol.

2.1.2.14 DL-RESET confirm

2.1.2.14.1 Function. This primitive is the service confirm primitive for the connection reset service.

2.1.2.14.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-RESET confirm(
                    source_address,
                    destination_address
                    )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection affected.

2.1.2.14.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to inform the network layer that a connection reset has been completed.

2.1.2.14.4 Effect on Receipt. The network layer entity may use this connection for data unit transfer.

2.1.2.14.5 Additional Comments. This primitive indicates that the remote LLC sublayer entity has acknowledged the reset.

2.1.2.15 DL-CONNECTION-FLOWCONTROL request

2.1.2.15.1 Function. This primitive is the service request primitive for the connection flow control service.

2.1.2.15.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-CONNECTION-FLOWCONTROL  request(
                                   source_address,
                                   destination_address,
                                   amount
                                   )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection to be flow controlled. The amount parameter specifies the amount of data the LLC sublayer entity is permitted to pass.

2.1.2.15.3 When Generated. This primitive is passed from the network layer to the LLC sublayer to request control of the flow of DL-DATA indication primitives associated with a connection from the LLC sublayer.

2.1.2.15.4 Effect on Receipt. Receipt of this primitive causes the LLC sublayer to adjust the amount of data that may be passed to the network layer.

2.1.2.15.5 Additional Comments. Control of the flow of data on a connection is independent of control of the flow on other connections. The amount of data permitted to be passed is dynamically updated by each request. If amount is specified as zero, then the associated flow is stopped. Specific implementations may allow amount to be specified in implementation specific units, and may allow amount to be specified as "infinite."

A possible logical sequence of primitives associated with a DL-CONNECTION-FLOWCONTROL request is illustrated in Fig 2-2 (a).

2.1.2.16 DL-CONNECTION-FLOWCONTROL indication

2.1.2.16.1 Function. This primitive is the service indication primitive for the connection flow control service.

2.1.2.16.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
DL-CONNECTION-FLOWCONTROL  indication(
                                      source_address,
                                      destination_address,
                                      amount
                                      )
```

The source_address and destination_address parameters specify the local and remote LSAPs of the connection to be flow controlled. The amount parameter specifies the amount of data that the network layer entity is permitted to pass to avoid data loss.

2.1.2.16.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to request control of the flow of DL-DATA request primitives associated with a connection from the network layer.

2.1.2.16.4 Effect on Receipt. Receipt of the primitive causes the network layer to adjust the amount of data that it is allowed to pass without data loss.

2.1.2.16.5 Additional Comments. Control of the flow of data on a connection is independent of control of the flow on other connections. The amount of data permitted to be passed is dynamically updated by each indication. If amount is specified as zero, then the associated flow is stopped. Specific implementations may allow amount to be specified in implementation-specific units, and may allow amount to be specified as "infinite."

A possible logical sequence of primitives associated with a DL-CONNECTION-FLOWCONTROL indication is illustrated in Fig 2-2 (b).

2.2 LLC Sublayer/MAC Sublayer Interface Service Specification. This section specifies the services required of the MAC sublayer by the LLC sublayer to allow the local LLC sublayer entity to exchange LLC data units with peer LLC sublayer entities. The services are described in an abstract way and do not imply any particular implementation or any exposed interface.

NOTE: Work is in progress to produce a single service specification that is common to all the MAC sublayers; when this is available, it will be referenced in this document instead of the current MAC service text.

2.2.1 Overview of Interactions

MA-UNITDATA request

MA-UNITDATA indication

MA-UNITDATA-STATUS indication

2.2.2 Detailed Service Specification

2.2.2.1 MA-UNITDATA request

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

2.2.2.1.2 Semantics of the Service Primitive. The semantics of the primitive are as follows:

```
MA-UNITDATA request(
                    source_address,
                    destination_address,
                    data,
                    priority,
                    service_class
                    )
```

The source_address parameter shall specify an individual MAC sublayer entity address. The destination_address parameter shall specify either an individual or a group MAC sublayer entity address. Together they must contain sufficient information to create the SA and DA fields that are appended to the frame by the local MAC sublayer entity as well as any physical layer address information (e.g., transmit frequency in broadband applications). The data parameter specifies the MAC service data unit to be transmitted by the MAC sublayer entity, which includes the DSAP, SSAP, C, and information (if present) fields as specified in Section 3, as well as sufficient information for the MAC sublayer entity to determine the length of the data unit. The priority parameter specifies the priority desired for the data unit transfer. The service_class parameter specifies the class of service desired for the data unit transfer.

2.2.2.1.3 When Generated. This primitive is generated by the LLC sublayer entity whenever a MSDU must be transferred to a peer LLC sublayer entity or entities. This can be as a result of a request from higher layers of protocol, or from a MSDU generated internally to the LLC sublayer, such as required by Type 2 operation.

2.2.2.1.4 Effect of Receipt. The receipt of this primitive shall cause the MAC sublayer entity to append all MAC specified fields, including DA, SA, and any fields that are unique to the particular medium access method, and pass the properly formatted frame to the lower layers of protocol for transfer to the peer MAC sublayer entity or entities.

2.2.2.1.5 Additional Comments. A possible logical sequence of primitives associated with successful MAC service data unit transfer is illustrated in Fig 2-2 (c).

2.2.2.2 MA-UNITDATA indication

2.2.2.2.1 Function. This primitive defines the transfer of a MSDU from the MAC sublayer entity to the LLC sublayer entity, or entities in the case of group addresses. In the absence of errors, the contents of the data parameter are logically complete and unchanged relative to the data parameter in the associated MA-UNITDATA request primitive.

2.2.2.2.2 Semantics of the Service Primitive. The semantics of the primitive are as follows:

```
MA-UNITDATA  indication(
                         source_address,
                         destination_address,
                         data,
                         reception_status,
                         priority,
                         service_class
                         )
```

The source_address parameter must be an individual address as specified by the SA field of the incoming frame. The destination_address parameter shall be either an individual or a group address as specified by the DA field of the incoming frame. The data parameter specifies the MAC service data unit as received by the local MAC entity. The reception_status parameter indicates the success or failure of the incoming frame. The priority parameter specifies the priority desired for this data unit transfer. The service_class parameter specifies the class of service desired for the data unit transfer.

2.2.2.2.3 When Generated. The MA-UNITDATA indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities to indicate the arrival of a frame at the local MAC sublayer entity. Frames are reported only if at the MAC sublayer they are validly formatted, received without error, and their destination address designates the local MAC sublayer entity.

2.2.2.2.4 Effect of Receipt. The effect of receipt of this primitive by the LLC sublayer is dependent on the validity and content of the frame.

2.2.2.2.5 Additional Comments. If the local MAC sublayer entity is designated by the destination_address parameter of an MA-UNITDATA request primitive, the indication primitive will also be invoked by the MAC sublayer entity to the local LLC sublayer 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-UNITDATA indication primitives at all stations in the network including the station that generated the request).

2.2.2.3 MA-UNITDATA-STATUS indication

2.2.2.3.1 Function. This primitive has local significance and shall provide the LLC sublayer with status information for a previous associated MA-UNITDATA request primitive.

2.2.2.3.2 Semantics of the Service Primitive. The semantics of this primitive are as follows:

