

IEEE Standard for Medical Device Communications—Transport Profile—Connection Mode— Amendment 1: Corrections and Clarifications

Sponsor

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

Approved 21 September 2000

IEEE-SA Standards Board

Abstract: This amendment sets forth a number of clarifications and corrections to IEEE Standard for Medical Device Communications—Transport Profile—Connection Mode.

Keywords: bedside communications controller (BCC), bedside medical device, Data Link layer, device communications controller (DCC), flow control, frame, 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.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2000 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 23 October 2000. Printed in the United States of America.

Print: ISBN 0-7381-2613-6 SH94879
PDF: ISBN 0-7381-2614-4 SS94879

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

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. 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 societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

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

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

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

IEEE is the sole entity that may authorize the use of certification marks, trademarks, or other designations to indicate compliance with the materials set forth herein.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

(This introduction is not part of IEEE Std 1073.3.1a-2000, IEEE Standard for Medical Device Communications—Transport Profile—Connection Mode—Amendment 1: Corrections and Clarifications.)

This amendment provides corrections and clarifications to IEEE Std 1073.3.1-1994. This is the first amendment to IEEE Std 1073.3.1-1994.

IEEE Std 1073.3.1-1994 defines a communications transport profile. This profile is for a local area network (LAN) for the interconnection of computers and medical devices. Specifically, IEEE 1073.3.1-1994 describes connection-oriented communications services and protocols.

IEEE Std 1073.3.1-1994 and IEEE Std 1073.3.1a-2000 are part of the family of IEEE 1073 standards. They are compatible with the upper layer standards (see IEEE P1073.1 and IEEE P1073.2).

The primary users of IEEE Std 1073.3.1a-2000 are technical personnel who are creating or interfacing with a medical device communications system. Familiarity with IEEE 1073 family of standards is recommended. Familiarity with communications and networking technologies is also recommended.

The corrections and clarifications in this amendment were identified during early implementations of IEEE Std 1073.3.1-1994.

In addition to editorial changes for punctuation, spelling, labeling, and typographical errors, this amendment makes corrections to the following areas:

- The maximum number of data bytes
- Bit numbering
- HDLC field formats
- XID values for time of day and options
- Frame check validation
- Error counting
- Responsibilities of BCC vs. DCC
- Connection establishment logic
- Frame delimiting
- Service parameters
- State tables
- Source and destination addresses
- Validity checking criteria for XID frames
- Private parameter byte ordering
- Initialization
- Physical layer events

The amendment provides clarification for

- Term definitions and use
- Operations performed during connection establishment
- Bit order
- Unused XID parameters
- Use of Unnumbered Information frames
- Use of the data indication primitive
- Operation of date/time-of-day and data link timers

- Disconnect handling
- The soft ID parameter
- Minimum response time

Some features are made optional with this amendment:

- The model number extension parameter
- Capability for a BCC to notify regarding the intent to disconnect

Unused or incorrect elements of IEEE Std 1073.3.1-1994 are removed:

- XID user data
- Erroneous event entries
- References to interrupts for a DCC without interrupts

IEEE Std 1073.3.1a-2000 also replaces references to national standards with the international versions, where available.

Participants

At the time this amendment was completed, the working group had the following membership:

Robert J. Kennelly, *Chair*

Tom Canup
Jim Cominsky

Todd Cooper
Kenneth J. Fuchs
Mike Glass

Bill Hawley
Mark Poler

Other individuals who have contributed to this amendment include:

Allen Farquhar

Carol Ann Pellegrini

David Walson

The following members of the balloting committee voted on this amendment:

Grace Esche
Allen Farquhar
Kenneth J. Fuchs
Robert J. Kennelly

S. H. Meij
Melvin Reynolds
Gary S. Robinson

M. Michael Shabot
Charles Sidebottom
Lars Steubesand
Monica Vago

When the IEEE-SA Standards Board approved this amendment on 21 September 2000, it had the following membership:

Donald N. Heirman, *Chair*

James T. Carlo, *Vice Chair*

Judith Gorman, *Secretary*

Satish K. Aggarwal
Mark D. Bowman
Gary R. Engmann
Harold E. Epstein
H. Landis Floyd
Jay Forster*
Howard M. Frazier
Ruben D. Garzon

James H. Gurney
Richard J. Holleman
Lowell G. Johnson
Robert J. Kennelly
Joseph L. Koepfinger*
Peter H. Lips
L. Bruce McClung
Daleep C. Mohla

James W. Moore
Robert F. Munzner
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Akio Tojo
Donald W. Zipse

*Member Emeritus

Also included is the following nonvoting IEEE-SA Standards Board liaison:

Alan Cookson, *NIST Representative*

Donald R. Volzka, *TAB Representative*

Don Messina

IEEE Standards Project Editor

Contents

1.	Overview	1
1.1	Scope	1
1.2	Purpose	2
1.3	Standards compatibility	2
2.	References	2
3.	Definitions	3
5.	Communications nodes on the subnet	3
5.2	Bedside communications controller (BCC)	3
5.3	Device communications controller (DCC)	3
6.	Inactive Transport layer	4
6.1	Service specification	4
6.1.1	Overview of T_user and transport interactions	9
6.1.2	Detailed service specifications	9
6.2	PDU specification	9
6.3	Mode of operation	9
7.	Inactive Network layer	11
7.1	Service specification	11
7.1.2	Service specification	11
8.	Data Link layer	12
8.2	Service specification	12
8.2.1	Overflow of DL_user and DL interactions	12
8.2.2	Detailed service specifications	13
8.3	PDU specification	18
8.3.2	DLPDU format	18
8.3.4	Flag	18
8.3.5	Invalid DLPDU	18
8.4	DL elements of procedures	19
8.4.2	Control field formats	19
8.4.3	Control field parameters	20
8.5	DL procedure descriptions	23
8.5.6	Procedures for information transfer	23
8.5.8	Timer functions	23
8.6	Interrupt requests and acknowledgments	24
8.6.1	DCC interrupt request function	24
8.7	Formal description of procedures	24
8.7.2	State event tables	24

9. Labelling and conformance requirements.....	109
9.3 Conformance requirements.....	109
9.3.1 BCC statement of conformance table	109
9.3.2 DCC statement of conformance table.....	109
Annex A (normative) Systems and data link management	110
A.1 Introduction.....	110
A.5 Management DL addressing	110
A.11 Parameter format.....	110
Annex B (informative) High-level data link control (HDLC) field formats.....	111
Annex C (informative) Cyclic redundancy checking—implementation notes for the frame checking sequence (FCS)	114

IEEE Standard for Medical Device Communications—Transport Profile—Connection Mode— Amendment 1: Corrections and Clarifications

NOTE—The editing instructions contained in this amendment define how to merge the material contained herein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in ***bold italic***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make small corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~striketrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make large changes in existing text, subclauses, tables, or figures by removing existing material and replacing it with new material. Editorial notes will not be carried over into future editions because the changes will be incorporated into the base standard.

1. Overview

1.1 Scope

Change item a) in 1.1 to the following:

- a) Uses ~~EIA-485~~ ISO/IEC 8482:1993 for data transfer.

