

IEEE Standard for Medical Device Communications—Transport Profile—Connection Mode

Sponsor

**Medical Information Bus Committee
of the
Engineering in Medicine and Biology Society**

Approved December 13, 1994

IEEE Standards Board

Abstract: A local area network (LAN) for the interconnection of computers and medical devices is defined by the specifications and guidelines set forth in this standard. The functions, features, and protocols of the intra-room communications subnet of a bedside communications network known as the Medical Information Bus (MIB) are defined. This communications subnet is the functional equivalent for the MIB of the Transport, Network, Data Link, and Physical layers of the Organization for International Standards (ISO) Reference Model for Open Systems Interconnection (OSI). This standard defines the services and protocols for the MIB Transport, Network, and Data Link layers.

Keywords: bedside communications controller (BCC), bedside medical device, Data Link layer, device communications controller (DCC), frame, flow control, high-level data link control (HDLC), Medical Information Bus (MIB), Network layer, Physical layer, subnetwork, Transport layer

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

Copyright © 1995 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1995. Printed in the United States of America.

ISBN 1-55937-504-3

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

IEEE Standards documents are developed within the 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 that 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 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 may involve the use of patented technology. Their approval by the Institute of Electrical and Electronics Engineers does not mean that using such technology for the purpose of conforming to such standards is authorized by the patent owner. It is the obligation of the user of such technology to obtain all necessary permissions.

Introduction

(This introduction is not a part of IEEE Std 1073.3.1-1994, IEEE Standard for Medical Device Communications—Transport Profile—Connection Mode.)

This communications subnet is a member of the 1073 family of standards for computer communication with medical devices, known as the Medical Information Bus (MIB). Collectively, the MIB family of standards covers the entire seven layers of the communications architecture of the Open Systems Interconnect (OSI) Reference Model. This particular standard specifies the services and protocols for the Data Link, Network, and Transport layers, also known as layers 2–4.

This standard is intended to facilitate fast, reliable, and robust communications with a variety of medical devices at a single patient bedside in an integrated manner. The services and protocols defined address the special requirements of the medical users in the patient bedside environment. These include automatic detection of device connections (and disconnections) and establishment of communications without any action on the part of the user other than plugging in the communications cable; logical association of devices with a particular geographical location (a particular bed, and, by inference, a particular patient); and unique identification of devices on the network without address settings, switches, or unique codes in read-only memory.

The services and protocols of this standard are optimized for use in the medical bedside environment. As such, this communications subnet addresses the special requirements of the medical environment in which it is intended to operate. Consequently, this standard should be considered a “special purpose” communications facility.

Nonetheless, for the services defined, this is a self-contained and stand-alone specification for the lower four layers of the OSI Reference Model. There are no restrictions on the user entity which makes use of this standard. The scope of this standard does not preclude the use of the standard for domains outside of the medical bedside environment, but it is not the intent to define the requirements for, nor to optimize this standard for, those other domains.

At the time this standard was completed, the P1073.3.1 Working Group had the following membership:

Robert J. Kennelly, *Chair*

John J. Harrington, *Vice Chair*

Mike Glass, *Lower Layers Chair*

Charles Brill
Don Eddleman
Yossi Elaz
John Fioriglio
David F. Franklin
Kenneth Fuchs
Reed Gardner
Lee Golter

Harald Greiner
Bill Hawley
Earl F. Hoerner
Gideon Kantor
Randy Krohn
Ron Levin
Scott Morin
Katherine Nichols

Dave Ostler
S. Mark Poler
M. Michael Shabot
Bob Strehlow
Charles Turini
Cliff Weber
Jan Wittenber
Dale Zeskind

The following persons were on the balloting committee:

J. A. P. M. Ansems
George W. Beeler
Jeffrey B. Beers
Thomas Booker
Branko Celler
Todd Cooper
Phil Curran
James R. “Bob” Davis
Alan Figler
Paul Fitzmeyer
Michael Floetotto
David F. Franklin
Kenneth Fuchs
Elmer Gabrieli
John Gage
Reed Gardner

Shaun J. Glenn
Lee Golter
Harald Greiner
Charles Gropper
Gene Gryziecki
James Han
John J. Harrington
Bill Hawley
Earl F. Hoerner
Jorg Kampmann
Constance Kelly
Timothy J. Kelly
Randy Krohn
Christopher Lynch
Paul Oehler
James R. Peterson

S. Mark Poler
Vince C. Price
Robin Roehl
Anders B. Rorholt
Ricardo F. Ruiz
Carlos H. Salvador
Pascal Schirato
M. Michael Shabot
Andrew L. Spector
Mike Spicer
Sunder Subramanian
Alessandro Taddei
Helmut Thiemann
Charles Turini
Jan Wittenber
Dale Zeskind

When the IEEE Standards Board approved this standard on December 13, 1994, it had the following membership:

Wallace S. Read, *Chair*

Donald C. Loughry, *Vice Chair*

Andrew G. Salem, *Secretary*

Gilles A. Baril
Bruce B. Barrow
José A. Berrios de la Paz
Clyde R. Camp
James Costantino
Stephen L. Diamond
Donald C. Fleckenstein
Jay Forster*
Ramiro Garcia

Donald N. Heirman
Richard J. Holleman
Jim Isaak
Ben C. Johnson
Sonny Kasturi
Lorraine C. Kevra
E. G. “Al” Kiener
Ivor N. Knight

Joseph L. Koepfinger*
D. N. “Jim” Logothetis
L. Bruce McClung
Marco W. Migliaro
Mary Lou Padgett
Arthur K. Reilly
Ronald H. Reimer
Gary S. Robinson
Leonard L. Tripp

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
James Beall
Richard B. Engelman
Robert E. Hebner

Valerie E. Zelenty
IEEE Standards Project Editor

Contents

CLAUSE	PAGE
1. Overview.....	1
1.1 Scope.....	1
1.2 Purpose.....	1
1.3 Standards compatibility	2
2. References.....	2
3. Definitions.....	3
4. Abbreviations and acronyms.....	8
5. Communications nodes on the subnet.....	10
5.1 Overview	10
5.2 Bedside communications controller (BCC)	11
5.3 Device communications controller (DCC)	11
6. Inactive Transport layer	11
6.1 Service specification	11
6.2 PDU specification	19
6.3 Mode of operation.....	19
7. Inactive Network layer.....	22
7.1 Service specification	22
7.2 PDU specification	29
7.3 Mode of operation.....	30
8. Data Link layer	32
8.1 Table of capabilities.....	32
8.2 Service specification	32
8.3 PDU specification	49
8.4 DL elements of procedure.....	53
8.5 DL procedure descriptions.....	64
8.6 Interrupt requests and acknowledgments.....	74
8.7 Formal description of procedures	75
9. Labelling and conformance requirements.....	175
9.1 Implementation requirements	175
9.2 Labelling requirements	175
9.3 Conformance requirements.....	175
Annex (normative) Systems and data link management.....	177

IEEE Standard for Medical Device Communications—Transport Profile—Connection Mode

1. Overview

1.1 Scope

The specifications and guidelines set forth in this standard are intended to define a local area network (LAN) for the interconnection of computers and medical devices. This standard defines the functions, features, and protocols of the intra-room communications subnet of a bedside communications network known as the Medical Information Bus (MIB). This communications subnet is the functional equivalent for the MIB of the Transport, Network, Data Link, and Physical layers of the Organization for International Standards (ISO) Reference Model for Open Systems Interconnection (OSI). This standard defines the services and protocols for the MIB Transport, Network, and Data Link layers.

The Physical Layer, specified in IEEE Std 1073.4.1-1994¹

- a) Uses EIA-485 for data transfer.
- b) Provides DC power distribution for device communications hardware.
- c) Provides a pair of special function signals for the purposes of providing device connection indications, interrupt requests, time synchronization, and frame delimiting.
- d) Specifies a unique MIB connector and cable.

The MIB Data Link layer offers services for data line connection, data transfer, and disconnection using high-level data link control (HDLC) procedures. The Transport and Network layers are inactive [i.e., there are no functions for packet assembly and disassembly (PAD) or end-to-end acknowledgments or any other services usually associated with Transport and Network layers]. See figure 1.

1.2 Purpose

The purpose of this subnet is to provide connection-oriented communications services. It is functional as a stand-alone network for any application entity that can directly use the transport services offered. The LAN is designed to provide a logical association between a patient and those medical devices connected to, monitoring, or otherwise data-related to that patient and in close physical proximity. New device connections are automatically detected and communications are established with those new devices without the need of any user input. The network is intended to provide a highly reliable data transfer medium with a priority on providing quantitative throughput to each attached node. The network is intended to be highly

¹Information on references can be found in clause 2.

robust in an environment where devices are frequently connected to and disconnected from the network. It responds to a changing device topology by establishing new connections and detecting disconnections in a manner that is transparent to the clinical user. Hence, in regard to the device nodes on the network, it is a completely automatic and self-regulating communications subnet.

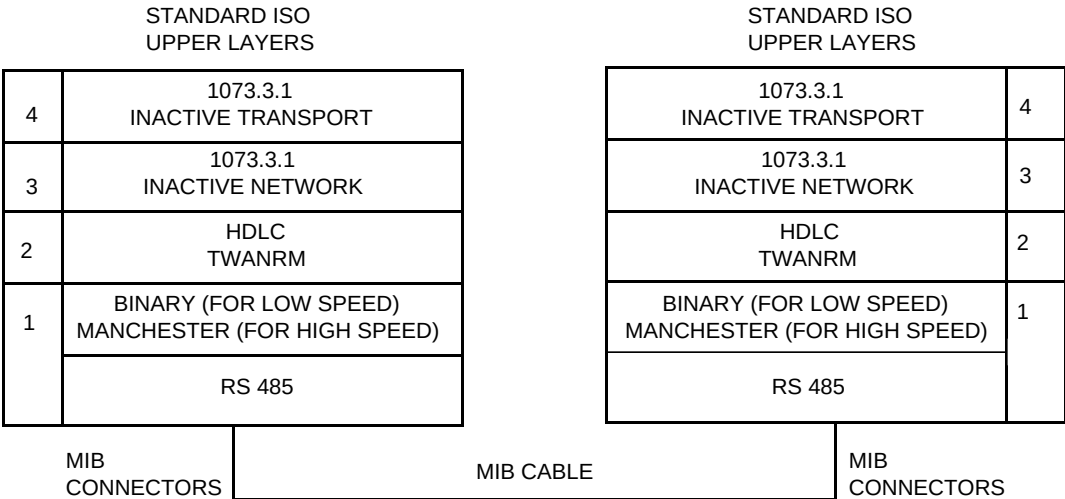


Figure 1—OSI layer representation of IEEE 1073.3.1

1.3 Standards compatibility

The polling protocol procedures defined in clause 8 utilize some of the concepts and principles, as well as commands and responses, of the link control procedures known as two-way alternate normal response mode (TWANRM), as defined in ISO/IEC 4335: 1993. TWANRM defines a half-duplex data link in which there is a primary station and a secondary station. The secondary station shall only transmit in response to a transmission by the primary station. For the purpose of this standard, the bedside communications controller (BCC) is the primary station, while the device communications controller (DCC) is the secondary station. The frame structure defined for the data link procedures (as described in clause 8) utilizes the same basic structure and format as defined in ISO/IEC 3309: 1993 with the same field usage, except that some fields are unused and the poll bit usage is defined by this standard. Use of the exchange identification (XID) frame for negotiation purposes during connection establishment conforms with usage described in ISO/IEC 8885: 1993. The Physical layer media specification and data transfer coding are called out in IEEE Std 1073.4.1-1994 and EIA-485 (1983).

2. References

This standard shall be used in conjunction with the following standards. When the following standards are superseded by an approved revision, the revision shall apply.

EIA-485 (1983), Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems.²

IEEE Std 1073.4.1-1994, IEEE Standard for Medical Device Communications—Physical Layer Interface—Cable Connected.³

²EIA publications are available from Global Engineering, 1990 M Street NW, Suite 400, Washington, DC, 20036, USA.
³IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

ISO 7498: 1984, Information processing systems—Open Systems Interconnection—Basic Reference Model (and Technical Corrigendum 1: 1988, and Addendum 1: 1987, Connectionless-mode transmission).⁴

ISO 8601: 1988, Data elements and interchange formats—Information interchange—Representation of dates and times (and Technical Corrigendum 1: 1991).