```
MA-UNITDATA-STATUS  indication(
                                  source_address,
                                  destination_address,
                                  transmission_status,
                                  provided_priority,
                                  provided_service_class
                                  )
```


The source_address parameter must be an individual MAC sublayer entity address as specified in the associated MA-UNITDATA request primitive. The destination_address parameter shall be either an individual or a group MAC sublayer entity address as specified in the associated MA-UNITDATA request primitive. The transmission_status parameter is used to pass status information back to the local requesting LLC sublayer entity. The types of status that can be associated with this primitive are dependent on the particular implementation as well as the type of MAC sublayer that is used (e.g., "excessive collisions" may be a status returned by a CSMA/CD MAC sublayer entity).

The provided_priority parameter specifies the priority that was used for the associated data unit transfer. The provided_service_class parameter specifies the class of service provided for the data unit transfer.

2.2.2.3.3 When Generated. The MA-UNITDATA-STATUS indication primitive is passed from the MAC sublayer entity to the LLC sublayer to indicate the status of the service provided for a previous associated MA-UNIT-DATA request primitive.

2.2.2.3.4 Effect of Receipt. The effect of receipt of this primitive by the LLC sublayer is dependent upon the type of operation employed by the LLC sublayer entity.

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

2.3 LLC Sublayer/LLC Sublayer Management Function Interface Service Specification. (This matter is the subject of further ongoing study and resolution.)

3. LLC PDU Structure

3.1 General. This section defines in detail the LLC PDU structure for data communication systems using bit-oriented procedures. It defines the relative positions of the various components of the PDU. It defines the method for representing data link layer service access point addresses (to or from network layer entities). It defines a partition of these addresses into individual and group addresses. Details of the control and information field allocation are specified in Section 5.

3.2 LLC PDU Format. All LLC PDUs shall conform to the format shown in Fig 3-1.

3.3 Elements of the LLC PDU

3.3.1 Address Fields. Each LLC PDU shall contain two address fields: the Destination Service Access Point (DSAP) address field and the Source Service Access Point (SSAP) address field, in that order. Each address field shall

Fig 3-1
LLC PDU Format

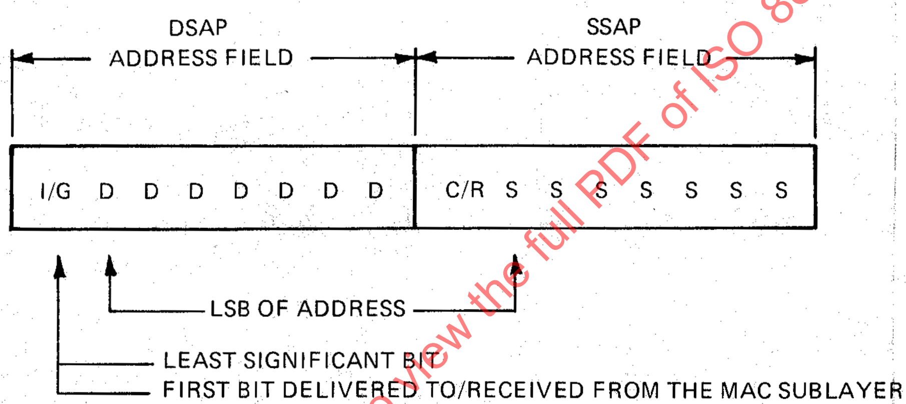
DSAP Address	SSAP Address	Control	Information
8 bits	8 bits	8 or 16 bits	M * 8 bits

DSAP Address = destination service access point address field
SSAP Address = source service access point address field
Control = control field (16 bits for formats that include sequence numbering, and 8 bits for formats that do not (see 5.2))
Information = information field
* = multiplication
M = an integer value equal to or greater than 0. (Upper bound of M is a function of the medium access control methodology used.)

contain only a single address. The DSAP address field shall identify the one or more service access points for which the LLC information field is intended. The SSAP address field shall identify the specific service access point from which the LLC information field was initiated.

3.3.1.1 Address Field Representation. The representation of each address field shall be as shown in Fig 3-2a and 3-2b:

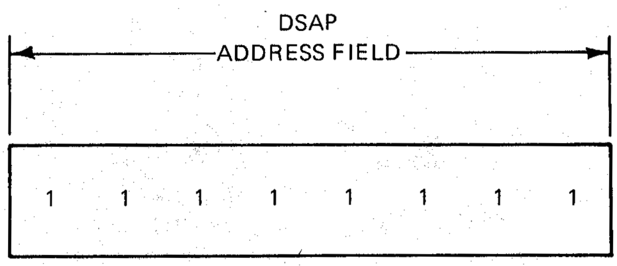
Fig 3-2a
DSAP and SSAP Address Field Formats



I/G = 0	INDIVIDUAL DSAP
I/G = 1	GROUP DSAP
C/R = 0	COMMAND
C/R = 1	RESPONSE

XODDDDDD	DSAP ADDRESS
XOSSSSSS	SSAP ADDRESS
X1DDDDDD	RESERVED FOR ISO DEFINITION
X1SSSSSS	RESERVED FOR ISO DEFINITION

Fig 3-2b
Global DSAP Address Field Format



(1) Each address field shall contain one octet.

(2) Each address field shall contain 7 bits of actual address and one bit that shall be used in the DSAP address field to identify the DSAP address as either an individual or a group address (called the address type designation bit) and in the SSAP address field to identify that the LLC PDU is a command or a response (called the command/response identifier bit).

(3) The address type designation bit shall be located in the least significant bit position of the DSAP address field. If this bit is "0", it shall indicate that the address is an individual DSAP address. If this bit is "1", it shall indicate that the address is a group DSAP address that identifies none, one or more, or all of the service access points that are serviced by the LLC entity.

(4) The command/response identifier bit shall be located in the least significant bit position of the SSAP address field. If this bit is "0", it shall indicate that the LLC PDU is a command. If this bit is "1", it shall indicate that the LLC PDU is a response.

3.3.1.2 Address Usage. An individual address shall be usable as both an SSAP and a DSAP address; a null address shall be usable as both an SSAP and a DSAP address; a group address shall be usable only as a DSAP address.

All "1"s in the DSAP address field (i.e., the address type designation bit set to "1", and the seven address bits set to "1") is predefined to be the "Global" DSAP address. This DSAP address designates a group consisting of all DSAPs actively being serviced by the underlying MAC service access point address(es).

All "0"s in the DSAP or SSAP address field, (i.e., the address type designation bit set to "0", and the seven address bits set to "0") is predefined to be the "Null" address. The Null service access point address designates the LLC that is associated with the underlying MAC service access point address, and is NOT used to identify any service access point to the network layer or any service access point to an associated layer management function.

Addresses 01000000 and 11000000 are designated as the individual and the group addresses, respectively, for an LLC sublayer management function at the station. Other addresses with the next to low-order bit set to "1" are reserved for ISO definition.

3.3.2 Control Field. The control field shall consist of one or two octets that shall be used to designate command and response functions, and shall contain sequence numbers when required. The content of this field shall be as described in Section 5.

3.3.3 Information Field. The information field shall consist of any integral number (including zero) of octets.

3.3.4 Bit Order. Addresses, commands and responses, and sequence numbers shall be delivered to/received from the MAC sublayer least significant bit first (i.e., the first bit of a sequence number that is delivered/received shall have the weight 2**0). The information field shall be delivered to the MAC sublayer in the same bit order as received from the network layer. The information field shall be delivered to the network layer in the same bit order as received from the MAC sublayer.

3.3.5 Invalid LLC PDU. An invalid LLC PDU shall be defined as one which meets at least one of the following conditions:

(1) It is identified as such by the PHY or the MAC sublayer.
(2) It is not an integral number of octets in length.
(3) It does not contain two properly formatted address fields, one control field, and optionally an information field in their proper order.
(4) Its length is less than 3 octets (one-octet control field) or 4 octets (two-octet control field).

Invalid LLC PDUs shall be ignored.

4. LLC Types and Classes of Procedure

4.1 General. LLC defines two types of operation for data communication between service access points.

(1) *Type 1 Operation.* With Type 1 operation, PDUs shall be exchanged between LLCs without the need for the establishment of a data link connection. In the LLC sublayer these PDUs shall not be acknowledged, nor shall there be any flow control or error recovery in the Type 1 procedures.

(2) *Type 2 Operation.* With Type 2 operation, a data link connection shall be established between two LLCs prior to any exchange of information-bearing PDUs. The normal cycle of communication between two Type 2 LLCs on a data link connection shall consist of the transfer of PDUs containing information from the source LLC to the destination LLC, acknowledged by PDUs in the opposite direction.

With Type 2 operation, the control of traffic between the source LLC and the destination LLC shall be effected by means of a numbering scheme, which shall be cyclic within a modulus of 128 and measured in terms of PDUs. An independent numbering scheme shall be used for each source/destination LLC pair. Each such pairing shall be defined to be a logical point-to-point data link connection between data link layer service access points and shall take into account the DA and SA addressing that is part of the MAC sublayer. The acknowledgment function shall be accomplished by the destination LLC informing the source LLC of the next expected sequence number. This shall be done either in a separate PDU, not containing information, or within the control field of a PDU containing information.

LLC Type 2 procedures shall be applicable to balanced data link connections. A balanced data link connection shall involve two participating LLCs. For control purposes, each LLC shall assume responsibility for the organization of its data flow and for the link error recovery operations for the transmissions that it originates. Each LLC shall be capable of sending and receiving both command and response PDUs.

For the transfer of data between LLCs in Type 2 operation, Fig 4-1 depicts the data link control functions utilized. The data source in each LLC shall control the data sink in the other LLC by the use of command PDUs. The information shall flow from the data source to the data sink, and any acknowledgments shall always be sent in the opposite direction. The poll-type command PDUs shall be utilized by each LLC station to solicit specific acknowledgments and status responses from the other LLC.

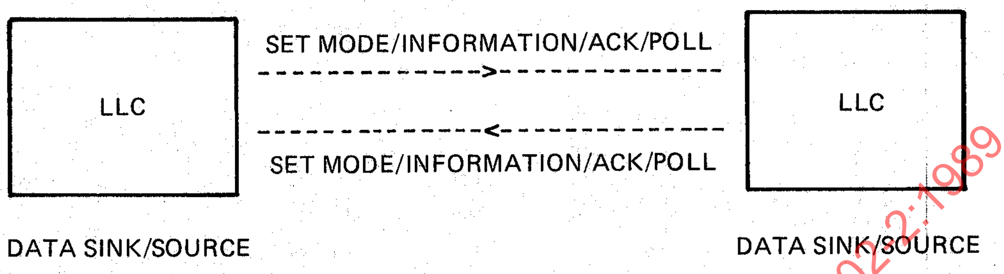


Fig 4-1
Balanced Data Link Connection Configuration

NOTE: The need for a reliable transaction type service has been identified as a subject for urgent further study. The desired service is one that includes a basic acknowledgment function that will indicate that a sent PDU has been received and accepted by the peer LLC sublayer.

4.2 Classes of LLC (Conformance Clause). Two classes of LLC are defined. A Class I LLC shall support Type 1 operation only, whereas a Class II LLC shall support both Type 1 and Type 2 operations, as indicated by X in Fig 4-2.

This means all LLCs on a local area network shall have Type 1 operation in common. In a Class II LLC, the support of Type 1 operation shall be totally independent of the modes or change of modes of the Type 2 operation in that same LLC. A Class II LLC shall be capable of going back and forth between Type 1 operation and Type 2 operation on a PDU-to-PDU basis in the same SAP, if necessary.

Fig 4-2
Classes of LLC

CLASSES OF LLC	TYPE OF OPERATION 1	TYPE OF OPERATION 2
I	X	
II	X	X

4.2.1 Class I LLC. Class I LLCs shall support Type 1 operation only. Class I service shall be applicable to individual, group, global, and null DSAP addressing, and applications requiring no data link layer acknowledgment or flow control procedures. The set of command PDUs and response PDUs supported in Class I service shall be

	Commands	Responses
Type 1:	UI	
	XID	XID
	TEST	TEST

4.2.2 Class II LLC. Class II LLCs shall support both Type 1 operation and Type 2 operation. In a Class II station, the operation of the Type 1 procedures and the Type 2 procedures are completely independent. The set of command PDUs and response PDUs supported in Class II service shall be

	Commands	Responses
Type 1:	UI	
	XID	XID
	TEST	TEST
Type 2:	I	I
	RR	RR
	RNR	RNR
	REJ	REJ
	SABME	UA
	DISC	DM
		FRMR

5. LLC Elements of Procedure

5.1 General. This section specifies the elements of the local area network logical link control (LLC) procedures for code-independent data communication using the LLC PDU structure (see Section 3).

These LLC elements of procedure are defined specifically in terms of the actions that shall occur in the LLC on receipt of commands, and occasionally on receipt of a reply to a command, over a logical data link (Type 1) and a data link connection (Type 2). Each element of procedure is utilized by only one of the two types of operation (Type 1 or Type 2) that are defined in Section 4.

5.2 Control Field Formats. The three formats defined for the control field (Fig 5-1) shall be used to perform numbered information transfer, numbered

Fig 5-1
LLC PDU Control Field Formats

LLC PDU CONTROL FIELD BITS

	1	2	3	4	5	6	7	8	9	10 - 16
INFORMATION TRANSFER COMMAND/RESPONSE (I-FORMAT PDU)	0	N(S)							P/F	N(R)
SUPERVISORY COMMANDS/RESPONSES (S-FORMAT PDUs)	1	0	S	S	X	X	X	X	P/F	N(R)
UNNUMBERED COMMANDS/RESPONSE (U-FORMAT PDUs)	1	1	M	M	P/F	M	M	M		

N(S) = Transmitter send sequence number (Bit 2 = low-order bit)
N(R = Transmitter receive sequence number (Bit 10 = low-order bit)
S = Supervisory function bit
M = Modifier function bit
X = Reserved and set to zero
P/F = Poll bit—command LLC PDU transmissions
Final bit—response LLC PDU transmissions
(1 = Poll/Final)

supervisory transfer, unnumbered control, and unnumbered information transfer functions. The numbered information transfer and supervisory transfer functions apply only to Type 2 operation. The unnumbered control and unnumbered information transfer functions apply either to Type 1 or Type 2 operation (but not both) depending upon the specific function selected.

5.2.1 Information Transfer Format–I. The I-format PDU shall be used to perform a numbered information transfer in Type 2 operation. Except where otherwise specified (e.g., UI, TEST, FRMR, and XID), it shall be the only LLC PDU that may contain an information field. The functions of N(S), N(R), and P/F shall be independent; i.e., each I-format PDU shall have an N(S) sequence number, an N(R) sequence number which shall or shall not acknowledge additional I-format PDUs at the receiving LLC, and a P/F bit that shall be set to "1" or "0".

5.2.2 Supervisory Format–S. The S-format PDU shall be used to perform data link supervisory control functions in Type 2 operation, such as acknowledging I-format PDUs, requesting retransmission of I-format PDUs, and requesting a temporary suspension of transmission of I-format PDUs. The functions of N(R) and P/F shall be independent; i.e., each S-format PDU shall have an N(R) sequence number that shall or shall not acknowledge additional I-format PDUs at the receiving LLC, and a P/F bit that shall be set to "1" or "0".

5.2.3 Unnumbered Format–U. The U-format PDUs shall be used in either Type 1 or Type 2 operation, depending upon the specific function utilized, to provide additional data link control functions and to provide unsequenced information transfer. The U-format PDUs shall contain no sequence numbers, but shall include a P/F bit that shall be set to "1" or "0".

5.3 Control Field Parameters. The various parameters associated with the control field formats are described in the following sections.

5.3.1 Type 1 Operation Parameters. The only parameter that exists in Type 1 operation is the Poll/Final (P/F) bit. The P/F bit set to "1" shall only be used in Type 1 operation with the XID and TEST command/response PDU functions. The Poll (P) bit set to "1" shall be used to solicit (poll) a correspondent response PDU with the F bit set to "1" from the addressed LLC. The Final (F) bit set to "1" shall be used to indicate that response PDU that is sent by the LLC as the result of a soliciting (poll) command PDU (P bit set to "1").

5.3.2 Type 2 Operation Parameters. The various parameters associated with the control field formats in Type 2 operation are described in the following sections.

5.3.2.1 Modulus. Each I PDU shall be sequentially numbered with a number that shall have a value between 0 and MODULUS minus ONE (where MODULUS is the modulus of the sequence numbers). The modulus shall equal 128 for the Type 2 LLC control field format. The sequence numbers shall cycle through the entire range.

The maximum number of sequentially numbered I PDUs that may be outstanding (i.e., unacknowledged) in a given direction on a data link connection at any given time shall never exceed one less than the modulus of the sequence numbers. This restriction shall prevent any ambiguity in the association of sent I PDUs with sequence numbers during normal operation and/or error recovery action.

5.3.2.2 LLC PDU State Variables and Sequence Numbers. A station LLC shall maintain a send state variable V(S) for the I PDUs it sends and a receive state variable V(R) for the I PDUs it receives on each data link connection. The operation of V(S) shall be independent of the operation of V(R).

5.3.2.2.1 Send State Variable V(S). The send state variable shall denote the sequence number of the next in-sequence I PDU to be sent on a specific data link connection. The send state variable shall take on a value between 0 and MODULUS minus ONE (where MODULUS equals 128 and the numbers cycle through the entire range). The value of the send state variable shall be incremented by one with each successive I PDU transmission on the associated data link connection, but shall not exceed N(R) of the last received PDU by more than MODULUS minus ONE.

5.3.2.2.2 Send Sequence Number N(S). Only I PDUs shall contain N(S), the send sequence number of the sent PDU. Prior to sending an I PDU, the value of N(S) shall be set equal to the value of the send state variable for that data link connection.

5.3.2.2.3 Receive State Variable V(R). The receive state variable shall denote the sequence number of the next in-sequence I PDU to be received on a specific data link connection. The receive state variable shall take on a value between 0 and MODULUS minus ONE (where MODULUS equals 128 and the numbers cycle through the entire range). The value of the receive state variable associated with a specific data link connection shall be incremented by one whenever an error-free, in-sequence I PDU is received whose send sequence number N(S) equals the value of the receive state variable for the data link connection.