Change item c) in 1.1 to the following:

- c) Provides a pair of special function signals for the purposes of providing device connection indications, interrupt requests, and time synchronization, ~~and frame delimiting.~~

1.2 Purpose

Replace figure 1 in 1.2 with the following:

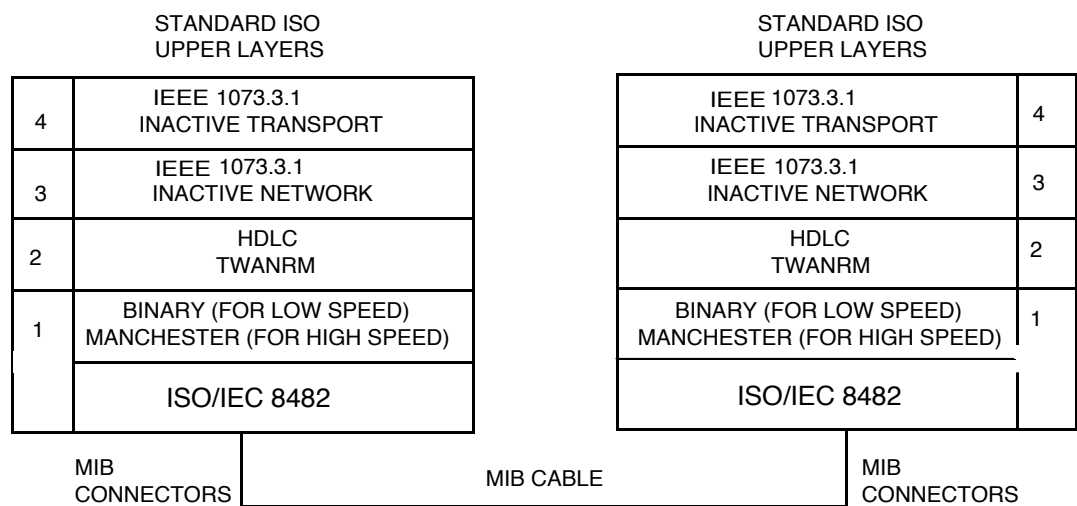


Figure 1—OSI layer representation of IEEE Std 1073.3.1-1994

1.3 Standards compatibility

Change the last sentence in 1.3 to the following:

The Physical layer media specification and data transfer coding are called out in IEEE Std 1073.4.1-1994 and ~~EIA-485 (1983)~~ ISO/IEC 8482:1993.

2. References

Delete the following reference in Clause 2 and its associated footnote (Footnote 2), then renumber the remaining footnotes:

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

Insert the following references in Clause 2, in alphabetical order:

ISO/IEC 8482:1993, Information technology—Telecommunications and information exchange between systems—Twisted pair multipoint interconnections.

ISO/IEC 8802-1:1994, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 1: Overview of Local Area Network Standards.

3. Definitions

Replace the following definitions in Clause 3:

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.)

3.46 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, it is required that there 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 provided to MIB connector (BCC only), +12 V used from MIB connector (DCC only), BCC capability, DCC capability, DCC connection determination sensing, capability for DCC interrupt function, capability for DCC sync reception, capability for DCC sync transmission, and standard version/revision.

3.48 physically connected: The state when a BCC port is physically connected to a DCC via an interconnecting MIB cable. This is indicated to both the DCC and the BCC by voltage levels.

3.50 physical protocol service unit (PhSDU): A group of octets that are transferred between a Ph_user entity and a Ph_entity. In this standard, the Ph_user entity is the data link entity.

Delete definition 3.47 and renumber the remaining definitions in Clause 3.

5. Communications nodes on the subnet

5.2 Bedside communications controller (BCC)

Change the first sentence in 5.2 to the following:

The BCC is the ~~subnet~~ communication controller for the ~~network at the bedside~~ subnet.

5.3 Device communications controller (DCC)

Replace the text in 5.3 with the following:

The DCC is the interface for each device onto the MIB network at the bedside. Logical connections are initiated by the DCC. Following connection establishment, the DCC normally communicates by responding to periodic polls by the BCC. A DCC may initiate out of band communication by issuing an interrupt request to the 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

Replace the first paragraph in 6.1 with the following:

The following subclauses, 6.1.1 and 6.1.2, specify the services offered by the MIB inactive transport entity to the transport user (T_user), as viewed by the transport user. These services allow a local T_user entity to establish a connection-oriented communications channel with a remote T_user entity and exchange packets over the channel. 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

Replace the text in 6.1.1.1 with the following:

The DCC 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 directed 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 DCC transport (T) entity 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 BCC T_entity to the BCC T_user entity 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 from the BCC T_user entity to the BCC T_entity to convey the results of the previous associated T_CONNECT.indication primitive. The T_CONNECT.confirm primitive is passed from the DCC T_entity to the DCC T_user entity to convey the results of the previous associated T_CONNECT.request.

6.1.1.2 Transport data transfer phase

Change the first sentence in 6.1.1.2 as follows:

All application messages are sent using ~~data transfer service, using~~ the T_Data request primitive.

6.1.2 Detailed service specifications

6.1.2.1 T_CONNECT.request

6.1.2.1.2 Semantics of the service primitive

Replace the text in 6.1.2.1.2 with the following:

The primitive shall provide the following parameters.

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

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

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

6.1.2.1.3 When generated

Replace the text in 6.1.2.1.3 with the following:

This primitive is issued by the DCC T_user entity to the T_entity in order to establish a transport connection with a remote TSAP.

6.1.2.2 T_CONNECT.indication

6.1.2.2.2 Semantics of the service primitive

Replace the text in 6.1.2.2.2 with the following:

The primitive shall provide the following parameters.

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

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

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

6.1.2.2.3 When generated

Replace the text in 6.1.2.2.3 with the following:

This primitive is passed from the T_entity to the T_user entity 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

Replace the text in 6.1.2.2.4 with the following:

The BCC T_user entity shall either accept the connection to the DCC T_user entity, by issuing a T_CONNECT.response; or reject the connection, by issuing a T_DISCONNECT.request.

6.1.2.2.5 Additional comments

Replace the text in 6.1.2.2.5 with the following:

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

6.1.2.3 T_CONNECT.response

6.1.2.3.3 When generated

Replace the text in 6.1.2.3.3 with the following:

The primitive is passed from the T_user entity to the T_entity 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

Replace the text in 6.1.2.3.4 with the following:

The local T_entity completes the connection phase; the T_user may use the connection for data transfer.

6.1.2.4 T_CONNECT.confirm

6.1.2.4.2 Semantics of the service primitive

Replace the text in 6.1.2.4.2 with the following:

The primitive shall provide the following parameters.

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

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

6.1.2.4.3 When generated

Replace the text in 6.1.2.4.3 with the following:

This primitive is passed from the T_entity to the T_user entity 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

Replace the text in 6.1.2.4.4 with the following:

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

6.1.2.5 T-DATA.request

Replace the head for 6.1.2.5 with the following:

6.1.2.5 T_DATA.request**6.1.2.5.2 Semantics of the service primitive**

Replace the last sentence in 6.1.2.5.2 with the following:

The TSDU parameter specifies the TSDU to be transferred by the T_entity.

6.1.2.5.3 When generated

Replace the text in 6.1.2.5.3 with the following:

This primitive is passed from the T_user entity to the T_entity to request that TSDU be transferred to a remote TSAP.

6.1.2.5.4 Effects on receipt

Replace the text in 6.1.2.5.4 with the following:

The receipt of this primitive by the T_entity causes the T_entity to transfer the TSDU using transport data transfer services.