ISO/IEC 3309: 1993, Information technology—Telecommunications and information exchange between systems—High-level data link control (HDLC) procedures—Frame structure.

ISO/IEC 4335: 1993, Information technology—Telecommunications and information exchange between systems—High-level data link control (HDLC) procedures—Elements of procedures.

ISO/IEC 7809: 1993, Information technology—Telecommunications and information exchange between systems—High-level data link control (HDLC) procedures—Classes of procedures.

ISO/IEC 8072: 1994, Information technology—Open Systems Interconnection—Transport service definition.

ISO/IEC 8073: 1992, Information technology—Telecommunications and information exchange between systems—Open Systems Interconnection—Protocol for providing the connection-mode transport service (and Technical Corrigendum 1: 1993).

ISO/IEC 8802-2: 1994 [ANSI/IEEE Std 802.2, 1994 Edition], Information technology—Telecommunications and information exchange between systems—Local and Metropolitan area networks—Specific requirements—Part 2: Logical link control.⁵

ISO/IEC 8885: 1993, Information technology—Telecommunications and information exchange between systems—High-level data link control (HDLC) procedures—General purpose XID frame information field content and format.

3. Definitions

For the purposes of this standard, the following definitions apply.

3.1 abort: To terminate the transmission of a frame before it has been completely transmitted.

3.2 address field: The part of a protocol data unit (PDU) that contains an address that identifies one or more addressable entities. The address may be a single-source address, single-destination address, or multiple-destination address multicast). (For this standard, the address field comprises the first two octets of each HDLC frame.)

3.3 baud rate: For a given encoding scheme and data rate, the maximum number of signal transitions transmitted on a serial interface in a 1 s period.

3.4 bedside: Those medical devices that directly interact with, monitor, provide treatment to, or are in some way associated with a single patient.

3.5 bedside communications controller (BCC): A communications controller, typically located at a patient bedside, that serves to interface between one or more medical devices. The BCC may be embedded into

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

⁵This publication is available from the ISO Central Secretariat. It is also available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

local display, monitoring, or control equipment. Alternatively, it may comprise part of a communications router to a remote hospital host computer system.

3.6 binary encoding: An encoding scheme for serial communications in which the symbol for a logic “0” or “1” assumes a specified voltage level for the full duration of a signalling bit period.

3.7 confirm: A primitive provided from one layer entity to another layer entity for the purpose of verifying that the entity generating the confirm primitive responded to a request primitive originating from the entity that received the confirm primitive.

3.8 connect: Indicates that a DCC has been physically attached to a BCC port, generally by means of an interconnecting cable. Connect does not necessarily imply that communications have been logically established between the BCC and the DCC.

3.9 control field: The sequence of eight (or sixteen, if extended) bits immediately following the address field of a frame. This octet identifies the HDLC frame type.

3.10 data bit: A single entity of information that is transmitted across a serial signalling media. A bit assumes one of two values: logic “0” or logic “1.” A data bit may convey control, address, information, or frame check sequence (FCS) data.

3.11 data link (DL): The services required to allow a local entity to establish a connection-oriented communications channel with a remote entity and exchange packets over the channel.

3.12 data link service access point (DLSAP): The point at which logical connection between Data Link layers occurs. In IEEE 1073, both BCCs and DCCs have DLSAPs.

3.13 data rate: The rate at which a data path (e.g., a channel) carries data, measured in bits per second (b/s).

3.14 date/time-of-day clock: A clock that shall be maintained by the BCC, providing an indication of time, from year to milliseconds, with a resolution of 1 ms.

3.15 delimiter: A nondata symbol transmitted on a serial link to delineate the beginning or end of a frame of data.

3.16 device: A medical instrument or other device used to generate data on a particular patient.

3.17 device communications controller (DCC): A communications interface associated with a medical device. A DCC may support one or more physically distinct devices acting as a single network communications unit. Its purpose is to provide a point-to-point serial communication link to a BCC.

3.18 differential signal: A signal that is conveyed between two separate conductors, instead of one active conductor and signal ground. The magnitude of the differential signal is the difference between the two signals, rather than the voltages between the two individual signals and ground.

3.19 electromagnetic compatibility (EMC): The requirements for electromagnetic emission and susceptibility dictated by the physical environment and regulatory governing bodies in whose jurisdiction a piece of equipment is operated. Compatibility requirements are thus site specific, and their definition is not within the scope of this standard.

3.20 encoding: The representation of data bits and nondata information for the purpose of signal transmission across a serial communications medium. Nondata information includes indications of start and end of octets and frame transmission.

3.21 entity: An object that provides an implementation of the functions defined by this standard. In an Open Systems Interconnection (OSI) communication environment, there are separate entities for implementing the functions of separate layers.

3.22 eXchange IDentification (XID): A frame exchange used during the logical connection (initialization) sequence for the purpose of transferring Data Link layer and upper layer parameters between the primary station (BCC) and the secondary station (DCC).

3.23 flow control: A mechanism used in open systems to regulate data communications to ensure that no data is lost in the case of insufficient buffer size, or other limited resources. Flow control is done using receive not ready (RNR), receive ready (RR), and the flow control primitives. Flow control in devices or systems entails operations from the Data Link through the Application layer.

3.24 frame: A continuous transmission of octets from one station (BCC or DCC) to the other station. A Physical layer frame is also referred to as a physical protocol data unit (PhPDU). The physical service data unit (PhSDU) passed between the Data Link layer and the Physical layer consists of the data octets portion of the frame. The PhSDU consists of an integral number of binary octets. The frame consists of these octets, plus other encoded symbols that are added by the Physical layer. For low-speed operation, each octet consists of a binary-encoded start bit, eight data bits, and a stop bit. The first octet of a low-speed frame consists of a flag octet. The last octet of a low-speed frame is either a flag octet or an abort octet. For high-speed operation, each octet consists of eight Manchester biphase-encoded data bits. For high-speed operation, there is also a start delimiter, and either an end delimiter or an abort delimiter, indicating the beginning and end of individual frames, respectively. The operation of concatenating the delimiters to the data octets is performed by the Physical layer.

3.25 frame check sequence (FCS): The field immediately preceding the closing delimiter of a frame. The FCS used is the 16 b polynomial defined by the cyclic redundancy check sequence specified by ITU-T (CRC-ITU-T). This field allows the detection of errors by the receiving station.

3.26 frame validity checking: Verification, by a receiving station, of correct frame transmission by the transmitting station. Frame checking entails verifying for correct encoding of all transmitted delimiters and start and stop bits, data bits, and octet encoding. For the Medical Information Bus (MIB), frame validity checking is performed by both the Physical layer and the Data Link layer.

3.27 high-level data link control (HDLC): A set of Data Link layer communication protocols defined by ISO/IEC 3309: 1993, ISO/IEC 4335: 1993, ISO/IEC 7809: 1993, and ISO/IEC 8885: 1993. These standards define a multiplicity of point-to-point and multidrop protocols. These include both master/slave and peer-to-peer types of data links, employing both half-duplex and full-duplex methodologies. (For the data link portion of this standard, a particular subset, known as TWANRM, is utilized. TWANRM defines a half-duplex master/slave variation of HDLC).

3.28 host: A device to which other devices (peripherals) are connected and that generally controls those devices.

3.29 host computer: A computer, attached to a network, providing primary services such as computation, data base access, and special programs or programming languages.

3.30 idle state: The inactive (nontransmitting) state of a serial link transmitter. The output of an inactive transmitter shall present a high impedance to a bidirectional serial link.

3.31 indication: A mechanism informing an entity of the occurrence of an event in a lower layer entity. Alternatively, an indication may provide evidence of a request by a remote station entity.

3.32 information (I) field: The sequence of octets occurring between the control field and the end of the logical link control (LLC) PDU. The information field consists of I, TEST, and unnumbered information (UI). PDUs are not interpreted at the LLC sublayer.

3.33 intentional disconnect: A primitive passed to the upper layers of both BCCs and DCCs to allow the device or system to recognize that an upcoming network disconnection is intentional. The method to trigger this primitive is left to the device or system designer.

3.34 interrupt: A mechanism provided to allow a DCC to request service from a BCC prior to the next scheduled polling time. Interrupt is also used to establish initial connection for high-speed DCCs.

3.35 layer: A subdivision of the OSI architecture, constituted by subsystems of the same rank.

3.36 link error counter (LEC): A counter maintained by a network station to track all data link errors. The maximum permissible value is a system parameter.

3.37 logically connected: The state following a successful negotiation between a DCC and a BCC for network connection. The trigger for this state is a set normal response mode (SNRM) being sent from a BCC to a DCC.

3.38 Manchester biphas-L encoding: A signal transmission method defined for the representation of binary data bits. Manchester biphas-L encoding specifies two “half-bits,” such that there is a guaranteed mid-bit transition in the transmitted signal. The transition is defined to be positive for encoding a logic “0,” and negative for encoding a logic “1.”

3.39 medical devices: Devices, apparatus, or systems that are used to prevent, diagnose, or treat diseases in humans. These devices do not normally enter metabolic pathways. They provide support for electronic communications.

3.40 Medical Information Bus (MIB): The informal name for the IEEE 1073 family of standards.

3.41 network loading unit (NLU): The time required to transmit one octet over the MIB Physical layer.

3.42 network service access point (NSAP): The point at which an entity can request connection to the network. In IEEE 1073, both BCCs and DCCs have an NSAP.

3.43 normal response mode (NRM): The secondary station (a DCC) initiates transmission only after receiving express permission to do so from the primary station (the BCC). *Syn:* command/response.

3.44 octet: A group of eight adjacent bits.

3.45 Open Systems Interconnection (OSI): Refers to ISO 7498: 1984. A model that provides a common basis for the coordination of standards development for the purpose of systems interconnection, while allowing existing standards to be placed into perspective within the overall reference model. The OSI model defines a seven-layer functional model, including descriptions of the functions defined for each layer.

3.46 PH_CHARACTERISTICS: Physical layer characteristics indication primitive. A set of attributes used to delineate the optional capabilities of a BCC or DCC Physical layer. This set of capabilities is required to be labeled on a BCC or DCC. In addition, there is required to be a mechanism for providing this information to the BCC's or DCC's MIB upper layers. The list of attributes includes the following capabilities: low-speed (2400 Bd or 9600 Bd) operation, high-speed (1 Mb/s) operation, +12 V used from MIB connector, BCC capability, capability for DCC interrupt function, and capability for DCC sync function.

3.47 physical interface data unit (PhsDU): An octet (8 data bits) that is communicated across the interface between a BCC or a DCC Physical layer and Data Link layer.

3.48 physically connected: The state when an IEEE 1073 connector is connected at both ends of the cable. This is indicated to both the DCC and the BCC by voltage levels.

3.49 physical protocol data unit (PhPDU): A frame, consisting of both nondata symbols (delimiters or start and stop bits) and data symbols (data bits), that is transmitted from a BCC to a DCC or from a DCC to a BCC. A PhPDU includes mechanisms indicating the start and end of transmission. PhPDUs may be preceded and followed by idle periods on the serial transmission medium. There are conditions in which two consecutive PhPDUs may be transmitted contiguously by the same station.

3.50 physical protocol service unit (PhSDU): A symbol for a single data bit transmitted on the physical medium between a BCC and a DCC.

3.51 physical service user (PhS_User): Refers to the Data Link layer entity, which is the user of a BCC or DCC Physical layer entity.

3.52 poll/final (P/F) bit: A bit set by the BCC to “1” to solicit a response from a DCC. The P/F bit is set to “1” by the DCC to indicate that the final frame of the message has been transmitted.

3.53 primitive: A mechanism for communicating between the entities of different layers of the MIB communications standard (e.g., between the Physical layer and Data Link layer entities). A primitive may request a particular service, provide indication of a request, provide a response to an indication, or provide confirmation of a response.

3.54 protocol error counter (PEC): A counter contained by a network station to keep track of protocol errors such as invalid frames.

3.55 request: A type of primitive in which one layer entity solicits another layer entity to perform a particular function.

3.56 reset: An action that occurs when certain error conditions occur, or when error conditions exceed a preset value. Reset causes the Data Link layer to go to the offline state. Reconnection can then be requested by the DCC.