5.3.2.2.4 Receive Sequence Number N(R). All I-format PDUs and S-format PDUs shall contain N(R), the expected sequence number of the next received I PDU on the specified data link connection. Prior to sending an I-format PDU or S-format PDU, the value of N(R) shall be set equal to the current value of the associated receive state variable for that data link connection. N(R) shall indicate that the station sending the N(R) has received correctly all I PDUs numbered up through N(R)-1 on the specified data link connection.

5.3.2.3 Poll/Final (P/F) Bit. The poll (P) bit shall be used to solicit (poll) a response from the addressed LLC. The final (F) bit shall be used to indicate the response PDU sent as the result of a soliciting (poll) command.

The poll/final (P/F) bit shall serve a function in Type 2 operation in both command PDUs and response PDUs. In command PDUs the P/F bit shall be referred to as the P bit. In response PDUs it shall be referred to as the F bit. P/F bit exchange provides a distinct command/response linkage that is useful during both normal operation and recovery situations.

5.3.2.3.1 Poll Bit Functions. A command PDU with the P bit set to "1" shall be used on a data link connection to solicit a response PDU with the F bit set to "1" from the addressed LLC on that data link connection.

Only one PDU with a P bit set to "1" shall be outstanding in a given direction at a given time on the data link connection between any specific pair of LLCs. Before an LLC issues another PDU on the same data link connection with the P bit set to "1", the LLC shall have received a response PDU with the F bit set to "1" from the addressed LLC. If no valid response PDU is received within a system-defined P-bit timer time-out period, the (re)sending of a command PDU with the P bit set to "1" shall be permitted for error recovery purposes.

5.3.2.3.2 Final Bit Functions. A response PDU with the F bit set to "1" shall be used to acknowledge the receipt of a command PDU with the P bit set to "1".

Following the receipt of a command PDU with the P bit set to "1", the LLC shall send a response PDU with the F bit set to "1" on the appropriate data link connection at the earliest possible opportunity.

The LLC shall be permitted to send appropriate response PDUs with the F bit set to "0" at any medium access opportunity on an asynchronous basis (without the need for a command PDU).

5.4 Commands and Responses. This section defines the commands and associated responses. Sections 5.4.1 and 5.4.2 contain the definitions of the set of commands and responses (listed below) for each of the control field formats for Type 1 operation and for Type 2 operation, respectively.

The C/R bit, located in the low-order bit of the SSAP field, is used to distinguish between commands and responses. The following discussion of commands and responses assumes that the C/R bit has been properly decoded.

Commands	Responses
Information transfer format commands	**Information transfer format responses**
I — Information	I — Information
Supervisory format commands	**Supervisory format responses**
RR — Receive Ready	RR — Receive Ready
RNR — Receive Not Ready	RNR — Receive Not Ready
REJ — Reject	REJ — Reject
Unnumbered format commands	**Unnumbered format responses**
UI — Unnumbered Information	UA — Unnumbered Acknowledgment
DISC — Disconnect	DM — Disconnected Mode
SABME — Set Asynchronous Balanced Mode Extended	FRMR — Frame Reject
XID — Exchange Identification	XID — Exchange Identification
TEST — Test	TEST — Test

5.4.1 Type 1 Operation Commands and Responses. The Type 1 commands and responses are all U-format PDUs.

5.4.1.1 Type 1 Operation Commands. The U-format PDU command encodings for Type 1 operation are listed in Fig 5-2.

5.4.1.1.1 Unnumbered Information (UI) Command. The UI command PDU shall be used to send information to one or more LLCs. Use of the UI command PDU is not dependent on the existence of a data link connection between the destination and source LLCs, and its use will not affect the V(S) or V(R) variables associated with any data link connections. There is no LLC response PDU to the UI command PDU.

Reception of the UI command PDU is not acknowledged or sequence number verified; therefore, the data contained in a UI PDU may be lost if a logical data link exception (such as a transmission error or a receiver-busy condition) occurs during the sending of the command PDU.

A UI command PDU shall have either an individual, group, global, or null address as the destination DSAP address and the originator's individual address as the SSAP address.

5.4.1.1.2 Exchange Identification (XID) Command. The XID command PDU shall be used to convey the types of LLC services supported (for all LLC services) and the receive window size on a per data link connection basis to the destination LLC, and to cause the destination LLC to respond with the XID response PDU (see 5.4.1.2.1) at the earliest opportunity. The XID command PDU shall have no effect on any mode or sequence numbers maintained by the remote LLC. An XID command PDU shall have either an individual, group, global, or null address as the destination DSAP address and the originator's individual address as the SSAP address.

The information field of an XID basic format command PDU shall consist of an 8-bit XID format identifier field plus a 16-bit parameter field that is encoded to identify the LLC services supported plus the receive window size, as shown in Fig 5-3. The receive window size (k) is the maximum number that the send state variable V(S) can exceed the N(R) of the last received PDU.

Fig 5-2
Type 1 Operation Command Control Field Bit Assignments

FIRST CONTROL FIELD BIT DELIVERED
TO/RECEIVED FROM THE MAC SUBLAYER
↓

1	2	3	4	5	6	7	8	
1	1	0	0	P	0	0	0	UI COMMAND
1	1	1	1	P	1	0	1	XID COMMAND
1	1	0	0	P	1	1	1	TEST COMMAND

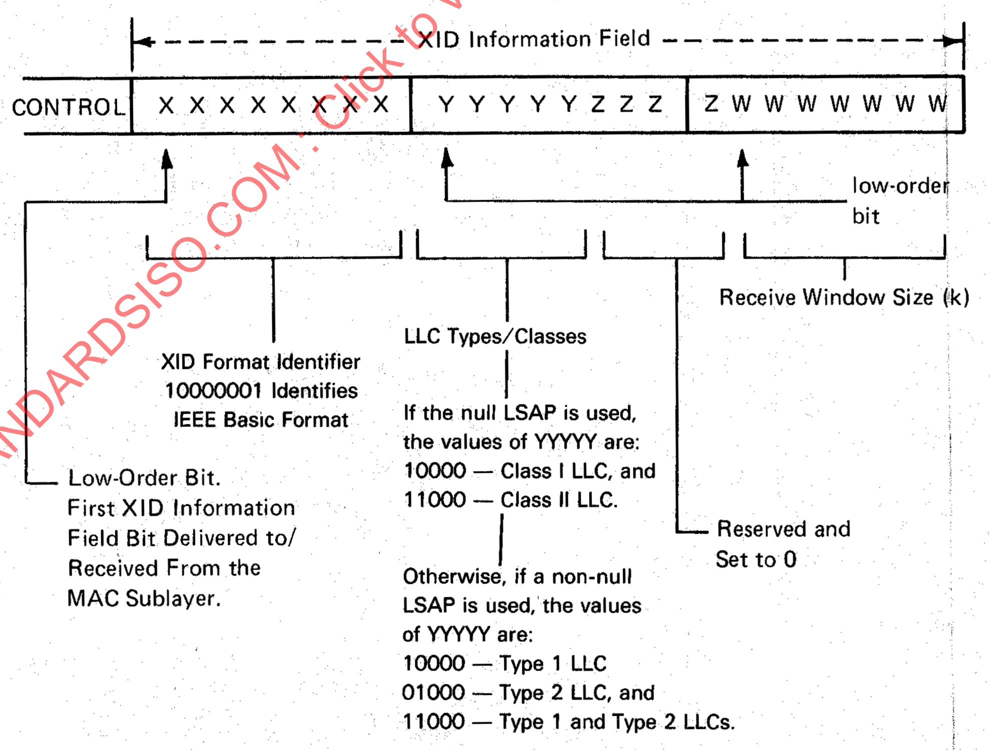
NOTE: Other uses of the XID PDU are for further study. In particular, the use of an unsolicited XID response PDU to announce the presence of a new LLC will be examined.

5.4.1.1.3 Test (TEST) Command. The TEST command PPU shall be used to cause the destination LLC to respond with the TEST response PPU (see 5.4.1.2.2) at the earliest opportunity, thus performing a basic test of the LLC to LLC transmission path. An information field is optional with the TEST command PDU. If present, however, the received information field shall be returned, if possible, by the addressed LLC in the TEST response PDU. The TEST command PDU shall have no effect on any mode or sequence numbers maintained by the remote LLC and may be used with an individual, group, global, or null DSAP address, and with an individual, group, or global DA address.

5.4.1.2 Type 1 Operation Responses. The U-format PDU response encodings for Type 1 operation are listed in Fig 5-4.

5.4.1.2.1 Exchange Identification (XID) Response. The XID response PDU shall be used to reply to an XID command PDU at the earliest opportunity. The XID response PDU shall identify the responding LLC and shall include an information field like that defined for the XID command PDU (see

Fig 5-3
XID Information Field Basic Format



FIRST CONTROL FIELD BIT DELIVERED
TO/RECEIVED FROM THE MAC SUBLAYER

1	2	3	4	5	6	7	8	
1	1	1	1	F	1	0	1	XID RESPONSE
1	1	0	0	F	1	1	1	TEST RESPONSE

Fig 5-4
Type 1 Operation Response Control Field Bit Assignments

5.4.1.1.2), regardless of what information is present in the information field of the received XID command PDU. The XID response PDU shall use an individual or null DSAP address and shall use an individual or null SSAP address. The XID response PDU shall have its F bit set to the state of the P bit in the XID command PDU.

5.4.1.2.2 Test (TEST) Response. The TEST response PDU shall be used to reply to the TEST command PDU. The TEST response PDU shall have its F bit set to the value of the P bit in the TEST command PDU. An information field, if present in the TEST command PDU, shall be returned in the corresponding TEST response PDU. If the LLC cannot accept an information field (e.g., buffering limitation), a TEST response PDU without an information field may be returned. Refer to 6.7 for specific details on TEST response usage.

5.4.2 Type 2 Operation Commands and Responses. Type 2 commands and responses consist of I-format, S-format, and U-format PDUs.

5.4.2.1 Information Transfer Format Command and Response. The function of the information, I, command and response shall be to transfer sequentially numbered PDUs containing an octet-oriented information field across a data link connection. The encoding of the I PDU control field for Type 2 operation shall be as listed in Fig 5-5.

The I PDU control field shall contain two sequence numbers: N(S), send sequence number, which shall indicate the sequence number associated with the I PDU; and N(R), receive sequence number, which shall indicate the sequence number (as of the time the PDU is sent) of the next expected I PDU to be received, and consequently shall indicate that the I PDUs numbered up through N(R)-1 have been received correctly.

See 5.3.2.3 for a description of P/F bit functions.

5.4.2.2 Supervisory Format Commands and Responses. Supervisory, S, PDUs shall be used to perform numbered supervisory functions such as acknowledgments, temporary suspension of information transfer, or error recovery.

PDUs with the S format shall not contain an information field and, therefore, shall not increment the send state variable at the sender or the receive

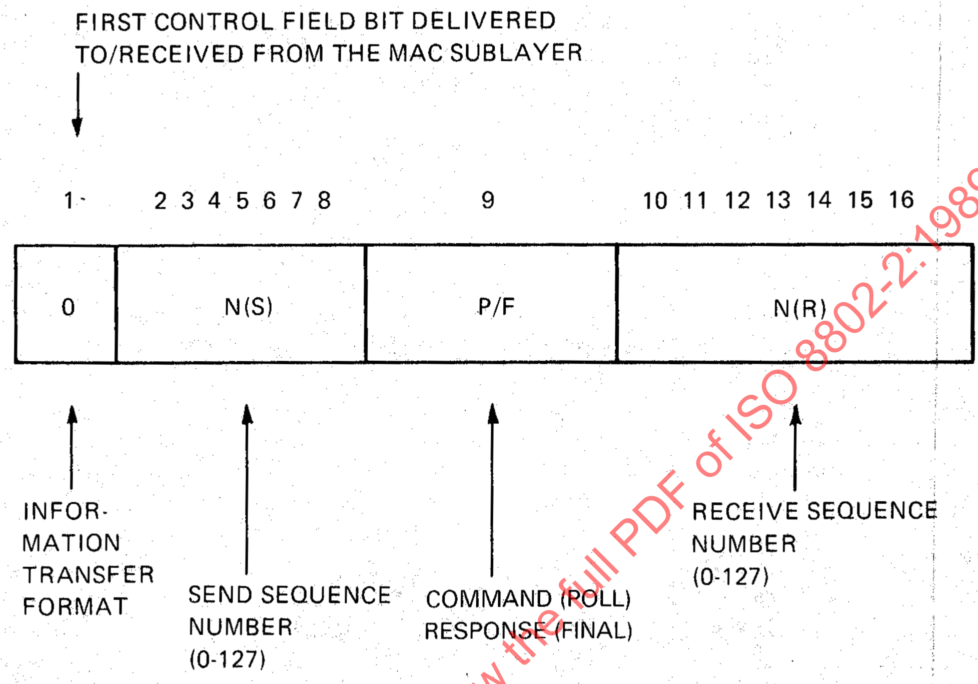


Fig 5-5
Information Transfer Format Control Field Bits

state variable at the receiver. The encoding of the S-format PDU control field for Type 2 operation shall be as shown in Fig 5-6.

An S-format PDU shall contain an N(R), receive sequence number, which shall indicate, at the time of sending, the sequence number of the next expected I PDU to be received, and consequently shall indicate that all received I PDUs numbered up through N(R)-1 have been received correctly.

When sent, an RR or REJ PDU shall indicate the clearance of any busy condition at the sending LLC that was indicated by the earlier sending of an RNR PDU.

See 5.3.2.3 for a description of the P/F bit functions.

5.4.2.2.1 Receive Ready (RR) Command and Response. The RR PDU shall be used by an LLC to indicate it is ready to receive an I PDU(s). I PDUs numbered up through N(R)-1 shall be considered as acknowledged.

5.4.2.2.2 Reject (REJ) Command and Response. The REJ PDU shall be used by an LLC to request the resending of I PDUs starting with the PDU numbered N(R). I PDUs numbered up through N(R)-1 shall be considered as acknowledged. It shall be possible to send additional I PDUs awaiting initial sending after the resent I PDU(s).

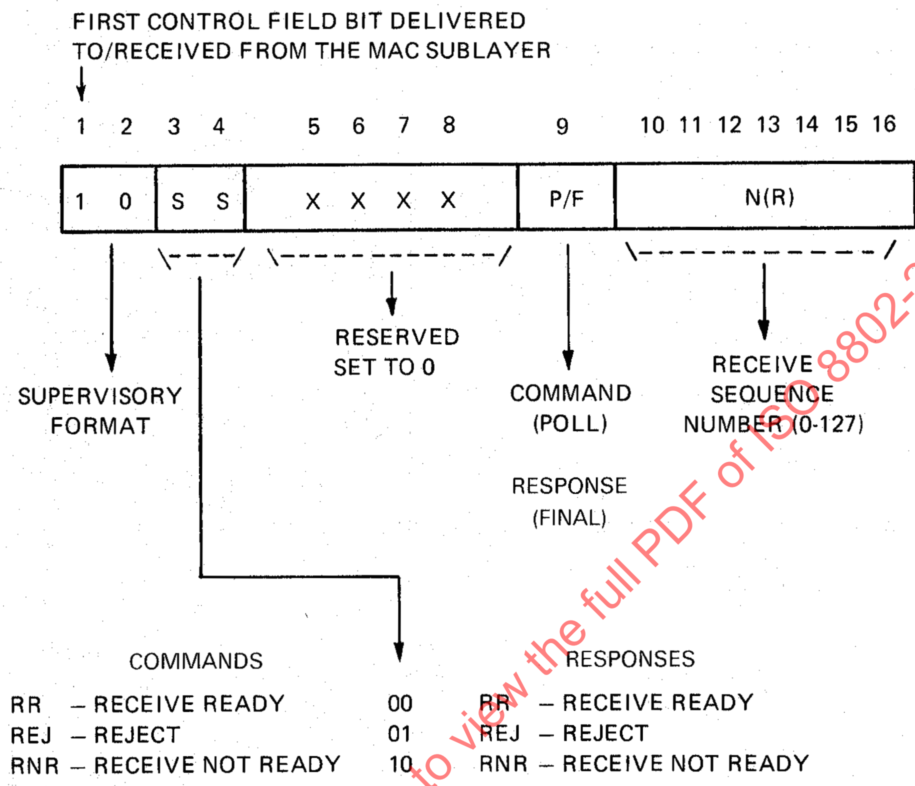


Fig 5-6
Supervisory Format Control Field Bits

With respect to each direction of sending on a data link connection, only one "sent REJ" condition shall be established at any given time. The "sent REJ" condition shall be cleared upon the receipt of an I PDU with an N(S) equal to the N(R) of the REJ PDU. The "sent REJ" condition may be reset in accordance with procedures described in 7.5.4.

5.4.2.2.3 Receive Not Ready (RNR) Command and Response. The RNR PDU shall be used by an LLC to indicate a busy condition (i.e., a temporary inability to accept subsequent I PDUs). I PDUs numbered up through N(R)-1 shall be considered as acknowledged. I PDUs numbered N(R) and any subsequent I PDUs received, if any, shall not be considered as acknowledged; the acceptance status of these PDUs shall be indicated in subsequent exchanges.

5.4.2.3 Unnumbered Format Commands and Responses. Unnumbered, U, commands and responses shall be used in Type 2 operation to extend the number of data link connection control functions. PDUs sent with the U format shall not increment the state variables on the data link connection at

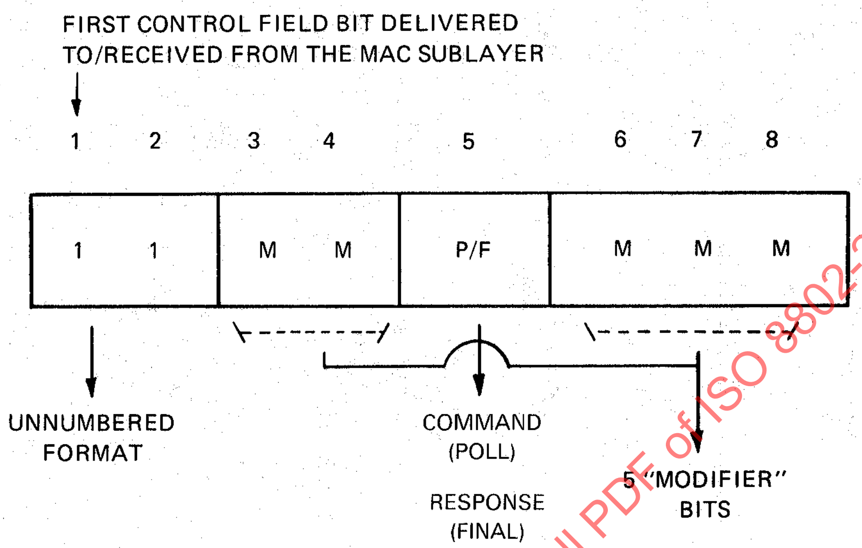


Fig 5-7a
Unnumbered Format Control Field Bits

FIRST CONTROL FIELD BIT DELIVERED
TO/RECEIVED FROM THE MAC SUBLAYER

1	2	3	4	5	6	7	8	
1	1	1	1	P	1	1	0	SABME COMMAND
1	1	0	0	P	0	1	0	DISC COMMAND
1	1	0	0	F	1	1	0	UA RESPONSE
1	1	1	1	F	0	0	0	DM RESPONSE
1	1	1	0	F	0	0	1	FRMR RESPONSE

Fig 5-7b
Unnumbered Command and Response Control Field Bit Assignments

either the sending or the receiving LLC. The encoding of the U-format command/response PDU control field shall be as shown in Fig 5-7a.

The U-format command and response encodings for Type 2 operation are listed in Fig 5-7b.

See 5.3.2.3 for a description of the P/F bit functions.

5.4.2.3.1 Set Asynchronous Balanced Mode Extended (SABME) Command. The SABME command PDU shall be used to establish a data link connection to the destination LLC in the asynchronous balanced mode. No information shall be permitted with the SABME command PDU. The destination LLC shall confirm receipt of the SABME command PDU by sending a UA response PDU on that data link connection at the earliest opportunity according to whether a DL-CONNECT response or a DL-DISCONNECT request primitive is passed from the network layer to the LLC sublayer. Upon acceptance of the SABME command PDU, the destination LLCs send and receive state variables shall be set to zero. If the UA response PDU is received correctly, then the initiating LLC shall also assume the asynchronous balanced mode with its corresponding send and receive state variables set to zero.

Previously sent I PDUs that are unacknowledged when this command is actioned shall remain unacknowledged. Whether or not an LLC resends the contents of the information field of unacknowledged outstanding I PDUs shall be decided at a higher layer.

5.4.2.3.2 Disconnect (DISC) Command. The DISC command PDU shall be used to terminate an asynchronous balanced mode previously set by a SABME command PDU. It shall be used to inform the destination LLC that the source LLC is suspending operation of the data link connection and the destination LLC should assume the logically disconnected mode. No information field shall be permitted with the DISC command PDU. Prior to actioning the command, the destination LLC shall confirm the acceptance of the DISC command PDU by sending a UA response PDU on that data link connection.

Previously sent I PDUs that are unacknowledged when this command is actioned shall remain unacknowledged. Whether or not an LLC resends the contents of the information field of unacknowledged outstanding I PDUs shall be decided at a higher layer.

5.4.2.3.3 Unnumbered Acknowledgment (UA) Response. The UA response PDU shall be used by an LLC on a data link connection to acknowledge the receipt and acceptance of the SABME and DISC command PDUs. These received command PDUs shall not be actioned until the UA response PDU is sent. No information field shall be permitted with the UA response PDU.

5.4.2.3.4 Disconnect Mode (DM) Response. The DM response PDU shall be used to report status indicating that the LLC is logically disconnected from the data link connection and is, by system definition, in ADM. No information field shall be permitted with the DM response PDU.

5.4.2.3.5 Frame Reject (FRMR) Response. The FRMR response PDU shall be used by the LLC in the asynchronous balanced mode to report that one of the following conditions that is not correctable by resending the identical PDU resulted from the receipt of a PDU from the remote LLC:

(1) The receipt of a command PDU or a response PDU that is invalid or not implemented; examples of invalid PDUs include
 (a) the receipt of a supervisory or unnumbered PDU with an information field that is not permitted,
 (b) the receipt of an unsolicited F bit set to "1," and
 (c) the receipt of an unexpected UA response PDU.

(2) The receipt of an I PDU with an information field that exceeded the

established maximum information field length that can be accommodated by the receiving LLC for that data link connection.

(3) The receipt of an invalid N(R) from the remote LLC. [An invalid N(R) shall be defined as one that signifies an I PDU that has previously been sent and acknowledged, or signifies an I PDU that has not been sent and is not the next sequential I PDU awaiting to be sent.]

(4) The receipt of an invalid N(S) from the remote LLC. [An invalid N(S) shall be defined as an N(S) that is greater than or equal to the last sent N(R) + k, where k is the maximum number of outstanding I PDUs. The parameter k is the window size indicated in the XID PDU.]

The responding LLC shall send the FRMR response PDU at the earliest opportunity.

The LLC receiving the FRMR response PDU shall be responsible itself for initiating the appropriate mode setting or resetting corrective action by initializing both directions of transmissions on the data link connection, using the SABME and DISC command PDUs, as applicable.

An information field shall be returned with the FRMR response PDU to provide the reason for the PDU rejection. The information field shall contain the fields shown in Fig 5-8.

Fig 5-8
FRMR Information Field Format

FIRST CONTROL FIELD BIT DELIVERED TO/RECEIVED FROM THE MAC SUBLAYER

1 - - - - - - - - - - - - -16	17	18 - - 24	25	26 - - 32	33 - - 36	37 - - 40
REJECTED PDU CONTROL FIELD	0	V(S)	C/R	V(R)	WXYZ	V000

(1) Rejected PDU control field shall be the control field of the received PDU which caused the FRMR exception condition on the data link connection. When the rejected PDU is a U-format PDU, the control field of the rejected PDU shall be positioned in bit positions 1-8, with 9-16 set to 0.

(2) V(S) shall be the current send state variable value for this data link connection at the rejecting LLC (bit 18 = low-order bit).

(3) C/R set to "1" shall indicate that the PDU which caused the FRMR was a response PDU, and C/R set to "0" shall indicate that the PDU which caused the FRMR was a command PDU.

(4) V(R) shall be the current receive state variable value for this data link connection at the rejecting LLC (bit 26 = low-order bit).

