Copies of this document may be purchased from:
Global Engineering, 15 Inverness Way East,
Englewood, CO 80112-5704
Phone: (800) 854-7179 or (303) 792-2181 Fax: (303) 792-2192

dpANS X3.296-199x
X3T11/95-469/Project 1132-D/Rev 2.3

SBCON

Single-Byte Command Code Sets
CONnection architecture
(SBCON)
REV 2.3

working draft proposed
American National Standard
for Information Systems

September 30, 1996

Secretariat:
Information Technology Industry Council

ABSTRACT: This standard describes an input/output (I/O) and interconnection architecture including specification of fiber optic links, switched point-to-point topology, and I/O protocols for high bandwidth, high performance and long distance information exchange.

NOTE:

This is a draft proposed American National Standard of Accredited Standards Committee X3. As such, this is not a completed standard. The X3T11 Technical Committee may modify this document as a result of comments received during public review and its approval as a standard.

POINTS OF CONTACT:

Roger Cummings (X3T11 Chairman)
Distributed Processing Technology
140 Candace Drive
Maitland, FL 32751
(407) 830-5522 x348 Fax: (407) 260-5366
E-Mail: cummings_roger@dpt.com

Robert Dugan (SBCON Working Group Chairman and SBCON Editor)
IBM Corporation
522 South Road
Poughkeepsie, NY 12601-5400
(914) 435-7801 Fax: (914) 432-9413
E-Mail: dugan@pkedvm9.vnet.ibm.com

Carl Zeitler (X3T11 Vice Chairman)
IBM Corporation - AWD, MS 9250
11400 Burnet Road, Austin, TX 78758
(512) 838-1797 Fax: (512) 838-3822
E-Mail: zeitler@ausvm6.vnet.ibm.com

Richard Taborek (SBCON Editor)
Amdahl Corporation, MS 119
1250 E. Arques Ave., Sunnyvale, CA 94088-3470
(408) 746-6533 Fax: (408) 746-7000
E-Mail: rich.taborek@ivorytower.amdahl.com

Changes from Revision 2.2:

- Incorporated acceptable comments from ANSI editor during pre-edit stage.
- Incorporated proposed resolutions to First Public Review comments.
- Updated Clause 2, Normative References, per proposed ANSI X3T11 guidelines.
- Added ANSI X3T11 committee list to Foreword.

draft proposed American National Standard
for Information Technology —
Single-Byte Command Code Sets
CONnection architecture (SBCON)

Secretariat

Information Technology Industry Council

Approved ,199

American National Standards Institute, Inc.

Abstract

This standard describes an input/output (I/O) and interconnection architecture including specification of fiber optic links, switched point-to-point topology, and I/O protocols for high bandwidth, high performance and long distance information exchange.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgement of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Published by

American National Standards Institute
11 W. 42nd Street, New York, New York 10036

Copyright © 199x by American National Standards Institute
All rights reserved

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

Printed in the United States of America

INSERT CODE HERE

Page

Page

Page

Annexes

Tables

Figures

Page

Foreword

(This Foreword is not part of dpANS X3.296-199x.)

This standard describes an input/output (I/O) and interconnection architecture including specification of fiber optic links, link-interconnection topologies, and I/O protocols for high bandwidth, high performance and long distance information exchange.

This standard was developed by Task Group X3T11 of Accredited Standards Committee X3 during 1992. The standards approval process started in 1994. This standard includes three annexes, which are informative, and are not considered part of the standard.

Requests for interpretation, suggestions for improvement or addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by Accredited Standard Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, the X3 Committee had the following members:

Richard Gibson, Chair
Donald C. Loughry, Vice-Chair
Joanne M. Flanagan, Secretary

NOTE – The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However neither the developers nor the publisher have undertaken a patent search in order to identify which if any patents may apply to this standard. No position is taken with respect to the validity of any claim or any patent rights that may have been disclosed. Details may be obtained from the publisher concerning any statement of patents and willingness to grant a license on a nondiscriminatory basis and with reasonable terms and conditions to applicants desiring to obtain such a license.

Organization Represented	*Name of Representative*
Allen-Bradley Company	Ronald H. Reimer Joe Lenner (Alt.)
American Library Association	Paul E. Peters
American Nuclear Society	Geraldine C. Main Sally Hartzell (Alt.)
AMP, Inc.	Edward R. Kelly Edward Mikoski (Alt.)
Apple Computer, Inc.	Karen Higginbottom
Association of the Institute of Certification of Computer Professionals	Kennety Zemrowski Eugene M. Dwyer (Alt.)
AT&T	Thomas F. Frost Paul D. Bartoli (Alt.)
Boeing Company	Catherine Howells Gail Dohmen (Alt.)
Bull HN Information Systems, Inc.	David M. Taylor
Compaq Computer Corporation	James L. Barnes Keith Lucke (Alt.)
Digital Equipment Corporation Users Society	Stephen C. Jackson Mike Terrazas (Alt.)
Digital Equipment Corporation	Gary S. Robinson Del Shoemaker (Alt.)
Eastman Kodak	James D. Converse Michael Nier (Alt.)

Electronic Data Systems Corporation	Jerrold S. Foley
	Charles M. Durrett (Alt.)
GUIDE International	Frank Kirshenbaum
	Harold Kuneke (Alt.)
Hewlett-Packard	Donald C. Loughry
Hitachi America Ltd.	Kei Yamashita
	John Neumann (Alt.)
Hughes Aircraft Company	Harold L. Zebrack
IBM Corporation	Robert H. Follett
	Mary Anne Gray (Alt.)
Lawrence Berkeley Laboratory	David F. Stevens
	Robert L. Fink (Alt.)
National Communications Systems	Dennis Bodson
	George W. White (Alt.)
National Institute of Standards and Technology	Robert E. Rountree
	Michael D. Hogan (Alt.)
NCR Corporation	Thomas W. Kern
	A. R. Daniels (Alt.)
OMNICOM, Inc.	Harold C. Folts
	Cheryl C. Slobodian (Alt.)
Open Software Foundation	Fritz Schulz
Recognition Technology Users Association	Herbert F. Schantz
	G. Edwin Hale (Alt.)
SHARE, Inc.	Thomas B. Steel Jr.
	Gary Ainsworth (Alt.)
Sony Corporation	Michael Deese
Storage Technology Corporation	Joseph S. Zajaczkowski
	Samuel Cheatham (Alt.)
Sun Microsystems, Inc.	Scott K. Jameson
3M Company	Paul D. Jahnke
Unisys Corporation	Stephen P. Oksala
	John L. Hill (Alt.)
U. S. Department of Defense	William C. Rinehuls
	Thomas E. Bozek (Alt.)
	Fred Virtue (Alt.)
U. S. General Services Administration	Douglas K. Arai
	Larry L. Jackson (Alt.)
US West Corporation	Gary Dempsey
	Susan Capraro (Alt.)
USE, Inc.	Pete Epstein
Wang Laboratories, Inc.	Steve Brody
	Sarah Wagner (Alt.)
Wintergreen Information Services	John L. Wheeler
Xerox Corporation	Roy Pierce

Task Group X3T11 on Device Level Interfaces, which developed this standard, had the following participants:

Roger Cummings, Chair
Carl Zeitler, Vice-Chair

l.Ahmed
D.Allan
D.Allen
T.Anderson
T.Asami
B.Atkinson
R.Bailly
C.Bazaar
S.Berman
C.Binford
T.Bohman
P.Boulay
C.Brill
J.Bumblis
E.Cady
J.Calle
K.Chan
E.Chang
M.Chenery
K.Chennappan
H.Chin
P.Chow
T.Clay
R.Cook
J.Coomes
R.Cornelius
R.Dahlgren
S.Darnell
J.Dedek
D.Deel
M.Dorsett
P.Duclos
R.Dugan
L.Duong
B.Edge
S.Erler
F.Ferguson
S.Finch
M.Fitzpatrick
M.Fogg
D.Ford
M.Foster
C.Foster
K.Fredericks
G.Freeman
E.Frymoyer
B.Gallagher
B.Galloway
E.Gardner
M.Griffin
E.Grivna
D.Hagerman
B.Ham
T.Harrington
N.Harris
B.Harvie
V.Haydu
D.Hepner
J.Himes
M.Hoard
E.Jacques
T.Johnson
S.Joiner
M.Karg
R.Kembel
R.Kleckowski
0.Kornblum
D.LaFollette
L.Lamers
P.Lawthers
M.Leib
B.Levin
T.Lindsay
J.Lohmeyer
P.Maher
K.Malavalli
W.Martin
J.Mathis
V.Melendy
J.Merkwaz
C.Mulvey
J.Murdock
K.Nakamura
J.Oliver
T.Palkert
J.Parker
R.Pedersen
R.Petersen
G.Porter
R.Prentice
R.Purohit
G.Rara
J.Renwick
C.Rice
C.Rich
W.Rickard
R.Ronald
B.Rumer
P.Rupert
J.Schaefer
P.Seto
J.Sherfey
M.Shiflett
M.Shriki
B.Smith
R.Snively
T.Sprenkle
J.Stai
G.Stephens
B.Stoevhase
A.Stone
S.Swirhun
R.Taborek
J.Thatcher
L.Thorsbakken
J.Toy
S.Ucci
H.Van Deusen
S.Van Doorn
T.Voss
R.Wagner
P.Walford
N.Wanamaker
G.Warden
L.Whitfield
J.Williams
S.Wilson
M.Wingard
K.Witte
M.Zandy
C.Zeitler

The designated editors of this document were:

Schelto van Doorn
Robert Dugan
Ken Fredericks
Phil Mills
Mike O’Donnell
Richard Taborek

Introduction

SBCON describes an input/output (I/O) and interconnection architecture including specification of fiber optic links, link-interconnection topologies, and I/O protocols for high bandwidth, high performance and long distance information exchange. SBCON describes a master/slave environment supporting a single data-link protocol designed to transport multiple Single-Byte Command Code Sets (SBCCS). Single-Byte Command Code Sets are widely implemented in data processing systems where high I/O bandwidth and performance are required. Single-Byte Command Code Sets are applicable to a wide variety of device types.