6.1.2.5.5 Additional comments

Replace the text and figure in 6.1.2.5.5 with the following:

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

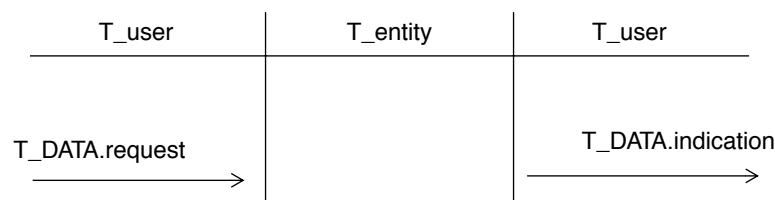


Figure 4—Timing sequence diagram for Transport layer data transfer primitives

6.1.2.6 T_DATA.indication**6.1.2.6.2 Semantics of the service primitive**

Replace the last sentence in 6.1.2.6.2 with the following:

The TSDU parameter specifies the transport service data unit that has been received by the local T_entity from the remote T_entity.

6.1.2.6.3 When generated

Replace the text in 6.1.2.6.3 with the following:

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

6.1.2.6.4 Effect on receipt

Replace the text in 6.1.2.6.4 with the following:

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

6.1.2.7 T_DISCONNECT.request

6.1.2.7.2 Semantics of the service primitive

Replace the text in 6.1.2.7.2 with the following:

The primitive shall provide the following parameter.

```
T_DISCONNECT.request(  
    )
```

6.1.2.7.3 When generated

Replace the text in 6.1.2.7.3 with the following:

This primitive is passed from the T_user entity to the T_entity when the T_user entity wishes to terminate a connection.

6.1.2.7.4 Effects on receipt

Replace the text in 6.1.2.7.4 with the following:

This primitive is passed by the T_user entity to the T_entity to be transmitted to the remote T_user entity. As a result of this protocol sequence, the transport connection will be terminated.

6.1.2.8 T_DISCONNECT.indication

6.1.2.8.2 Semantics of the service primitive

Replace the text in 6.1.2.8.2 with the following:

The primitive shall provide the following parameter.

```
T_DISCONNECT.indication(  
    reason  
    )
```

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_entity.

6.1.2.8.3 When generated

Replace the text in 6.1.2.8.3 with the following:

This primitive is passed from the T_entity to the T_user entity to signal an immediate disconnection.

6.2 PDU specification

6.2.2 TPDU format

Replace the first paragraph in 6.2.2 with the following:

The only transport primitives with associated PDUs are the data request and indication primitives. All transport PDUs shall consist of an information field with a single octet inactive Transport layer header (see figure 6).

6.3 Mode of operation

Replace figure 7 with the following:

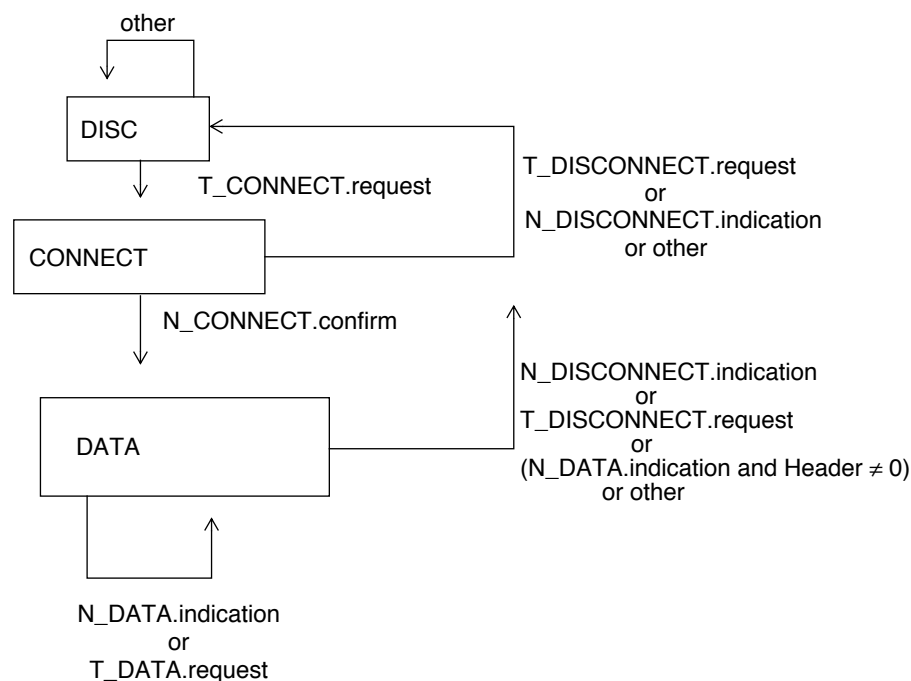


Figure 7—Inactive Transport layer DCC state diagram

Replace figure 8 with the following:

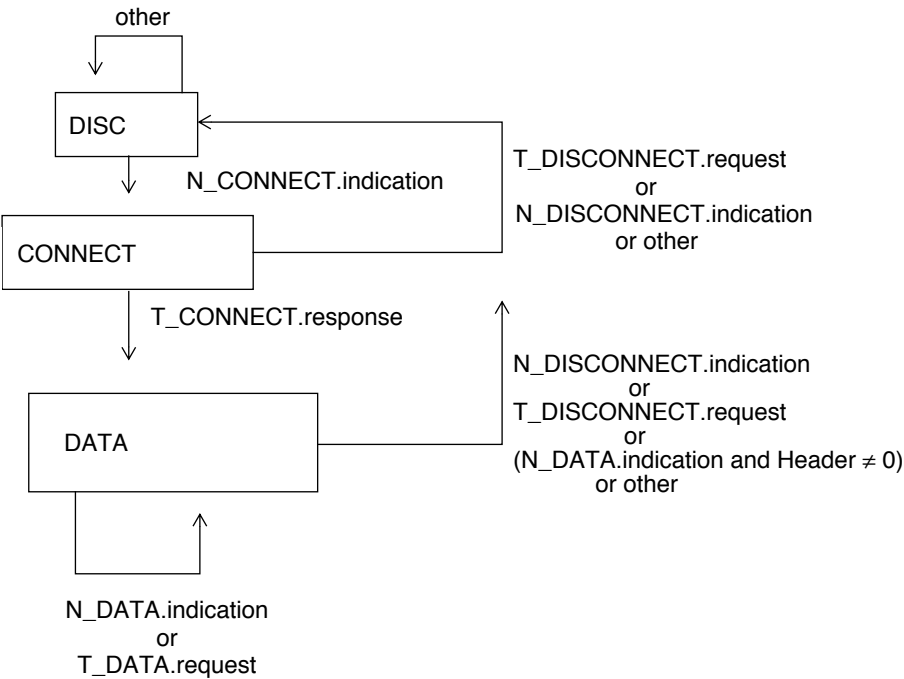


Figure 8—Inactive Transport layer BCC state diagram

Replace table 1 with the following:

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
DATA	T_DATA.request	N_DATA.request	DATA
	N_DATA.indication and Header = 0	T_DATA.indication	DATA
	N_DATA.indication and Header ≠ 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

Replace table 2 with the following:

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
DATA	T_DATA.request	N_DATA.request	DATA
	N_DATA.indication and Header = 0	T_DATA.indication	DATA
	N_DATA.indication and Header ≠ 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

7.1.2 Detailed service specifications

7.1.2.1 N_CONNECT.request

7.1.2.1.2 Semantics of the service primitive

Replace the text in 7.1.2.1.2 with the following:

The primitive shall provide the following parameters.

```

N_CONNECT.request(
    called address,
    calling address
)

```

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

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

7.1.2.2 N_CONNECT.indication

7.1.2.2.2 Semantics of the service primitive

Replace the text in 7.1.2.2.2 with the following:

The primitive shall provide the following parameters.