(5) W set to "1" shall indicate that the control field received and returned in bits 1 through 16 was invalid or not implemented. Examples of invalid PDU are defined as

(a) the receipt of a supervisory or unnumbered PDU with an information field which is not permitted;

(b) the receipt of an unsolicited F bit set to "1"; and

(c) the receipt of an unexpected UA response PDU.

(6) X set to "1" shall indicate that the control field received and returned in bits 1 through 16 was considered invalid because the PDU contained an information field which is not permitted with this command or response. Bit W shall be set to "1" in conjunction with this bit.

(7) Y set to "1" shall indicate that the information field received exceeded the established maximum information field length which can be accommodated by the rejecting LLC on that data link connection.

(8) Z set to "1" shall indicate that the control field received and returned in bits 1 through 16 contained an invalid N(R).

(9) V set to "1" shall indicate that the control field received and returned in bits l through l6 contained an invalid N(S). Bit W shall be set to "1" in conjunction with this bit.

6. LLC Description of the Type 1 Procedures

6.1 Modes of Operation. In Type 1 operation, no modes of operation are defined. An LLC using Type 1 procedures shall support the entire procedure set whenever it is operational on the local area network.

6.2 Procedure for Addressing. The address fields shall be used to indicate the source (SSAP) and destination (DSAP) of the LLC PDU. The first bit in the source address field (SSAP) shall be used to identify whether a command or a response is contained in the PDU.

Individual, group, global, and null addressing shall be supported for destination DSAP addresses. The source address field (SSAP) shall contain either an individual or null source address (see 3.3.1.2).

6.3 Procedure for the Use of the P/F Bit. A UI command PDU shall only be sent with the P bit set to"0". If a UI command PDU is received with the P bit set to "1", the LLC sublayer shall optionally discard it or pass it to the higher layer with a flag identifying that the P bit was set to "1". Since a UI PDU shall not be sent as a response PDU, procedures regarding the use of the F bit do not apply.

An XID command PDU shall have the P bit set to either "0" or "1". Upon receipt of an XID command PDU, the receiving LLC shall return an XID response PDU which has the F bit set equal to the value of the P bit contained in the incoming command PDU.

A TEST command PDU shall have the P bit set to either "0" or "1". Upon receipt of a TEST command PDU, the receiving LLC shall return a TEST response PDU which has the F bit set equal to the value of the P bit contained in the incoming command PDU.

6.4 Procedures for Logical Data Link Set-Up and Disconnection. Type 1 operation does not require any prior data link connection establishment (set-up), and hence no data link disconnection. Once the service access point has been enabled within the LLC, presumably by layer management's request, information may be sent to, or received from, a remote LLC service access point which is also participating in Type 1 operation.

6.5 Procedures for Information Transfer

6.5.1 Sending UI PDUs. Information transfer shall be accomplished by sending the UI command PDU with the P bit set to "0". Sending UI PDUs with the P bit set to "1" or as response PDUs is prohibited. It shall be possible to send the UI command PDU at any time.

6.5.2 Receiving UI PDUs. Reception of the UI command PDU shall not be acknowledged or sequence number verified by the logical data link procedures; therefore, the UI PDU may be lost if a logical data link exception occurs during the sending of the command PDU. It shall be possible to receive a UI command PDU at any time. However, local conditions at the receiver may result in the discarding of valid UI command PDUs by the receiving LLC. UI command PDUs that are received with the P bit set to "1" shall optionally be discarded or passed to the higher layer with a flag identifying that the P bit was set to "1".

UI PDUs that are response PDUs are invalid transmissions and shall be discarded by the receiving LLC.

6.6 Uses of the XID Command PDU and Response PDU. While the response to an XID command PDU shall be mandatory, the origination of an XID command PDU shall be optional. It shall be possible for the XID capabilities to be used as a part of some network control functions. As such, an XID command PDU may be sent on direction from a higher layer function, an administration function having access to the data link layer, or an automatic start-up function. However, it shall also be possible for a more capable implementation of LLC to incorporate the use of the XID function directly to make more efficient use of the protocol.

Some possible uses of the XID capabilities include:

(1) The XID command PDU with a null DSAP and a null SSAP is a way to solicit a response from any station (i.e., any DA). As such it represents a basic "Are You There?" test capability.

(2) The XID command PDU with a group DA or group DSAP address can be used to determine the group membership. In particular, the XID command PDU with a global DA address can identify all active stations.

(3) A duplicate address check can be made (see Table 6-1a).

(4) For Class II LLCs in ABM, an XID exchange can be used to identify the receive window size at each LLC for that data link connection.

NOTE: The use of an XID exchange for this purpose is not valid in the ADM state.

(5) An XID exchange with a null DSAP and a null SSAP can identify the class of each LLC.

(6) An XID exchange with a specific DSAP and a specific SSAP can identify the service types supported by those service access points.

(7) An LLC can announce its presence with a global DA address in an XID PDU.

6.7 Uses of the TEST Command PDU and Response PDU. The TEST function provides a facility to conduct loopback tests of the LLC to LLC transmission path. The initiation of the TEST function may be caused by an administration or management entity within the data link layer. Successful completion of the test consists of sending a TEST command PDU with a particular information field provided by this administration or management entity to the desig-

Table 6-1a
Station Component State Transitions

Current State	Event	Action(s)	Next State
DOWN_STATE	[ENABLE_WITH_DUPLICATE_ADDRESS_CHECK]	SEND_NULL_DSAP_XID_C START_ACK_TIMER RETRY_COUNT:=0 XID_R_COUNT:=0	DUPLICATE_ADDRESS_CHECK_STATE
	[ENABLE_WITHOUT_DUPLICATE_ADDRESS_CHECK]	REPORT_STATUS(STATION_UP)	UP_STATE
UP_STATE	DISABLE_REQUEST	REPORT_STATUS(STATION_DOWN)	DOWN_STATE
	RECEIVE_NULL_DSAP_XID_C	SEND_XID_R	UP_STATE
	RECEIVE_NULL_DSAP_TEST_C	SEND_TEST_R	UP_STATE
DUPLICATE_ADDRESS_CHECK_STATE (OPTIONAL)	[RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=0]	XID_R_COUNT:=XID_R_COUNT+1	DUPLICATE_ADDRESS_CHECK_STATE
	[RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=1]	REPORT_STATUS(DUPLICATE_ADDRESS_FOUND)	DOWN_STATE
	[RECEIVE_NULL_DSAP_XID_C]	SEND_XID_R	DUPLICATE_ADDRESS_CHECK_STATE
	[ACK_TIMER_EXPIRED_AND_RETRY_COUNT<MAXIMUM_RETRY]	SEND_NULL_DSAP_XID_C START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1 XID_R_COUNT:=0	DUPLICATE_ADDRESS_CHECK_STATE
	[ACK_TIMER_EXPIRED_AND_RETRY_COUNT=MAXIMUM_RETRY]	REPORT_STATUS(STATION_UP)	UP_STATE
	[DISABLE_REQUEST]	REPORT_STATUS(STATION_DOWN)	DOWN_STATE

nated destination LLC address and receiving, in return, the identical information field in a TEST response PDU.

Implementation of the TEST command PDU is optional but every LLC must be able to respond to a received TEST command PDU with a TEST response PDU. The length of the information field is variable from 0 to the largest size specified that each LLC on this local area network must support for normal data transfer.

It shall also be possible to send even larger information fields with the following interpretations. If the receiving LLC can successfully receive and return the larger information field, it will do so. If it cannot receive the entire information field but the MAC can detect a satisfactory FCS, the LLC shall discard the portion of the information field received, and may return a TEST response PDU with no information field. If the MAC cannot properly compute the FCS for the overlength information fields, the LLC shall discard the portion of the information field received, and shall give no response. Any TEST command PDU received in error shall be discarded and no response PDU sent. In the event of failure, it shall be the responsibility of the administration or management entity which initiated the TEST function to determine any future actions.

6.8 List of Logical Data Link Parameters. A number of logical data link parameters are defined, the range of values for which are determined on a system-by-system basis by the user at the time that the local area network is established.

The logical data link parameters for Type 1 operation shall be as follows:

6.8.1 Maximum Number of Octets in a UI PDU. Refer to the appropriate MAC protocol specification for any limitation on the maximum number of octets in a UI PDU. No restrictions are imposed by the LLC sublayer. However, in the interest of having a value that all users of Type I LLC may depend upon, all MACs must at least be capable of accommodating UI PDUs with information fields up to and including 128 octets in length.

6.8.2 Minimum Number of Octets in a PDU. The minimum length valid PDU shall contain exactly two service access point address fields and one control field in that order. Thus the minimum number of octets in a valid PDU shall be 3.

6.9 Precise Description of the Type 1 Procedures. If discrepancies appear to exist with the text found in the balance of Section 6, this subsection (6.9) shall be viewed as being the definitive description.

6.9.1 LLC Precise Specification. The operation of the LLC is logically divided into several components. Each component characterizes a set of protocol operations performed by an LLC entity and is defined using a protocol state machine description. These state machines do not specify particular implementation techniques; rather, they are intended to describe the "external" characteristics of an LLC entity, as perceived by an LLC entity in a remote station or by a higher layer protocol in the local station.

The LLC operation is described using the following three types of components:

(1) *Station Component*. This component is responsible for processing the events which affect the entire LLC entity. The station component handles PDUs addressed to the null DSAP address and processes the duplicate address check, if implemented. One station component shall exist for each MAC service access point present on the local area network.

(2) *Link Service Access Point (SAP) Component*. This component is responsible for processing the events which affect a specific operating service access point. One SAP component shall exist for each SAP within the LLC entity.

(3) *Connection Component*. This component is responsible for processing the events which affect a specific data link connection for Type 2 procedures only (see 7.9 below). One connection component shall exist for each data link connection supported in the LLC entity.

The operation of each component is described using a state machine description. These important points are assumed in these descriptions:

(a) The components are hierarchically related; i.e., the station component is the "parent" of the SAP component, which in turn is the "parent" of the connection component. See Fig 6-1.

(b) Each "parent" component has a state which provides the enabling conditions for its "child" component(s) to operate. If the parent component leaves this state, then the "child" component(s) are disabled.

(c) For each "parent" component, several "child" components are allowed to be concurrently operating once their "parent" enabling conditions have been satisfied.

(d) There exists for each MAC service access point one and only one LLC entity, consisting of the various operating components.

(e) In Class I LLC operation, each LLC can have zero or more SAPs being serviced (i.e., active) at any one time, independent of each other, which are differentiated by the DSAP address. The services of a SAP shall be provided by a separate service access point component.

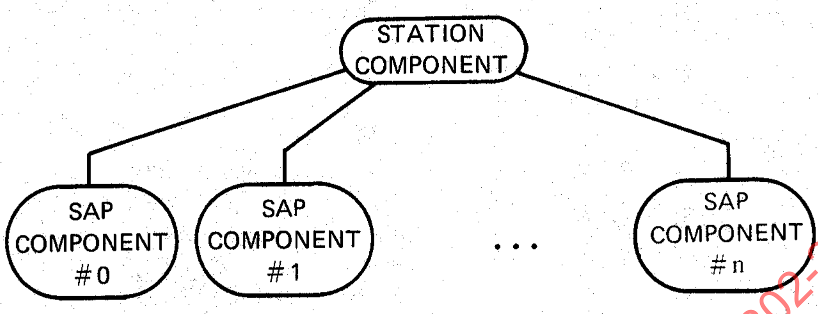
(f) In Class II LLC operation, the services of each SAP can also support zero or more concurrent data link connections (each designated by the logical concatenation of the MAC address (DA/SA) and the LLC address (DSAP/SSAP). Each data link connection is controlled by a separate connection component.

Each component description shall consist of these sections:

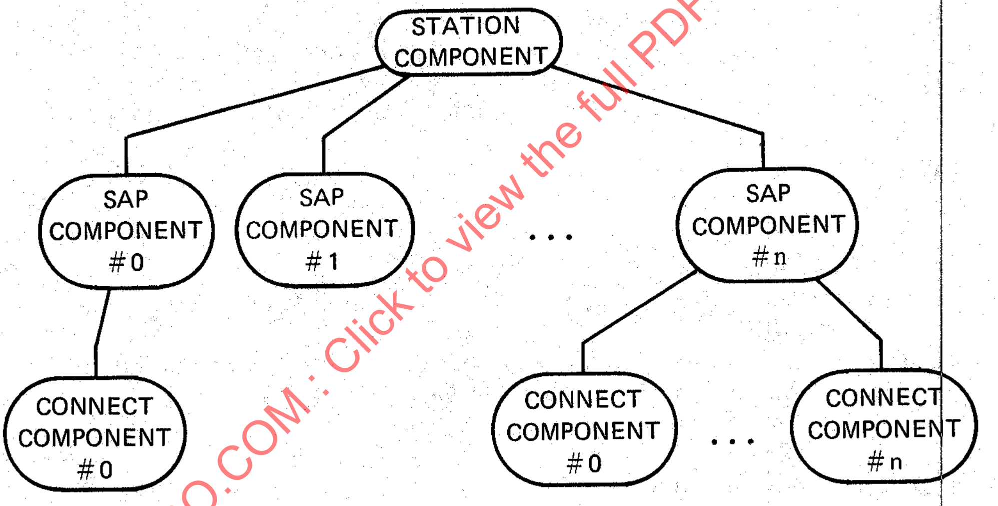
(i) *Component Overview*. This section shall discuss the overall purpose behind the component operation.

(ii) *Component State Transition Diagram*. This diagram shall graphically represent the component machine overview.

(iii) *Component State Transition Table*. This chart shall display a table of the state transitions, including columns for current state, event, action(s), and next state. This table shall define all valid events for each state as well as the resultant component action(s) and state change.



(a) Class I LLC Component Relationship



(b) Class II LLC Component Relationship

Fig 6-1
Component Relationships

(iv) *Component Event Description.* Each of the events which are used in the state transition table is explained.

(v) *Component Action Description.* Each of the actions which are used in the state transition table is explained.

(vi) *Component State Description.* Each of the states that are used in the state transition table are explained.

The following basic state machine operation rules apply:

(A) Events shall cause a state transition in the machine, and shall result in execution of some action(s) along with a state change (which may return to the same state).

(B) Events which are not listed as valid inputs to the current state of any of the operating components shall not cause
(I) actions or state changes; or
(II) PDU transmissions.
The station should perform some error recovery which is appropriate for the particular implementation.

(C) If an incoming PDU is destined for a DSAP which is not active (i.e., the appropriate component is not operating), it shall be considered to be an exception and dealt with in a manner appropriate for the receiving station.

6.9.2 Station Component Overview. The station component is responsible for handling all events that are directed to the LLC as a whole (i.e., events affecting all SAPs and connections serviced by that LLC). The station component shall begin in the DOWN state, optionally check for a duplicate station address, and potentially enter the UP state; see Fig 6-2 and Table 6-1a. The UP state of the LLC station component provides the enabling conditions for the operation of the service access point (SAP) components.

The station component shall be capable of receiving and responding to the XID and TEST command PDUs. It shall optionally be capable of initiating the XID command PDU, if duplicate address checking is performed by the LLC entity in a particular implementation; see Table 6-1b. These PDUs shall use the null DSAP address to denote that the station component is being referenced.

The performance of the duplicate address check requires that the station component be prepared to receive its own XID PDUs. The definition of the MAC operation provides for the ability to simultaneously transmit and receive. Since the DA=SA in the XID PDUs can be used for duplicate address check, the MAC will recognize its own address and pass the PDU to the station component. The station component will respond to an XID command PDU with an XID response PDU, regardless of whether it originated from itself or a remote LLC. The station component provides the duplicate address check by maintaining a count of received XID response PDUs. If more than one XID response PDU is received, then at least one other identical MAC DA exists on the LAN. See Fig 6-2 and Table 6-1a for details.

6.9.2.1 Station Component State Descriptions

(1) **DOWN_STATE.** The station component is powered off, not initialized, and/or disabled from operating in the local area network.

(2) **DUPLICATE_ADDRESS_CHECK_STATE.** The station component is in the process of checking for duplicate MAC addresses on the LAN. The main purpose of this state shall be to allow the LLC station component to verify that this station's MAC address is unique on the LAN. The station component shall send XID command PDUs with identical MAC DA and SA addresses, and shall wait for a possible XID Response PDU indicating the existence of other stations with identical MAC link addresses.

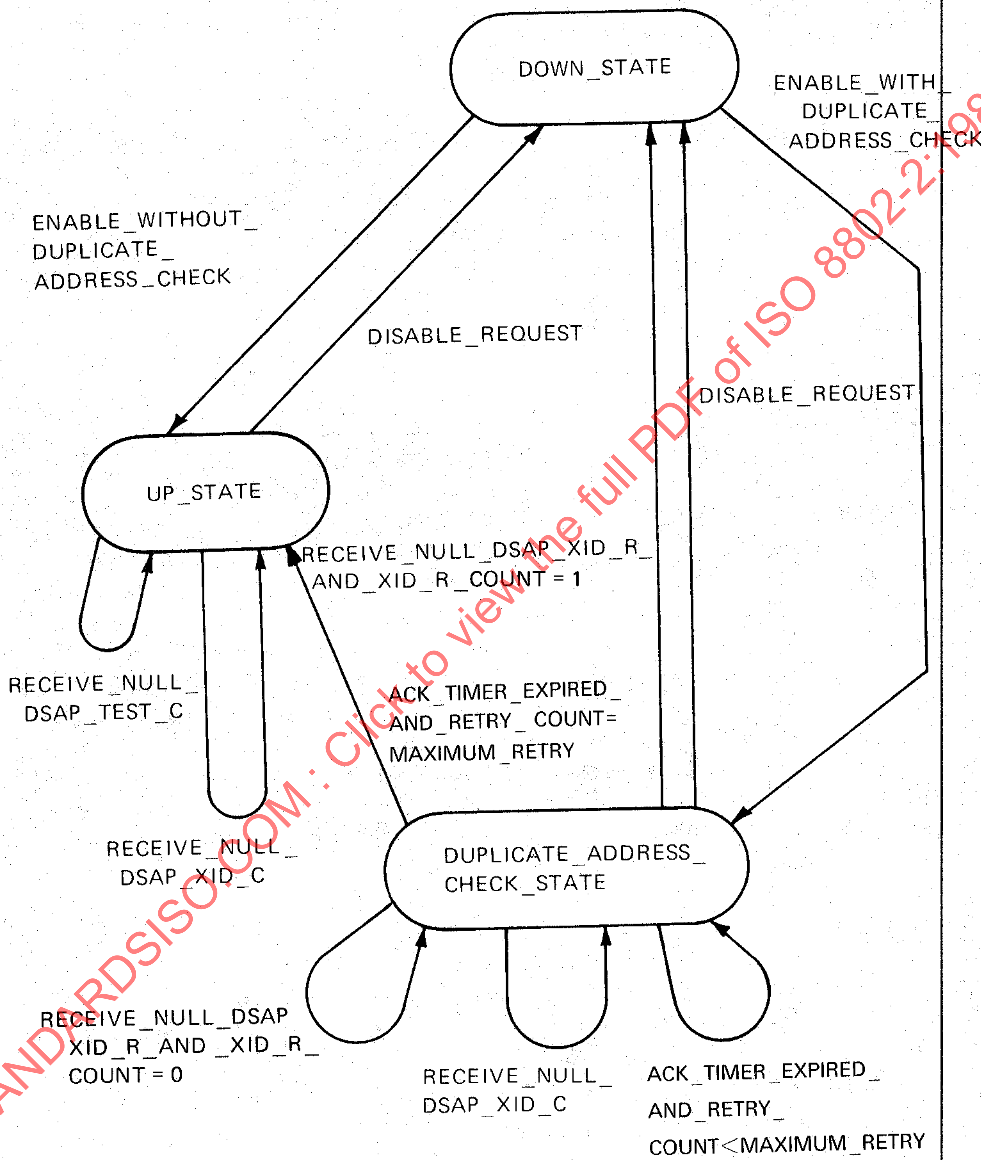


Fig 6-2
Station Component State Diagram

Table 6-1b
Station Component Options

Description	States Omitted	Other Requirements
No duplicate address check	DUPLICATE_ADDRESS_ CHECK_STATE	Omit: ENABLE_WITH_DUPLICATE_ ADDRESS_CHECK ACK_TIMER_EXPIRED_AND_ RETRY_COUNT < MAXIMUM_ RETRY ACK_TIMER_EXPIRED_AND_ RETRY_COUNT = MAXIMUM_ RETRY RECEIVE_NULL_DSAP_ XID_R_AND_XID_R_ COUNT = 1 RECEIVE_NULL_DSAP_ XID_R_AND_XID_R_ COUNT = 0
Optional use of duplicate address check	none	Omit: none
Always perform duplicate address check	none	Omit: ENABLE_WITHOUT_ DUPLICATE_ADDRESS_ CHECK

(3) **UP_STATE.** The station component is enabled, powered on, initialized, and operating in the local area network. The LLC shall allow SAPs to exchange LLC PDUs on the medium.

6.9.2.2 Station Component Event Descriptions

(1) **ENABLE_WITH_DUPLICATE_ADDRESS_CHECK.** Station component user has initialized/enabled the station equipment, and has requested that the LLC check for MAC service access point address duplications before participating in data link communications.

(2) **ENABLE_WITHOUT_DUPLICATE_ADDRESS_CHECK.** Station component user has initialized/enabled the station equipment, but duplicate MAC service access point address checking by the LLC is not supported/desired.