ANSI X3.230-1994, Fibre Channel, describes a multiprotocol I/O interface and it can be either master/slave or peer-to-peer. Fibre Channel provides extensions over SBCON, which support SBCCS, such as I/O operation multiplexing, connectionless service, scalable bandwidth, as well as concurrent support of other FC-4’s in addition to SBCCS. Because of its wide applicability, Fibre Channel may be considered as the evolution of the SBCON I/O Interface definition, and it is perceived that users of this standard would eventually migrate applications to Fibre Channel and ANSI X3.271-1996, FC-SB, which specifies the mapping of SBCCS to Fibre Channel.

draft proposed American National Standard for Information Systems —

Single-Byte Command Code Set CONnection (SBCON) architecture

1 Scope

SBCON describes an input/output (I/O) and interconnection architecture. SBCON specifies fiber optic links, switched point-to-point topology, and I/O protocols for high bandwidth, high performance and long distance information exchange.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents can be obtained from ANSI: Approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT, EIA/TIA), and approved and draft foreign standards (including BSI, JIS, and DIN). For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at http://www.ansi.org.

2.1 Approved references

ANSI X3.230-1994*, Fibre Channel Physical and Signaling Interface (FC-PH)*

ANSI X3.271-199x, *Fibre Channel—Single-Byte Command Code Sets (SBCCS) Mapping Protocol (FC-SB)*

ANSI Z136.2-1988, *Standard for the safe use of optical fibre communication systems utilizing laser diode and LED sources*

FOTP-6 (EIA/TIA-455-6B),[1,2] *Cable Retention Test Procedure for Fiber Optic Cable Interconnecting (March 1992)*

FOTP-27 (EIA/TIA-455-27), *Methods for Measuring Outside (Uncoated) Diameter of Optical Waveguide Fibers*

FOTP-29 (EIA/TIA-455-29A), *Refractive Index Profile, Transverse Interference Method: 1st Ed. Aug.1981, 2nd Ed. Oct. 1989. (Measures*

---

[1] All FOTP-xx are EIA/TIA-455-xxx and all OFSTP-xx are EIA/TIA-526-xxx. All FOTP and OFSTP references are as of 12/16/92. Note that some are listed as EIA-zzz-xx and some as EIA/TIA-zzz-xx. The reason for this is related to timing of the document development and/or revision since TIA became an accredited organization.

[2] Fiber Optic Test Procedure (FOTP) and Optical Fiber System Test Practice (OFSTP) standards are developed and published by the Electronics Industries Association under the EIA/TIA-455 and the EIA/TIA-526 series of standards. Copies may be obtained by contacting:

Global Engineering, 15 Iverness Way East,
Englewood, CO 80112-5704
Phone:(800) 854-7179 or (303) 792-2181, Fax:(303) 792-2192

core diameter, numerical aperture, and refractive index profile of multimode fiber) Reaffirmed 04/01/91 until 10/94

FOTP-30 (EIA/TIA-455-30B), *Frequency Domain Measurement of Multimode Optical Fiber Information Transmission Capacity: 1st Ed. Sept. 1982, 2nd Ed. Aug. 1988, 3rd Ed. Oct. 1991*

FOTP-34 (EIA/TIA-455-34), *Interconnection Device Insertion Loss Test, May 1985*

FOTP-44 (EIA/TIA-455-44A), *Refractive Index Profile, Refracted Ray Method: 1st Ed. Jan. 1984, 2nd Ed. Oct. 1989, 3rd Ed. Sept. 1992. (Measures core diameter, numerical aperture, and refractive index profile of multimode fiber)*

FOTP-45 (EIA/TIA-455-45B), *Method for Measuring Optical Fiber Geometry Using a Laboratory Microscope: 1st Ed. Sept. 1984, 2nd Ed. Aug, 1988, 3rd Ed. June 1992*

FOTP-47 (EIA/TIA-455-47B), *Output Farfield Radiation Pattern Measurement: 1st Ed. Sept. 1983, 2nd Ed. May 1989, 3rd Ed. Aug. 1992 (Measures numerical aperture of multimode fiber)*

FOTP-48 (EIA/TIA-455-48B), *Diameter Measurement of Optical Fibers Using Laser-Based Measurement Instruments: 1st Ed. Dec. 1983, 2nd Ed. Oct. 1987, 3rd Ed. Dec. 1990*

FOTP-51 (EIA/TIA-455-51A), *Pulse Distortion Measurement of Multimode Glass Optical fiber Information Transmission Capacity: 1st Ed. Sept. 1983, 2nd Ed. May 1991*

FOTP-54 (EIA/TIA-455-54A), *Mode Scrambler Requirements for Overfilled Launching Conditions to Multimode Fibers: 1st Ed. Sept. 1982, 2nd Ed. Nov. 1990*

FOTP-58 (EIA/TIA-455-58A), *Core Diameter Measurement of Graded-Index Optical Fibers, Nov. 1990*

FOTP-80 (EIA/TIA-455-80), *Cutoff Wavelength of Uncabled Single-Mode Fiber by Transmitted Power*

FOTP-107 (EIA/TIA-107), *Return Loss for Fiber Optic Components (February 1989)*

FOTP-127 (EIA/TIA-455-127), *Spectral Characteristics of Multimode Laser Diodes Performance, Nov. 1991*

FOTP-164 (EIA/TIA-455-164), *Single-Mode Fiber, Measurement of Mode Field Diameter by Far-Field Scanning*

FOTP-167 (EIA/TIA-455-167), *Mode Field Diameter Measurement - Variable Aperture Method in the Far-Field*

FOTP-168 (EIA/TIA-455-168A), *Chromatic Dispersion Measurement of Multimode Graded-Index and Single-Mode Optical Fibers by Spectral Group Delay Measurement in the Time Domain: 1st Ed. July 1987, 2nd Ed. March 1992*

FOTP-170 (EIA/TIA-455-170), *Cable Cutoff wavelength of Single-Mode Fiber by Transmitted Power*

FOTP-171 (EIA/TIA-455-171), *Attenuation by Substitution Measurement- For Short Length Multimode and Single-Mode Fiber Cable Assemblies, July 1986*

FOTP-176 (EIA/TIA-455-176), *Measurement Method for Optical Fiber Geometry by Automated Grey-Scale Analysis.*

FOTP-177 (EIA/TIA-455-177), *Numerical Aperture Measurement of Graded-Index Optical Fibers: 1st Ed. Nov. 1989, 2nd Ed. Aug. 1992 (“Umbrella” document, indicating factors required by FOTP-29, FOTP-44, and FOTP-47 to map to each other)*

FOTP-187 (EIA/TIA-455-187), *Engagement and Separation Force Measurement of Fiber Optic Connector Sets, June 1991*

EIA/TIA-492AAAA, *Detail Specification for 62.5 µm Core Diameter/125 µm Cladding Diameter Class 1a Multimode, Graded Index Optical Waveguide Fibers*

EIA/TIA-492BAAA, *Detail Specification for Class IVa Dispersion-Unshifted Single-Mode Optical Waveguide Fibers Used in Communications Systems*

OFSTP-2 (EIA/TIA-526-2), *Effective Transmitter Output Power Coupled into Single-Mode Fiber Optic Cable, Sept.1992*

OFSTP-3 (EIA/TIA-526-3), *Fiber Optic Terminal Equipment Receiver Sensitivity and Maximum Receiver Input, Oct. 1989*

OFSTP-4 (EIA/TIA-526-4), *Optical Eye Pattern Measurement Procedure (in final approval stage)*

OFSTP-7 (EIA/TIA-526-7), *Optical Power Loss Measurement of Installed Single-Mode Fiber Cable Plant, Jan. 1993 Draft (unpublished)*

OFSTP-11 (EIA/TIA-526-11), *Measurement of Single-Reflection Power Penalty for Fiber Optic Terminal Equipment, Dec. 1991*

OFSTP-14 (EIA/TIA-526-14), *Optical Power Loss Measurement of Installed Multimode Fiber Cable Plant, Nov. 1990*

EIA/TIA 568, *Commercial Building Telecommunications Wiring Standard.*

IEC 825-1984, *Radiation safety of laser products, equipment classification, requirements and user's guide*

JIS C 5973, *FO4 Type Connectors for Optical Fiber Cards*

2.2 Other References

Food and Drug Administration (FDA)/Department of Health and Human Services (DHHS) Regulations 21 CFR Chapter I, Subchapter J, Part 1040.10, Performance standards for light-emitting products

3 Definitions and conventions

For SBCON, the following definitions, conventions, abbreviations, acronyms, and symbols apply.