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

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

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

7.1.2.4 N_CONNECT.confirm

7.1.2.4.2 Semantics of the service primitive

Replace the text in 7.1.2.4.2 with the following:

The primitive shall provide the following parameters.

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

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

8. Data Link layer

8.2 Service specification

Change the first three sentences in 8.2 as follows:

The following subclauses specify the services offered by the DL ~~layer entity~~ 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 specified by ISO/IEC 8802-2:1994. These services allow a local DL_user ~~layer-local~~ entity to establish a connection-oriented communications channel with a remote DL_user entity and exchange packets over that channel.

8.2.1 Overflow of DL_user and DL interactions

8.2.1.1 DL connection phase

Change the first sentence in 8.2.1.1 as follows:

The DL_user ~~layer entity~~ is responsible for initiating the DL connection.

Change the last paragraph in 8.2.1.1 as follows:

The DL_CONNECT.request primitive is passed to the DL ~~layer~~ entity 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~~ entity 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~~ entity to convey the results of the previous associated DL_CONNECT.indication primitive. The DL_CONNECT.confirm primitive is passed from the DCC DL ~~layer~~ entity to convey the results of the previous associated DL_CONNECT.request.

8.2.2 Detailed service specifications

8.2.2.1 DL_CONNECT.request

8.2.2.1.2 Semantics of the service primitive

Replace the text in 8.2.2.1.2 with the following:

The primitive shall provide the following parameter.

```
DL_CONNECT.request (  
    )
```

NOTE—There is no DLSDU parameter. The parameters for the data link parameter negotiation subfield and private parameter negotiation subfield, including the bandwidth negotiation variables Tx-Length, Rx-Length, and polling period, are transferred to and from the data link entity by means of data link systems management.

8.2.2.2 DL_CONNECT.indication

8.2.2.2.2 Semantics of the service primitive

Replace the text in 8.2.2.2.2 with the following:

The primitive shall provide the following parameter.

```
DL_CONNECT.indication (  
    )
```

NOTE—There is no DLSDU parameter. The parameters for the data link parameter negotiation subfield and private parameter negotiation subfield, including the bandwidth negotiation variables Tx-Length, Rx-Length, and polling period, are transferred to and from the data link entity by means of data link systems management.

8.2.2.4 DL_CONNECT.confirm

8.2.2.4.2 Semantics of the service primitive

Replace the text in 8.2.2.4.2 with the following:

The primitive shall provide the following parameter.

```
DL_CONNECT.confirm(  
    )
```

NOTE—There is no DLSDU parameter. The parameters for the data link parameter negotiation subfield and private parameter negotiation subfield, including the bandwidth negotiation variables Tx-Length, Rx-Length, and polling period, are transferred to and from the data link entity by means of data link system management.

8.2.2.5 DL_DATA.request

8.2.2.5.2 Semantics of the service primitive

Replace the text in 8.2.2.5.2 with the following:

The primitive shall provide the following parameters.

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

The source address identifies the data link source service access point (SSAP) of the local data link entity. The value of the source address is 40h for DL_user data, 02h for data link management data, or 80h for MIB systems management data.

The destination address identifies the data link destination service access point (DSAP) of the remote data link entity. The value of the destination address is 40h for individually addressed DL_user data, 41h for group addressed DL_user data, 02h for individually addressed data link management data, 03h for group addressed data link management data, 80h for individually addressed MIB systems management data, or 81h for group addressed MIB systems management data.

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

8.2.2.6 DL_DATA.indication

8.2.2.6.2 Semantics of the service primitive

Replace the second and third paragraphs of 8.2.2.6.2 with the following:

The source address identifies the data link SSAP of the remote data link entity. The value of the source address is 40h for DL_user data, 02h for data link management data, or 80h for MIB systems management data.

The destination address identifies the data link DSAP of the local data link entity. The value of the destination address is 40h for individually addressed DL_user data, 41h for group addressed DL_user data, 02h for individually addressed data link management data, 03h for group addressed data link management data, 80h for individually addressed MIB systems management data, or 81h for group addressed MIB systems management data.

8.2.2.7 DL_DATA.confirm

8.2.2.7.2 Semantics of the service primitive

Replace the last two paragraphs in 8.2.2.7.2 with the following:

The source address identifies the data link SSAP of the remote data link entity. The value of the source address is 40h for DL_user data, 02h for data link management data, or 80h for MIB systems management data.

The destination address identifies the data link DSAP of the local data link entity. The value of the destination address is 40h for DL_user data, 02h for data link management data, or 80h for MIB systems management data.

8.2.2.8 DL_UNITDATA.request

8.2.2.8.2 Semantics of the service primitive

Replace the text in 8.2.2.8.2 with the following:

The primitive shall provide the following parameters.

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

The source address identifies the data link SSAP of the local data link entity. The value of the source address is 40h for DL_user data, 02h for data link management data, or 80h for MIB systems management data.

The destination address identifies the data link DSAP of the remote data link entity. The value of the destination address is 40h for individually addressed DL_user data, 41h for group addressed DL_user data, 02h for individually addressed data link management data, 03h for group addressed data link management data, 80h for individually addressed MIB systems management data, or 81h for group addressed MIB systems management data.

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

8.2.2.9 DL_UNITDATA.indication

8.2.2.9.2 Semantics of the service primitive

Replace the second and third paragraphs of 8.2.2.9.2 with the following:

The source address identifies the data link SSAP of the remote data link entity. The value of the source address is 40h for DL_user data, 02h for data link management data, or 80h for MIB systems management data.

The destination address identifies the data link DSAP of the local data link entity. The value of the destination address is 40h for individually addressed DL_user data, 41h for group addressed DL_user data, 02h for individually addressed data link management data, 03h for group addressed data link management

data, 80h for individually addressed MIB systems management data, or 81h for group addressed MIB systems management data.

8.2.2.10 DL_CONNECTION_FLOWCONTROL.request

8.2.2.10.2 Semantics of the service primitive

Replace the text in 8.2.2.10.2 with the following:

The primitive shall provide the following parameter.

```
DL_CONNECTION_FLOWCONTROL.request(  
                                     amount  
                                   )
```

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.

8.2.2.11 DL_CONNECTION_FLOWCONTROL.indication

8.2.2.11.2 Semantics of the service primitive

Replace the text in 8.2.2.11.2 with the following:

The primitive shall provide the following parameter.

```
DL_CONNECTION_FLOWCONTROL.indication(  
                                     amount  
                                   )
```

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.

8.2.2.12 DL_DISCONNECT.request

8.2.2.12.2 Semantics of the service primitive

Replace the text in 8.2.2.12.2 with the following:

The primitive shall provide the following parameter.

```
DL_DISCONNECT.request(  
                      )
```

8.2.2.13 DL_DISCONNECT.indication**8.2.2.13.2 Semantics of the service primitive**

Replace the first three paragraphs in 8.2.2.13.2 with the following:

The primitive shall provide the following parameters.