3.57 response: A reply represented in the control field of a response protocol data unit (PDU). It advises the address destination logical link control (LLC) of the action taken by the source LLC to one or more command PDUs.

3.58 sampling rate: For the purposes of “de-bouncing” and noise rejection, the receiving station (BCC or DCC) of a special function signal shall sample the signal at a specified rate for a specified length of time in order to determine the “filtered” logic sense of the signal. The clock rate used is referred to as the sampling rate.

3.59 SENSE/INT/SYNC-IN: A special function signal from the BCC to the DCC in a BCC-to-DCC interconnection cable used for three purposes. The signal allows the DCC to sense if it has been physically connected to a BCC port, allows a BCC to send sync pulses to the DCC, and allows the BCC to send interrupt deactivate pulses to the DCC.

3.60 SENSE/INT/SYNC-OUT: A special function signal from the DCC to the BCC in a BCC-to-DCC interconnection cable used for three purposes. The signal allows the BCC port to sense if a DCC has been physically connected, allows a DCC to send sync pulses to the BCC, and allows the DCC to send interrupt activate and interrupt deactivate pulses to the BCC.

3.61 sequence number: Each I frame is sequentially numbered with a number listed in the control field, from 0 to 7. The sequence numbers cycle through the entire range.

3.62 set normal response mode (SNRM): An HDLC message sent by a BCC to a DCC when a successful connection to the network has occurred.

3.63 signal transition distortion: The deviation from the ideal time between signal transitions of a signal on a serial communication link. Contributing factors to signal transition distortion include clock oscillator frequency differences, variations in transmitter delay times and rise/fall times, and phase shifts introduced by cables. For a 1 Mb/s Manchester-encoded signal, the ideal times between transitions are 500 ns and 1 μ s. A transition is defined as the time that a differential signal passes through 0 V.

3.64 station: A physical device that may be attached to a shared medium local area network (LAN) for the purpose of transmitting and receiving information on that shared medium. A data station is identified by a destination address. *syn:* data station.

3.65 subnetwork: A functional unit comprised of a single dual bus pair and those access units (AUs) attached to it. Subnetworks are physically formed by connecting adjacent nodes with transmission links.

3.66 supervisory format: The format used to perform data link supervisory control such as acknowledging I frames, requesting retransmission of I frames, and requesting a temporary suspension of transmission of I frames.

3.67 sync: A function in which a BCC may precisely synchronize a common fiduciary (now) maintained by multiple DCCs to the value of its own real-time clock, to an accuracy of approximately 1 ms. The BCC implements this function by transmitting a 3 μ s pulse to one or more DCCs. A sync pulse may also be transmitted by a BCC or a DCC for reasons other than time synchronization.

3.68 transmission format: The specified arrangement of delimiter symbols or start and stop symbols, and data bit symbols that comprise a complete transmitted signal frame (PhPDU). Data bit symbols are always arranged in octets (8 b groupings). The definition of the transmission format also includes the selected encoding scheme, which is binary for low-speed operation, and Manchester biphas-L for high-speed operation.

3.69 two-way alternating (TWA): A subset of HDLC defining half-duplex, rather than simultaneous, two-way communications.

3.70 unintentional disconnect: Any disconnection from the network, at either end, that is not preceded by an intentional disconnect primitive, within the specified time period.

3.71 unnumbered (U) format: The format used to provide additional data link control functions and unnumbered information transfer. This format shall contain no sequence numbers, but shall include a P/F bit that may be set to “1” or “0.”

4. Abbreviations and acronyms

ACK	ACKnowledge
AU	Access Unit
BCC	Bedside Communications Controller
CRC-ITU-T	Cyclic Redundancy Check sequence specified by ITU-T
DCC	Device Communications Controller
DISC	DISConnect
DL	Data Link

DLPDU	Data Link Protocol Data Unit
DLSAP	Data Link Service Access Point
DLSDU	Data Link Service Data Unit
DM	Disconnected Mode
EMC	Electromagnetic Compatibility
F	Final
FCS	Frame Check Sequence
GI	Group Identifier
HDLC	High-Level Data Link Control
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union—Telecommunication Standardization Sector (formerly CCITT)
LAN	Local Area Network
LEC	Link Error Counter
LLC	Logical Link Control
LSB	Least Significant Bit
MDDL	Medical Device Data Language
MIB	Medical Information Bus
MLEC	Maximum Link Error Counter
MPEC	Maximum Protocol Error Counter
MSB	Most Significant Bit
N	Network
NDM	Normal Disconnected Mode
NLU	Network Loading Unit
NPDU	Network Protocol Data Unit
N(R)	Receive sequence Number
NRM	Normal Response Mode
N(S)	Send sequence Number
NSAP	Network Service Access Point
NSDU	Network Service Data Unit
OSI	Open Systems Interconnection
P	Poll
PAD	Packet Assembly and Disassembly
PDU	Protocol Data Unit
PEC	Protocol Error Counter
P/F	Poll/Final
PhPDU	Physical Protocol Data Unit
PhSDU	Physical Service Data Unit
PhsDU	Physical Interface Data Unit
PhS_User	Physical Service User
PI	Parameter Identifier
RAC	IEEE Registration Authority Committee
RD	Request Disconnection
RIM	Request Initialization Mode
RNR	Receive Not Ready
RR	Receive Ready
S	Supervisory
SAP	Service Access Point
SIM	Set Initialization Mode
SNRM	Set Normal Response Mode

SSAP	Source Service Access Point
T	Transport
TPDU	Transport Protocol Data Unit
TSAP	Transport Service Access Point
TSDU	Transport Service Data Unit
TWA	Two-Way Alternating
TWANRM	Two-Way Alternating Normal Response Mode
U	Unnumbered (format)
UA	Unnumbered Acknowledge
UD	User Data
UDI	User Data Identifier
UI	Unnumbered Information
UP	Unnumbered Poll
V(R)	Receive state Variable
V(S)	Send state Variable
XID	eXchange IDentification

5. Communications nodes on the subnet

5.1 Overview

The IEEE 1073.3 subnet defines a star topology, requiring each device to have its own connection directly into the network. On the communications subnet, there are two types of communications nodes allowed on the network.

- a) The bedside communications controller (BCC) is the primary node and functions as the subnet controller and the hub of the star.
- b) Device communication controllers (DCCs) are secondary nodes and variable in number up to the loading capacity of the BCC and/or number of physical port sockets.

The devices connect to the MIB through the DCC. The BCC can interface directly to a local host computer, or to a remote host computer over an open system network. The portion of the BCC performing IEEE 1073.3 operations would be the same in both situations. Note, however, that the BCC would also include internet-working functions in the latter case. Unless otherwise noted, all references to a BCC in this standard refer only to those components performing the functions of IEEE 1073.3 (see figure 2).

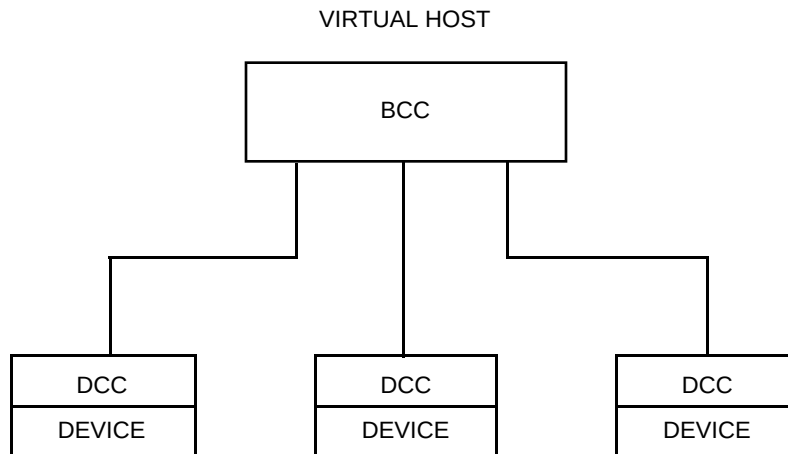


Figure 2—IEEE 1073.3 DCC-to-BCC connection topology

5.2 Bedside communications controller (BCC)

The BCC is the subnet controller for the network at the bedside. It functions as a gateway between all those devices at a patient's bedside and a host computer system desiring to exchange messages with those devices. There may be more than one BCC associated with a particular patient, but there is never more than one patient per BCC.

The BCC functions as the primary station in the HDLC two-way alternate normal response mode (TWANRM). As such, it is responsible for polling all connected DCCs. In addition, it must poll empty ports to detect new DCC connections. For connected devices, the BCC shall delay for no longer than 1 s between successive polls to a device, and shall not exceed a maximum polling rate of 250 Hz.

The BCC is a logical and functional entity, and as such no particular implementation is implied by this standard.

5.3 Device communications controller (DCC)

The DCC is the interface for each device onto the MIB network at the bedside. All connections are initiated by the DCC. The DCC communicates only by responding to the polls of a BCC. The DCC in this standard refers only to those components that perform the functions of the IEEE 1073.3 subnet.

The DCC is a logical and functional entity, and as such no particular implementation is implied by this standard.

6. Inactive Transport layer

6.1 Service specification

The following subclauses specify the services offered by the MIB inactive Transport layer to the transport user (T_user), as viewed by the transport user. These services allow a T_user layer local entity to establish a connection-oriented communications channel with a remote T_user entity and exchange packets over the channel. These services are meant to follow the complete ISO/IEC 8072: 1994 Transport service definition, Class 0. For the purposes of this standard, if there are any differences between this standard and ISO/IEC 8072: 1994, this standard shall have precedence. The services are described in an abstract way and do not imply any particular implementation.

6.1.1 Overview of T_user and transport interactions

6.1.1.1 Transport connection phase

The T_user layer is responsible for initiating a transport connection. Within the MIB, new connections are always started by the DCC requesting connection via the T_CONNECT request service primitive. The automatic connect feature of the network is dependent upon the DCC being informed by the device that connection is desired.

The service primitives associated with transport connection are

T_CONNECT.request
T_CONNECT.indication
T_CONNECT.response
T_CONNECT.confirm

The T_CONNECT.request primitive is passed to the transport (T) layer to request that a logical connection be established between a local transport service access point (TSAP) and a remote TSAP. The T_CONNECT indication is passed from the T layer to indicate the results of an attempt by a remote entity to establish a connection to a local TSAP. The T_CONNECT.response primitive is passed to the N layer to convey the results of the previous associated T_CONNECT.indication primitive. The T_CONNECT.confirm primitive is passed from the T layer to convey the results of the previous associated T_CONNECT.request.

6.1.1.2 Transport data transfer phase

All application messages are sent using data transfer service, using the T_Data request primitive. This class of service is connection oriented. Data is transmitted using transport service data units (TSDUs). This service is subject to flow control by higher layers.

The primitives associated with data transfer are

T_DATA.request
T_DATA.indication

The T_DATA.request primitive is passed to the T layer to request that a TSDU be sent. The T_DATA.indication primitive is passed from the T layer to indicate the arrival of a TSDU.

6.1.1.3 Transport disconnection phase

Either the DCC or the BCC can initiate a disconnect. Disconnection is not subject to the response of a higher layer, but is treated as an automatic sequence once either end of the connection initiates it.

The primitives associated with disconnection are

T_DISCONNECT.request
T_DISCONNECT.indication

The T_DISCONNECT request is passed to the N layer to initiate the disconnect sequence. The T_DISCONNECT indication is passed from the T layer to indicate the termination of a connection by the remote entity.

6.1.2 Detailed service specifications

The following subclauses describe 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 shall be available to the receiving entity. No specific implementation is implied by this standard.

6.1.2.1 T_CONNECT.request**6.1.2.1.1 Function**

This primitive is the service request primitive for the transport connection establishment process.