3.1 Definitions

The following cross-references are used in these definitions:

- **contrast with.** This refers to a term that has an opposed or substantively different meaning.
- **see.** This makes reference to multiple-word terms in which this term appears.
- **see also.** This makes reference to terms that have a related, but not synonymous, meaning.
- **synonym.** This indicates that the term has the same meaning as a preferred term, which is defined here.

3.1.1 adapter: (1) Hardware that provides some transitional functions between two or more devices; (2) A mechanism for attaching parts, for example, parts having different diameters; (3) In an SBCON environment, link hardware used to join different optical fiber connector types. Contrast with *coupler*.

3.1.2 attenuation: The power loss expressed in units of dB.

3.1.3 attenuation coefficient: In fiber optics, the rate of decrease in magnitude of average optical power with respect to distance along the fiber, usually expressed in decibels per kilometer (dB/km).

3.1.4 average power: The optical power measured using an average reading power meter when an SBCON I/O interface is transmitting a specified code sequence as defined in the test procedure.

3.1.5 bandwidth: Maximum effective transfer rate for a given set of physical variants such as communication model, payload size, link baud rate, and overhead specified by SBCON protocol.

3.1.6 Baud: The symbol rate per second.

3.1.7 bit error rate (BER): The statistical probability of a transmitted bit being erroneously received in a communication system. The BER is measured by counting the number of erroneous bits at the output of a receiver and dividing by the total number of bits.

3.1.8 byte: A group of eight bits.

3.1.9 cable plant: All passive communications elements (e.g., optical fiber, connectors, splices, etc.) between a transmitter and a receiver.

3.1.10 calculated link loss: In an SBCON environment, the total optical attenuation (loss) calculated for a specific link, the value of which cannot be more than the maximum loss allowed for that link. See also *maximum allowable link loss*.

3.1.11 center wavelength (laser): The nominal value of the central wavelength of the operating, modulated laser. This is the wavelength (see FOTP-127) where the effective optical power resides.

3.1.12 center wavelength (LED): The average of the two wavelengths measured at the half amplitude points of the power spectrum.

3.1.13 channel: The entity that controls one channel path. Each channel controls an interface to one or more control units.

3.1.14 channel-command-word (CCW): (1) A control block which contains an I/O request; (2) A structure of a specific system architecture which specifies the command to be executed along with parameters.

3.1.15 channel image: A single instance of a channel having the logical appearance of a channel.

3.1.16 channel path: A single interface between a channel subsystem and one or more control units along which signals and data can be sent to perform I/O operations.

3.1.17 channel program: One or more channel command words that control a specific sequence of channel operations.

3.1.18 channel subsystem (CSS): An entity including one or more channels which contains common facilities for the control of I/O operations.

3.1.19 character: Any transmission character associated by transmission code with a data byte or special code.

3.1.20 checking block code (CBC): One or more redundant bits, called check bits, appended to a group of data bits. CBC is separate from customer data. Contrast with *cyclic redundancy check*.

3.1.21 cladding. In an optical cable, the region of low refractive index surrounding the core. See also *optical fiber*.

3.1.22 code bit: The smallest time period used for transmission on the media.

3.1.23 code balance: The numerical sum of the 1 bits in any 10 bits in the transmitted bit stream divided by 10 (e.g., the bit stream b‘1110100011’ has a code balance of 6/10 = 60%).

3.1.24 code violation: An error condition that occurs when a received transmission character cannot be decoded to a data byte or special-code using the validity checking rules specified by the transmission-code.

3.1.25 comma: The seven bit sequence ‘0011111’ or ‘1100000’ in an encoded stream.

3.1.26 comma character: A transmission character containing a comma.

3.1.27 connection: (1) Between a channel and control unit, an association established after the successful exchange of a request/response frame pair containing the proper delimiters. The delimiters include a connect-SOF delimiter in the request frame and both passive-SOF and passive-EOF delimiters in the response frame; (2) In a dynamic switch, an association established between two switch ports that provides a communication path between them. See also *dynamic connection* and *static connection*.

3.1.28 control unit (CU): A entity that controls the reading, writing, or displaying of data at one or more devices. Devices are indirectly attached to channels through a control unit.

3.1.29 control unit image: A single instance of a control unit having the logical appearance of a control unit.

3.1.30 coupler: In an SBCON environment, link hardware used to join identical optical fiber connector types. Contrast with *adapter*.

3.1.31 current running disparity: The running disparity present at a transmitter when encoding of a data byte or special-code is initiated, or at a receiver when decoding of a transmission character is initiated.

3.1.32 cutoff wavelength: In fiber optics, the wavelength at which a particular waveguide mode ceases to be a bound mode.

NOTE – In a single-mode fiber, concern is with the cutoff wavelength of the second order mode.

3.1.33 cyclic redundancy check (CRC): (1) A redundancy check in which the check key is generated by a cyclic algorithm; (2) A system of error checking performed at both the sending and receiving station after a block check character has been accumulated. Contrast with *checking block code*.

3.1.34 data character: Any transmission character considered valid by the transmission code and equated to a valid data byte (i.e., not a special character).

3.1.35 decoding: Validity checking of received transmission characters and generation of valid data bytes and special codes from those characters.

3.1.36 delimiter: An ordered set used to indicate a frame boundary.

3.1.37 device: A mechanical, electrical, or electronic contrivance with a specific purpose.

3.1.38 device addressing: In an SBCON I/O interface, one of two levels of addressing, this level pertaining to an I/O device and identifying that device to the channel or control unit that has been determined through link-level addressing. Contrast to *link-level addressing*.

3.1.39 disparity: The difference between the number of ones and zeros in a transmission character.

3.1.40 dispersion: A term used to denote pulse broadening and distortion. The two general categories of dispersion are modal dispersion, due to the difference in the propagation velocity of the propagation modes in a multimode fiber, and chromatic dispersion, due to the difference in propagation of the various spectral components of the optical source.

3.1.41 dispersion-unshifted fiber: A single-mode fiber that has a nominal zero-dispersion wavelength in the 1300 nanometer transmission window, and has a dispersion coefficient approximated by a differentiated Sellmeier equation.

3.1.42 distribution panel: In an SBCON environment, a panel that provides a central location for the attachment of trunk and jumper cables and can be mounted in a rack or wiring closet, or on a wall.

3.1.43 duplex connector: In an SBCON environment, an optical fiber component that terminates two cable fibers in one housing and provides physical keying for attachment to a duplex receptacle.

3.1.44 duplex receptacle: In an SBCON environment, a fixed or stationary optical fiber component that provides physical keying for attachment to a duplex connector.

3.1.45 dynamic connection: In a dynamic switch, a connection between two switch ports, established or removed by the switch and that, when active, appears as one continuous link. The duration of the connection depends on the protocol defined for the frames transmitted through the switch ports and on the state of the switch ports Contrast with *static connection.*

3.1.46 dynamic connectivity: In a dynamic switch, the capability that allows connections to be established and removed at any time.

3.1.47 dynamic switch: A hardware unit that provides the capability to physically interconnect any two links that are attached to it.

3.1.48 dynamic-switch port: The link attachment point on a dynamic switch. Synonym: *switch port*.

3.1.49 encoding: Generation of transmission characters from valid data bytes and special codes.

3.1.50 error: A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. Contrast with *failure* and *fault*.

3.1.51 excess connector loss: The link loss associated with connecting a single-mode jumper cable (nominal MFD of 9 µm) to a single-mode trunk cable having a nominal MFD of 10 µm.

3.1.52 extended binary-coded decimal interchange code (EBCDIC): A coded character set consisting of 8-bit coded characters.

3.1.53 extinction ratio: The ratio (in dB) of the average optical energy in a logic one level to the average optical energy in a logic zero level measured under modulated conditions at the specified baud rate.

3.1.54 eye opening: The time interval across the eye, measured at the 50% normalized eye amplitude.

3.1.55 failure: An uncorrected hardware error. Contrast with *error* and *fault*.

3.1.56 fault: An accidental condition that causes a functional unit to fail to perform its required function. Contrast with *error* and *failure*.

3.1.57 fiber: See *optical fiber*.

3.1.58 fiber optic cable: See *optical cable*.

3.1.59 fiber optic subassembly. A component that contains a serializer, deserializer, transmitter, and receiver. See also *transmitter-receiver subassembly (TRS)*.