(3) **ACK_TIMER_EXPIRED_AND_RETRY_COUNT<MAXIMUM_RETRY.** Acknowledgment timer has expired and retry count is less than maximum retry limit.

(4) **ACK_TIMER_EXPIRED_AND_RETRY_COUNT=MAXIMUM_RETRY.** Acknowledgment timer has expired and retry count is equal to the maximum retry limit.

(5) **RECEIVE_NULL_DSAP_XID_C.** An XID command PDU with the null DSAP address has been received.
(6) **RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=0.** A single XID response PDU with the null DSAP address has been received.
(7) **RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=1.** A second XID response PDU with the null DSAP address have been received.
(8) **RECEIVE_NULL_DSAP_TEST_C.** A TEST command PDU with the null DSAP address has been received.
(9) **DISABLE_REQUEST.** Station user has requested that the equipment be disabled from operating on the medium.

6.9.2.3 Station Component Action Descriptions

(1) **START_ACK_TIMER.** Start the acknowledgment timer. This allows the LLC to determine that it has not received an acknowledgment from the remote station within a specified response time.
(2) **RETRY_COUNT:=0.** Initialize the retry counter.
(3) **RETRY_COUNT:=RETRY_COUNT+1.** Increment the retry counter.
(4) **XID_R_COUNT:=0.** Initialize the XID response PDU counter.
(5) **XID_R_COUNT:=XID_R_COUNT+1.** Increment the XID response PDU counter.
(6) **SEND_NULL_DSAP_XID_C.** The LLC shall send an XID command PDU with null SSAP and null DSAP addresses and with identical MAC DA and SA addresses.
(7) **SEND_XID_R.** The LLC shall send an XID response PDU, using the SSAP address of the XID command PDU as the DSAP address of the response PDU, and using a null SSAP address.
(8) **SEND_TEST_R.** The LLC shall send a TEST response PDU, using the SSAP address of the TEST command PDU as the DSAP address of the response PDU, and using a null SSAP address.
(9) **REPORT_STATUS.** The LLC shall be able to report data link status conditions, with the following valid reasons:
  (a) **STATION_UP.** LLC entity is now operational.
  (b) **STATION_DOWN.** The LLC entity is now non-operational.
  (c) **DUPLICATE_ADDRESS_FOUND.** LLC entity has detected another LLC entity on the LAN with a MAC service access point address identical to its own.

6.9.3 Service Access Point (SAP) Component Overview. The service access point (SAP) component handles all LLC Type 1 PDU traffic for a particular DSAP address in the local station component. The local service access point user is able to activate and deactivate the operation of each individual SAP component in the station component (see Table 6-2 and Fig 6-3). Once active, the SAP component shall process Type 1 LLC PDUs addressed to the DSAP and send Type 1 LLC PDUs either by service access point user request or as a result of some LLC protocol action.

For Class II stations, the ACTIVE state of the SAP component provides the activating conditions for Type 2 LLC connection component services (see

Table 6-2
Service Access Point Component State Transitions

Current State	Event	Action(s)	Next State
INACTIVE_STATE	SAP_ACTIVATION_REQUEST	REPORT_STATUS(SAP_ACTIVE)	ACTIVE_STATE
ACTIVE_STATE	RECEIVE_UI	UNITDATA_INDICATION	ACTIVE_STATE
	UNITDATA_REQUEST	SEND_UI	ACTIVE_STATE
	XID_REQUEST	SEND_XID_C	ACTIVE_STATE
	RECEIVE_XID_C	SEND_XID_R	ACTIVE_STATE
	RECEIVE_XID_R	XID_INDICATION	ACTIVE_STATE
	TEST_REQUEST	SEND_TEST_C	ACTIVE_STATE
	RECEIVE_TEST_C	SEND_TEST_R	ACTIVE_STATE
	RECEIVE_TEST_R	TEST_INDICATION	ACTIVE_STATE
	SAP_DEACTIVATION_REQUEST	REPORT_STATUS(SAP_INACTIVE)	INACTIVE_STATE

Fig 6-1). Any attempt to make a data link connection, either by the user or a remote LLC, while the SAP component is ACTIVE, shall be passed to the Type 2 LLC connection component and ignored by the SAP component (this includes the handling of the disconnect mode for a Type 2 LLC connection component).

6.9.3.1 Service Access Point (SAP) Component State Descriptions

(1) **INACTIVE_STATE.** LLC SAP component is not active, functioning, or operational. No PDUs are accepted and/or sent.
(2) **ACTIVE_STATE.** LLC SAP component is active, functioning, and operational. PDUs are received and sent.

6.9.3.2 Service Access Point (SAP) Component Event Description

(1) **SAP_ACTIVATION_REQUEST.** The SAP user has requested that the particular LLC SAP component be activated and begin logical data link operation of the Type 1 services.
(2) **SAP_DEACTIVATION_REQUEST.** The SAP user has requested that the particular LLC SAP component be deactivated and no longer allowed to operate on the logical data link.
(3) **XID_REQUEST.** The SAP user has requested that the LLC SAP component send an XID command PDU to one or more remote SAPs.

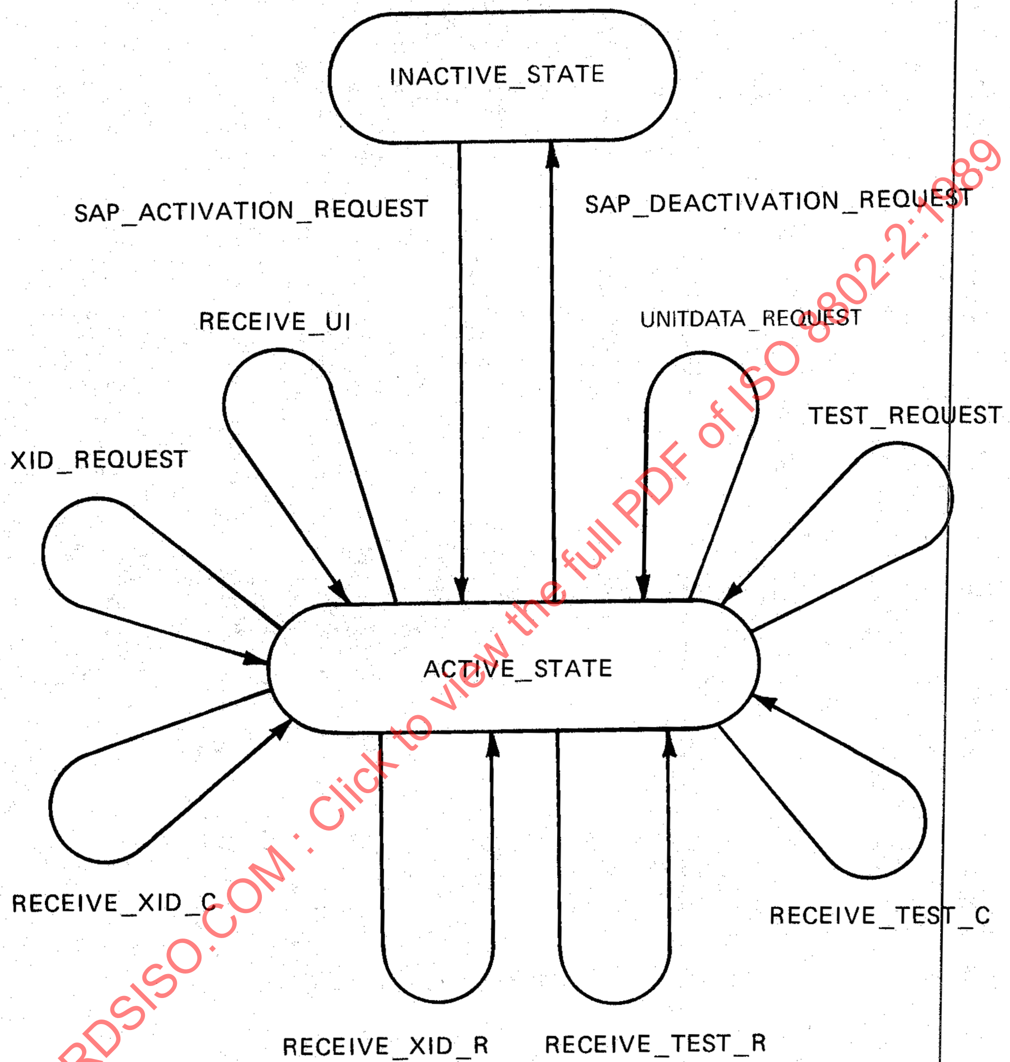


Fig 6-3
Service Access Point Component State Diagram

(4) **TEST_REQUEST.** The SAP user has requested that the LLC SAP component send a TEST command PDU to one or more remote SAPs.
(5) **RECEIVE_UI.** The local SAP component has received a UI PDU from a remote SAP.
(6) **UNITDATA_REQUEST.** The SAP user has requested that a data unit be passed to a remote LLC SAP, via a UI PDU.

(7) **RECEIVE_XID_C.** The local SAP component has received an XID command PDU from a remote SAP.
(8) **RECEIVE_XID_R.** The local SAP component has received an XID response PDU from a remote SAP.
(9) **RECEIVE_TEST_C.** The local SAP component has received a TEST command PDU from the remote SAP.
(10) **RECEIVE_TEST_R.** The local SAP component has received a TEST response PDU from the remote SAP.

6.9.3.3 Service Access Point (SAP) Component Action Descriptions

(1) **UNITDATA_INDICATION.** LLC SAP component has received a UI PDU from a remote SAP. The service data unit is given to the SAP user.
(2) **SEND_UI.** A UI PDU is sent to one or more remote SAPs in response to a user request to send a service data unit.
(3) **SEND_XID_C.** LLC SAP component shall send an XID command PDU to remote SAPs in response to a SAP user request to identify other SAPs.
(4) **SEND_XID_R.** LLC SAP component shall send an XID response PDU to remote SAPs in response to a received XID command PDU.
(5) **SEND_TEST_C.** LLC SAP component shall send a TEST command PDU in response to SAP user request to test a remote SAP.
(6) **SEND_TEST_R.** LLC SAP component shall send a TEST response PDU in response to a remote LLC TEST command PDU.
(7) **REPORT_STATUS.** The LLC SAP component shall be able to report data link status conditions for the particular SAP component with the following valid reasons:
  (a) **SAP_ACTIVE.** The SAP_ACTIVATION_REQUEST has been successfully processed and the component is now operational.
  (b) **SAP_INACTIVE.** The SAP_DEACTIVATION_REQUEST has been successfully processed and the component is now deactivated.
(8) **XID_INDICATION.** LLC SAP component has received an XID response PDU from a remote SAP. An indication of this event is passed to the SAP user, and may also return the XID information field.
(9) **TEST_INDICATION.** LLC SAP component has received a TEST response PDU from a remote SAP. An indication of this event is passed to the SAP user, and may also return the TEST information field.

7. LLC Description of the Type 2 Procedures

7.1 Modes. In Type 2 operation, two modes of operation are defined, an operational mode and a non-operational mode.

7.1.1 Operational Mode. The one operational mode shall be the asynchronous balanced mode (ABM).

ABM is a balanced operational mode where a data link connection has been established between two service access points. Either LLC shall be able to send commands at any time and initiate response transmissions without receiving explicit permission from the other LLC. Such an asynchronous transmission shall contain one or more LLC PDUs and shall be used for information field transfer and/or to indicate status changes in the LLC (for example, the number of the next expected information LLC PDU, transition from a ready to a busy condition or vice versa, occurrence of an exception condition).

ABM consists of a data link connection phase, an information transfer phase, a data link resetting phase, and a data link disconnection phase.

7.1.2 Non-operational Mode. The one non-operational mode shall be the asynchronous disconnected mode (ADM).

ADM differs from the operational mode (ABM) in that the data link connection is logically disconnected from the physical medium; i.e., no information (user data) shall be sent or accepted.

ADM is defined to prevent a data link connection from appearing on the physical medium in a fully operational mode during unusual situations or exception conditions since such operation could cause the following:

(1) sequence number mismatch between the LLCs on the data link connection, or

(2) ambiguity in one LLC as to the status of another LLC.

A data link connection shall be system predefined as to the condition(s) that cause it to assume the asynchronous disconnected mode (ADM).

Examples of possible conditions (in addition to receiving a DISC command PDU) which shall cause a data link connection to enter ADM are

(a) the power is turned on,

(b) the data link layer logic is manually reset, or

(c) the data link connection is manually switched from a local (home) condition to the connected-on-the-data-link (on-line) condition.

A LLC on a data link connection in ADM shall be required to monitor transmissions received from its MAC for the purpose of

(i) accepting and responding to one of the mode setting command PDUs (SABME, DISC), or

(ii) sending a DM response PDU at a medium access opportunity, when required.

In addition, since the LLC has the ability to send command PDUs at any time, the LLC may send an appropriate mode setting command PDU.

An LLC in ADM receiving a DISC command PDU shall respond with the DM response PDU. An LLC in ABM receiving a DISC command PDU shall respond with the unnumbered acknowledgment (UA) response PDU if it is capable of actioning the command.

An LLC in ADM shall not establish a frame reject exception condition (see 5.4.2.3.5 and 7.6). ADM consists of a data link disconnected phase.

7.2 Procedure for Addressing. The address fields shall be used to indicate the source (SSAP) and destination (DSAP) of the PDU. The first bit in the source address field (SSAP) shall be used to identify whether a command or response is contained in the PDU.

A single data link connection can be established between any two service access points on the local area network. This data link connection is identified by a pair of "complete" data link addresses, each of which consists of a logical concatenation of the implicit physical address (not contained in the frame structure), the MAC address (DA/SA), and the LLC address (DSAP/SSAP). In order for a receiving DSAP to correctly identify the data link connection associated with an incoming PDU, the receiving DSAP must have access to the "complete" data link address information for the remote service access point.

7.3 Procedures for the Use of the P/F Bit. The LLC receiving a command PDU (SABME, DISC, RR, RNR, REJ, or I) with the P bit set to "1", shall send a response PDU with the F bit set to "1".

The response PDU returned by a LLC to a SABME or DISC command PDU with the P bit set to "1" shall be a UA or DM response PDU with the F bit set to "1". The response PDU returned by an LLC to an I, RR, or REJ command PDU with the P bit set to "1" shall be an I, RR, REJ, RNR, DM, or FRMR response PDU with the F bit set to "1". The response PDU returned by an LLC to an RNR command PDU with the P bit set to "1" shall be an RR, REJ, RNR, DM, or FRMR response PDU with the F bit set to "1".

NOTE: The P bit is usable by the LLC in conjunction with the timer recovery condition (see 7.5.9).

7.4 Procedures for Data Link Set-Up and Disconnection

7.4.1 Data Link Connection Phase. Either LLC shall be able to take the initiative to initialize the data link connection.

When the LLC wishes to initialize the link, it shall send the SABME command PDU and start the acknowledgment timer (see 7.8.1 below). Upon reception of the UA response PDU, the LLC shall have reset both its send and receive state variables V(S) and V(R) to 0 for the corresponding data link connection, shall stop its acknowledgment timer, and shall enter the information transfer phase.

When receiving the DM response PDU, the LLC which originated the SABME command PDU shall stop its acknowledgment timer, shall not enter the information transfer phase, and shall report to the higher layer for appropriate action.

For a description of the actions to be followed upon receipt of a SABME or DISC command PDU, see 7.4.5. Other Type 2 PDUs received (commands and responses) while attempting to connect shall be ignored by the LLC.

Should the Acknowledgement Timer run out before reception of the UA or DM response PDU, the LLC shall resend the SABME command PDU and restart the acknowledgment timer. After resending the SABME command PDU N2 times, the sending LLC shall stop sending the SABME command PDU and shall report to the higher layer for the appropriate error recovery action to initiate. The value of N2 is defined in 7.8.2 below.

When receiving an SABME command PDU, the LLC shall pass the indication to the network layer to indicate that an establishment of the data link connection is being requested from the remote LLC.

Thereafter, if the LLC receives a notification from the network layer to accept the connection, it shall return a UA response PDU to the remote LLC and set both its send and receive state variables V(S) and V(R) to 0 for the corresponding data link connection and enter the information transfer phase. The return of the UA response PDU shall take precedence over any other response PDU for the same data link connection which may be pending at the LLC. It shall be possible to follow the UA response PDU with additional LLC PDUs, if pending.

If the LLC receives a notification from the network layer not to enter the indicated phase, it shall return a DM response PDU to the remote LLC and remain in the link disconnected mode.

7.4.2 Information Transfer Phase. After having sent the UA response PDU to an SABME command PDU or having received the UA response PDU to a sent SABME command PDU, the LLC shall accept and send I-format and S-format PDUs according to the procedures described in 7.5 below.

When receiving an SABME command PDU while in the information transfer phase, the LLC shall conform to the resetting procedure described in 7.6.

7.4.3 Data Link Disconnection Phase. During the information transfer phase, either LLC shall be able to initiate disconnecting of the data link connection by sending a DISC command PDU.

When the LLC wishes to disconnect the data link connection, it shall send the DISC command PDU and start the acknowledgment timer (see 7.8.1). Upon reception of the UA or DM response PDU from the remote LLC, the LLC shall stop its acknowledgment timer and enter the link disconnected mode.

Should the acknowledgment timer run out before reception of the UA or DM response PDU, the LLC shall resend the DISC command PDU and restart the acknowledgment timer. After sending the DISC command PDU N2 times, the sending LLC shall stop sending the DISC command PDU, shall enter the data link disconnected phase, and shall report to the higher layer for the appropriate error recovery action to initiate. The value of N2 is defined in 7.8.2.

When receiving a DISC command PDU, the LLC shall return a UA response PDU and enter the data link disconnected phase. The return of the UA response PDU shall take precedence over any other response PDU for the same data link connection which may be pending at the LLC.

7.4.4 Data Link Disconnected Phase. After having received a DISC command PDU from the remote LLC and returned a UA response PDU, or having received the UA response PDU to a sent DISC command PDU, the LLC shall enter the data link disconnected phase.

In the disconnected phase, the LLC shall be able to initiate data link connection. In the disconnected phase, the LLC shall react to the receipt of an SABME command PDU as described in 7.4.1 above and shall send a DM response PDU in answer to a received DISC command PDU.

When receiving any other Type 2 command PDU with the P bit set to "1" in the disconnected phase, the LLC shall send a DM response PDU with the F bit set to "1". Other Type 2 PDUs received in the disconnected phase shall be ignored by the LLC.

7.4.5 Contention of Unnumbered Mode Setting Command PDUs. A contention situation in a LLC shall be resolved in the following way.

If the sent and received mode setting command PDUs are the same, each LLC shall send the UA response PDU at the earliest opportunity. Each LLC shall enter the indicated phase either after receiving the UA response PDU, or after its acknowledgment timer expires.

If the sent and received mode setting command PDUs are different, each LLC shall enter the data link disconnected phase and shall issue a DM response PDU at the earliest opportunity.

7.5 Procedures for Information Transfer. The procedures which apply to the transfer of I PDUs in each direction on a data link connection during the information transfer phase are described below.

In the following, "number one higher" is in reference to a continuously repeated sequence series, i.e., 127 is one higher than 126 and 0 is one higher than 127 for modulo 128 series.

7.5.1 Sending I PDUs. When the LLC has an I PDU to send (i.e., an I PDU not already sent, or having to be resent as described in 7.5.5 below), it shall send the I PDU with an N(S) equal to its current send state variable V(S), and an N(R) equal to its current receive state variable V(R) for that data link connection. At the end of sending the I PDU, the LLC shall increment its send state variable V(S) by one.

If the acknowledgment timer is not running at the time that an I PDU is sent, the acknowledgment timer shall be started.

If the data link connection send state variable V(S) is equal to the last value of N(R) received plus k (where k is the maximum number of outstanding I PDUs, see 7.8.4) the LLC shall not send any new I PDUs on that data link connection, but shall be able to resend an I PDU as described in 7.5.6 or 7.5.9.