6.1.2.1.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
T_CONNECT.request (
    called address,
    calling address,
    TSDU
)
```

The called address is always set to 0; this parameter is not used by this service.

The calling address is always set to 0; this parameter is not used by this service.

The TSDU parameter will include the XID information field passed down from the DCC upper layers and transmitted in the DCC's XID response frame.

6.1.2.1.3 When generated

This primitive is passed from the T_user layer to the T_layer when the T_user entity wishes to establish a transport connection to a remote TSAP.

6.1.2.1.4 Effects on receipt

The receipt of this primitive by the T layer causes the local T entity to initiate the establishment of a connection with the remote T entity.

6.1.2.1.5 Additional comments

A possible logical sequence of primitives associated with transport connection is illustrated in figure 3. This service primitive is only available to the DCC.

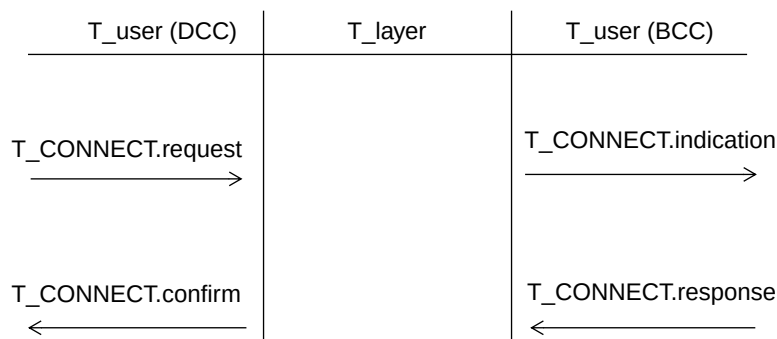


Figure 3—Timing sequence diagram for Transport layer connection primitives

6.1.2.2 T_CONNECT.indication**6.1.2.2.1 Function**

This primitive is the service indication primitive for the network connection process.

6.1.2.2.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
T_CONNECT.indication (  
    called address,  
    calling address,  
    TSDU  
)
```

The called address is always set to 0; this parameter is not used by this service.

The calling address is always set to 0; this parameter is not used by this service.

The TSDU parameter contains the last XID information field that originated from the DCC and resulted in a successful connection establishment bandwidth negotiation.

6.1.2.2.3 When generated

This primitive is passed from the T_layer to the T_user layer to indicate that a remote T_user entity desires a connection with a local T_user entity.

6.1.2.2.4 Effects on receipt

The T_user layer entity is expected, but is not required, to respond with a T_CONNECT.response.primitive (accepted) or a T_DISCONNECT.request (declined) to either establish a connection with, or terminate the transport connection request from, the remote T_user entity.

6.1.2.2.5 Additional comments

A possible logical sequence of primitives associated with transport connection is illustrated in figure 4. This service primitive is only available to the BCC.

6.1.2.3 T_CONNECT.response

6.1.2.3.1 Function

This primitive is the service response primitive for the transport connection process.

6.1.2.3.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
T_CONNECT.response (  
    responding address  
)
```

The responding address is always set to 0; this parameter is not used by the IEEE 1073.3.1 transport service.

6.1.2.3.3 When generated

This primitive is passed from the T_user layer to the T_layer to indicate that the local T_user entity accepts a connection with a remote T_user entity.

6.1.2.3.4 Effects on receipt

The local T layer entity completes the connection phase; the T_user may use the connection for data transfer.

6.1.2.3.5 Additional comments

A possible logical sequence of primitives associated with transport connection is illustrated in figure 4. This service primitive is only available to the BCC.

6.1.2.4 T_CONNECT.confirm

6.1.2.4.1 Function

This primitive is the service confirm primitive for the transport connection process.

6.1.2.4.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
T_CONNECT.confirm (  
    responding address,  
    TSDU  
)
```

The responding address is always set to 0; this parameter is not used by the IEEE 1073.3.1 transport service.

The TSDU will reflect the most recent values of the XID information field that were transferred from the DCC to the BCC. These will correspond to the values contained in the TSDU parameter associated with the most recent T_CONNECT.request primitive that was passed from the T_user entity prior to the last transmission of an XID response frame.

6.1.2.4.3 When generated

This primitive is passed from the T_layer to the T_user layer to indicate that the remote T_user entity accepts the previous connection request service primitive from the local T_user entity.

6.1.2.4.4 Effects on receipt

The T_user layer can now use this network connection for data unit transfer.

6.1.2.4.5 Additional comments

A possible logical sequence of primitives associated with DL connection is illustrated in figure 3. This service primitive is only available to the DCC.

6.1.2.5 T-DATA.request

6.1.2.5.1 Function

This primitive is the service request primitive for the TSDU transfer service.

6.1.2.5.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
T_DATA.request (
    TSU
)
```

The TSU parameter specifies the TSU to be transferred by the T layer entity.

6.1.2.5.3 When generated

This primitive is passed from the T_user layer to the T layer to request that TSU be transferred to a remote TSAP.

6.1.2.5.4 Effects on receipt

The receipt of this primitive by the T layer causes the T layer entity to transfer the TSU using transport data transfer services.

6.1.2.5.5 Additional comments

A possible logical sequence of primitives for successful data transfer is shown in figure 4.

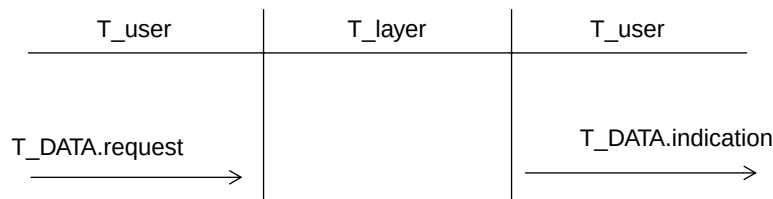


Figure 4—Timing sequence diagram for Transport layer data transfer primitives

6.1.2.6 T_DATA.indication

6.1.2.6.1 Function

This primitive is the service indication primitive for the data transfer service.

6.1.2.6.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
T_DATA.indication(
    TSU
)
```

The TSU parameter specifies the transport service data unit that has been received by the local T layer entity from the remote T layer entity.

6.1.2.6.3 When generated

This primitive is passed by the T layer to the T user layer to indicate the arrival of a TSDU from the specified remote T_user layer entity.

6.1.2.6.4 Effect on receipt

The effect of receipt of the primitive by the T_user layer is unspecified.

6.1.2.6.5 Additional comments

In the absence of errors, the contents of the TSDU parameter are logically complete and unchanged relative to the TSDU parameter in the associated (remote) T_DATA.request primitive.

6.1.2.7 T_DISCONNECT.request

6.1.2.7.1 Function

This primitive is the service request primitive for the disconnection service.

6.1.2.7.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
T_DISCONNECT.request (
    TSDU
)
```

The TSDU parameter will always be empty; IEEE 1073.3 does not support data transfer on disconnection.

6.1.2.7.3 When generated

This primitive is passed from the T_user layer to the T layer when the T_user layer wishes to terminate a connection.

6.1.2.7.4 Effects on receipt

Receipt of this primitive causes the T layer to immediately terminate the connection.

6.1.2.7.5 Additional comments

All unacknowledged TSDUs are discarded. The logical disconnect sequence is a destructive process. No guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Orderly disconnection (i.e., without loss of data) is the responsibility of a higher layer service.

A possible logical sequence of primitives associated with a successful logical disconnect sequence is illustrated in figure 5.

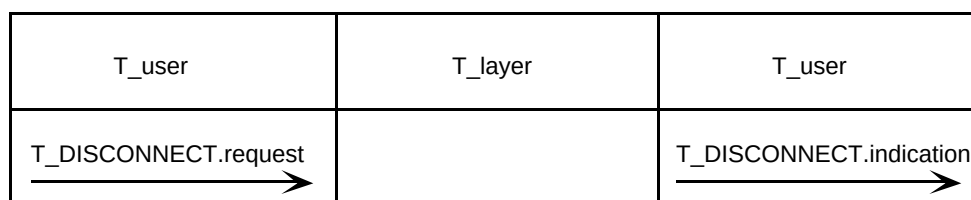


Figure 5—Timing sequence diagram for Transport layer disconnect primitives

6.1.2.8 T_DISCONNECT.indication

6.1.2.8.1 Function

This primitive is the service indication for the disconnection service.

6.1.2.8.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
T_DISCONNECT.indication (
    reason,
    TSDU
)
```

The reason parameter specifies the reason for the disconnection, which may include a request by a remote entity, or a condition internal to the T layer.

Except for the case of a failed DL XID bandwidth negotiation, IEEE 1073.3 does not support data transfer during the disconnection phase. In all other instances, the value of the TSDU parameter shall always be empty. In the case of a failed XID bandwidth negotiation, the TSDU parameter shall contain the value of the XID information field received from the remote T_entity.

6.1.2.8.3 When generated

This primitive is passed from the T layer to the T_user layer to signal an immediate disconnection.

6.1.2.8.4 Effects on receipt

The T_user entity may no longer use the indicated connection for data transfer. Any data transmissions not acknowledged shall be deemed not received and appropriate action shall be taken.

6.1.2.8.5 Additional comments

All unacknowledged TSDUs are discarded. The logical disconnect sequence is a destructive process. No guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Orderly disconnection (i.e., without loss of data) is the responsibility of a higher layer service.

A possible logical sequence of primitives associated with a successful logical disconnect sequence is illustrated in figure 5.

6.2 PDU specification

6.2.1 General

The following subclauses define in detail the transport PDU structure.

6.2.2 TPDU format

All transport PDUs shall consist of an information field with a single octet inactive Transport layer header (see figure 6).

HEADER	INFORMATION
0	TSDU

WHERE

HEADER = INACTIVE TRANSPORT LAYER HEADER
INFORMATION = INFORMATION FIELD
TSDU = TRANSPORT SERVICE DATA UNIT

Figure 6—Transport PDU format

6.2.3 Elements of the TPDU

6.2.3.1 Inactive Transport layer header

All frames shall start with a single-octet header field that shall contain a zero value. This header is provided to allow future IEEE 1073.3 versions to support a functional Transport layer by providing a nonzero header.

6.2.3.2 Information field

Information may be any sequence of octets. This field will contain the TSDU.

6.2.4 Invalid TPDU

An invalid TPDU is defined as a TPDU without a zero value in the header field. Receipt of an invalid TPDU will raise an exception to the T_user.

6.3 Mode of operation

The MIB inactive Transport layer is a service-mapped layer. All primitives received from the Session layer are directly mapped onto Network layer service. The only exceptions are the T_DATA primitives. These primitives will manage a single-octet header containing a zero. This octet is removed from the network PDU before passing the PDU to the Session layer, and added to the session PDU before passing the PDU to the Network layer. If a nonzero header is received, the connection will be aborted (see figures 7 and 8).

The state tables for the DCC inactive Transport layer and the BCC inactive Transport layer are given in tables 1 and 2, respectively.