3.1.60 fiber optic test procedure (FOTP): Standards developed and published by the Electronic Industries Association (EIA) under the EIA-RS-455 series of standards.

3.1.61 frame: The major unit of transmission on a link. Frames are used to transfer control information, data, commands, or status between a channel and control unit.

3.1.62 frame reception: The act of receiving a frame over a link performed by a link-level facility after the receiver has acquired synchronization.

3.1.63 full duplex: Pertaining to transmission in both directions at the same time. Contrast with *half duplex*.

3.1.64 fusion splice: In fiber optics, a splice accomplished by the application of localized heat sufficient to fuse or melt the ends of two lengths of optical fiber, forming a continuous, single fiber. Contrast with *mechanical splice*.

3.1.65 graded-index fiber: An optical fiber with a refractive index that varies with radial distance from the center of the fiber.

3.1.66 half duplex: Pertaining to transmission in only one direction at a time. Contrast with *full duplex*.

3.1.67 idle-function: The occurrence of a K28.5 character that is not part of an ordered set.

3.1.68 ignored: A field that is not interpreted by the receiver.

3.1.69 image: A group of related processes. Examples of an image are a single system or a single logical partition of a system behind an interface.

3.1.70 image pair: The relationship between two images behind separate interfaces.

3.1.71 initialization: The operations required for setting a system or device to a starting state, before the use of a data medium, or before implementation of a process.

3.1.72 initiation frame: A frame defined to make a connection.

3.1.73 input/output (I/O): (1) Pertaining to a device whose parts can perform an input and/or output process; (2) Pertaining to a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process.

3.1.74 insertion loss: In fiber optics, the total optical power loss caused by insertion of an optical component such as a connector, splice, or coupler.

3.1.75 interface: A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics as appropriate. The concept includes the specification of the connection of two entities having different functions.

3.1.76 interface connection: An optical connector which connects the media to the SBCON transmitter or receiver. The connection consists of a receptacle and a plug.

3.1.77 intersymbol interference: The effect on a sequence of symbols in which the symbols are distorted by transmission through a limited bandwidth medium to the extent that adjacent symbols interfere with each other.

3.1.78 I/O operation: An operation involving the transfer of data between a channel and I/O device.

3.1.79 jacket: In an optical cable, the outermost layers of protective covering.

3.1.80 initial program load (IPL): The initialization procedure that causes an operating system to commence operation**.**

3.1.81 jitter: Deviations from the ideal timing of an event which occur at high frequencies. Low frequency deviations are tracked by the clock recovery and do not directly affect the timing allocations within a bit cell. Jitter is not tracked by the clock recovery and directly affects the timing allocations in a bit cell. For SBCON the lower cutoff frequency for jitter is defined as the bit rate divided by 2500. Jitter is customarily subdivided into deterministic and random components.

3.1.82 jitter, deterministic (DJ): Timing distortions caused by normal circuit effects in the transmission system. Deterministic jitter is often subdivided into duty cycle distortion (DCD) caused by propagation differences between the two transitions of a signal and data dependent jitter (DDJ) caused by the interaction of the limited bandwidth of the transmission system components and the symbol sequence.

3.1.83 jitter, random (RJ): Jitter due to thermal noise which may be modeled as a Gaussian process. The peak-to-peak value of RJ is of a probabilistic nature and thus any specified value yields an associated BER.

3.1.84 jumper cable: In an SBCON environment, an optical cable having two conductors that provides physical attachment between two devices or between a device and a distribution panel. Contrast with *trunk cable*.

3.1.85 jumper assembly loss: The loss associated with connecting a jumper cable assembly between two reference cables compared to connecting the reference cables directly.

3.1.86 laser chirp: A phenomenon in lasers where the wavelength of the emitted light changes during modulation.

3.1.87 link: (1) In an SBCON environment, the physical connection and transmission medium used between an optical transmitter and an optical receiver. A link consists of two conductors, one used for sending and the other for receiving, thereby providing a duplex communication path. (2) In an SBCON I/O interface, the physical connection and transmission medium used between a channel and control unit, a channel and a dynamic switch, a control unit and a dynamic switch, or between two dynamic switches.

3.1.88 link-level facility: The SBCON I/O interface entity that manages the attached link, including the transmitter and the receiver. Processes performed by the link-level facility include: link-level initialization, offline, connection-recovery, and link-failure operations; performs link-level functions and protocols; and manages logical paths.

3.1.89 link-level addressing: In an SBCON I/O interface, one of two levels of addressing, this level pertaining to link-level functions and identifying the channel path between the channel and control unit. Contrast to *device addressing*.

3.1.90 maximum allowable link loss: In an SBCON environment, the maximum amount of link attenuation (loss), expressed in decibels, that can exist without causing a possible failure condition. See also calculated link loss.

3.1.91 mechanical splice: In fiber optics, a splice accomplished by fixtures or materials rather than thermal fusion. Index matching material can be applied between two fiber ends. Contrast with fusion splice.

3.1.92 mode field diameter (MFD): A measure of the width of the guided optical power’s intensity distribution in a single-mode fiber.

3.1.93 mode-partition noise: Noise in a laser-based optical communication system caused by the changing distribution of laser energy partitioning itself among the laser modes (or lines) on successive pulses in the data stream. The effect is a different center wavelength for the successive pulses resulting in arrival time jitter attributable to chromatic dispersion in the fiber.

3.1.94 multimode optical fiber: A graded-index or step-index optical fiber that allows more than one bound mode to propagate. Contrast with single-mode optical fiber.

3.1.95 numerical aperture: The sine of the radiation or acceptance half angle of an optical fiber, multiplied by the refractive index of the material in contact with the exit or entrance face (see FOTP-177).

3.1.96 optical cable: A fiber, multiple fibers, or a fiber bundle in a structure built to meet optical, mechanical, and environmental specifications. See also *jumper cable, optical cable assembly,* and *trunk cable*.

3.1.97 optical cable assembly: An optical cable that is connector-terminated. Generally, an optical cable that has been terminated by a manufacturer and is ready for installation. See also *jumper cable* and *optical cable*.

3.1.98 optical fall time: The time interval required for the falling edge of an optical pulse to transition between specified percentages of the signal amplitude. For lasers the transitions are measured between the 80% and 20% points. For LED media the specification points are 90% and 10%.

3.1.99 optical fiber: Any filament or fiber, made of dielectric material, that guides light.

3.1.100 optical fiber splice: A joint that couples optical power between two fibers. See also *fusion splice* and *mechanical splice*.

3.1.101 optical fiber system test practice (OFSTP): Standards developed and published by the Electronic Industries Association (EIA) under the EIA/TIA-526 series of standards.

3.1.102 optical path penalty: A link penalty to account for those effects other than attenuation.

3.1.103 optical power: Synonym for *radiant power*.

3.1.104 optical receiver: Hardware that converts an optical signal to an electrical logic signal. Contrast with *optical transmitter*.

3.1.105 optical reference plane: The plane that defines the optical boundary between the plug and the receptacle.

3.1.106 optical return loss (ORL): The ratio (expressed in units of dB) of optical power incident upon a component port or an assembly to the optical power reflected by that component when that component or assembly is introduced into a link or system.

3.1.107 optical rise time: The time interval required for the rising edge of an optical pulse to transition between specified percentages of the signal amplitude. For lasers the transitions are measured between the 20% and 80% points. For LED media the specification points are 10% and 90%.

3.1.108 optical transmitter: Hardware that converts an electrical logic signal to an optical signal. Contrast with *optical receiver*.

3.1.109 optional: Features that are not required by the standard. However, if any optional feature defined by the standard is implemented, it shall be implemented according to the standard.

3.1.110 ordered set: Specific combinations of special characters and data characters which provide frame delineation, control of dynamic connections, and synchronization between the transmitter and receiver circuits at opposite ends of a link.

3.1.111 parallel-I/O interface: A precursor to the SBCON I/O interface (a.k.a. FIPS-60, OEMI).

3.1.112 physical-contact connector: In an SBCON environment, an optical fiber connector type having no air gap, thereby providing a low-loss junction point.

3.1.113 plug: The male cable half of the interface connector which terminates an optical signal transmission cable.

3.1.114 point-to-point topology: A network topology that provides one bi-directional communication path between a channel and control unit and does not include switching facilities. Contrast with *switched-point-to-point topology*.

3.1.115 radiant power: In fiber optics, the time rate of flow of radiant energy, expressed in watts. The prefix is often dropped and the term power is used.

3.1.116 receiver: (1) The portion of a link-level facility or dynamic-switch port dedicated to receiving an encoded bit stream from a fiber, converting this bit stream into transmission characters, and decoding these characters using the rules specified by SBCON. (2) An electronic circuit (Rx) that converts a signal from the media to an electrical serial logic signal.

3.1.117 receiver overload: The condition of exceeding the maximum acceptable value of the received average optical power to achieve a BER < 10^{-12}.

3.1.118 receiver sensitivity: The minimum acceptable value of average received signal to achieve a BER < 10^{-12}. It takes into account power penalties caused by use of a transmitter with a worst-case output. In the case of an optical path it does not include power penalties associated with dispersion, jitter, effects related to the modal structure of the source or reflections from the optical path. These effects are specified separately in the allocation of maximum optical path penalty.