When a local LLC data link connection is in the busy condition, the LLC shall still be able to send I PDUs, provided that the remote LLC on this data link connection is not busy itself. When the LLC for a particular data link connection is in the FRMR exception condition, it shall stop transmitting I PDUs on that data link connection.

7.5.2 Receiving an I PDU. When the LLC data link connection is not in a busy condition and receives an I PDU whose send sequence number is equal to the receive state variable V(R), the LLC shall accept the information field of this PDU, increment by one its receive state variable V(R), and act as follows:

(1) If an I PDU is available to be sent, the LLC shall be able to act as in 7.5.1 above and acknowledge the received I PDU by setting N(R) in the control field of the next sent I PDU to the value of the receive state variable V(R). The LLC shall also be able to acknowledge the received I PDU by sending an RR PDU with the N(R) equal to the value of the receive state variable V(R).

(2) If no I PDU is available to be sent by the LLC, then the LLC shall either

(a) send an RR PDU with the N(R) equal to the value of the receive state variable V(R) at the earliest opportunity; or

(b) if the received PDU was not a command PDU with the P bit set to "1", wait for some period of time bounded by the probability of the remote acknowledgment timer expiry, for either an I PDU to become available for transmission, or to accumulate additional I PDUs to be acknowledged in a single RR PDU, subject to window size constraints.

(3) If receipt of the I PDU caused the LLC to go into the busy condition with regard to any subsequent I PDUs, the LLC shall send an RNR PDU with the N(R) equal to the value of the receive state variable V(R). If an I PDU(s) is available to send, the LLC shall be able to send them as in 7.5.1 above prior to or following the sending of the RNR PDU.

When the LLC associated with a particular data link connection is in a busy condition, and receives an in-sequence I PDU, the LLC shall be able to ignore the information field contained in any received I PDU on that data link connection (see 7.5.8).

7.5.3 Reception of Incorrect PDUs. When the LLC receives an invalid PDU (see 3.3.5) or a PDU with an incorrect DSAP or SSAP address, this PDU shall be discarded entirely.

7.5.4 Reception of Out-of-Sequence PDUs. When the LLC receives an I PDU whose send sequence number is not in sequence, i.e., not equal to the current receive state variable V(R) but is within the receive window, the LLC shall discard the information field of the I PDU and send a REJ PDU with the N(R) set to the value of V(R). The LLC shall then discard the information field of all I PDUs until the expected I PDU is correctly received. When receiving the expected I PDU, the LLC shall acknowledge the PDU as described in 7.5.2 above. The LLC shall use the N(R) and P bit indications in the discarded I PDUs.

On a given data link connection, only one "sent REJ" exception condition from a given LLC to another given LLC shall be established at a time. A "sent REJ" condition shall be cleared when the requested I PDU is received. The "sent REJ" condition shall be able to be reset when a reject timer time-out function runs out. When the LLC perceives by reject timer time-out that the requested I PDU will not be received, because either the requested I PDU or the REJ PDU was in error or lost, the LLC shall be able to repeat the REJ PDU in order to re-establish the "sent REJ" condition up to N2 times. The value of N2 is defined in 7.8.2.

7.5.5 Receiving Acknowledgment. When correctly receiving an I-format or S-format PDU, even in the busy condition (see 7.5.8), the receiving LLC shall consider the N(R) contained in this PDU as an acknowledgment for all the I PDUs it has sent on this data link connection with an N(S) up to and including the received N(R) minus one. The LLC shall reset the Acknowledgment Timer when it correctly receives an I-format or S-format PDU with the N(R) higher than the last received N(R) (actually acknowledging some I PDUs).

If the timer has been reset and there are outstanding I PDUs still unacknowledged on this data link connection, the LLC shall restart the Acknowledgment Timer. If the timer then runs out, the LLC shall follow the procedures in 7.5.9 with respect to the unacknowledged I PDUs.

7.5.6 Receiving a REJ PDU. When receiving a REJ PDU, the LLC shall set its send state variable V(S) to the N(R) received in the REJ PDU control field. The LLC shall (re)send the corresponding I PDU as soon as it is available. If other unacknowledged I PDUs had already been sent on that data link connection following the one indicated in the REJ PDU, then those I PDUs shall be resent by the LLC following the resending of the requested I PDU.

If retransmission beginning with a particular PDU occurs due to checkpointing (see 7.5.9) and a REJ PDU is received which would also start retransmission with the same particular I PDU (as identified by the N(R) in the REJ PDU), the retransmission resulting from the REJ PDU shall be inhibited.

7.5.7 Receiving an RNR PDU. A LLC receiving an RNR PDU shall stop sending I PDUs on the indicated data link connection at the earliest possible time, and shall start the busy-state timer, if not already running. When the busy-state timer runs out, the LLC shall follow the procedure described in 7.5.9. In any case, the LLC shall not send any other I PDUs on that data link connection before receiving an RR or REJ PDU, or before receiving an I response PDU with the F bit set to "1", or before the completion of a resetting procedure on that data link connection.

7.5.8 LLC Busy Condition. A LLC shall enter the busy condition on a data link connection when it is temporarily unable to receive or continue to receive I PDUs due to internal constraints; for example, receive buffering limitations. When the LLC enters the busy condition, it shall send a RNR PDU at the earliest opportunity. It shall be possible to send I PDUs awaiting to be sent on that data link connection prior to or following the sending of the RNR PDU. While in the busy condition, the LLC shall accept and process supervisory PDUs and return an RNR response PDU with the F bit set to "1" if it receives a supervisory or I command PDU with the P bit set to "1" on the affected data link connection.

To indicate the clearance of a busy condition on a data link connection, the LLC shall send either an I response PDU with the F bit set to "1" if a P bit set to "1" is outstanding, an REJ response PDU, or an RR response PDU on the data link connection with N(R) set to the current receive state variable V(R), depending on whether or not the LLC discarded information fields of correctly received I PDUs. Additionally, the sending of a SABME command PDU or a UA response PDU shall indicate the clearance of a busy condition at the sending LLC on a data link connection.

7.5.9 Waiting Acknowledgment. The LLC maintains an internal retransmission count variable for each data link connection which shall be set to "0" when the LLC receives or sends a UA response PDU to an SABME command PDU, or when the LLC receives an RNR frame PDU, or when the LLC correctly receives an I-format or S-format PDU with the N(R) higher than the last received N(R) (actually acknowledging some outstanding I PDUs).

If the acknowledgment timer, busy-state timer, or, optionally, P-bit timer runs out, the LLC on this data link connection shall enter the timer recovery condition and add one to its retransmission count variable.

The LLC shall then start the P-bit timer and send an S-format command PDU with the P bit set to "1".

The timer recovery condition shall be cleared on the data link connection when the LLC receives a valid I-format or S-format PDU from the remote LLC, with the F bit set to "1".

If, while in the timer recovery condition, the LLC correctly receives a valid I-format or S-format PDU with the F bit set to "1" and with the N(R) within the range from the last value of N(R) received to the current send state variable inclusive, the LLC shall clear the timer recovery condition, set its send state variable to the received N(R), stop the P-bit timer, and resend any unacknowledged PDUs.

If, while in the timer recovery condition, the LLC correctly receives a valid I-format or S-format PDU with the P/F bit set to "0" and with a N(R) within the range from the last value of N(R) received to the current send state variable inclusive, the LLC shall not clear the timer recovery condition but shall treat the N(R) value received as an acknowledgment for the indicated previously transmitted I PDUs (see 7.5.5).

If the P-bit timer runs out in the timer recovery condition, the LLC shall add one to its retransmission count variable. If the retransmission count variable is not equal to N2, the LLC shall resend an S-format PDU with the P bit set to "1" and restart its P-bit timer.

If the retransmission count variable is equal to N2, the LLC shall initiate a resetting procedure (by sending a SABME command PDU) as described in 7.6 below. N2 is a system parameter (see 7.8.2).

7.6 Procedures for Resetting. The resetting phase is used to initialize both directions of information transfer according to the procedure described below. The resetting phase shall only apply during the asynchronous balanced mode ABM.

Either LLC shall be able to initiate a resetting of both directions by sending an SABME command PDU and starting its acknowledgment timer.

After receiving an SABME command PDU, the LLC shall return, at the earliest opportunity;

(1) a UA response PDU and reset its send and receive state variables V(S) and V(R) to 0 to reset the data link connection, or

(2) a DM response PDU if the data link connection is to be terminated.

The return of the UA or DM response PDU shall take precedence over any other response PDU for the same data link connection which may be pending at the LLC. It shall be possible to follow the UA PDU with additional LLC PDUs, if pending. If the UA PDU is received correctly by the initiating LLC, it shall reset its send and receive state variables V(S) and V(R) to 0 and stop its acknowledgment timer. This shall also clear all exception conditions which might be present at either of the LLCs involved in the reset. This exchange shall also indicate clearance of any busy condition that may have been present at either LLC involved in the reset.

If a DM response PDU is received, the LLC shall enter the data link disconnected phase, shall stop its acknowledgment timer and shall report to the higher layer for appropriate action. If the acknowledgment timer runs out before a UA or DM response PDU is received, the SABME command PDU shall be resent and the acknowledgment timer shall be started. After the timer runs out N2 times, the sending LLC shall stop sending the SABME command PDU, shall report to the higher layer for the appropriate error recovery actions to initiate, and shall enter the asynchronous disconnected mode. The value of N2 is defined in 7.8.2.

Other Type 2 PDUs (with the exception of the SABME and DISC command PDUs) which are received by the LLC before completion of the reset procedure shall be discarded.

Under certain FRMR exception conditions listed in 7.7, it shall be possible for the LLC to ask the remote LLC to reset the data link connection by sending a FRMR response PDU.

Upon reception of an FRMR response PDU (even during a FRMR exception condition) the LLC shall initiate a resetting procedure by sending a SABME command PDU, or shall initiate a disconnect procedure by sending a DISC command PDU.

After sending a FRMR response PDU, the LLC shall enter the FRMR exception condition. The FRMR exception condition shall be cleared when the LLC receives or sends an SABME or DISC command PDU or DM response PDU. Any other Type 2 command PDU received while in the FRMR exception condition shall cause the LLC to resend the FRMR response PDU with the same information field as originally sent.

In the FRMR exception condition, additional I PDUs shall not be sent, and received I-format PDUs and S-format PDUs shall be discarded by the LLC.

It shall be possible for the LLC to start its acknowledgment timer on the sending of the FRMR response PDU. If the timer runs out before the reception of an SABME or DISC command PDU from the remote LLC, it shall be possible for the LLC to resend the FRMR response PDU and restart its acknowledgment timer. After the acknowledgment timer has run out N2 times, the LLC shall reset the data link connection by sending a SABME command PDU. The value of N2 is defined in 7.8.2.

When an additional FRMR response PDU is sent while the Acknowledgment Timer is running, the timer shall not be reset or restarted.

7.7 FRMR Exception Conditions. The LLC shall request a resetting procedure (by sending a FRMR response PDU) as described in 7.6, when receiving, during the information transfer phase, a PDU with one of the conditions identified in 5.4.2.3.5. The coding of the information field of the FRMR response PDU which is sent is given in 5.4.2.3.5.

The LLC shall initiate a resetting procedure (by sending a SABME command PDU) as described in 7.6 when receiving a FRMR response PDU during the information transfer phase.

7.8 List of Data Link Connection Parameters. A number of data link connection parameters are defined, the range of values for which are determined on a system-by-system basis by the user at the time that the local area network is established.

The data link connection parameters for Type 2 operation shall be as follows:

7.8.1 Timer Functions. In Type 2 operation it is possible for a number of independent events to be taking place on a data link connection that could each employ a timing function. These timing functions are defined below, as identified in the text that describes Type 2 operation. It is understood that these timing functions can be realized by using a number of individual timers, or by using a single timer. If a single timing function is employed, it will be necessary for the designer to determine on an instance-by-instance basis when to reset and restart the timer and when to let it continue running based on the priority assigned to the individual actions that are in progress.

The periods of the timer functions shall take into account whether the timers are started at the beginning or the end of the event that initiated the timer (e.g., sending of a PDU by the LLC), and any delay introduced by the MAC sublayer.

The proper operation of the procedure shall require that the value of the timing functions be greater than the maximum time between the normal network operation of Type 2 PDUs and the reception of the corresponding Type 2 PDU returned as an answer to the initiating Type 2 PDU.

7.8.1.1 Acknowledgment Timer. The acknowledgment timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive an acknowledgment to one or more outstanding I PDUs or an expected response PDU to a sent unnumbered command PDU.

7.8.1.2 P-Bit Timer. The P-bit timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive a PDU with the F bit set to "1" in response to a sent Type 2 command with the P bit set to "1".

7.8.1.3 Reject Timer. The Reject Timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive a reply to a sent REJ PDU.

7.8.1.4 Busy-State Timer. The busy-state timer is a data link connection parameter that shall define the time interval during which the LLC shall wait for an indication of the clearance of a busy condition at the other LLC.

7.8.2 Maximum Number of Transmissions, N2. N2 is a data link connection parameter that indicates the maximum number of times that a PDU is sent following the running out of the acknowledgment timer, the P-bit timer, or the reject timer.

7.8.3 Maximum Number of Octets in an I PDU, N1. N1 is a data link connection parameter that denotes the maximum number of octets in an I PDU. Refer to the various MAC descriptions to determine the precise value of N1 for a given medium access method. LLC itself places no restrictions on the value of N1. However, in the interest of having a value for N1 that all users of Type 2 LLC may depend upon, all MACs must at least be capable of accommodating I PDUs with information fields up to and including 128 octets in length.

7.8.4 Maximum Number of Outstanding I PDUs, k. The maximum number (k) of sequentially numbered I PDUs that the LLC may have outstanding (i.e., unacknowledged) at any given time shall be a data link connection parameter which can never exceed 127.

7.8.5 Minimum Number of Octets in a PDU. A minimal length valid data link connection PDU shall contain exactly two address fields and one control field in that order. Thus the minimum number of octets in a valid data link connection PDU shall be 3 or 4, depending on whether the PDU is a U-format PDU, or an I-format or S-format PDU, respectively.

7.9 Precise Description of the Type 2 Procedures

7.9.1 General. If discrepancies appear to exist with the text found in the balance of Section 7, this subsection (7.9) shall be viewed as being the definitive description.

7.9.2 Connection Service Component Overview. The connection service component handles all LLC Type 2 PDU traffic for a specific data link connection (designated by a DA, DSAP — SA, SSAP pair). Once activated, the connection service component shall process Type 2 LLC PDUs addressed to the local service access point from the remote service access point and shall send Type 2 PDUs to the remote service access point as a result of either a service access point user request or as the result of some data link protocol action. (See Fig 7-1 and Table 7-1.)

When the service access point component (as described in 6.9) is placed in the ACTIVE state, all the connection service components associated with the service access point are placed in the ADM (asynchronous disconnected mode) state. When the service access point component leaves the ACTIVE state, all of the associated connection service components are deactivated, regardless of the current state of the connection service component.