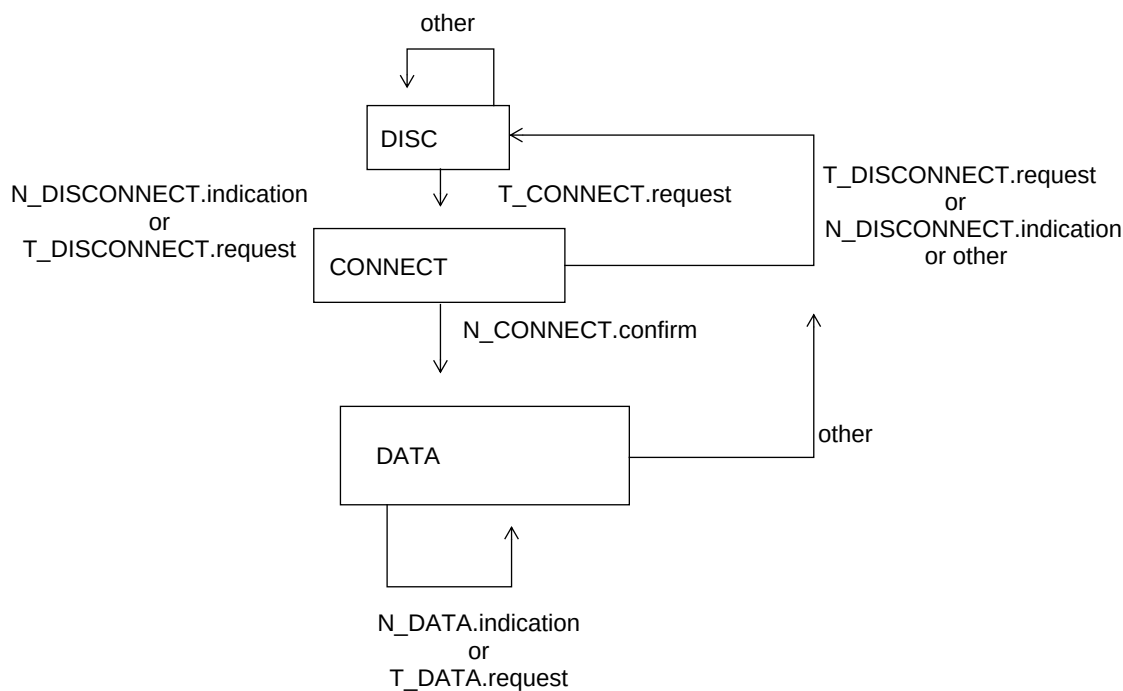


Figure 7—Inactive Transport layer DCC state diagram

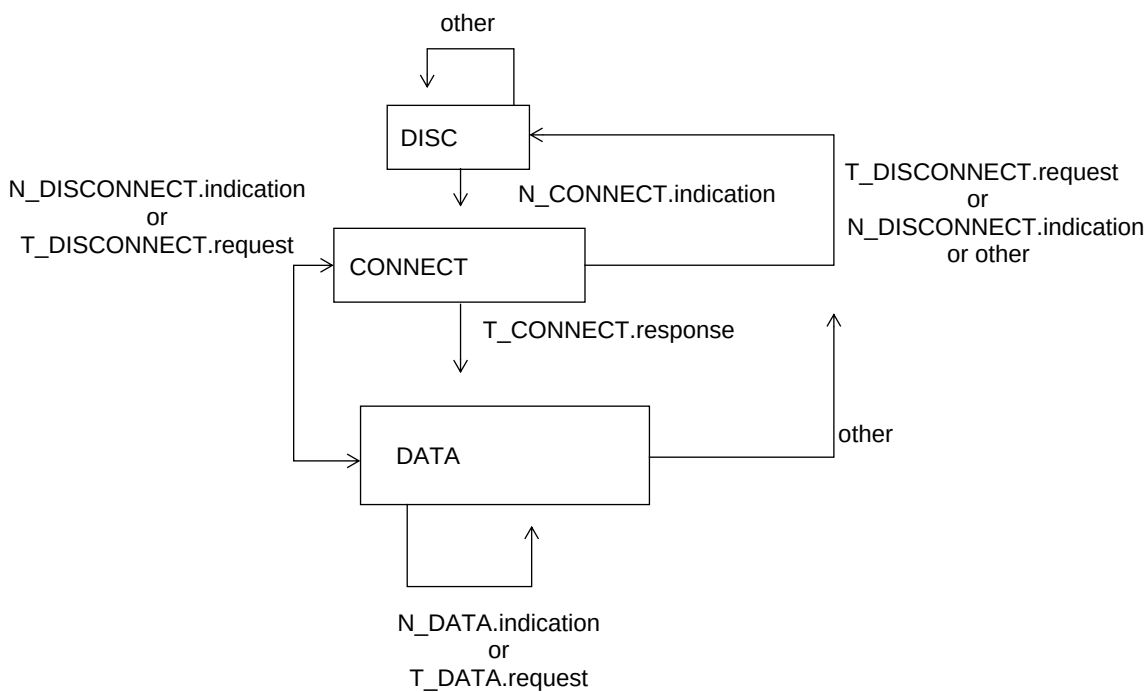


Figure 8—Inactive Transport layer BCC state diagram

Table 1—DCC Transport layer states

State	Inputs	Outputs	Next state
DISC	T_CONNECT.request	N_CONNECT.request	CONNECT
	other	T_DISCONNECT.indication	DISC
CONNECT	N_CONNECT.confirm	T_CONNECT.confirm	DATA
	T_DISCONNECT.request	N_DISCONNECT.request	DISC
	N_DISCONNECT.indication	T_DISCONNECT.indication	DISC
	other	T_DISCONNECT.indication	DISC
DATA	T_DATA.request	N_DATA.request	DATA
	N_DATA.indication and Header = 0	T_DATA.indication	DATA
	N_DATA.indication and Header $\neq$ 0	T_DISCONNECT.indication and N_DISCONNECT.request	DISC
	N_DISCONNECT.indication	T_DISCONNECT.indication	DISC
	T_DISCONNECT.request	N_DISCONNECT.request	DISC
	other	T_DISCONNECT.indication	DISC

Table 2—BCC Transport layer states

State	Inputs	Outputs	Next state
DISC	N_CONNECT.indication	T_CONNECT.indication	CONNECT
	other	T_DISCONNECT.indication	DISC
CONNECT	T_CONNECT.response	N_CONNECT.response	DATA
	T_DISCONNECT.request	N_DISCONNECT.request	DISC
	N_DISCONNECT.indication	T_DISCONNECT.indication	DISC
	other	T_DISCONNECT.indication	DISC
DATA	T_DATA.request	N_DATA.request	DATA
	N_DATA.indication and Header = 0	T_DATA.indication	DATA
	N_DATA.indication and Header $\neq$ 0	T_DISCONNECT.indication and N_DISCONNECT.request	DISC
	N_DISCONNECT.indication	T_DISCONNECT.indication	DISC
	T_DISCONNECT.request	N_DISCONNECT.request	DISC
	other	T_DISCONNECT.indication	DISC

7. Inactive Network layer

7.1 Service specification

The following subclauses specify the services offered by the MIB inactive Network layer to the network user (N_user), as viewed by the network user. These services allow an N_user layer local entity to establish a connection-oriented communications channel with a remote N_user entity and exchange packets over that channel. The services are described in an abstract way and no particular implementation is implied by this standard.

7.1.1 Overview of N_user and network interactions

7.1.1.1 Network connection phase

The N_user layer is responsible for initiating a network connection. Within the MIB, new connections are always started by the DCC requesting connection via the N_CONNECT.request service primitive. The automatic connect feature of the network is dependent upon the DCC being informed by the device that connection is desired.

The service primitives associated with network connection are

- N_CONNECT.request
- N_CONNECT.indication
- N_CONNECT.response
- N_CONNECT.confirm

The N_CONNECT.request primitive is passed to the Network (N) layer to request that a logical line connection be established between a local network service access point (NSAP) and a remote NSAP. The N_CONNECT.indication is passed from the N layer to indicate the results of an attempt by a remote entity to establish a connection to a local NSAP. The N_CONNECT.response primitive is passed to the N layer to convey the results of the previous associated N_CONNECT.indication primitive. The N_CONNECT.confirm primitive is passed from the N layer to convey the results of the previous associated N_CONNECT.request.

7.1.1.2 Network data transfer phase

All application messages are sent using the data transfer service, using the N_DATA.request primitive. This class of service is connection oriented. Data is transmitted using network service data units (NSDUs). This service is subject to flow control by higher layers.

The primitives associated with data transfer are

- N_DATA.request
- N_DATA.indication

The N_DATA.request primitive is passed to the N_layer to request that an NSDU be sent. The N_DATA.indication primitive is passed from the N layer to indicate the arrival of an NSDU.

7.1.1.3 Network disconnection phase

Either the device or the host computer can initiate a logical disconnect sequence. Disconnection is not subject to the response of a higher layer, but is treated as an automatic sequence once either end of the connection initiates it.

The primitives associated with the logical disconnect sequence are

N_DISCONNECT.request
N_DISCONNECT.indication

The N_DISCONNECT.request primitive is passed to the N layer to initiate the logical disconnect sequence. The N_DISCONNECT.indication is passed from the N layer to indicate the termination of a connection by the remote entity.

7.1.2 Detailed service specifications

The following subclauses describe 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 shall be available to the receiving entity. No specific implementation is implied by this standard.

7.1.2.1 N_CONNECT.request

7.1.2.1.1 Function

This primitive is the service request primitive for the network connection establishment process.

7.1.2.1.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
N_CONNECT.request (  
    called address,  
    calling address,  
    NSDU  
)
```

The called address is always set to 0; this parameter is not used by this service.

The calling address is always set to 0; this parameter is not used by this service.

The NSDU parameter will include the XID information field passed down from the DCC N_User entity and transmitted in the DCC's XID response frame.

7.1.2.1.3 When generated

This primitive is passed from the N_user layer to the N_layer when the N_user entity wishes to establish a transport connection to a remote NSAP.

7.1.2.1.4 Effects on receipt

The receipt of this primitive by the N layer causes the local N entity to initiate the establishment of a connection with the remote N entity.

7.1.2.1.5 Additional comments

A possible logical sequence of primitives associated with network connection is illustrated in figure 9. This service primitive is only available to the DCC.

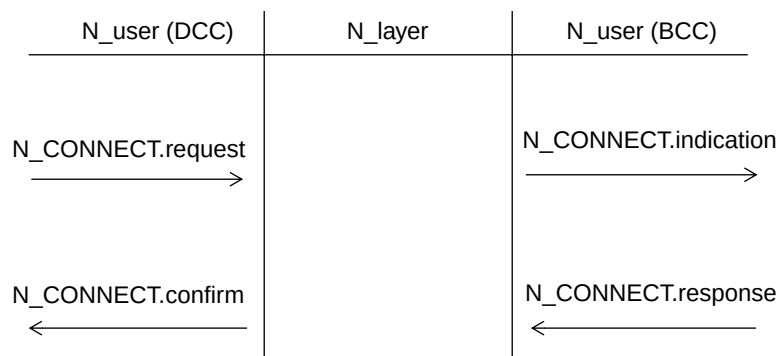


Figure 9—Timing sequence diagram for Network layer connection primitives

7.1.2.2 N_CONNECT.indication

7.1.2.2.1 Function

This primitive is the service indication primitive for the network connection process.

7.1.2.2.2 Semantics of the service primitive

The primitive shall provide the following parameters.

N_CONNECT.indication (
 called address,
 calling address,
 NSDU
)

The called address is always set to 0; this parameter is not used by the IEEE 1073.3.1 network service.

The calling address is always set to 0; this parameter is not used by the IEEE 1073.3.1 network service.

The NSDU parameter contains the XID information field that originated from the DCC.

7.1.2.2.3 When generated

This primitive is passed from the N_layer to the N_user layer to indicate that a remote N_user entity desires a connection with a local N_user entity.

7.1.2.2.4 Effects on receipt

The BCC N_user layer entity is expected, but is not required, to respond with an N_CONNECT.response primitive (accepted) or an N_DISCONNECT.request (declined) to either establish a connection with, or terminate the network connection request from, the remote N_user entity.

7.1.2.2.5 Additional comments

A possible logical sequence of primitives associated with network connection is illustrated in figure 9. This service primitive is only available to the BCC.

7.1.2.3 N_CONNECT.response

7.1.2.3.1 Function

This primitive is the service response primitive for the network connection process.

7.1.2.3.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
N_CONNECT.response (
    responding address
)
```

The responding address is always set to 0; this parameter is not used by the IEEE 1073.3.1 network service.

7.1.2.3.3 When generated

This primitive is passed from the N_user layer to the T layer to indicate that the local N_user entity accepts a connection with a remote N_user entity.

7.1.2.3.4 Effects on receipt

The local T layer entity completes the connection phase; the N_user may use the connection for data transfer.

7.1.2.3.5 Additional comments

A possible logical sequence of primitives associated with network connection is illustrated in figure 9. This service primitive is only available to the BCC.

7.1.2.4 N_CONNECT.confirm

7.1.2.4.1 Function

This primitive is the service confirm primitive for the network connection process.

7.1.2.4.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
N_CONNECT.confirm (
    responding address,
    NSDU
)
```

The responding address is always set to 0; this parameter is not used by the IEEE 1073.3.1 network service.

The NSDU will reflect the most recent value of the XID information subfield that was transmitted from the DCC to the BCC. This will correspond to the values contained in the TSDU parameter associated with the most recent N_CONNECT.request primitive that was passed from the N_User entity prior to the last transmission of an XID response frame.

7.1.2.4.3 When generated

This primitive is passed from the N_layer to the N_user layer to indicate that the remote N_user entity accepts the previous connection request service primitive from the local N_user entity.

7.1.2.4.4 Effects on receipt

The N_user layer can now use this network connection for data unit transfer.

7.1.2.4.5 Additional comments

A possible logical sequence of primitives associated with DL connection is illustrated in figure 9. This service primitive is only available to the DCC.

7.1.2.5 N_DATA.request

7.1.2.5.1 Function

This primitive is the service request primitive for the NSDU transfer service.

7.1.2.5.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
N_DATA.request (
    NSDU
)
```

The NSDU parameter specifies the NSDU to be transferred by the T layer entity.