3.1.119 receptacle: The female half of the interface connector.

3.1.120 reflections: Power returned by discontinuities in the physical link.

3.1.121 reserved: A field whose content and checking is specified by the accompanying text. Each bit in the reserved field is denoted by “r”.

NOTE – Reserved fields should not be checked or interpreted unless otherwise noted.

3.1.122 reserved ordered set: An ordered set not defined to have meaning by SBCON.

3.1.123 return loss: See optical return loss.

3.1.124 RIN_{12}: Laser noise in dB/Hz measured relative to the average optical power with 12dB return loss.

3.1.125 run length: Number of consecutive identical bits in the transmitted signal (e.g., the pattern 0011111010 has a maximum run length of five).

3.1.126 running disparity: A binary parameter indicating the cumulative disparity (positive or negative) of all previously issued transmission characters.

3.1.127 SBCON environment: The environment having an SBCON channel-to-control-unit I/O interface that uses optical cables as a transmission medium.

3.1.128 SBCON I/O interface: A type of channel path which allows attachment of one or more control units, with or without a dynamic switch, to a channel subsystem by using optical fiber links and provides protocols required for information transfer over the channel path.

3.1.129 sequence: A consecutive group of defined ordered sets that specify unique functions.

3.1.130 single-mode optical fiber: An optical fiber in which only the lowest-order bound mode (which can consist of a pair of orthogonally polarized fields) can propagate at the wavelength of interest. Contrast with *multimode optical fiber*.

3.1.131 special character: Any transmission character considered valid by the transmission code but not equated to a valid data byte. Special characters are provided by the transmission code for use in denoting special functions.

3.1.132 special code: A code which, when encoded using the rules specified by the transmission code, results in a special character. Special codes are typically associated with control signals related to protocol management (e.g., K28.5).

3.1.133 spectral width: (1) FWHM (Full Width Half Maximum) - The absolute difference between the wavelengths at which the spectral radiant intensity is 50 percent of the maximum power. This form is typically used for LED optical sources; (2) RMS - The weighted root mean square width of the optical spectrum. See FOTP-127. This form is typically used for laser optical sources.

3.1.134 splice: See *optical fiber splice*.

3.1.135 splice loss: Synonym for *insertion loss*.

3.1.136 static connection: In a dynamic switch, a connection between two switch ports that is not affected by transmission frames or sequences. This connection, which restricts those switch ports from communicating with any other switch ports, is established or removed by external means. Contrast with *dynamic connection*

3.1.137 strength member: In an optical cable, material that can be located either centrally or peripherally and that functions as a strain relief.

3.1.138 switched-point-to-point topology: A network topology that uses switching facilities to provide multiple communication paths between a channels and control units. Contrast with *point-to-point topology*.

3.1.139 symbol: The smallest entity transmitted on the media.

3.1.140 synchronization: Receiver identification of a transmission character boundary.

3.1.141 transmission character: Any encoded character (valid or invalid) transmitted across a physical interface. Valid transmission characters are specified by the transmission code and include data and special characters.

3.1.142 transmission code: A means of encoding data to enhance its transmission characteristics. The transmission code specified by SBCON is byte-oriented, with (1) valid data bytes and (2) special codes encoded into 10-bit transmission characters.

3.1.143 transmitter: (1) The portion of a link-level facility or dynamic-switch port dedicated to converting valid data bytes and special codes into transmission characters using the rules specified by the transmission code, converting these transmission characters into a bit stream, and transmitting this bit stream onto the transmission medium. (2) An electronic circuit (Tx) that converts an electrical logic signal to a signal suitable for the communications media.

3.1.144 transmitter-receiver subassembly (TRS): In an SBCON environment, the component that contains an optical transmitter and an optical receiver. See also *fiber optic subassembly*.

3.1.145 transceiver: A transmitter and receiver combined in one package.

3.1.146 trunk cable: In an SBCON environment, a cable consisting of multiple fiber pairs that do not directly attach to an active device. This cable usually exists between distribution panels and can be located within, or external to, a building. Contrast with *jumper cable*.

3.1.147 Upper Level Protocol (ULP): A protocol level immediately above the SBCON I/O interface.

3.1.148 valid frame: A frame received with a valid Start_of_Frame (SOF), a valid End_of_Frame (EOF), valid data characters, and proper Cyclic Redundancy Check (CRC) of the Information Field.

3.1.149 vendor specific: Functions, code values, and bits not defined by SBCON and set aside for private usage between parties using SBCON. Caution: Different implementations of SBCON may assign different meanings to these functions, code values, and bits.

3.2 Editorial conventions

In SBCON, a number of conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the following convention:

– a term consisting of multiple words with all letters lowercase and each word separated from the other by a dash (-) character. A word may also consist of an acronym or abbreviation which would be printed in uppercase. (e.g., link-level, CUE-with-busy, etc.).

All terms and words not conforming to the convention noted above have the normal technical English meanings.

Numbered items in SBCON do not represent any priority. Any priority is explicitly indicated.

In case of any conflict between figure, table, and text, the text, then tables, and finally figures take precedence. Exceptions to this convention are indicated in the appropriate sections.

In all of the figures, tables, and text of this document, the most significant bit of a binary quantity is shown on the left side. Exceptions to this convention are indicated in the appropriate sections.

The term *shall* is used to indicate a mandatory rule. If such a rule is not followed, the results are unpredictable unless indicated otherwise.

The fields or control bits which are not applicable shall be reset to zero.

If a field or a control bit in a frame is specified as *ignored*, the entity which receives the frame shall not check that field or control bit.

3.2.1 Binary notation

Binary notation is used to represent relatively short fields. For example, a three-bit flag-field-code containing a binary value of 010 is shown in binary format as b’010’.

3.2.2 Hexadecimal notation

Hexadecimal notation is used to represent relatively long fields. For example, a two-byte Source_Address field containing a binary value of 11000100 00000011 is shown in hexadecimal format as x’C403’.

3.3 Abbreviations, acronyms and symbols

Abbreviations, acronyms and symbols applicable to this International Standard are listed. Definitions of several of these items are included in 3.1. The index at the end of the standard provides help in locating these terms in the body of the standard.

3.3.1 Acronyms and other abbreviations

ACK	acknowledgment link-control function
AE	addressable element
ALA	acquire-link-address link-control function
AS	address-specific
BER	bit error rate
CBC	checking-block code
CC	chain-command
CCW	channel-command-word
CD	chain-data
CE	channel-end
CH	channel *and* chaining
CHPID	channel-path identifier
CI	channel-initiated
CIW	command-information-word
CL	class
CMR	command-response frame
CP	communication point
CR	command-retry
CRC	cyclic redundancy check
CSOF	connect-start-of-frame delimiter
CSS	channel subsystem
CT	command type
CTCA	channel-to-channel adapter
CU	control unit
CUB	control-unit-busy
CUE	control-unit-end
DASD	direct access storage device
dB	decibel
dBm	decibel (relative to 1 mw power)
DDTO	device-dependent-time-out
DE	device-end
DEOF	disconnect-end-of-frame delimiter
DHF	device-header-flag
DIB	device-information-block
DJ	deterministic jitter
DR	data-request
DU	data-chaining-update
E	end
EB	end-block
EBCDIC	extended binary-coded decimal interchange code
EIA	Electronic Industries Association
EMC	electromagnetic compatibility
EOF	end-of-frame
ELP	establish-logical-path link-control function
ES	supplemental-status
ET	entry type
EX	exponent
FFC	flag-field code
FLA	full link address
FRU	field-replaceable unit
GHz	GigaHertz = 1 billion cycles per second
hex	hexadecimal notation
HOML	high-order mode loss
Hz	Hertz = 1 cycle per second
IC	incident code
IDR	identifier-response link-control function
IFI	information-field identifier
IPL	initial program load
IPT	interface-protocol type
IQ	incident qualifier
ISA	identifiable subassembly
kHz	kiloHertz = 100,000 cycles per second
km	kilometer
L	level
LED	light emitting diode
LBY	link-level-busy link-control function
LED	light emitting diode
LID	link-incident-data link-control function
LIN	link-incident-notification link-control function
LOL	loss of light
LPE	logical-path-established link-control