The following points apply to the interpretation of the state tables:

(1) Flag variables are used to limit the number of states by maintaining the state of particular conditions affecting the connection component. The flags defined are
P_FLAG,
F_FLAG,
S_FLAG,
DATA_FLAG, and
REMOTE_BUSY.

(2) In the list of events, events of the form RECEIVE_XXX_YYY are listed. The interpretation is that this event is the reception of any command PDU or response PDU not specifically listed for that state.

(3) For some combinations of state and event(s) the table provides alternative actions. These are separated by horizontal dotted lines in the ACTIONS and NEXT STATE columns. Selection of an alternative is done on the basis of (a) local status, (b) the result of layer management action, or (c) implementation decision. There is no relationship between the order of alternatives between events, nor is it implied that the same alternative must be selected every time the event occurs.

(4) In the list of actions there is no implied ordering, unless one or more of the actions is conditional upon the value(s) of flag(s) which are modified by other actions. In this case the test(s) must be completed before the flag(s) are modified.

(5) In the list of actions, actions of the form SEND_XXX_RSP (F=l) are indicated. It should be noted that if some other response PDU (with the F bit set to "0") will be sent earlier, it is permissible to modify that PDU from F bit set to "0" to F bit set to "1", and to send the new PDU with the F bit set to "0". This could occur, for example, if an LLC implementation managed the queue of PDUs awaiting transmission.

(6) For simplicity, the state table has four timers: the ACK_TIMER for timing acknowledgments, the P_TIMER for timing the P/F cycle, the REJ_TIMER for timing the "sent REJ" condition, and the BUSY_TIMER for timing the "remote busy" condition. It should be noted that by the addition of appropriate flags a functionally equivalent state table can be developed that requires only one timer.

(7) Any START_TIMER action (re)starts the specified timer from zero, even if the timer is already running. When the time reaches its limit, the appropriate TIMER_EXPIRED condition is set and the timer stopped. The TIMER_EXPIRED condition is cleared when it is recognized by the connection component state machine. The STOP_TIMER action stops the timer if it is running or clears the TIMER_EXPIRED condition if the timer has already reached its limit.

(8) Events not recognized in a particular state are assumed to remain pending until any masking flag is modified or a transition is made to a state where they can be recognized.

NOTE: To ensure proper interpretation of the state table, the descriptions of the entries (see 7.9.2.1–7.9.2.3) should be read in concert with the state tables.

7.9.2.1 Connection Component State Descriptions

(1) **ADM.** The connection component is in the asynchronous disconnected mode. It can accept an SABME PDU from a remote LLC SSAP or, at the request of the service access point user, can initiate an SABME PDU transmission to a remote LLC DSAP, to establish a data link connection. It also responds to a DISC command PDU and to any command PDU with the P bit set to "1".

(2) **SETUP.** The connection component has transmitted an SABME command PDU to a remote LLC DSAP and is waiting for a reply.

(3) **NORMAL.** A data link connection exists between the local LLC service access point and the remote LLC service access point. Sending and reception of information and supervisory PDUs can be performed.

(4) **BUSY.** A data link connection exists between the local LLC service access point and the remote LLC service access point. I PDUs may be sent. Local conditions make it likely that the information field of received I PDUs will be ignored. Supervisory PDUs may be both sent and received.

(5) **REJECT.** A data link connection exists between the local LLC service access point and the remote LLC service access point. The local connection component has requested that the remote connection component resend a specific I PDU that the local connection component has detected as being out of sequence. Both I PDUs and supervisory PDUs may be sent and received.

(6) **AWAIT.** A data link connection exists between the local LLC service access point and the remote LLC service access point. The local LLC is performing a timer recovery operation and has sent a command PDU with the P bit set to "1", and is awaiting an acknowledgment from the remote LLC. I PDUs may be received but not sent. Supervisory PDUs may be both sent and received.

(7) **AWAIT_BUSY.** A data link connection exists between the local LLC service access point and the remote LLC service access point. The local LLC is performing a timer recovery operation and has sent a command PDU with the P bit set to "1," and is awaiting an acknowledgment from the remote LLC. I PDUs may not be sent. Local conditions make it likely that the information field of received I PDUs will be ignored. Supervisory PDUs may be both sent and received.

(8) **AWAIT_REJECT.** A data link connection exists between the local LLC service access point and the remote LLC service access point. The local connection component has requested that the remote connection component re-transmit a specific I PDU that the local connection component has detected as being out of sequence. Before the local LLC entered this state it was performing a timer recovery operation and had sent a command PDU with the P bit set to "1", and is still awaiting an acknowledgment from the remote LLC. I PDUs may be received but not transmitted. Supervisory PDUs may be both transmitted and received.

(9) **D_CONN.** At the request of the service access point user, the local LLC has sent a DISC command PDU to the remote LLC DSAP and is waiting for a reply.

(10) **RESET.** As a result of a service access point user request or the receipt of a FRMR response PDU, the local connection component has sent an SABME command PDU to the remote LLC DSAP to reset the data link connection and is waiting for a reply.

(11) **ERROR.** The local connection component has detected an error in a received PDU and has sent a FRMR response PDU. It is waiting for a reply from the remote connection component.

(12) **CONN.** The local connection component has received an SABME PDU from a remote LLC SSAP, and it is waiting for the local user to accept or refuse the connection.

(13) **RESET_CHECK.** The local connection component is waiting for the local user to accept or refuse a remote reset request.

(14) **RESET_WAIT.** The local connection component is waiting for the local user to indicate a RESET_REQUEST or a DISCONNECT_REQUEST.

7.9.2.2 Connection Service Component Event Description. In the list of events below, the value of the P or F bits in received commands and responses is listed as X. In the state transition table, values of 0, 1, or X are used. The latter indicates that either 0 or 1 may occur in the event.

(1) **CONNECT_REQUEST.** The user has requested that a data link connection be established with a remote LLC DSAP.

(2) **CONNECT_RESPONSE.** The user has accepted the data link connection.

(3) **DATA_REQUEST.** The user has requested that a data unit be sent to the remote LLC DSAP.

(4) **DISCONNECT_REQUEST.** The user has requested that the data link connection with the remote LLC DSAP be terminated.

(5) **RESET_REQUEST.** The user has requested that the data link connection with the remote LLC DSAP be reset.

(6) **RESET_RESPONSE.** The user has accepted the reset of the data link connection.

(7) **LOCAL_BUSY_DETECTED.** The local station has entered a busy condition and may not be able to accept I PDUs from the remote LLC SSAP.

(8) **LOCAL_BUSY_CLEARED.** The local station busy condition has ended and the station can accept I PDUs from the remote LLC SSAP.

(9) **RECEIVE_BAD_PDU.** The remote SSAP has sent to the local DSAP a command or response PDU which is not implemented, or has an information field when not permitted, or is an I PDU with an information field length greater than can be accommodated by the local LLC.

(10) **RECEIVE_DISC_CMD(P=X).** The remote SSAP has sent a DISC command PDU with the P bit set to "X" addressed to the local DSAP.

(11) **RECEIVE_DM_RSP(F=X).** The remote SSAP has sent a DM response PDU with the F bit set to "X" addressed to the local DSAP.

(12) **RECEIVE_FRMR_RSP(F=X).** The remote SSAP has sent a FRMR response PDU with the F bit set to "X" addressed to the local DSAP.
(13) **RECEIVE_I_CMD(P=X).** The remote SSAP has sent a I command PDU with the P bit set to "X" addressed to the local DSAP. Both the N(R) and N(S) fields are valid and the N(S) value is the expected sequence number.
(14) **RECEIVE_I_CMD(P=X)_WITH_UNEXPECTED_N(S).** The remote SSAP has sent an I command PDU with the P bit set to "X" addressed to the local DSAP. The N(S) field of the command does not contain the expected sequence number but is within the window size. The N(R) field is valid.
(15) **RECEIVE_I_CMD(P=X)_WITH_INVALID_N(S).** The remote SSAP has sent an I command PDU with the P bit set to "X" addressed to the local DSAP. The N(S) field of the command is invalid. The N(R) field is valid.
(16) **RECEIVE_I_RSP(F=X).** The remote SSAP has sent a I response PDU with the F bit set to "X" addressed to the local DSAP. Both the N(R) and N(S) fields are valid and the N(S) value is the expected sequence number.
(17) **RECEIVE_I_RSP(F=X)_WITH_UNEXPECTED_N(S).** The remote SSAP has sent an I response PDU with the F bit set to "X" addressed to the local DSAP. The N(S) field of the command does not contain the expected sequence number but is within the window size.
(18) **RECEIVE_I_RSP(F=X)_WITH_INVALID_N(S).** The remote SSAP has sent an I response PDU with the F bit set to "X" addressed to the local DSAP. The N(S) field of the response is invalid. The N(R) field is valid.
(19) **RECEIVE_REJ_CMD(P=X).** The remote SSAP has sent a REJ command PDU with the P bit set to "X" addressed to the local DSAP.
(20) **RECEIVE_REJ_RSP(F=X).** The remote SSAP has sent a REJ response PDU with the F bit set to "X" addressed to the local DSAP.
(21) **RECEIVE_RNR_CMD(P=X).** The remote SSAP has sent a RNR command PDU with the P bit set to "X" addressed to the local DSAP.
(22) **RECEIVE_RNR_RSP(F=X).** The remote SSAP has sent a RNR response PDU with the F bit set to "X" addressed to the local DSAP.
(23) **RECEIVE_RR_CMD(P=X).** The remote SSAP has sent a RR command PDU with the P bit set to "X" addressed to the local DSAP.
(24) **RECEIVE_RR_RSP(F=X).** The remote SSAP has sent a RR response PDU with the F bit set to "X" addressed to the local DSAP.
(25) **RECEIVE_SABME_CMD(P=X).** The remote SSAP has sent an SABME command PDU with the P bit set to "X" addressed to the local DSAP.
(26) **RECEIVE_UA_RSP(F=X).** The remote SSAP has sent a UA response PDU with the F bit set to "X" addressed to the local DSAP.
(27) **RECEIVE_XXX_CMD(F=X).** The remote SSAP has sent a Type 2 command PDU with the P bit set to "X" addressed to the local DSAP. The command is any command not specifically listed for that state.

(28) **RECEIVE_XXX_RSP(F=X).** The remote SSAP has sent a Type 2 response PDU with the F bit set to "X" addressed to the local DSAP. The response is any response not specifically listed for that state.
(29) **RECEIVE_XXX_YYY.** The remote SSAP has sent a Type 2 PDU addressed to the local DSAP. The PDU is any command or response not specifically listed for that state.
(30) **RECEIVE_ZZZ_CMD(P=X)_WITH_INVALID_N(R).** The remote SSAP has sent an I, RR, RNR, or REJ command PDU with the P bit set to "X" addressed to the local DSAP. The N(R) field of the command is invalid.
(31) **RECEIVE_ZZZ_RSP(F=X)_WITH_INVALID_N(R).** The remote SSAP has sent an I, RR, RNR, or REJ response PDU with the F bit set to "X" addressed to the local DSAP. The N(R) field of the response is invalid.
(32) **P_TIMER_EXPIRED.** The P/F cycle timer has expired.
(33) **ACK_TIMER_EXPIRED.** The acknowledgment timer has expired.
(34) **REJ_TIMER_EXPIRED.** The "sent REJ" timer has expired.
(35) **BUSY_TIMER_EXPIRED.** The remote-busy timer has expired.

In the state transition table some of the above events are qualified by the following conditions. The event is recognized only when the condition is true.

(36) **DATA_FLAG=1.** When DATA_FLAG has a value of one, data unit(s) from I PDUs were discarded during a local busy period.
(37) **DATA_FLAG=0.** When DATA_FLAG has a value of zero, data unit(s) from I PDUs were not discarded during a local busy period.
(38) **DATA_FLAG=2.** When DATA_FLAG has a value of two, the BUSY state was entered from the REJECT state, and the requested I PDU has not yet been received.
(39) **P_FLAG=1.** P_FLAG has a value of one when a command with the P bit set to "1" has been sent and a response with the F bit set to "1" is expected.
(40) **P_FLAG=0.** P_FLAG has a value of zero when a response PDU with the F bit set to "1" is not expected.
(41) **P_FLAG=F.** P_FLAG has a value equal to the F bit in the response PDU received.
(42) **REMOTE_BUSY=1.** When REMOTE_BUSY has a value of one, an RNR PDU has been received from the remote connection component to indicate that I PDUs should not be sent. DATA_REQUEST events are not recognized until this flag is reset to zero.
(43) **REMOTE_BUSY=0.** When REMOTE_BUSY has a value of zero, sending of I PDUs is possible.
(44) **RETRY_COUNT<N2.** The number of retries is less than the maximum number of retries.
(45) **RETRY_COUNT>=N2.** The number of retries has reached the maximum number permissible.
(46) **S_FLAG=1.** In the SETUP, RESET, and RESET_WAIT states, an S_FLAG value of one indicates that an SABME PDU has been received.
(47) **S_FLAG=0.** In the SETUP, RESET, and RESET_WAIT states, an S_FLAG value of zero indicates that an SABME PDU has not been received.

(48) **INITIATE_P/F_CYCLE.** The local LLC wants to initiate a P/F cycle. (This is only required if the local LLC is not generating other command PDUs for some reason.)

7.9.2.3 Connection Component Action Description. In the list of actions described below the value of the P or F bits in the transmitted commands and responses is listed as X. In the state transition table, values of 0, 1, or X are used. The latter indicates that either 0 or 1 may be used.

(1) **CLEAR_REMOTE_BUSY.** If REMOTE_BUSY has a value of one, then set REMOTE_BUSY to zero to indicate the remote LLC is now able to accept I PDUs, stop the BUSY_TIMER, inform the user by issuing REPORT_STATUS (REMOTE_NOT_BUSY) and, provided the local LLC is in the NORMAL, REJECT, or BUSY state, start the (re)sending of any I PDUs that were waiting for the remote busy to be cleared.

(2) **CONNECT_INDICATION.** Inform the user that a connection has been requested by a remote LLC SSAP.

(3) **CONNECT_CONFIRM.** The connection service component indicates that the remote network entity has accepted the connection.

(4) **DATA_INDICATION.** The connection service component passes the data unit from the received I PDU to the user.

(5) **DISCONNECT_INDICATION.** Inform the user that the remote network entity has initiated disconnection of the data link connection.

(6) **RESET_INDICATION.** Inform the user that either the remote network entity or the remote LLC component has initiated a reset of the data link connection, or that the local LLC has determined that the data link connection is in need of reinitialization. The valid results are

REMOTE. The remote network entity or remote peer has initiated a reset of the data link connection.

LOCAL. The local LLC has determined that the data link connection is in need of reinitialization.

(7) **RESET_CONFIRM.** The connection service component indicates that the remote network entity has accepted the reset.

(8) **REPORT_STATUS.** Report the status of the data link connection to the sublayer management function. Permissible status values are

FRMR_RECEIVED. The local connection service component has received a FRMR response PDU.

FRMR_SENT. The local connection service component has received an invalid PDU, and has sent a FRMR response PDU.

REMOTE_BUSY. The remote LLC DSAP is busy. The local connection service component will not accept a DATA_REQUEST.

REMOTE_NOT_BUSY. The remote LLC DSAP is no longer busy. The local connection service component will now accept a DATA_REQUEST.

(9) **IF_F=1_CLEAR_REMOTE_BUSY.** If the I PDU is a response with the F bit set to "1" in response to a command PDU with the P bit set to "1", then perform the CLEAR_REMOTE_BUSY action.

(10) **IF_DATA_FLAG=2_STOP_REJ_TIMER.** If DATA_FLAG has a value

of two, indicating that a REJ PDU has been sent, stop the "sent REJ" timer.

(11) **SEND_DISC_CMD(P=X).** Transmit a DISC command PDU with the P bit set to "X" to the remote LLC DSAP.

(12) **SEND_DM_RSP(F=X).** Send a DM response PDU with the F bit set to "X" to the remote LLC DSAP.

(13) **SEND_FRMR_RSP(F=X).** Send a FRMR response PDU with the F bit set to "X" to the remote LLC DSAP.

(14) **RE-SEND_FRMR_RSP(F=0).** Send the same FRMR response PDU with the same information field as sent earlier to the remote LLC DSAP. Set the F bit to "0".

(15) **RE-SEND_FRMR_RSP(F=P).** Send the same FRMR response PDU with the same information field as sent earlier to the remote LLC DSAP. Set the F bit equal to the P bit of the received command PDU.

(16) **SEND_I_CMD(P=1).** Send an I command PDU with the P bit set to "1" to the remote LLC DSAP with the data unit supplied by the user with the DATA_REQUEST. Before transmission, copy the current values of the send state variable V(S) and the receive state variable V(R) into the N(S) and N(R) fields, respectively, of the I PDU and increment (modulo 128) the send state variableV(S).

(17) **RE-SEND_I_CMD(P=1).** Start resending all the unacknowledged I PDUs for this data link connection beginning with the N(R) given in the received PDU. Send the first as a command with the P bit set to "1". If the queue contains more than one I PDU, the balance must be sent as commands with the P bit set to "0", or as responses with the F bit set to "0".