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

8.2.2.14 DL_RESET.request**8.2.2.14.2 Semantics of the service primitive**

Replace the text in 8.2.2.14.2 with the following:

The primitive shall provide the following parameter.

```
DL_RESET.request(  
    )
```

8.2.2.15 DL_RESET.indication**8.2.2.15.2 Semantics of the service primitive**

Replace the text in 8.2.2.15.2 with the following:

The primitive shall provide the following parameter.

```
DL_RESET.indication(  
    )
```

8.2.2.16 DL_RESET.confirm**8.2.2.16.2 Semantics of the service primitive**

Replace the text in 8.2.2.16.2 with the following:

The primitive shall provide the following parameter.

```
DL_RESET.confirm(  
    )
```

8.3 PDU specification

8.3.2 DLPDU format

Change the description of INFORMATION in figure 20 to the following:

INFORMATION = INFORMATION FIELD. FOR INFORMATION (I), UNNUMBERED INFORMATION (UI), AND EXCHANGE IDENTIFICATION (XID) FRAMES ONLY. THE MAXIMUM LENGTH INFORMATION FIELD IS 1496 1498 OCTETS. THE XID INFORMATION FIELD IS DEFINED IN 8.4.3.7.2. FOR THE I AND UI FRAMES, THE INFORMATION FIELD CONSISTS OF A 1-OCTET DSAP ADDRESS, A 1-OCTET SSAP ADDRESS, AND UP TO 1496 OCTETS OF DL USER DATA.

8.3.4 Flag

8.3.4.1 Control octet transparency for low-speed operation

Insert the following at the end of 8.3.4.1:

NOTE—In accordance with subclause 4.5.2.2 of ISO/IEC 3309:1993, the sending data link entity shall compute the value of its FCS prior to performing its octet-insertion operation. The receiving data link entity shall perform its octet-extraction operation prior to checking the received FCS.

8.3.4.8 Bit order

Replace the text in 8.3.4.8 with the following:

All octets prior to the FCS, including flag octets and control escape octets (for low-speed operation), HDLC address octets, HDLC control octet, DSAP address, SSAP address, information field, and sequence numbers, shall be transmitted LSB first. The FCS field shall be transmitted starting with the coefficient of the highest term.

NOTE—In accordance with ISO/IEC 3309:1993 and ISO/IEC 4335:1993, the bit designated as “bit 1” for flag octets and control escape octets (for low-speed operation), HDLC address octets, and HDLC control octet is transmitted first. The first bit of an octet or HDLC sequence number (N(S) or N(R)) that is transmitted/received shall be designated as the LSB of the field and have the weight of 2^0 .

8.3.5 Invalid DLPDU

Change item e) in 8.3.5 as follows:

- e) Verifying that the number of ~~words~~ octets in the information field does not exceed 1498.

Insert the following after item g) in 8.3.5:

- h) For an XID frame that is transmitted as part of the connection establishment frame sequence, the receiving DCC or BCC shall perform additional validity checking, as follows:
 - 1) For the DL parameter negotiation subfield, verify for correct overall subfield octet count; that all parameters are included; for correct values of the GI and GL parameters per this standard; and for correct values (per this standard) of the PI, PL, and PV attributes for the Class, Options, Group Address, Window-Length Tx, and Window-Length Rx options. For the Tx-Length and Rx-Length parameters, the PI and PL attributes shall be verified to be in accordance with this standard, and the values of PV shall be within the allowable range.

- 2) For the private parameter negotiation subfield, the receiving DCC or BCC shall verify for the correct value of GI, and that the overall number of received octets is in agreement with the value indicated by the GL parameter.
- 3) The BCC shall verify that the DCC transmits the following parameters of the DL private parameter subfield: Version/revision of the IEEE standard, DCC desired polling period, DCC model number, DCC firmware revision level, and DCC configuration/status octets. In addition, the BCC shall verify that the DCC does not transmit any of the following parameters: fraction of poll group available, system manager address, and BCC soft ID number.
- 4) The DCC shall verify that the BCC transmits the following parameters of the DL private parameter subfield: Version/revision of the IEEE standard, date/time-of-day, BCC minimum polling period, fraction of poll group bandwidth available, BCC model number, BCC firmware revision level, and BCC configuration/status octet.
- 5) For any required parameters [per item h3) and per item h4)], or any optional parameters transmitted, the values of PI must be in accordance with this standard, the values of PL must be in accordance with this standard for those parameters where PL is specified, and the number of received octets for the parameter's value attribute must be in agreement with the transmitted value of PL for each parameter transmitted.
- 6) The PV attribute of the standard version/revision parameter shall consist of valid ASCII octets.
- 7) For the BCC's date/time-of-day parameter, and the DCC's optional date/time-of-day parameter (if transmitted), the receiving DCC or BCC shall verify for correct ASCII octet encoding, all values within allowable ranges, and an ASCII decimal point (".") for the fifteenth octet in the string.
- 8) The receiving DCC or BCC shall verify that the value of the BCC minimum polling period, or the DCC desired polling period, is in the allowable range for the particular data rate, per table 8.
- 9) For the final octet of the BCC's configuration/status octets, the DCC shall verify that bits 5, 6, and 7 are logic "0."
- i) For an XID frame that is periodically transmitted by the BCC for the purpose of time synchronization, the DCC shall verify that the formatting of the PI, PL, and PV fields is correct. This includes verifying for ASCII encoding, values within allowable ranges, and an ASCII decimal point (".") for the fifteenth octet in the string.

8.4 DL elements of procedures

8.4.2 Control field formats

Replace the paragraph below figure 23 with the following, delete table 6 and table 7, and renumber the remaining tables:

The control fields for all BCC command frames and DCC response frames shall be formatted in accordance with ISO/IEC 4335:1993 (E). For both BCC and DCC, this includes Information (I) frames, and Receive ready (RR) and Receive not ready (RNR) supervisory frames. For BCC, this also includes the Set normal response mode (SNRM), Disconnect (DISC), Set initialization mode (SIM), Unnumbered poll (UP), Unnumbered information (UI), and eXchange IDentification (XID) unnumbered command frames. For DCC, this also includes the Unnumbered acknowledgement (UA), Disconnect mode (DM), Request initialization mode (RIM), Unnumbered information (UI), and eXchange IDentification (XID) unnumbered response frames.

8.4.3 Control field parameters

8.4.3.7 Unnumbered commands and responses

8.4.3.7.2 Exchange information (XID) command and response

Change the title of 8.4.3.7.2 as follows:

8.4.3.7.2 Exchange information eXchange IDentification (XID) command and response

Replace the second paragraph in 8.4.3.7.2 with the following:

The XID information field is divided among three subfields: a format identifier, a parameter negotiation DL layer subfield, and a private parameter DL subfield. The BCC and DCC share a common format identifier and DL layer parameter negotiation subfield. In addition, portions of the private parameter DL subfields are common to the BCC and DCC.

Insert the following after the fourth paragraph in 8.4.3.7.2:

The BCC or DCC shall not transmit the PI (parameter identifier), PL (parameter length), or PV (parameter value) for any nonapplicable parameters or unimplemented optional parameters.

NOTE—This is consistent with subclause 5.2.3 of ISO/IEC 8885:1991. An example of a nonapplicable parameter is the SYSTEM MANAGER ADDRESS parameter for a DCC. An example of an unimplemented optional parameter is the BCC OR DCC GLOBAL IDENTIFIER NUMBER for a BCC or DCC that does not implement this option.