	function
LPR	logical-path-removed link-control function
LR	long-record
LRJ	link-level-reject link-control function
LW	long wavelength
m	meter
Mb	Mega bit
MB	Mega Byte
MBaud	Mega Baud
MBd	Mega Baud
MFD	mode field diameter
MHz	MegaHertz = 1 million cycles per second
MIHPTO	missing-interrupt-handler-primary-time-out
MIHSTO	missing-interrupt-handler-secondary-time-out
MM	multimode
ms	millisecond
µs	microsecond
NA	numerical aperture
N/A	not applicable
ND	node descriptor
NDR	number-of-data-requests
NDR-R	number-of-data-requests, read operation
NDR-W	number-of-data-requests, write operation
NED	node-element descriptor
NEQ	node-element qualifier
nm	nanometer
NOS	not-operational sequence
NQ	node qualifier
ns	nanosecond
NS	node selector
OFSTP	optical fiber system test practice
ORL	optical return loss
OLS	offline sequence
PBY	port-busy link-control function
PEOF	passive-end-of-frame delimiter
ppm	parts per million
PRJ	port-reject link-control function
ps	picosecond
PSOF	passive-start-of-frame delimiter
QTUF	queuing-time-unit factor
R	reserved
RC	request-command-retry
RD	running disparity
RDY	ready
RFI	Radio Frequency Interference
RID	request-node-identifier link-control function
RIIN	reflection induced intensity noise
RIN	relative intensity noise
RIR	request-incident-record link-control function
RJ	random jitter
RLP	remove-logical-path link-control function
RMS	root mean square
RO	request-unit-check-with-overrun
RU	request-unit-check
Rx	receiver
SA	supplemental-status-available
SAT	subassembly type
SBCCS	Single-Byte Command Code Sets
SCI	statically connected switch interface
SCN	state-change-notification link-control function
SDC	self-describing component
SM	single mode
S/N	signal-to-noise ratio
SOF	start-of-frame
TIN	test-initialization link-control function
TIR	test-initialization-result link-control function
TRS	transmitter-receiver subassembly
Tx	transmitter
UC	unit-check
UD	unconditional-disconnect sequence
UDR	unconditional-disconnect-response sequence
UI	unit interval = 1 bit period
ULP	upper level protocol
XV	transfer-count-valid

3.3.2 Symbols

Unless indicated otherwise, the following symbols have the listed meaning.

8B/10B	8-bit to 10-bit code
µ	micro (e.g., µm = micrometer)
λ	wavelength

4 I/O interface overview

The SBCON I/O interface is a type of channel path which allows attachment of one or more control units to the channel subsystem by using optical fiber links and provides protocols required for information transfer over that channel path. This standard provides a description of the physical and logical interface and the protocols which govern the transfer of information over the interface.

4.1 Channel-path elements

The channel path provides the communication path between a channel and one or more control units. The physical elements that make up a channel path are a channel, possibly one or two dynamic switches, one or more control units, and one or more links.

4.1.1 Channel

The channel subsystem directs the transfer of information between I/O devices and main storage and provides the common controls for the attachment of different I/O devices by means of one or more channel paths.

A channel image, illustrated in figure 1, has the logical appearance of a channel. Each channel image appears to be an independent channel although all channel images on a specific SBCON I/O interface share the same facilities and physical paths. The channel facilities common to a SBCON I/O interface perform certain functions for all sharing channel images (for example, link synchronization and acquisition of a link address) and may perform functions for a single sharing channel image (for example, establishment of a logical path and performance of device-level functions). When the channel facilities common to multiple channel images perform a function for a single channel image, these facilities are dedicated to that channel image for the time necessary to perform the function; when no longer needed, these facilities are available to be used by another sharing channel image.

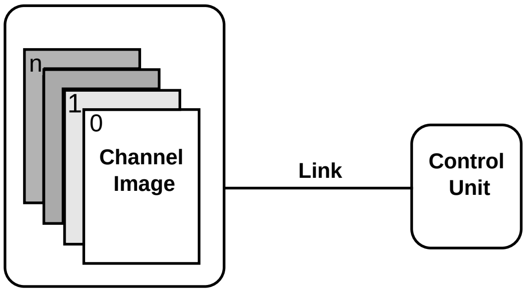


Figure 1 – Channel images

The term channel is used when describing actions and functions performed by the channel facilities on behalf of either all sharing channel images or one channel image. The channel image describes a particular configuration and the relationships among logical images in that configuration. The term channel is used throughout this document to mean a channel for the SBCON I/O interface, that is, a channel that meets the requirements specified in this standard.

Certain manual controls are included with the channel. In particular, the online/offline state of the link interface and powering on and off are subject to manual controls. The number, type, and operation of these controls are model-dependent. An SBCON system architecture example is shown in figure 2.

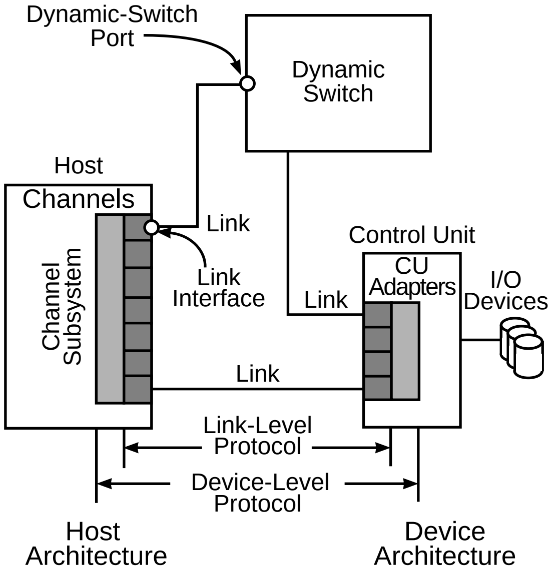


Figure 2 – System architecture example

4.1.2 Link

The transmission medium for the SBCON I/O interface is called a link. A link is a point-to-point pair of conductors (optical fibers) that physically interconnects a control unit and a channel, a channel and a dynamic switch, a control unit and a dynamic switch, or, in some cases, a dynamic switch and another dynamic switch. The two conductors of a link provide a simultaneous two-way communication path. One conductor is for transmitting information and the other is for receiving information. The physical elements of the link are described in clause 5. A link is attached to a channel or control unit by means of the link interface of that channel or control unit and to a dynamic switch by means of a dynamic-switch port.

4.1.3 Control unit

A control unit for the SBCON I/O interface provides the logical capability necessary to operate and control one or more I/O devices and adapts, through the use of common facilities, the characteristics of each I/O device to the link interface provided by the channel. These common facilities provide for the execution of I/O operations, indications concerning the status of the I/O device and control unit, control of the timing of data transfer over the channel path, and certain levels of I/O-device control.

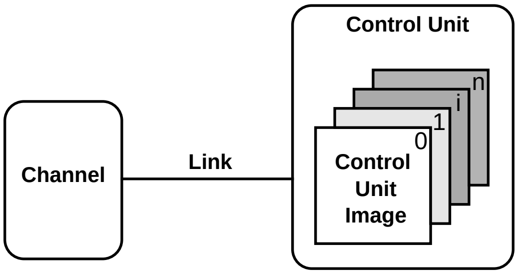


Figure 3 – Control unit images

A control-unit image has the logical appearance of a control unit (see figure 3). Each control-unit image appears to be an independent control unit, although these control-unit images may share the same facilities and physical paths. Link-level and device-level protocols operate for each control-unit image independent of all other control-unit images except for resolving contention for the control-unit common facilities. The control-unit facilities common to a link interface are dedicated to a control-unit image (pertaining to a particular logical path) only for the duration of a particular device-level function.

NOTE – The term “control unit” is used to refer to either a single control-unit image or to the control-unit facilities common to a link interface. A separate term should not be used because in most cases there is no need to distinguish these two functions.

Up to 256 I/O devices can be attached to each control unit or control-unit image.

A control unit may have more than one link interface in order to allow attachment to more than one link, each from a different channel or from a port on the same or a different dynamic switch. When the control-unit link interface is attached to a link from a dynamic switch, the control unit and its I/O devices are physically accessible over that interface to all channels also attached to links from that dynamic switch.

When the link interface on a control unit is shared among multiple channel paths, each channel path is logically represented separately within the control unit. This logical relationship is called the logical path (see 4.2.4). The number of logical paths that a control unit permits is model-dependent.

A control unit permits at least one logical path for each operational interface. The number of logical paths beyond the minimum that a control unit permits is model-dependent, and depends on the required connectivity, availability, and performance.

A control unit can be connected to more than one channel at the same time over different link interfaces. When a control unit is connected to more than one channel at the same time, each connection may be for the same or a different I/O device.

Certain manual controls are included with a control unit. In particular, the online/offline state of the link interface and powering on and off are subject to manual controls. The number, type, and operation of these controls are model-dependent.

4.1.4 Dynamic switch

When a dynamic switch is present in a channel path, a channel-path configuration called a switched-point-to-point configuration is provided. The dynamic switch provides the capability to physically interconnect any two links that are attached to it. The link attachment point on the dynamic switch will be called a dynamic-switch port.

Protocols are defined to allow a channel path attached to a dynamic switch to be shared dynamically. Multiple channels are attached to a single dynamic switch in order to permit sharing of the I/O devices also attached to the same dynamic switch.

Only two dynamic-switch ports are interconnected in a single connection, but multiple physical connections may exist simultaneously within the same dynamic switch. The interconnection of two dynamic-switch ports established by the dynamic switch does not affect the existing interconnection of any other pair of dynamic-switch ports, nor does it affect the ability of the dynamic switch to remove those connections.