7.1.2.5.3 When generated

This primitive is passed from the N_user layer to the N layer to request that NSDU be transferred to a remote NSAP.

7.1.2.5.4 Effects on receipt

The receipt of this primitive by the N layer causes the N layer entity to transfer the NSDU using network data transfer services.

7.1.2.5.5 Additional comments

A possible logical sequence of primitives for successful data transfer is shown in figure 10.

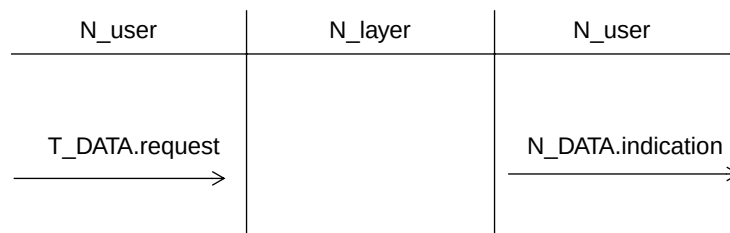


Figure 10—Timing sequence diagram for Network layer data transfer primitives

7.1.2.6 N_DATA.indication

7.1.2.6.1 Function

This primitive is the service indication primitive for the data transfer service.

7.1.2.6.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
N_DATA.indication (  
    NSDU  
)
```

The NSDU parameter specifies the network service data unit that has been received by the local N layer entity from the remote N layer entity.

7.1.2.6.3 When generated

This primitive is passed by the N layer to the N user layer to indicate the arrival of an NSDU from the specified remote N_user layer entity.

7.1.2.6.4 Effect on receipt

The effect of receipt of the primitive by the N_user layer is unspecified.

7.1.2.6.5 Additional comments

In the absence of errors, the contents of the NSDU parameter are logically complete and unchanged relative to the NSDU parameter in the associated (remote) N_DATA.request primitive. A possible logical sequence of primitives for successful data transfer is shown in figure 10.

7.1.2.7 N_DISCONNECT.request

7.1.2.7.1 Function

This primitive is the service request primitive for the disconnection service.

7.1.2.7.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
N_DISCONNECT.request (  
    )
```

7.1.2.7.3 When generated

This primitive is passed from the N_user layer to the N layer when the N_user layer wishes to terminate a connection.

7.1.2.7.4 Effects on receipt

Receipt of this primitive causes the N layer to immediately terminate the connection.

7.1.2.7.5 Additional comments

All unacknowledged NSDUs are discarded. The logical disconnect sequence is a destructive process. No guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Orderly disconnection (i.e., without loss of data) is the responsibility of a higher layer service.

A possible logical sequence of primitives associated with a successful logical disconnect sequence is illustrated in figure 11.

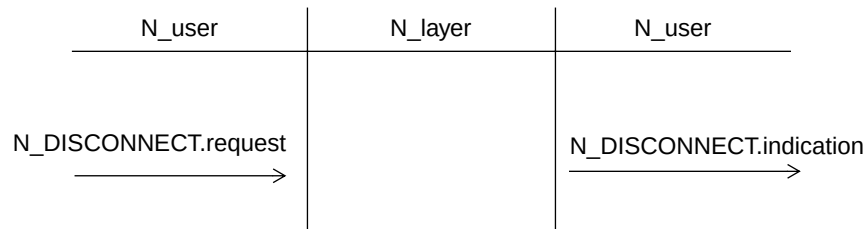


Figure 11—Timing sequence diagram for Network layer disconnect primitives

7.1.2.8 N_DISCONNECT.indication

7.1.2.8.1 Function

This primitive is the service indication for the disconnection service.