Replace figure 24 in 8.4.3.7.2 with the following:



FI - Format Identifier = 82h = General purpose XID information field identifier

Figure 24—Overview of XID information field

8.4.3.7.2.1 DL parameter negotiation subfields

Change the table in figure 25 to the following:

GI	GL	PI PL PV	PI PL PV	PI PL PV	PI PL PV	PI PL PV	PI PL PV	PI PL PV
80h	<u>001AH</u>	Class	Options	Group address	Tx- Length	Rx- Length	Window- Length Tx	Window- Length Rx

Change the following descriptions in figure 25 as shown:

Class Class of procedures (PI = 02h, PL = 02h, for BCC: PV = 0020h 0022h, for DCC: PV = 0024h)

Options HDLC optional features (PI = 03h, PL = 03h; ~~For low-speed (2400 Bd or 9600 Bd) operation, PV = 04C460h; For BCC or low-speed DCC operation, PV = 14C4B8h; For high-speed (1 MB/s) DCC operation, PV = 00C460h 00C4B8h~~)

8.4.3.7.2.2 DL private parameter negotiation subfields

Replace figure 26 and the accompanying Notes with the following:

GI	GL	PI PL PV	PI PL PV	PI PL PV	PI PL PV	PI PL PV
F0h	USER DEFINED	VERSION/ REVISION OF IEEE STD 1073.3.1	DATE/TIME- OF-DAY	BCC MINIMUM POLLING PERIOD OR DCC DESIRED POLLING PERIOD	BCC ONLY: FRACTION OF POLL GROUP BANDWIDTH AVAILABLE (^f AVAILABLE)	SYSTEM MANAGER ADDRESS (BCC ONLY)

PI PL PV	PI PL PV	PI PL PV	PI PL PV	PI PL PV	PI PL PV	PI PL PV
BCC OR DCC GLOBAL IDENTIFIER NUMBER	BCC OR DCC MODEL NUMBER	BCC OR DCC MODEL NUMBER EXTENSION	BCC OR DCC FIRMWARE REVISION LEVEL	BCC SOFT ID NUMBER (BCC ONLY)	DCC SOFT ID NUMBER	BCC OR DCC CONFIG- URATION/ STATUS OCTETS

Figure 26—XID information field, DL private parameter subfields

NOTES for figure 26

Group Identifier (GI), Group Length (GL): GI = F0h, GL = User Defined (2-octet binary)

For all XID DL private parameter negotiation subfield PVs (parameter values), the octet-transmission ordering is most-significant octet to least-significant octet.

Standard version/revision: PI = 01h, PL = variable, PV = corresponds to the standard and revision level, encoded in ASCII bytes. This parameter includes information regarding the version and revision level of the IEEE 1073.3 transport profile and the IEEE 1073.4 Physical layer standards that are implemented within a BCC or DCC.

NOTE—For this release of this standard, PL = 13h, PV = 1073.3.1a,1073.4.1a.

Date/Time-of-day: PI = 02h, PL = 15h. PV: the date and time of day are encoded in accordance with ISO 8601:1988. Date/time-of-day is encoded as a 21-character ASCII string, as follows: CCYYMMDDHHMMSS.FFFFFFFF,

where

CC	is century
YY	is year
first MM	is month
DD	is day
HH	is hour
second MM	is minute
SS	is second
.	is decimal point, separating seconds subfield from fractional seconds subfield.
FFFFFFF	is fractional seconds (resolution = 1 μ s/LSB). Note that if some or all of these digits are not used, the unused digits shall be assigned values of 0.

This subfield represents the current time of day when embedded in a BCC XID frame, and the time of the last previously received XID frame, when embedded in a DCC XID response frame.

This parameter is embedded into an XID frame that is transmitted by the BCC during the connection establishment frame sequence and once every five seconds thereafter for the purpose of synchronizing the DCC's date/time-of-day clock with the BCC's date/time-of-day clock. If the BCC's value for the date/time-of-day clock is not valid, the BCC shall transmit a value of all zeros for this attribute. Use of this parameter is optional for DCCs.

Polling period: PI = 03h, PL = 02h. PV is the BCC's minimum polling period, or the DCC requested polling period, in milliseconds. The format for the polling period parameter is 2 binary octets. The BCC may have separate minimum poll-

ing periods for each of the three data rates. For each of the three data rates, the minimum and maximum polling periods are defined in table 8.

Fraction of BCC poll group bandwidth available ($f_{\text{AVAILABLE}}$) (BCC only): PI = 04, PL = 2; PV = the fraction of a multiplexed BCC's poll group bandwidth that is currently available for use by a DCC connected to a particular BCC port. This fraction is encoded as two binary octets, representing a fractional value between 0.0000000 and 0.9999847, with a resolution of 0.0000153 per LSB. The BCC may maintain one or more polling groups. Each polling group may include DCCs at the same and/or different data rates.

System manager address (BCC only): PI = 05, PL = Nh, PV = the address of the system management entity. Note that N = the number of octets in the system manager's address. This is an optional parameter that may be transmitted by the BCC, with a default to not send it. In order to transmit it, the value must be set by means of the BCC DL system management. This parameter is not sent by the DCC.

BCC or DCC global identifier number: PI = 06, PL = 08; PV is the 64 b global identifier (EUI-64) parameter for the BCC or DCC. Use of this parameter is optional. As shown in figure 27, this parameter consists of a 3-octet (24 b) company_id number followed by a 5-octet (40 b) extension identifier. The individual company_id number is assigned and administered by the IEEE Registration Authority Committee (RAC), and represents a unique code for a particular manufacturing company or other organization.

The 40 b extension identifier number is assigned by the particular manufacturer or other organization. The length of this parameter shall be 5 octets. The format of this parameter is left to the discretion of the BCC or DCC manufacturer. It is the responsibility of the individual communication controller manufacturer to assign a unique (non-duplicate) extension identifier number to each BCC or DCC.

No more than one 64 b global identifier value shall be contained within each component that is manufactured.

BCC or DCC model number: PI = 07, PL = TBD, PV = defined by the BCC or DCC manufacturer.

BCC or DCC model number extension: PI = 08, PL = TBD, PV = defined by the BCC or DCC manufacturer. This parameter is optional for BCCs and DCCs.

BCC or DCC firmware revision level: PI = 09, PL = TBD, PV = defined by the BCC or DCC manufacturer.

BCC soft ID number (BCC only): PI = 10, PL = TBD, PV = user assigned. This parameter, which is only transmitted by BCCs, allows unique, dynamic, user-assigned identification of BCCs within an institution.

DCC soft ID number: PI = 11, PL = user defined, PV = user assigned. This parameter is optional. This parameter may be assigned and initially transmitted from the BCC to the DCC. It may be subsequently transmitted by the DCC. If the DCC has no storage of a previously assigned soft ID number, it shall transmit a value of zero for this parameter. If the BCC transmits a soft ID parameter, and a value for the DCC soft ID parameter has been assigned by the DCC's local DL system management user, the value of the DCC soft ID received from the BCC shall override the locally assigned value. This parameter allows unique, dynamic, user-assigned identification of DCCs within an institution.

BCC or DCC configuration/status octets: PI = 12, PL = User defined (minimum of 1 for a BCC, and 2 for a DCC). PV consists of 1 (for a BCC), 2 (for a DCC), or more binary-encoded octets. The last octet for a BCC, or last two octets for a DCC, shall be encoded as shown in figure 28.

Replace part a) of figure 28 with the following:

BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8
2400 Bd LOW-SPEED BCC CAPABILITY	9600 Bd LOW-SPEED BCC CAPABILITY	HIGH- SPEED BCC CAPABILITY	BCC INTENT TO PHYSICALLY DISCONNECT NOTIFICATION CAPABILITY	LOGIC "0"	LOGIC "0"	LOGIC "0"	BCC FAIL

a) Last octet for a BCC

Figure 28—Encoding of configuration/status octets

8.4.3.7.2.3 User data subfield

Delete 8.4.3.7.2.3, including figure 29.