When a connection is established, two dynamic-switch ports and their respective point-to-point links are interconnected within the dynamic switch such that the two links appear as one continuous link for the duration of the connection. When frames are received by one of two connected switch ports, the frames are passed from one switch port to the other for transmission on the other switch port's link.

The dynamic switch can form a connection between two switch ports in one of two ways: dynamic or static.

The dynamic switch can establish or remove a dynamic connection between two dynamic-switch ports based on the information provided by certain frame delimiters or other sequences received at the dynamic-switch ports and based on conditions present at each of these dynamic-switch ports. When a connection is to be made, addressing information in the frame is used to determine the port to which the port initiating the connection is to be connected.

A start-of-frame (SOF) delimiter is found at the beginning of every frame to signal whether to establish a dynamic connection, and an end-of-frame (EOF) delimiter is normally found at the end of a frame to signal whether to retain or remove a dynamic connection. If a defined sequence is recognized by one of two dynamically connected switch ports, the dynamic connection will be removed.

The dynamic connection normally lasts until a disconnect-EOF delimiter is received or a defined sequence being sent or received causes the connection to be removed. A static connection between two dynamic-switch ports is established or removed by means of the facilities of the dynamic-switch control unit. When a static connection exists between two dynamic-switch ports, the ports are in the static state (see 10.9.7). The static state of the dynamic-switch port is not affected by information transferred on either link of the static connection.

A dynamic switch may have a maximum of 254 ports; one or more for the dynamic-switch control unit and the rest (up to 253) for switch ports. The number of ports implemented depends on configuration requirements and is model-dependent. A connection between two ports provides simultaneous two-way information transfer between the two ports. The maximum number of possible simultaneous connections is equal to half the number of dynamic-switch ports implemented, or, if the quantity of dynamic-switch ports implemented is an odd number, half of one less than the maximum number of dynamic-switch ports implemented. Certain conditions in the dynamic switch may preclude the implemented maximum number of connections from being made.

A dynamic-switch port does not have a link address; however, an association between a dynamic-switch port and either a destination link address or a source link address is provided by the dynamic switch.

NOTE – When a static connection is established between two switch ports, any existing dynamic connection with these switch ports is removed. Therefore, before a static connection is established, system activity with the affected dynamic-switch ports should be quiesced. Similarly, when a static connection is removed, activity at both switch ports involved in the static connection should be quiesced before the connection is removed.

4.1.5 Dynamic-switch control unit

A dynamic switch includes an element for controlling the allowed connections and for communicating with the channel for purposes of controlling the configuration. For purposes of accepting channel commands and signaling status, this element has the appearance of a control unit and is referred to in this document as the dynamic-switch control unit. The dynamic-switch control unit is assigned one or more link addresses. Through the facilities provided by the dynamic-switch control unit, static connections may be made or removed, a port may be placed in the offline state, and the ability of ports to be dynamically connected with other ports may otherwise be altered.

Certain conditions cause the dynamic switch to have initiative to report state change by sending state-change notification to each link-level facility which is potentially affected by the state change. The conditions are described in 9.1.13.

4.1.6 Link-level facility

The link-level facility manages the attached link, including the transmitter and the receiver; processes initialization, offline, connection-recovery, and link-failure operations; performs link-level functions and protocols; and manages logical paths. Transmitting and receiving frames and sequences on the link are controlled by the link-level-facility. When a link-level facility is in a particular state, the appropriate link-level function and protocol or the appropriate procedure which uses a sequence will be performed, depending on the conditions present.

The link-level facility provides the capability to communicate with another link-level facility or a dynamic-switch port. The link-level-facility considers the physical path as a two-way communication path in a switched-point-to-point configuration. For management of the link, a link-level facility uses monolog communication or dialog communication. Monolog communication occurs when an initiation frame has been sent or received and a frame has not been received or transmitted, respectively. Dialog communication occurs when an initiation frame has been sent or received (that is, a link-level facility having monolog communication) and a frame that does not have a disconnect-EOF delimiter has been received or transmitted, respectively. Regardless of the channel-path configuration, a two-way communication path must exist before a link-level facility considers a dialog communication to exist. The interconnection appears to each link-level facility as one continuous link for the duration of the dialog communication. The link-level facility uses the type of SOF delimiter that started frame transmission or reception, the frame-terminating conditions that ended frame transmission or reception, and the conditions present to determine whether monolog communication exists, dialog communication exists, or monolog or dialog communication is removed.

4.2 Channel-path configurations

Two channel-path configurations are provided: point-to-point and switched-point-to-point. Identical protocols are used for transferring information in the two types of configuration.

A static connection formed between a channel and a control unit in a configuration with a dynamic switch appears the same as a point-to-point configuration (see 10.9.7). A static connection or a point-to-point configuration will be permitted between a channel and control unit only when a single control unit is defined on the channel path or when multiple control-unit images all share the same link-level facility. Identical functions and protocols are used for transferring information on channel paths, regardless of the type of configuration.

In both types of configuration, a link will be the common element of attachment. A link provides only two points of attachment; one at each end of the link. Because a link provides separate paths for transmission and reception, contention to transmit on a link from either end does not occur; this means that the link will always be available to the channel or control unit for transmission. However, transmission without contention from one end of a link does not eliminate contention for shared resources required for receiving that transmission at the other end of the link. This contention can be for link-level, device-level, or device resources. Protocols have been defined to handle each of these situations.

Contention will occur, for example, when two or more control units on the same dynamic switch simultaneously send a frame to the same channel. In this example, the channel receives the frame from only one of the control units; the other control units receive responses of switch busy.

The type of channel-path configuration depends on the I/O-configuration requirements of a particular system and is model-dependent.

4.2.1 Point-to-point configuration

A channel path that consists of a single link interconnecting one or more control-unit images to one or more channel images forms a point-to-point configuration (see figure 4).

The channel common facilities and the control-unit common facilities resolve contention among the control-unit images for access to the link.

A maximum of one link can be attached to the channel in a point-to point channel-path configu-

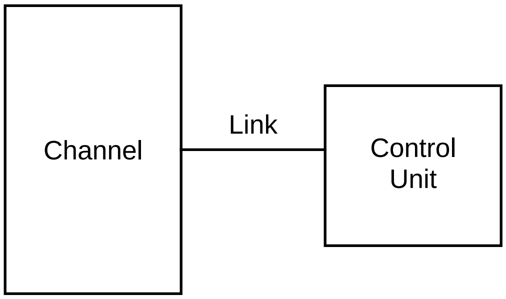


Figure 4 – Point-to-point configuration

ration. The maximum number of control-unit images that can be addressed over the link will be 16; therefore, the maximum number of I/O devices that can be addressed over a channel path configured point-to-point is equal to 16 times 256, or 4096.

4.2.2 Switched-point-to-point configuration

A channel path that consists of one link which interconnects one or more channel images with a dynamic switch and one or more links, each of which interconnects a dynamic switch with one or more control-unit images, forms a switched-point-to-point configuration (see figure 5).

Unlike a point-to-point configuration, a switched-point-to-point configuration does not form a physical connection to a channel when a control unit is attached to a link from the dynamic switch. A static connection presents the appearance of a point-to-point configuration. A control unit becomes physically connected to a channel when the dynamic-switch port to which its link is attached becomes connected to the port to which the link from the channel is attached. When the connection is removed, the control unit will no longer be physically connected to the channel (see 4.1.4). Only one control unit at a time can be physically connected to the channel, and similarly, only one channel at a time can be physically connected to the same control unit over the same interface; that is, the dynamic switch allows only a single physical connection to exist between two dynamic-switch ports.

Channels and control units can be attached to the links from a dynamic switch in any combination, depending on configuration requirements and depending on available resources in the dynamic switch. When multiple channel facilities are attached to links from a dynamic switch, each channel facility may be simultaneously connected to a control unit, with each connection providing the full data-transfer capability of a single link. Sharing a control unit through a dynamic switch means that communication with the control unit can take place over one link interface in the case where the control unit has only one link to the dynamic switch or over multiple link interfaces in the case where a control unit has more than one link to the dynamic switch. Sharing a control unit through a dynamic switch can occur between two channels on the same system, even if the same path from the control unit to the dynamic switch is used.

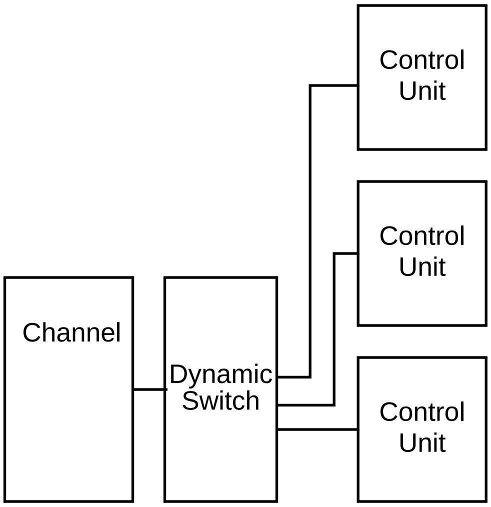


Figure 5 – Switched-point-to-point configuration

Only one dynamic switch will provide dynamic switching on a channel path. The physical path between a channel and a control unit may have two dynamic switches, connected in series, provided one of the two connections is static. The other connection will be either static or dynamic. If the other connection is static, then one to 16 control-unit images may be defined on the channel path. If the other connection is dynamic, then one to 16 control-unit images may be defined for each link accessible through the dynamic switch (see 4.1.4).