(18) **RE-SEND_I_CMD(P=1)_OR_SEND_RR.** Start resending all the unacknowledged I PDUs for this data link connection beginning with the N(R) given in the received PDU. Send the first as a command with the P bit set to "1". If the queue contains more than one I PDU the balance must be sent as commands with the P bit set to "0" or as responses with the F bit set to "0". It is permissible to send a RR command PDU with the P bit set to "1" to the remote LLC DSAP before starting the resending of the I PDUs. In this case, the first I PDU is sent as a command with the P bit set to "0", or as a response with the F bit set to "0". If no I PDU is ready to send, a RR command PDU with the P bit set to "1" must be sent to the remote LLC DSAP.

(19) **SEND_I_XXX(X=0).** Send either an I response PDU with the F bit set to "0" or an I command PDU with the P bit set to "0" to the remote LLC DSAP with the data unit supplied by the user with the DATA_REQUEST. Before transmission, copy the current values of the send state variable V(S) and the receive state variable V(R) into the N(S) and N(R) fields, respectively, of the I PDU and increment (modulo 128) the send state variable V(S).

(20) **RE-SEND_I_XXX(X=0).** Start resending all the unacknowledged I PDUs for this data link connection beginning with the N(R) given in the

received PDU. They must be sent as either commands with the P bit set to "0" or as responses with the F bit set to "0".

(21) **RE-SEND_I_XXX(X=0)_OR_SEND_RR.** Start resending all the unacknowledged I PDUs for this data link connection beginning with the N(R) given in the received PDU. They must be sent as either commands with the P bit set to "0" or as responses with the F bit set to "0". It is permissible to send either a RR response PDU with the F bit set to "0" or a RR command PDU with the P bit set to "0" to the remote LLC DSAP before starting the resending of the I PDUs. If no I PDU is ready to send, either an RR response PDU with the F bit set to "0" or an RR command PDU with the P bit set to "0" must be sent to the remote LLC DSAP.

(22) **RE-SEND_I_RSP(F=1).** Start resending all the unacknowledged I PDUs for this data link connection beginning with the N(R) given in the received PDU. Send the first as a response with the F bit set to "1". If the queue contains more than one I PDU, the balance must be transmitted as commands with the P bit set to "0" or as responses with the F bit set to "0".

(23) **SEND_REJ_CMD(P=1).** Send a REJ command PDU with the P bit set to "1" to the remote LLC DSAP.

(24) **SEND_REJ_RSP(F=1).** Send a REJ response PDU with the F bit set to "1" to the remote LLC DSAP.

(25) **SEND_REJ_XXX(X=0).** Send either a REJ response PDU with the F bit set to "0" or a REJ command PDU with the P bit set to "0" to the remote LLC DSAP.

(26) **SEND_RNR_CMD(P=1).** Send a RNR command PDU with the P bit set to "1" to the remote LLC DSAP.

(27) **SEND_RNR_RSP(F=1).** Send a RNR response PDU with the F bit set to "1" to the remote LLC DSAP.

(28) **SEND_RNR_XXX(X=0).** Send either a RNR response PDU with the F bit set to "0" or a RNR command PDU with the P bit set to "0" to the remote LLC DSAP.

(29) **SET_REMOTE_BUSY.** If REMOTE_BUSY is zero, then set REMOTE_BUSY to one to indicate the remote LLC is in the busy state and is not able to accept I PDUs, start the BUSY_TIMER, inform the sublayer management function by using REPORT_STATUS (REMOTE_BUSY) and stop any (re)sending of I PDU that is in progress.

(30) **OPTIONAL_SEND_RNR_XXX(X=0).** It is permissible to send a RNR command PDU with the P bit set to "0" or a RNR response PDU with the F bit set to "0" to the remote LLC DSAP in case the remote LLC did not receive the first RNR sent when the busy state was entered.

(31) **SEND_RR_CMD(P=1).** Send a RR command PDU with the P bit set to "1" to the remote LLC DSAP.

(32) **SEND_ACKNOWLEDGE_CMD(P=1).** Under all conditions it is permissible to send a RR command PDU with the P bit set to "1" to the remote LLC DSAP. If no I PDU is ready to send, the RR command PDU with the P bit set to "1" must be sent to the remote LLC DSAP. (This RR PDU may be delayed by a time bounded by the ACK_TIMER value, to wait for the

generation of an I PDU.) However, if an I PDU is ready to send, and can be modified to a command with the P bit set to "1", then the RR command PDU does not need to be sent.

(33) **SEND_RR_RSP(F=1).** Send a RR response PDU with the F bit set to "1" to the remote LLC DSAP.

(34) **SEND_ACKNOWLEDGE_RSP(F=1).** Under all conditions it is permissible to send a RR response PDU with the F bit set to "1" to the remote LLC DSAP. If no I PDU is ready to send, the RR response PDU with the F bit set to "1" must be sent to the remote LLC DSAP. However, if an I PDU is ready to send, and can be modified to a response with the F bit set to "1," then the RR response PDU does not need to be sent.

(35) **SEND_RR_XXX(X=0).** Send either a RR response PDU with the F bit set to "0" or a RR command PDU with the P bit set to "0" to the remote LLC DSAP.

(36) **SEND_ACKNOWLEDGE_XXX(X=0).** Under all conditions it is permissible to send either a RR response PDU with the F bit set to "0" or a RR command PDU with the P bit set to "0" to the remote LLC DSAP. If no I PDU is ready to send, either an RR response with the F bit set to "0" or an RR command PDU with the P bit set to "0" must be sent to the remote LLC DSAP. (This RR PDU may be delayed, by a time bounded by the ACK_TIMER value, to wait for the generation of an I PDU.) However, if an I PDU is ready to send, then the RR PDU does not need to be sent.

(37) **SEND_SABME_CMD(P=X).** Send a SABME command PDU with the P bit set to "X" to the remote LLC DSAP.

(38) **SEND_UA_RSP(F=X).** Send a UA response PDU with the F bit set to "X" to the remote LLC DSAP.

(39) **S_FLAG:=0.** Set S_FLAG to zero to indicate that a SABME PDU has not been received from the remote LLC while the local connection service component is in the RESET, SETUP, or RESET_WAIT state.

(40) **S_FLAG:=1.** Set S_FLAG to one to indicate that a SABME PDU has been received from the remote LLC while the local connection service component is in the RESET, SETUP, or RESET_WAIT state.

(41) **START_P_TIMER.** Start the P/F cycle timer from zero; if the P_FLAG is zero, initialize RETRY_COUNT to zero, and set P_FLAG to one.

(42) **START_ACK_TIMER.** Start the acknowledgment timer from zero.

(43) **START_REJ_TIMER.** Start the "sent REJ" timer from zero.

(44) **START_ACK_TIMER_IF_NOT_RUNNING.** If the acknowledgment timer is not currently running, then start the acknowledgment timer from zero.

(45) **STOP_ACK_TIMER.** Stop the acknowledgment timer.

(46) **STOP_P_TIMER.** Stop the P/F cycle timer and set P_FLAG to zero.

(47) **STOP_REJ_TIMER.** Stop the "sent REJ" timer.

(48) **STOP_ALL_TIMERS.** Stop the P/F cycle timer, the "sent REJ" timer, the remote-busy timer, and the acknowledgment timer.

(49) **STOP_OTHER_TIMERS.** Stop the P/F cycle timer, the "sent REJ" timer, and the remote-busy timer.

(50) **UPDATE_N(R)_RECEIVED.** If the N(R) of the received PDU acknowledges the receipt of one or more previously unacknowledged I PDUs, update the local record of N(R)_RECEIVED, set RETRY_COUNT to zero, and stop the acknowledgment timer. If unacknowledged I PDUs still exist, start the acknowledgment timer if it was stopped.

NOTE: If some form of SEND_I_PDU is initiated at the same time as UPDATE_N(R)_RECEIVED, then the acknowledgment timer is always started if it was stopped.

(51) **UPDATE_P_FLAG.** If the received PDU was a response with the F bit set to "1", set the P_FLAG to zero and stop the P/F cycle timer.

(52) **DATA_FLAG:=2.** Set the DATA_FLAG to two to record that the BUSY state was entered with a REJ PDU outstanding.

(53) **DATA_FLAG:=0.** Set the DATA_FLAG to zero to indicate that the data units from received I PDUs were not discarded during a local busy period.

(54) **DATA_FLAG:=1.** Set the DATA_FLAG to one to indicate that the data units from received I PDUs were discarded during a local busy period.

(55) **IF_DATA_FLAG=0_THEN DATA_FLAG:=1.** If the DATA_FLAG had been zero, indicating that no data units had been discarded, set it to one to indicate that data units have now been discarded.

(56) **P_FLAG:=0.** Initialize the P_FLAG to zero. This indicates that the reception of a response PDU with the F bit set to "1" is not expected.

(57) **P_FLAG:=P.** Set the P_FLAG to the value of the P bit in the command PDU being sent.

(58) **REMOTE_BUSY:=0.** Set REMOTE_BUSY to zero to indicate that the remote LLC is able to accept I PDUs.

(59) **RETRY_COUNT:=0.** Initialize RETRY_COUNT to zero.

(60) **RETRY_COUNT:=RETRY_COUNT+1.** Increment RETRY_COUNT by one.

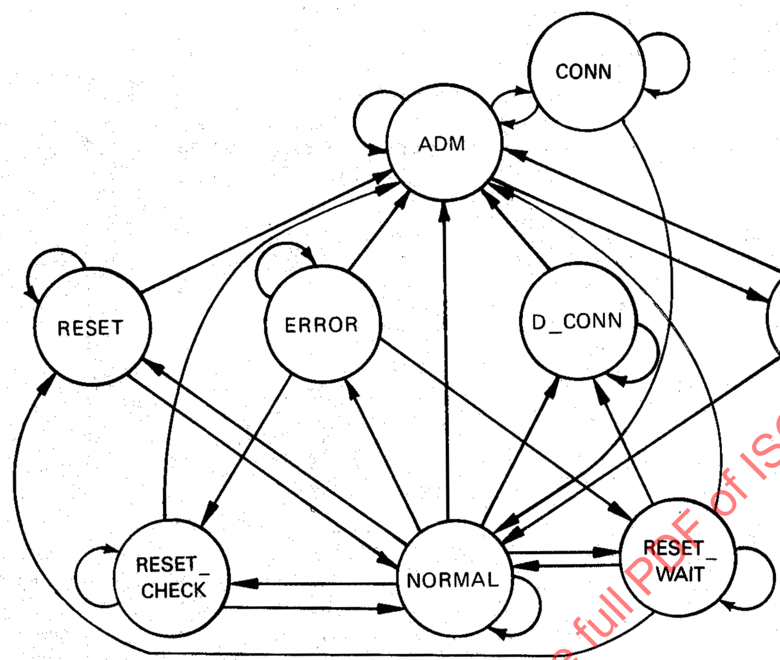
(61) **V(R):=0.** Initialize the receive state variable. This is the expected sequence number of the next I PDU received.

(62) **V(R):=V(R)+1.** Increment (modulo 128) the receive state variable. This is the expected sequence number of the next I PDU received.

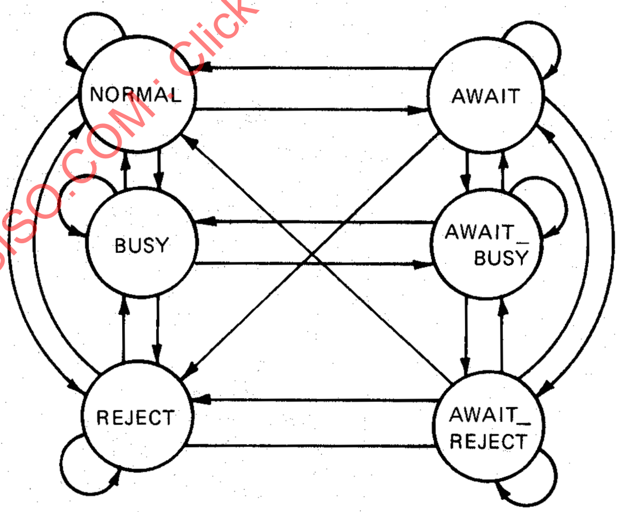
(63) **V(S):=0.** Initialize the send state variable. This is the sequence number of the next I PDU to be sent.

(64) **V(S):=N(R).** Reset the send state variable to the value specified by the N(R) field of the PDU just received.

(65) **F_FLAG:=P.** Set the F_FLAG to the value of the P bit received. This is the value of the F bit to be sent in UA or DM PDUs.



(a) Data Link Establishment, Disconnection, and Resetting States



(b) Information Transfer (Connected) States

Fig 7-1
Connection Component State Diagram

Table 7-1
Connection Component State Transitions

Current State	Event	Action(s)	Next State
ADM	CONNECT_REQUEST	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=0 S_FLAG:=0	SETUP
		.... DISCONNECT_INDICATION	ADM
	RECEIVE_SABME_CMD(P=X)	CONNECT_INDICATION P_FLAG:=P	CONN
	RECEIVE_DISC_CMD(P=X)	SEND_DM_RSP(F=P)	ADM
	RECEIVE_XXX_CMD(P=1)	SEND_DM_RSP(F=1)	ADM
	RECEIVE_XXX_CMD(P=0) or RECEIVE_XXX_RSP(F=X)		ADM
CONN	CONNECT_RESPONSE	SEND_UA_RSP(F=F_FLAG) V(S):=0 V(R):=0 RETRY_COUNT:=0 P_FLAG:=0 REMOTE_BUSY:=0	NORMAL
	DISCONNECT_REQUEST	SEND_DM_RSP(F=F_FLAG)	ADM
	RECEIVE_SABME_CMD(P=X)	F_FLAG:=P	CONN
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATION	ADM
	RECEIVE_XXX_YYY		CONN
RESET_ WAIT	RESET_REQUEST and S_FLAG=0	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=0	RESET
	RESET_REQUEST and S_FLAG=1	SEND_UA_RSP(F=F_FLAG) V(S):=0 V(R):=0 RETRY_COUNT:=0 P_FLAG:=0 REMOTE_BUSY:=0 RESET_CONFIRM	NORMAL

Current State	Event	Action(s)	Next State
RESET_ WAIT	DISCONNECT_REQUEST and S_FLAG=0	SEND_DISC_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=0	D_CONN
	DISCONNECT_REQUEST and S_FLAG=1	SEND_DM_RSP(F=F_FLAG)	ADM
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATION	ADM
	RECEIVE_SABME_CMD(P=X)	S_FLAG:=1 F_FLAG:=P	RESET_WAIT
	RECEIVE_DISC_CMD(P=X)	SEND_DM_RSP(F=P) DISCONNECT_INDICATION	ADM
	RECEIVE_XXX_YYY		RESET_WAIT
RESET_ CHECK	RESET_RESPONSE	SEND_UA_RSP(F=F_FLAG) V(S):=0 V(R):=0 RETRY_COUNT:=0 P_FLAG:=0 REMOTE_BUSY:=0	NORMAL
	DISCONNECT_REQUEST	SEND_DM_RSP(F=F_FLAG)	ADM
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATION	ADM
	RECEIVE_SABME_CMD(P=X)	F_FLAG:=P	RESET_ CHECK
	RECEIVE_DISC_CMD(P=X)	SEND_DM_RSP(F=P) DISCONNECT_INDICATION	ADM
	RECEIVE_XXX_YYY		RESET_ CHECK
SETUP	RECEIVE_SABME_CMD(P=X)	V(S):=0 V(R):=0 RETRY_COUNT:=0 SEND_UA_RSP(F=P) S_FLAG:=1	SETUP

Current State	Event	Action(s)	Next State
SETUP	RECEIVE_UA_RSP(F=X) and P_FLAG=F	STOP_ACK_TIMER V(S):=0 V(R):=0 RETRY_COUNT:=0 UPDATE_P_FLAG CONNECT_CONFIRM REMOTE_BUSY:=0	NORMAL
	ACK_TIMER_EXPIRED and S_FLAG=1	P_FLAG:=0 CONNECT_CONFIRM REMOTE_BUSY:=0	NORMAL
	RECEIVE_DISC_CMD(P=X)	SEND_DM_RSP(F=P) DISCONNECT_INDICATION STOP_ACK_TIMER	ADM
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATION STOP_ACK_TIMER	ADM
	RECEIVE_XXX_YYY		SETUP
	ACK_TIMER_EXPIRED and RETRY_COUNT<N2 and S_FLAG=0	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1	SETUP
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 and S_FLAG=0	DISCONNECT_INDICATION	ADM
RESET	RECEIVE_SABME_CMD(P=X)	V(S):=0 V(R):=0 RETRY_COUNT:=0 S_FLAG:=1 SEND_UA_RSP(F=P)	RESET
	RECEIVE_UA_RSP(F=X) and P_FLAG=F	STOP_ACK_TIMER V(S):=0 V(R):=0 RETRY_COUNT:=0 UPDATE_P_FLAG RESET_CONFIRM REMOTE_BUSY:=0	NORMAL
	ACK_TIMER_EXPIRED and S_FLAG=1	P_FLAG:=0 RESET_CONFIRM REMOTE_BUSY:=0	NORMAL

Current State	Event	Action(s)	Next State
RESET	RECEIVE_DISC_CMD(P=X)	SEND_DM_RSP(F=P) DISCONNECT_INDICATION STOP_ACK_TIMER	ADM
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATION STOP_ACK_TIMER	ADM
	RECEIVE_XXX_YYY		RESET
	ACK_TIMER_EXPIRED and RETRY_COUNT<N2 and S_FLAG=0	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1	RESET
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 and S_FLAG=0	DISCONNECT_INDICATION	ADM
D_CONN	RECEIVE_SABME_CMD(P=X)	SEND_DM_RSP(F=P) STOP_ACK_TIMER	ADM
	RECEIVE_UA_RSP(F=X) and P_FLAG=F	STOP_ACK_TIMER	ADM
	RECEIVE_DISC_CMD(P=X)	SEND_UA_RSP(F=P)	D_CONN
	RECEIVE_DM_RSP(F=X)	STOP_ACK_TIMER	ADM
	RECEIVE_XXX_YYY		D_CONN
	ACK_TIMER_EXPIRED and RETRY_COUNT<N2	SEND_DISC_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1	D_CONN
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2		ADM
ERROR	RECEIVE_SABME_CMD(P=X)	RESET_INDICATION(REMOTE) STOP_ACK_TIMER	RESET_ CHECK
	RECEIVE_DISC_CMD(P=X)	SEND_UA_RSP(F=P) DISCONNECT_INDICATION STOP_ACK_TIMER	ADM
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATION STOP_ACK_TIMER	ADM

Current State	Event	Action(s)	Next State
ERROR	RECEIVE_FRMR_RSP(F=X)	RESET_INDICATION(LOCAL) STOP_ACK_TIMER REPORT_STATUS(FRMR_RECEIVED) S_FLAG:=0	RESET_WAIT
	RECEIVE_XXX_CMD(P=X)	RE-SEND_FRMR_RSP(F=P) START_ACK_TIMER	ERROR
	RECEUVE_XXX_RSP(F=X)		ERROR
	ACK_TIMER_EXPIRED and RETRY_COUNT<N2	RE-SEND_FRMR_RSP(F=0) START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1	ERROR
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2	S_FLAG:=0 RESET_INDICATION(LOCAL)	RESET_WAIT
NORMAL or BUSY or REJECT or AWAIT or AWAIT_BUSY or AWAIT_REJECT	DISCONNECT_REQUEST	SEND_DISC_CMD(P=X) P_FLAG:=P START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0	D_CONN
	RESET_REQUEST	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0 S_FLAG:=0	RESET
	RECEIVE_SABME_CMD(P=X)	RESET_INDICATION(REMOTE) F_FLAG:=P STOP_ALL_TIMERS	RESET_CHECK
	RECEIVE_DISC_CMD(P=X)	SEND_UA_RSP(F=P) DISCONNECT_INDICATION STOP_ALL_TIMERS	ADM
	RECEIVE_FRMR_RSP(F=X)	STOP_ALL_TIMERS RESET_INDICATION(LOCAL) REPORT_STATUS(FRMR_RECEIVED) S_FLAG:=0	RESET_WAIT
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATION STOP_ALL_TIMERS	ADM

Current State	Event	Action(s)	Next State
NORMAL or BUSY or REJECT or AWAIT or AWAIT_ BUSY or AWAIT_ REJECT	RECEIVE_ZZZ_CMD(P=X)_ WITH_INVALID_N(R) or RECEIVE_I_CMD(P=X)_ WITH_INVALID_N(S)	SEND_FRMR_RSP(F=P) REPORT_STATUS(FRMR_SENT) START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0	ERROR
	RECEIVE_ZZZ_RSP(F=X)_ WITH_INVALID_N(R) or RECEIVE_I_RSP(F=X)_ WITH_INVALID_N(S) or RECEIVE_UA_RSP(F=X) or RECEIVE_XXX_RSP(F=1) and P_FLAG=0 or RECEIVE_BAD_PDU	SEND_FRMR_RSP(F=0) REPORT_STATUS(FRMR_SENT) START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0	ERROR
	P_TIMER_EXPIRED and RETRY_COUNT>=N2 or ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 or REJ_TIMER_EXPIRED and RETRY_COUNT>=N2 or BUSY_TIMER_EXPIRED and RETRY_COUNT>=N2	STOP_ALL_TIMERS RESET_INDICATION(LOCAL) S_FLAG:=0	RESET_WAIT
NORMAL	DATA_REQUEST and REMOTE_BUSY=0 and P_FLAG=0	SEND_I_CMD(P=1) START_P_TIMER START_ACK_TIMER_ IF_NOT_RUNNING .... SEND_I_XXX(X=0) START_ACK_TIMER_ IF_NOT_RUNNING	NORMAL .... NORMAL
	DATA_REQUEST and REMOTE_BUSY=0 and P_FLAG=1	SEND_I_XXX(X=0) START_ACK_TIMER_ IF_NOT_RUNNING	NORMAL
	LOCAL_BUSY_DETECTED and P_FLAG=0	SEND_RNR_CMD(P=1) START_P_TIMER DATA_FLAG:=0 .... SEND_RNR_XXX(X=0) DATA_FLAG:=0	BUSY .... BUSY

Current State	Event	Action(s)	Next State
NORMAL	LOCAL_BUSY_DETECTED and P_FLAG=1	SEND_RNR_XXX(X=0) DATA_FLAG:=0	BUSY
	RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=0 or RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=0 or RECEIVE_I_RSP(F=1)_ WITH_UNEXPECTED_N(S) and P_FLAG=1	SEND_REJ_XXX(X=0) UPDATE_N(R)_RECEIVED UPDATE_P_FLAG START_REJ_TIMER IF_F=1_CLEAR_REMOTE_BUSY	REJECT
		SEND_REJ_CMD(P=1) UPDATE_N(R)_RECEIVED START_P_TIMER START_REJ_TIMER IF_F=1_CLEAR_REMOTE_BUSY	REJECT
	RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=1 or RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=1	SEND_REJ_XXX(X=0) UPDATE_N(R)_RECEIVED START_REJ_TIMER	REJECT
	RECEIVE_I_CMD(P=1)_ WITH_UNEXPECTED_N(S)	SEND_REJ_RSP(F=1) UPDATE_N(R)_RECEIVED START_REJ_TIMER	REJECT
	RECEIVE_I_RSP(F=X) and P_FLAG=F or RECEIVE_I_CMD(P=0) and P_FLAG=0	V(R):=V(R)+1 DATA_INDICATION SEND_ACKNOWLEDGE_CMD(P=1) START_P_TIMER UPDATE_N(R)_RECEIVED IF_F=1_CLEAR_REMOTE_BUSY	NORMAL
		V(R):=V(R)+1 DATA_INDICATION UPDATE_P_FLAG SEND_ACKNOWLEDGE_XXX(X=0) UPDATE_N(R)_RECEIVED IF_F=1_CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_I_RSP(F=0) and P_FLAG=1 or RECEIVE_I_CMD(P=0) and P_FLAG=1	V(R):=V(R)+1 DATA_INDICATION SEND_ACKNOWLEDGE_XXX(X=0) UPDATE_N(R)_RECEIVED	NORMAL

Current State	Event	Action(s)	Next State
NORMAL	RECEIVE_I_CMD(P=1)	V(R):=V(R)+1 DATA_INDICATION SEND_ACKNOWLEDGE_RSP(F=1) UPDATE_N(R)_RECEIVED	NORMAL
	RECEIVE_RR_CMD(P=0) or RECEIVE_RR_RSP(F=0) or RECEIVE_RR_RSP(F=1) and P_FLAG=1	UPDATE_P_FLAG UPDATE_N(R)_RECEIVED CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_RR_CMD(P=1)	SEND_ACKNOWLEDGE_RSP(F=1) UPDATE_N(R)_RECEIVED CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_RNR_CMD(P=0) or RECEIVE_RNR_RSP(F=0) or RECEIVE_RNR_RSP(F=1) and P_FLAG=1	UPDATE_P_FLAG UPDATE_N(R)_RECEIVED SET_REMOTE_BUSY	NORMAL
	RECEIVE_RNR_CMD(P=1)	SEND_RR_RSP(F=1) UPDATE_N(R)_RECEIVED SET_REMOTE_BUSY	NORMAL
	RECEIVE_REJ_CMD(P=0) and P_FLAG=0 or RECEIVE_REJ_RSP(F=X) and P_FLAG=F	V(S):=N(R) UPDATE_N(R)_RECEIVED UPDATE_P_FLAG RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY .... V(S):=N(R) UPDATE_N(R)_RECEIVED START_P_TIMER RE-SEND_I_CMD(P=1) CLEAR_REMOTE_BUSY	NORMAL .... NORMAL
	RECEIVE_REJ_CMD(P=0) and P_FLAG=1 or RECEIVE_REJ_RSP(F=0) and P_FLAG=1	V(S):=N(R) UPDATE_N(R)_RECEIVED RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_REJ_CMD(P=1)	V(S):=N(R) UPDATE_N(R)_RECEIVED RE-SEND_I_RSP(F=1) CLEAR_REMOTE_BUSY	NORMAL