7.1.2.8.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```

N_DISCONNECT.indication (
    reason,
    NSDU
)
  
```

The reason parameter specifies the reason for the disconnection, which may include a request by a remote entity, or a condition internal to the N layer.

Except for the case of a failed DL XID bandwidth negotiation, IEEE 1073.3 does not support data transfer during the disconnection phase. In all other instances, the value of the NSDU parameter shall be empty. In the case of a failed XID bandwidth negotiation, the NSDU parameter shall contain the value of the XID information field received from the remote N_entity.

7.1.2.8.3 When generated

This primitive is passed from the N layer to the N_user layer to signal an immediate disconnection.

7.1.2.8.4 Effects on receipt

The N_user entity may no longer use the indicated connection for data transfer. Any data transmissions not acknowledged shall be deemed not received and appropriate action shall be taken.

7.1.2.8.5 Additional comments

All unacknowledged NSDUs are discarded. The logical disconnect sequence is a destructive process. No guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Orderly disconnection (i.e., without loss of data) is the responsibility of a higher layer service.

A possible logical sequence of primitives associated with a successful logical disconnect sequence is illustrated in figure 11.

7.2 PDU specification

7.2.1 General

The following subclauses define in detail the network PDU structure.

7.2.2 Network protocol data unit (NPDU) format

All NPDUs shall consist of an information field with a single-octet inactive Network layer header (see figure 12).

HEADER	INFORMATION
0	NSDU

WHERE

HEADER	= INACTIVE NETWORK LAYER HEADER
INFORMATION	= INFORMATION FIELD
NSDU	= NETWORK SERVICE DATA UNIT

Figure 12—NPDU format

7.2.3 Elements of the NPDU

7.2.3.1 Inactive Network layer header

All frames shall start with a single-octet header field that shall contain a zero value. This header is provided to allow future IEEE 1073.3 versions to support a functional Network layer by providing a nonzero header.

7.2.3.2 Information field

Information may be any sequence of octets. This field will contain the NSDU.

7.2.4 Invalid NPDU

An invalid NPDU is defined as an NPDU without a zero value in the header field. Receipt of an invalid NPDU will raise an exception to the N_user.

7.3 Mode of operation

The MIB inactive Network layer is a service-mapped layer. All primitives received from the Network layer are directly mapped onto DL layer services. The only exceptions are the N_Data primitives. These primitives will manage a single-octet header containing a zero. This octet is removed from data link protocol data units (DLPDUs) before passing the PDU to the Network layer and is added to the network PDU before passing the PDU to the DL layer. If a nonzero header is received, the connection will be aborted (see tables 3 and 4 and figures 13 and 14).

Table 3—DCC Network layer states

State	Inputs	Outputs	Next state
DISC	N_CONNECT.request	DL_CONNECT.request	CONNECT
	other	N_DISCONNECT.indication	DISC
CONNECT	DL_CONNECT.confirm	N_CONNECT.confirm	DATA
	N_DISCONNECT.request	DL_DISCONNECT.request	DISC
	DL_DISCONNECT.indication	N_DISCONNECT.indication	DISC
	other	N_DISCONNECT.indication	DISC
DATA	N_DATA.request	DL_DATA.request	DATA
	DL_DATA.indication and Header = 0	N_DATA.indication	DATA
	DL_DATA.indication and Header ≠ 0	N_DISCONNECT.indication and DL_DISCONNECT.request	DISC
	DL_DISCONNECT.indication	N_DISCONNECT.indication	DISC
	N_DISCONNECT.request	DL_DISCONNECT.request	DISC
	other	N_DISCONNECT.indication	DISC

Table 4—BCC Network layer states

State	Inputs	Outputs	Next state
DISC	DL_CONNECT.indication	N_CONNECT.indication	CONNECT
	other	N_DISCONNECT.indication	DISC
CONNECT	N_CONNECT.response	DL_CONNECT.response	DATA
	N_DISCONNECT.request	DL_DISCONNECT.request	DISC
	DL_DISCONNECT.indication	N_DISCONNECT.indication	DISC
	other	N_DISCONNECT.indication	DISC
DATA	N_DATA.request	DL_DATA.request	DATA
	DL_DATA.indication and Header = 0	N_DATA.indication	DATA
	DL_DATA.indication and Header ≠ 0	N_DISCONNECT.indication and DL_DISCONNECT.request	DISC
	DL_DISCONNECT.indication	N_DISCONNECT.indication	DISC
	N_DISCONNECT.request	DL_DISCONNECT.request	DISC
	other	N_DISCONNECT.indication	DISC

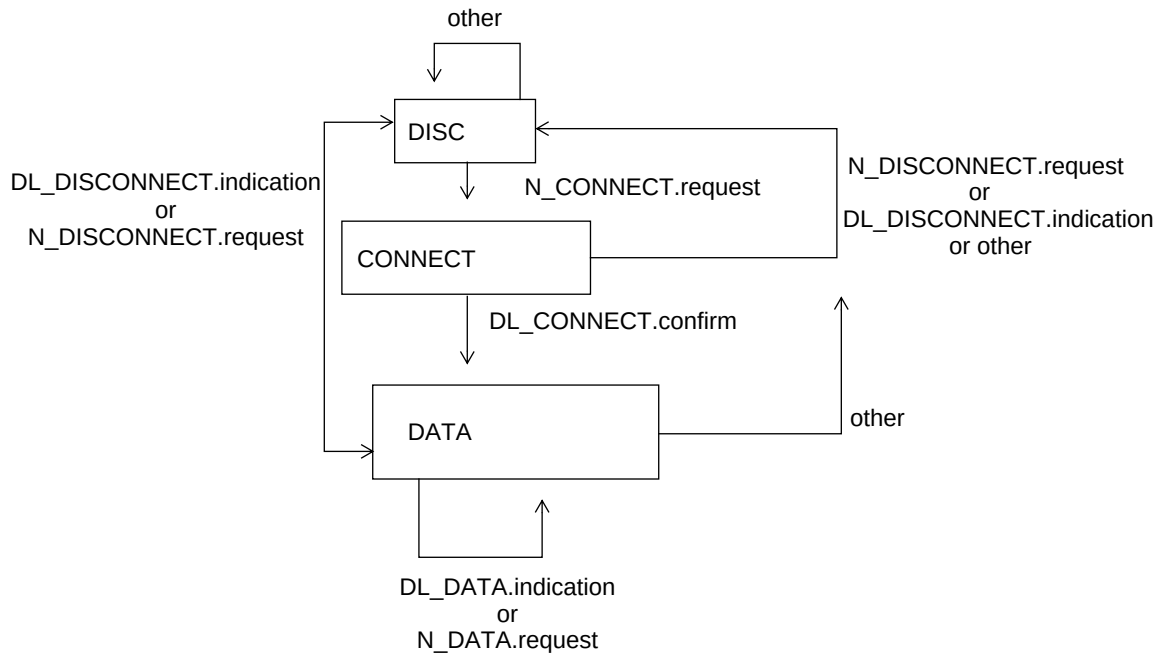


Figure 13—Inactive Network layer DCC state diagram

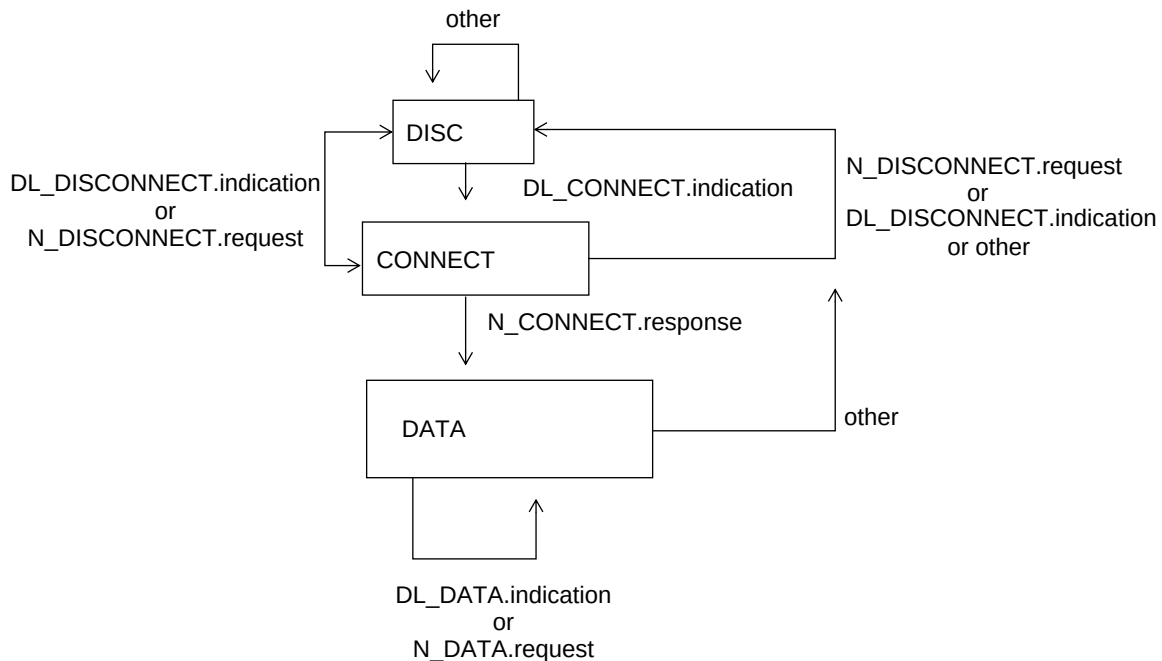


Figure 14—Inactive Network layer BCC state diagram

8. Data Link layer

8.1 Table of capabilities

Table 5 lists the required and optional capabilities for BCCs, high-speed DCCs, and low-speed DCCs.

Table 5—Table of capabilities

Capability	Requirement for BCC	Requirement for high-speed DCC	Requirement for low-speed DCC
2400 Bd low-speed operation	Required	Optional (see Note 1)	Optional (see Note 1)
9600 Bd low-speed operation	Required	Optional (see Note 1)	Optional (see Note 1)
High-speed (1 Mb/s) operation	Required	Required (see Note 1)	Optional (see Note 1)
Connection sense function	Required	Required	Optional (see Note 2)
Interrupt function	Required	Required (see Note 2)	Optional (see Note 2)
BCC-to-DCC sync function	Required	Required (see Note 2)	Optional (see Note 2)
DCC-to-BCC sync function	Required	Required (see Note 2)	Optional (see Note 2)
Date/time-of-day capability	Required	Optional	Optional
Intent to disconnect notification capability	Optional	Optional	Optional
Logical connection establishment capability	Required	Required	Required
XID negotiation capability	Required	Required	Required
Connection mode data transfer	Required	Required	Required
Logical disconnect sequencing	Required	Required	Required
Flow control	Required	Required	Required
Data link reset	Required	Required	Required
FCS data integrity checking	Required	Required	Required

NOTES

1—The DCC shall provide either 2400 Bd low-speed, 9600 Bd low-speed, and/or high-speed (1 Mb/s) operation. A DCC may provide operation at multiple data rates. However, a DCC may not change its data rate between the time of logical connection establishment and the time of physical disconnect.

2—Connection sensing, and interrupt and sync capability are required for a high-speed DCC and are optional for a low-speed DCC.

8.2 Service specification

The following subclauses specify the services offered by the DL layer to the data link user (DL_user), as viewed by the DL user. These are based closely on the Logical Link layer service specifications, as defined by ISO/IEC 8802-2: 1994. These services allow a DL_user layer local entity to establish a connection-oriented communications channel with a remote DL_user entity and exchange packets over that channel. The services also support connectionless data transfers over the DL. The services are described in an abstract way and no particular implementation is implied by this standard.

8.2.1 Overview of DL_user and DL interactions

8.2.1.1 DL connection phase

The DL_user layer is responsible for initiating DL connection. The BCC is notified when a DCC has been physically connected to a particular port by receiving a PH_CONNECT.indication primitive from the Physical layer. Within the MIB, new DL connections are always initiated by the DCC requesting connection via the DL_CONNECT.request service primitive. The automatic connect feature of the network is dependent upon the DCC being informed by the device that connection is desired.

The service primitives associated with data link connection are

- DL_CONNECT.request
- DL_CONNECT.indication
- DL_CONNECT.response
- DL_CONNECT.confirm

The DL_CONNECT.request primitive is passed to the DL layer to request that a logical link connection be established between a local data link service access point (DLSAP) and a remote DLSAP. The DL_CONNECT.indication is passed from the BCC DL layer to indicate the results of an attempt by a remote entity to establish a connection to a local DLSAP. The DL_CONNECT.response primitive is passed to the BCC DL layer to convey the results of the previous associated DL_CONNECT.indication primitive. The DL_CONNECT.confirm primitive is passed from the DCC DL layer to convey the results of the previous associated DL_CONNECT.request.

8.2.1.2 DL data transfer phase

All application messages are sent using the data transfer service, using the DL_DATA.request, DL_DATA.indication, and DL_DATA.confirm primitives. This class of service is connection oriented and provider confirmed. In addition, this service is subject to flow control by higher layers.

The DL_entity shall implement unnumbered frames. The unnumbered frames are initiated and reported by means of the DL_UNITDATA primitives. The DL_UNITDATA primitives provide an unconfirmed service.

The primitives associated with data transfer are

- DL_DATA.request
- DL_DATA.indication
- DL_DATA.confirm
- DL_UNITDATA.request
- DL_UNITDATA.indication

The DL_DATA.request primitive is passed to the DL layer to request that a data link service data unit (DLSDU) be sent. The DL_DATA.indication primitive is passed from the DL layer to indicate the arrival of a validated DLSDU. The DL_DATA.confirm primitive provides confirmation to the DL_user layer that a frame transmitted by a BCC or a DCC was correctly received by the other station. The DL_UNITDATA.request primitive is a request from the DL_user layer to transmit an unnumbered frame, such as an XID frame by the BCC, or a UI frame, as part of an interrupt request sequence, by the DCC. The DL_UNITDATA.indication primitive provides indication to the DL_user layer that a valid unnumbered frame has been received.

8.2.1.3 Data connection flow control

The DL_user entity informs the DL_entity whether or not it is capable of receiving information frames. Similarly, the DL entity has the responsibility of informing the DL_user entity if the other station is capable of receiving information frames.

The primitives associated with data connection flow control are

DL_CONNECTION_FLOWCONTROL.request
DL_CONNECTION_FLOWCONTROL.indication

DL_CONNECTION_FLOWCONTROL.request is used by the DL_user layer to inform the DL_entity whether or not it is ready to receive information frames. The DL_CONNECTION_FLOWCONTROL.indication primitive is issued to the DL_user layer to inform the DL_user entity whether or not the remote station (BCC or DCC) can receive information frames. Both primitives include source address, destination address, and amount primitives.

The port addresses 1–125 are used as the port addresses for individual DCCs. The BCC has an assigned port address of 126. The port address 0 is the HDLC “no station” address, and is used in the response frames from DCCs that have not been assigned port addresses during connection establishment sequences. The port address 127 is the HDLC broadcast address and is transmitted by the BCC in unnumbered poll (UP) frames during connection establishment and interrupt acknowledge sequences.

An amount parameter of zero denotes a “receiver_not_ready” or “remote_not_ready” condition. An amount parameter of greater than zero denotes a “receiver_ready” or “remote_ready” condition.

8.2.1.4 DL disconnection phase

Either the DCC or the BCC can initiate a disconnect. Disconnection is not subject to the response of a higher layer, but is treated as an automatic sequence once either end of the connection initiates it.

The primitives associated with a logical disconnect sequence are

DL_DISCONNECT.request
DL_DISCONNECT.indication

The DL_DISCONNECT.request is passed to the DL layer to initiate the disconnect sequence. The DL_DISCONNECT.indication is passed from the DL layer to indicate the termination of a connection by the remote entity. The DL_DISCONNECT.indication primitive includes a reason parameter. The possible reasons for a logical disconnect include intentional physical disconnection, unintentional physical disconnection, excessive protocol errors, excessive time-out errors, and others.

8.2.1.5 Reset function

The DL entity shall implement a provider-confirmed reset function. The reset sequence may be originated by either the BCC or the DCC.

The primitives associated with the reset function are

DL_RESET.request
DL_RESET.indication
DL_RESET.confirm

DL_RESET.request is issued by the DL_user layer. When received by the BCC, the BCC shall initiate a reset sequence by transmitting a set initialization mode (SIM) frame. Reception of this frame results in a DL_RESET.indication being issued by the DCC DL_entity. When the BCC receives the DCC's unnumbered acknowledge (UA) response frame, its DL_entity issues a DL_RESET.confirm to its DL_user entity.

When the DCC DL_user layer transmits a DL_RESET.request to the DL_entity, the DCC shall respond with a request initialization mode (RIM) response frame at the earliest opportunity. The BCC shall then transmit a SIM command frame to the DCC during its next poll cycle. Upon reception of the SIM frame, the DCC DL_entity shall transmit a UA frame to the BCC and issue a DL_RESET.confirm primitive to the DL_user layer.

Note that following a reset command sequence, the BCC shall transmit an XID command frame to the DCC. Therefore, the reset primitives may be used as a mechanism to initiate a renegotiation of the polling parameters Tx-Length, Rx-Length, and polling rate by either the BCC or the DCC.

8.2.2 Detailed service specifications

The following subclauses describe 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 shall be available to the receiving entity. No specific implementation is implied by this standard.

8.2.2.1 DL_CONNECT.request

8.2.2.1.1 Function

This primitive is the service request primitive for the connection-oriented link establishment process.

8.2.2.1.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
DL_CONNECT.request (
    DLSDU
)
```

The DLSDU parameter contains the contents of the XID information field for the response frame to be transmitted by the DCC during its connection establishment sequence. This includes the bandwidth negotiation variables Tx-Length, Rx-Length, and polling period.

8.2.2.1.3 When generated

This primitive is passed from the DL_user layer to the DL layer when the DL_user entity wishes to establish a logical link connection to a remote DLSAP.

8.2.2.1.4 Effects on receipt

The receipt of this primitive by the DL layer causes the local DL entity to initiate the establishment of a connection with the remote DL entity.

8.2.2.1.5 Additional comments

A possible logical sequence of primitives associated with DL connection is illustrated in figure 15. This service primitive is only available to the DCC.

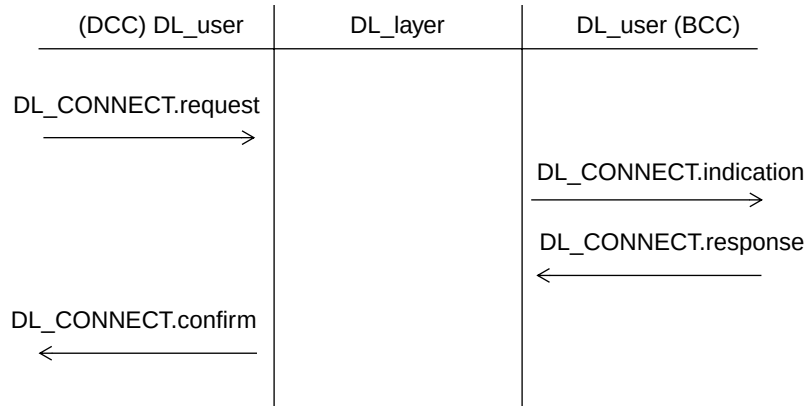


Figure 15—Timing sequence diagram DL layer connection primitives

8.2.2.2 DL_CONNECT.indication

8.2.2.2.1 Function

This primitive is the service indication primitive for the DL connection process.

8.2.2.2.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```

DL_CONNECT.indication(
    DLSDU
)
    
```

The DLSDU parameter contains the contents of the last XID frame received from the DCC during its connection establishment sequence. This includes the bandwidth negotiation variables Tx-Length, Rx-Length, and polling period.

8.2.2.2.3 When generated

This primitive is passed from the DL layer to the DL_user layer to indicate that a remote DL_user entity desires a connection with a local DL_user entity.

8.2.2.2.4 Effects on receipt

The DL_user layer entity will respond with a DL_CONNECT.response primitive to accept, or a DL_DISCONNECT.request primitive to reject, the connection with the remote DL_user entity.

8.2.2.2.5 Additional comments

A possible logical sequence of primitives associated with DL connection is illustrated in figure 15. This service primitive is only available to the BCC.

8.2.2.3 DL_CONNECT.response

8.2.2.3.1 Function

This primitive is the service response primitive for the DL connection process.

8.2.2.3.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
DL_CONNECT.response (  
    )
```

8.2.2.3.3 When generated

This primitive is passed from the DL_user layer to the DL layer to indicate that the local DL_user entity accepts a connection with a remote DL_user entity.