8.4.3.7.9 Request initialization mode (RIM) response

Insert the following after 8.4.3.7.9:

8.4.3.7.10 Unnumbered Information (UI) commands and responses

The UI frame types are used only during interrupt acknowledgment frame sequences. During an interrupt acknowledge sequence, the BCC responds to the DCC's interrupt request by transmitting a UP frame. The DCC responds to this frame with the transmission of a UI frame. The Information field of the DCC's UI frame consists of the DSAP address octet, the SSAP address octet, and up to 1496 octets comprising the DL_user information field. The BCC then acknowledges reception of the DCC's UI frame by transmitting a UI frame. The Information field of the BCC's UI frame consists only of the DSAP address octet and the SSAP address octet. This frame does not include a DL_user information field.

8.5 DL procedure descriptions**8.5.6 Procedures for information transfer****8.5.6.9 Interframe time space**

Replace the following row in table 9 on page 73:

Table 9—Timing constant definitions for the network loading model

Parameter	Description	Value for 2400 Bd low-speed operation	Value for 9600 Bd low-speed operation	Value for high-speed (1 Mb/s) operation
$L_{O/H}$	The maximum number of overhead bit times associated with the transmission of a command frame by the BCC and a response frame by the DCC. This allows for either the BCC or DCC (but not both) to transmit a contiguous frame pair, consisting of an I-frame, followed by an RNR frame. This includes address, control, and FCS octets, flag octets for low-speed frames, and start and stop delimiter times for high-speed frames. Note that for low-speed frames, $L_{O/H}$ does not include the start and stop bits transmitted with each octet.	160	160	135

8.5.8 Timer functions**8.5.8.2 DCC timer**

Change the second paragraph in 8.5.8.2 to the following:

The station maintains an internal LEC variable for each DL connection which shall be set to "0" when the station receives a UA response PDU to an SNRM command PDU. DCC DL_entity receives a DL_CONNECT.request from the DL_user entity.

8.6 Interrupt requests and acknowledgments

8.6.1 DCC interrupt request function

Insert the following at the end of the first paragraph in 8.6.1:

If the BCC's value for the date/time-of-day clock is not valid, the BCC shall transmit a value of all zeros for the date/time-of-day parameter.

8.6.1.1 BCC interrupt acknowledgment

Change items c) and d) in 8.6.1.1 as follows:

- c) The DCC shall transmit a UI response frame. The response time between the end of the received broadcast UP frame and the beginning of the UI response frame shall be a minimum of 350 μ s, with a maximum of 1.0 ms for high-speed operation, ~~and 100 ms for low-speed operation.~~ For low-speed operation, there is no minimum response time, and a maximum response time of 100 ms. The UI response frame shall contain the DCC's own port HDLC address field and information ~~conveying~~ notifying of the cause of the interrupt request. When the BCC receives the UI response frame, a DL_UNITDATA.indication primitive is issued to the BCC upper layers.
- d) The BCC shall acknowledge reception of the DCC's UI frame from the DCC by the transmission of a UI command frame to the responding DCC's HDLC port address. The Information field of this UI frame consists only of the DSAP address octet and the SSAP address octet. This frame does not include a DL_user information field.

8.7 Formal description of procedures

8.7.2 State event tables

8.7.2.1 DCC state tables

8.7.2.1.1 Events

Delete the following events from 8.7.2.1.1:

PH_CONNECTION.indication: Receive a physical connection indication from the physical layer.

PH_FRAME_START.indication: Indication from the DCC physical layer that a low-speed BCC port has asserted its FRAME* output (low) to signify the beginning of a command frame.

PH_FRAME_END.indication: Indication from the physical layer that a low-speed BCC port has cleared its FRAME* output (high) to signify the end of a command frame.

Insert the following event in 8.7.2.1.1 after PH_DISCONNECT.indication:

PH_DATA.indication: Indication from the physical entity to signal the arrival of a PhSDU from the Ph_entity.

Insert the following at the end of 8.7.2.1.1:

Valid reception of any frame type listed previously results in the link error counter (LEC) being cleared to zero.

Following reception of a valid I, RR, or RNR frame, the string variable `Frame_type` is assigned the value equal to the type of the received frame (i.e., `Frame_type := I, RR, or RNR`).

8.7.2.1.2 Constants and variables

Insert the following at the end of 8.7.2.1.2:

old_flow_amount: Previous value of “amount” parameter for `DL_CONNECTION_FLOWCONTROL.indication` primitive.

Frame_type: String variable set equal to the most recent type of HDLC Information (I) or Supervisory (RR, RNR) frame received.

8.7.2.1.3 Actions

Insert the following after the paragraph describing state transition in 8.7.2.1.3:

PH_DATA.request: Request to the physical entity to transfer a PhSDU to the remote `DL_entity`.

Change the third sentence in the paragraph describing $x()$ in 8.7.2.1.3 as follows:

At the start of a frame to be transmitted, the DCC `DL_entity` issues a `PH_ACTIVATE.request` primitive and a `PH_DATA.request` primitive to the `Ph_entity` and sets `Transmit_flag` to true.

Insert the following at the end of 8.7.2.1.3:

For the case of a transmitted frame pair consisting of an I- and an RNR frame, the two frames shall be transmitted contiguously. There are two separate FCSS, one transmitted at the end of the I- frame, the other transmitted at the end of the RNR frame. To initiate the transmission of the I- frame, `PH_ACTIVATE.request` and `PH_DATA.request` are issued. At this time, the `Transmit_flag` is set to true. For low-speed operation, the `DL_entity` shall transmit a total of three flag octets: one at the start of the I- frame, one delineating both the end of the I- frame and the start of the RNR frame, and one at the end of the RNR frame. The RNR frame is transmitted immediately following the end of the I- frame. The `DL_entity` does not issue a `PH_ACTIVATE.request` or `PH_DEACTIVATE.request`, or change the state of the `Transmit_flag` following transmission of the I- frame. Following the transmission of the RNR frame, the `DL_entity` issues a `PH_DEACTIVATE.request` primitive to the `Ph_entity` and clears the `Transmit_flag` to false.

8.7.2.1.4 Functions

Replace line 2 through line 20 in 8.7.2.1.4 with the following:

Process `N(R)` (process received ACK on incoming I, RR, RNR)

if `IframeSent = true` then

 if $N(R) = (V(S) + 1) \text{ modulo } 8$ then

`DL_DATA.confirm`

`IframeSent := false`

$V(S) := (V(S) + 1) \text{ modulo } 8$

`Data_counter := Data_counter - 1`

`N(R)_ok := true`

`PEC := 0`

 else

```

    if Frame_type = RNR then
        if N(R) = V(S) then
            N(R)_ok := true
            PEC := 0
        else
            N(R)_ok := false
        end
    else
        N(R)_ok := false
    end
end
end
else
    if N(R) = V(S) then
        N(R)_ok := true
        PEC := 0
    else
        N(R)_ok := false
    end
end
end

```

Process N(S) (process send sequence number on incoming I frame)

```

if V(R) = N(S) then
    N(S)_ok := true
    if receive_ready = true then
        DL_DATA.indication (DLSDU)
        V(R) := (V(R) + 1) modulo 8
    end
else
    N(S)_ok := false
end
end

```

Change the paragraph in 8.7.2.1.4 on page 81 that begins with **High_speed** := true or false (fixed value) as follows:

High_speed := true or false (fixed value)
 Rate := 2400 Bd, or 9600 Bd, or 4E6 1 Mb/s (fixed value)
 PH_DATA_RATE.request (speed, rate)
 Physical_connect_sense_capability := true or false (fixed value)