8.2.2.3.4 Effects on receipt

The local DL layer completes the connection.

8.2.2.3.5 Additional comments

A possible logical sequence of primitives associated with DL connection is illustrated in figure 15. This service primitive is only available to the BCC.

8.2.2.4 DL_CONNECT.confirm

8.2.2.4.1 Function

This primitive is the service confirm primitive for the DL connection process.

8.2.2.4.2 Semantics of the service primitive

The primitive shall provide the following parameter.

```
DL_CONNECT.confirm (  
    DLSDU  
    )
```

The DLSDU will reflect the most recent contents of the last XID frame transmitted by the DCC to the BCC during the connection establishment sequence. This includes the bandwidth negotiation parameters Tx-Length, Rx-Length, and polling period. These will correspond to the values contained in the DLSDU parameter associated with the most recent DL_CONNECT.request primitive that was passed from the DL_user entity prior to the last transmission of an XID response frame.

8.2.2.4.3 When generated

This primitive is passed from the DL layer to the DL_user layer to indicate that the remote DL_user entity accepts the previous connection request service primitive from the local DL_user entity.

8.2.2.4.4 Effects on receipt

The DL_user layer can now use this network connection for data transfer.

8.2.2.4.5 Additional comments

A possible logical sequence of primitives associated with DL connection is illustrated in figure 15. This service primitive is only available to the DCC.

8.2.2.5 DL_DATA.request

8.2.2.5.1 Function

This primitive is the service request primitive for the data transfer service.

8.2.2.5.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
DL_DATA.request (  
    source address,  
    destination address,  
    DLSDU  
)
```

The source address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the remote DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The DLSDU parameter specifies the DLSDU to be transferred from the local DL layer entity to the remote DL layer entity.

8.2.2.5.3 When generated

This primitive is passed from the DL_user layer to the DL layer to request that a DLSDU be transferred to a remote DLSAP.

8.2.2.5.4 Effects on receipt

The receipt of this primitive by the DL layer causes the local DL layer entity to transfer the DLSDU to the remote DL layer entity.

8.2.2.5.5 Additional comments

A possible logical sequence of primitives in successful data transfer is illustrated in figure 16a.

8.2.2.6 DL_DATA.indication

8.2.2.6.1 Function

This primitive is the service indication primitive for the data transfer service.

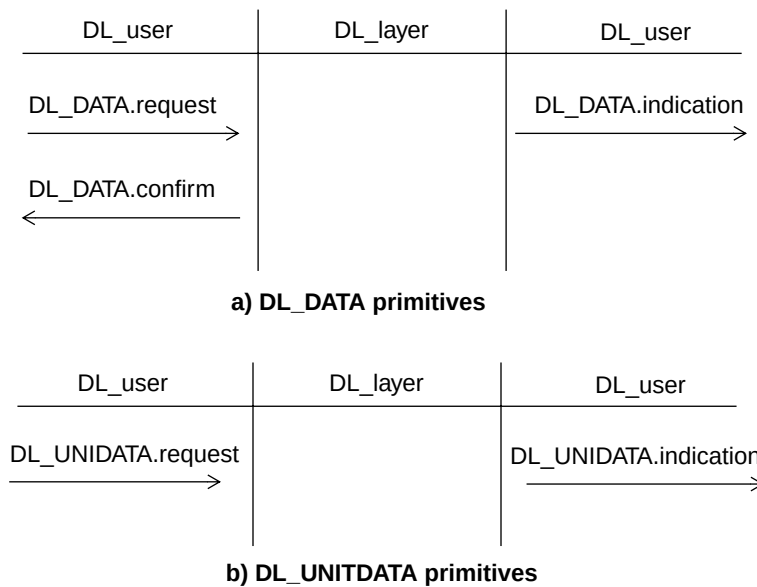


Figure 16—Timing sequence diagram DL layer data transfer primitives

8.2.2.6.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
DL_DATA.indication(
    source address,
    destination address,
    DLSDU
)
```

The source address identifies the port address of the remote DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The DLSDU parameter specifies the DLSDU that has been transferred from the remote DL layer entity to the local DL layer entity.

8.2.2.6.3 When generated

This primitive is passed by the DL layer to the DL_user layer to indicate the arrival of a valid DLSDU from the specified remote DL_user layer entity.

8.2.2.6.4 Effects on receipt

The effect of receipt of the primitive by the DL_user layer is unspecified.

8.2.2.6.5 Additional comments

In the absence of errors, the contents of the DLSDU parameter are logically complete and unchanged relative to the DLSDU parameter in the associated (remote) DL_DATA.request primitive. A possible logical sequence of primitives in successful data transfer is illustrated in figure 16a.

8.2.2.7 DL_DATA.confirm

8.2.2.7.1 Function

This primitive is the service-provider-confirmed service primitive for the connection-oriented data transfer service.

8.2.2.7.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
DL_DATA.confirm (  
    source address,  
    destination address  
)
```

The source address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the remote DL entity for the confirmed DLSDU. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

8.2.2.7.3 When generated

This primitive is passed by the DL layer to the DL_user layer when the local transmitting DL entity has confirmed correct reception of a previously transmitted frame by the local DL entity. Correct reception of a previously transmitted frame is determined by reception of a correct receive sequence number N(R) by the local DL entity.

8.2.2.7.4 Effects on receipt

The effect of receipt of the primitive by the DL_user layer is unspecified.

8.2.2.7.5 Additional comments

A possible logical sequence of primitives in successful data transfer is illustrated in figure 16a.

8.2.2.8 DL_UNITDATA.request

8.2.2.8.1 Function

This primitive is the service request primitive for connectionless (unnumbered frame) transfer service.

8.2.2.8.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
DL_UNITDATA.request (
    source address,
    destination address
    DLSDU
)
```

The source address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the remote DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The DLSDU parameter specifies the DLSDU to be transferred from the local DL layer entity to the remote DL layer entity.

8.2.2.8.3 When generated

This primitive is passed from the DL_user layer to the DL layer to request that a DLSDU be transferred to a remote DLSAP by means of an unnumbered frame.

8.2.2.8.4 Effects on receipt

The receipt of this primitive by the DL layer causes the DL layer entity to transfer the DLSDU by means of an unnumbered frame.

8.2.2.8.5 Additional comments

The DL_UNITDATA.request primitive may be used by the DCC to initiate an interrupt request sequence. The DL_UNITDATA.request primitive may be used by the BCC to initiate a sequence of XID frames.

A possible logical sequence of primitives in successful data transfer by means of an unnumbered frame is illustrated in figure 16b.

8.2.2.9 DL_UNITDATA.indication

8.2.2.9.1 Function

This primitive is the service indication primitive for the unnumbered data transfer service.

8.2.2.9.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
DL_UNITDATA.indication (
    source address,
    destination address,
    DLSDU
)
```

The source address identifies the port address of the remote DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The DLSDU parameter specifies the DLSDU that has been transferred from the remote DL layer entity to the local DL layer entity.

8.2.2.9.3 When generated

This primitive is passed by the DL layer to the DL_user layer to indicate the arrival of a valid DLSDU (unnumbered frame) from the specified remote DL_user layer entity.

8.2.2.9.4 Effects on receipt

The effect of receipt of this primitive by the DL_user layer is unspecified.

8.2.2.9.5 Additional comments

In the absence of errors, the contents of the DLSDU parameter are logically complete and unchanged relative to the DLSDU parameter in the associated (remote) DL_DATA.request primitive. A possible logical sequence of primitives in successful data transfer is illustrated in figure 16b.

8.2.2.10 DL_CONNECTION_FLOWCONTROL.request

8.2.2.10.1 Function

This primitive is the service request primitive for DL flow control.

8.2.2.10.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
DL_CONNECTION_FLOWCONTROL.request(  
    source_address,  
    destination_address,  
    amount  
)
```

The source address identifies the port address of the remote DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The amount parameter specifies whether or not the DL entity can pass data to the DL_user layer. If the value of the amount parameter is greater than zero, the DL_user entity is able to accept data passed from the DL entity. If the value of the amount parameter is equal to zero, the DL_user entity is not able to accept data passed from the DL entity.

8.2.2.10.3 When generated

This primitive is passed from the DL_user layer to the DL layer when the DL_user layer needs to enable or disable the reception of information (I) frames, based on available buffer space or other resources.

8.2.2.10.4 Effect on receipt

Receipt of this primitive causes the DL layer to set or clear its receive_ready Boolean variable. Receive_ready is set to true if the amount parameter is greater than zero, and cleared to false if the amount parameter is equal to zero.

8.2.2.10.5 Additional comments

A possible logical sequence of primitives associated with flow control is illustrated in figure 17.



Figure 17—Timing sequence diagram for DL layer flow control primitives

8.2.2.11 DL_CONNECTION_FLOWCONTROL.indication

8.2.2.11.1 Function

This primitive is the service indication primitive for DL flow control.

8.2.2.11.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
DL_CONNECTION_FLOWCONTROL.indication(
    source_address,
    destination_address,
    amount
)
```

The source address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the remote DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The amount parameter specifies whether or not the DL entity is able to accept data from the DL_user layer. If the value of the amount parameter is greater than zero, the DL entity is indicating that it is able to accept data passed from the DL_user entity. If the value of the amount parameter is equal to zero, the DL entity is indicating that it is not able to accept data passed from the DL_user entity.

8.2.2.11.3 When generated

This primitive is passed from the DL layer to the DL_user layer after reception of a numbered frame with the poll/final (P/F) bit set to one from the remote DL_entity. If the last frame received was an RR frame, or an I frame with the P/F bit set to one, the amount parameter will be greater than zero, indicating that the remote DL_entity is able to receive information frames (remote_ready = true). If the last frame received was an RNR frame, the amount parameter will be equal to zero, indicating that the remote DL_entity is not currently able to receive information frames (remote_ready = false).

8.2.2.11.4 Effect on receipt

The effect of receipt of this primitive by the DL_user is unspecified.

8.2.2.11.5 Additional comments

The value of the local DL_entity's remote_ready Boolean variable will be true if the remote DL_entity is able to receive information frames, and false if the remote DL_entity is not able to receive information frames.

A possible logical sequence of primitives associated with flow control is illustrated in figure 17.

8.2.2.12 DL_DISCONNECT.request

8.2.2.12.1 Function

This primitive is the service request primitive for the logical disconnect sequence.

8.2.2.12.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```
DL_DISCONNECT.request (  
    source_address,  
    destination_address  
)
```

The source address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the remote DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

8.2.2.12.3 When generated

This primitive is passed from the DL_user layer to the DL layer when the DL_user layer wishes to terminate a connection.

8.2.2.12.4 Effect on receipt

Receipt of this primitive causes the DL layer to terminate the connection.

8.2.2.12.5 Additional comments

All unacknowledged DLSDUs are discarded. The logical disconnect sequence is a destructive process. No guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Orderly disconnection (i.e., without loss of data) is the responsibility of a higher layer service.

A possible logical sequence of primitives associated with a successful logical disconnect sequence is illustrated in figure 18.



Figure 18—Timing sequence diagram DL layer disconnection primitives

8.2.2.13 DL_DISCONNECT.indication

8.2.2.13.1 Function

This primitive is the service indication primitive for the logical disconnect sequence.

8.2.2.13.2 Semantics of the service primitive

The primitive shall provide the following parameters.

```

DL_DISCONNECT.indication(
    source_address,
    destination_address,
    reason,
    DLSDU
)
  
```

The source address identifies the port address of the local DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The destination address identifies the port address of the remote DL entity. The port address is between 1 and 125 for the DCC, and is designated as 126 for the BCC.

The reason parameter specifies the reason for the disconnection, which may include an intentional or unintentional physical disconnect, a request by the remote entity, or an error or other condition internal to the DL layer.

Except for the case of a failed DL XID bandwidth negotiation, IEEE 1073.3 does not support data transfer during the disconnection phase. In all other instances, the value of the DLSDU parameter shall be empty. In the case of a failed XID bandwidth negotiation, the DLSDU parameter shall contain the value of the most recent XID information field received from the remote DL_entity.

8.2.2.13.3 When generated

This primitive is passed from the DL layer to signal an immediate disconnect.