Change the paragraph in 8.7.2.1.4 on page 81 that begins with **DL_DISCONNECT.indication** as follows:

```

DL_DISCONNECT.indication
(reason)
Process Restore_remaining_f_AVAILABLE
state transition...PHYSICALLY DISCONNECTED
end

```

Replace the first two lines in the Process Reset_rqst function in 8.7.2.1.4 with:

```
Process Reset_request  
If Transmit_flag = true then
```

Insert the following at the end of 8.7.2.1.4:

```
Process Flow_indication  
If old_flow_amount = 0 then  
    If amount = 1 then  
        DL_CONNECTION_FLOWCONTROL.indication (amount := 1)  
        old_flow_amount := 1  
    end  
else  
    If amount = 0 then  
        DL_CONNECTION_FLOWCONTROL.indication (amount := 0)  
        old_flow_amount := 0  
    end  
end
```

8.7.2.1.5 Initialization

Insert the following in 8.7.2.1.5 after the line Reset_flag:= false:

```
old_flow_amount := 1  
  
IframeSent := false
```

Delete the following variable in 8.7.2.1.5:

```
PH_INTERRUPT_DEACTIVATE.request
```

8.7.2.1.6 High-speed DCC state table

Insert the following after the first sentence in 8.7.2.1.6:

Note that timers will be reset as the result of any “state transitions” back to the current state.

Replace table 10 in 8.7.2.1.6 with the following:

Table 10—States for high-speed DCC

State	Event	Action	Next state
PHYSICALLY DISCONNECTED	DL_CONNECT.request	DL_DISCONNECT.indication (reason := Physically disconnected) LEC := 0, PEC := 0 state transition	PHYSICALLY DISCONNECTED
	PH_CONNECTION.indication	Connect_pending := false state transition	LOGICALLY DISCONNECTED (HIGH SPEED)
	DL_DISCONNECT.request	DL_DISCONNECT.indication (reason := Physically disconnected), state transition	PHYSICALLY DISCONNECTED
LOGICALLY DISCONNECTED (HIGH SPEED)	DL_CONNECT.request	IRQST = 0 Timer := 200 ms LEC := 0 PEC := 0 Data_counter := 0 Unitdata_counter := 0 V(R) := V(S) := 0 Connect_pending := true PH_INTERRUPT_ACTIVATE.request timer = 1.2 s state transition	WAIT FOR UP (CONN RQST/HIGH SPEED)
	DL_DATA. request	DISCONNECT.indication (reason := erroneous DCC data request) state transition	LOGICALLY DISCONNECTED (HIGH SPEED)
	DL_DISCONNECT.request	DL_DISCONNECT.indication (reason := disconnect request from DL_user while in disconnect state) LEC := 0, PEC := 0 state transition	LOGICALLY DISCONNECTED (HIGH SPEED)
	other	no response state transition	LOGICALLY DISCONNECTED (HIGH SPEED)
	PH_DISCONNECT.indication	Process Ph_disconnect	
	DL_CONNECTION_ FLOWCONTROL.request, with source address = 126 (BCC), destination address = DCC's address	Process Flowcontrol	
	Intentional_disconnect-timer (60 s) timed out	Intentional_disconnect := false state transition	LOGICALLY DISCONNECTED (HIGH SPEED)

Table 10—States for high-speed DCC (continued)

State	Event	Action	Next state
WAIT FOR UP (CONN RQST/ HIGH SPEED)	Intentional_disconnect-timer (60 s) timed out	Intentional_disconnect := false state transition	WAIT FOR UP (CONN RQST/ HIGH SPEED)
	rUP	state transition	DELAY 350 μ s (CONN REQUEST)
	Timer timed out (1 s)	LEC := LEC + 1 If LEC $\geq$ MLEC then DL_DISCONNECT.indication (reason := time-out counter exceeded max waiting for connection establish- ment UP) LEC := 0, PEC := 0 state transition else PH_INTERRUPT_ACTIVATE.request state transition end	LOGICALLY DISCONNECTED (HIGH SPEED) WAIT FOR UP (CONN RQST/HIGH SPEED)
	PH_DISCONNECT.indication	Process Ph_disconnect	
	DL_CONNECTION_ FLOWCONTROL.request, with source address = 126 (BCC), destination address = DCC's address, and amount = 0	receive_ready := false state transition	WAIT FOR UP (CONN RQST/HIGH SPEED)
	DL_CONNECTION_ FLOWCONTROL.request, with source address = 126 (BCC), destination address = DCC's address, and amount > 0	receive_ready := true state transition	WAIT FOR UP (CONN RQST/HIGH SPEED)
	DL_DISCONNECT.request	DL_DISCONNECT.indication (reason := disconnect request from DL_user layer) LEC := 0, PEC := 0 state transition	LOGICALLY DISCONNECTED (HIGH SPEED)
	PH_INTERRUPT_DEACTIVATE. indication	DL_DISCONNECT.indication (reason := time-out counter exceeded max waiting for connection establishment UP) LEC := 0, PEC := 0 state transition	LOGICALLY DISCONNECTED (HIGH SPEED)
	other	PEC := PEC + 1.2 if PEC $\geq$ MPEC then DL_DISCONNECT.indication (reason := PEC exceeded max) LEC := 0, PEC := 0 state transition else Timer := 1.2 s state transition end	LOGICALLY DISCONNECTED (HIGH SPEED) WAIT FOR UP (CONN RQST HIGH SPEED)

Table 10—States for high-speed DCC (continued)

State	Event	Action	Next state
DELAY 350 μ s/ CONN RQST	PH_INTERRUPT_DEACTIVATE. indication	DL_DISCONNECT.indication (reason := premature interrupt deactivate during connection establishment) LEC := 0, PEC := 0 state transition	LOGICALLY DISCONNECTED (HIGH SPEED)
	Intentional_disconnect-timer (60 s) timed out	Intentional_disconnect := false state transition	DELAY 350 μ s/ CONN RQST
	Timed out (350 μ s)	xDM (address := 0) Connect_pending := false Reset_flag := false Timer := 1 s state transition	WAIT FOR XID
	PH_DISCONNECT.indication	Process Ph_disconnect	
	DL_CONNECTION_ FLOWCONTROL.request, with source address = 126 (BCC), destination address = DCC's address, and amount = 0	receive_ready := false state transition	DELAY 350 μ s/ CONN RQST
	DL_CONNECTION_ FLOWCONTROL.request, with source address = 126 (BCC), destination address = DCC's address, and amount > 0	receive_ready := true state transition	DELAY 350 μ s/ CONN RQST
	DL_DISCONNECT.request	xRD DL_DISCONNECT.indication (reason := requested by DCC DL_user) LEC := 0, PEC := 0 state transition	LOGICALLY DISCONNECTED (HIGH SPEED)
WAIT FOR XID	rDISC	xUA Start Intentional_disconnect_timer (60 s) Intentional_disconnect := true state transition	WAIT FOR INTER- RUPT DEACTIVATE/ DISCONNECT
	rXID	Physically_disconnected := false xXID (address from received XID frame) Process f _{AVAILABLE} If Successful_XID := false then DL_DISCONNECT.indication (reason := failed XID negotiation) LEC := 0, PEC := 0 end state transition PEC := 0	WAIT FOR SNRM
	Intentional_disconnect-timer (60 s) timed out	Intentional_disconnect := false state transition	WAIT FOR XID
	PH_DISCONNECT.indication	Process Ph_disconnect	