The channel common facilities and the control unit common facilities resolve contention among the control-unit images for access to the link.

In the switched-point-to-point configuration, contention created by two or more dynamic-switch ports receiving a frame requiring a dynamic connection to the same dynamic-switch port and its respective link will be resolved by the facilities within the dynamic switch, based on the conditions present. This contention will be resolved independent of whether a channel or control unit is attached to the link from each port involved. When the contention is resolved, only one of the contending ports becomes dynamically connected to the required port, while the other contending ports indicate on their respective links that a port-busy condition exists at the dynamic switch (see 4.1.4).

A maximum of one link can be attached to the channel in a switched-point-to-point channel path configuration. A dynamic switch can attach up to 254 links. The dynamic-switch control unit always occupies at least one position on the dynamic switch thus reducing the number of links that can be attached to 253 or less. The maximum number of control-unit images that can be addressed over each link attached to the dynamic switch will be 16; therefore, the maximum number of I/O devices that can be addressed over a channel path configured switched-point-to-point is equal to 253 times 16 times 256, or 1,036,288. However, for every additional channel attached to a link from the dynamic switch, the maximum number of control-unit images that will be addressed is reduced by 16 and therefore, the maximum number of I/O devices that will be addressed is reduced by 16 times 256, or 4,096.

4.2.3 Physical path

The communication path between a channel and a control unit will be composed of two different parts, the physical channel path, or, for brevity, physical path, and the logical channel path, or, for brevity, logical path. The physical path is the link, or the interconnection of two links by a dynamic switch, that provides the physical transmission path between a channel and a control unit. The logical path is the relationship that exists between a channel and a control unit for device-level communication during execution of an I/O operation and presentation of status (see 4.2.4).

The physical path between the channel and the control unit is a two-way communication path that can be either:

a) a single point-to-point link;

b) two point-to-point links interconnected through one dynamic switch;

c) three point-to-point links interconnected through two dynamic switches.

In channel-path configurations interconnected through two dynamic switches, the connection through at least one of the dynamic switches must be a static connection.

When the physical path consists of an uninterrupted two-way communication path to the control unit, the control unit and the channel are considered physically connected. The physical connection in a point-to-point configuration exists at the moment the control unit is attached to the channel. The physical connection to a channel with a switched-point-to-point channel-path configuration exists only when a two-way communication path is made available through the dynamic switch. When the two-way communication path through the dynamic switch is removed, the physical connection no longer exists. (See 10.7.)

Only one control unit at a time can be physically connected to the channel, and similarly, only one channel at a time can be physically connected to the same control unit over the same link interface; that is, the dynamic switch allows only a single physical connection to exist between two dynamic-switch ports.

4.2.4 Logical path

A logical path is the relationship between a channel image and a control-unit image that designates the physical path to be used for device-level communication between these images. The logical path is established as part of the channel and control-unit initialization procedures by the exchange of link-level frames (see 10.1.2.3 and 10.1.3.3). When the logical path is established, device-level communication will be allowed on that logical path. All device-level protocols depend upon the existence and identity of logical paths. Device-level protocols are executed over an established logical path by means of the exchange of device frames between the channel and the control unit. When a logical path is not established, only link-level communication can occur on the physical path.

A logical path is identified within a link-level facility by the combination of an eight-bit link address assigned to the channel, an eight-bit link address assigned to the control unit, a four-bit logical address assigned to the channel image establishing the logical path, and a four-bit logical address assigned to the control-unit image to which the logical path is being established. Both the channel and the control-unit recognize a logical path by the same combination of addresses. The two eight-bit link addresses define the link-level facilities and the physical path that are associated with the logical path, and the two four-bit logical addresses identify the logical images for which the logical path exists. With the addressing structure described in 8.4, a channel image may have one logical path to each control-unit image. The number of logical paths a channel permits will be model-dependent.

Even though multiple logical paths will be associated with the same link interface on the control unit, the channel paths represented by these logical paths are treated as if each were associated with a separate link interface; each logical path represents a logical relationship to a channel, which includes the physical path that can be created to that channel. When multiple control-unit images are provided, their presence will be apparent only through associated logical paths.

I/O operations, system-resets, and path-groups for a particular system are identified by means of the logical path established by the channel for that system. To a control unit, each logical path represents a different channel image.

4.3 Information formats

Information will be transferred on a link in a serially transmitted synchronous bit stream. On the link, the major unit of transmission is the frame. Frames are used to transfer control information, data, commands, or status between a channel and a control unit. Additionally, sets of frames are used to initiate an I/O operation, perform data transfer, and end an I/O operation.

The SBCON I/O interface can handle the data-transfer requirements of a wide range of I/O-device types by providing for the selection of the optimum combination of frame size, frame rate, and quantity of data per data request. The contents of a frame are shown in figure 6.

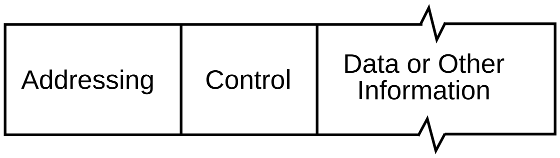


Figure 6 – Frame contents

The other units for transmitting information on a link are:

a) sequences, which are special groups of encoded characters to indicate certain conditions that cannot be indicated using frames;

b) the idle character, which is a special encoded character sent on a link when no other information is being sent.

4.4 Protocols

Two levels of protocol are defined: the link-level and the device-level. These levels represent logical collections of protocols necessary for transferring information on a channel path. Protocols for the SBCON I/O interface are classified as either link-level or device-level, depending on whether they are for the purpose of managing a channel path and exchanging information over that channel path or for the purpose of managing the execution of an I/O operation. The link-level describes the physical characteristics of a channel path and the associated protocols required for the transmission and reception of frames. The device-level primarily relates to the protocols associated with the execution of an I/O operation for a specific I/O device.

Link-level and device-level protocols operate for each control-unit or channel image independent of all other images sharing common facilities associated with the link except for resolving contention for these common facilities. The common facilities are dedicated to a control-unit or channel image (pertaining to a particular logical path) only for the duration of performing a particular device-level function.

The presence of multiple channel images or control-unit images only becomes apparent through the logical paths that are established. All device-level functions and protocols depend on the existence of and identity of these logical paths. When the logical path is known, the particular control-unit image for which the link-level and device-level facilities are dedicated at any one time is known.

The execution of an I/O operation requires link-level and device-level protocols to be present in both the channel and the control unit. Information exchanged between a channel and control unit as a result of executing an I/O operation will be transferred under the control of the device-level protocols with the aid of the link-level protocols for sending and receiving frames. The failure to satisfy protocols at either level results in an error being recognized.

Most link-control functions are invoked during or immediately after the initialization of the channel or control unit and are not invoked afterward unless certain link-recovery or reconfiguration procedures occur. Link-level functions will be initiated without the use of device-level functions or protocols. This provides a communication capability at the link-level, which, in some cases, will be necessary in order to complete certain initialization procedures and to establish the links of the channel path as operational. Link protocols are used to determine structure, size, and integrity of the frame, independent of whether the frame is a device frame or link-control frame, whenever a frame is sent. Link protocols also determine the physical path on which to send the frame, and, if a dynamic switch is present on the channel path, the switch connection which forms that physical path.

Figure 7 shows the relationship between the link-level, device-level, physical path, and logical path.

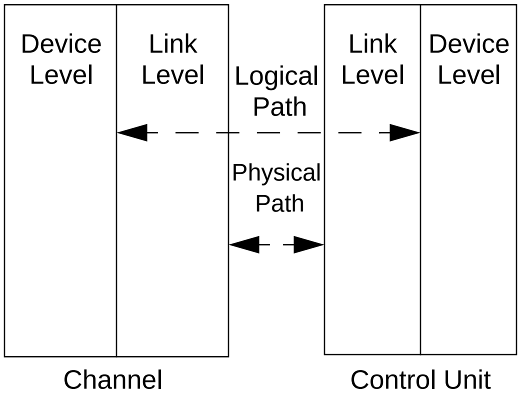


Figure 7 – Level and path relationships

4.4.1 Link-level

All communications over a channel path use link-level protocols. Link-level protocols are used to establish and maintain the physical and logical paths in order to provide for transmission and reception of frames and sequences. They include functions such as acquiring link addresses, generating the basic frame structure, coordinating the protocols for frame transmission, and checking the integrity of information sent in frames. (See clauses 8, 9 and 10.)

Certain functions require the exchange of information between the channel and control unit at the link-level; these functions are performed by means of link-control frames. Initialization is such a function. Before communication over a channel path can occur, and before an I/O operation can be executed, some form of initialization must be completed according to the requirements of the system and according to the specifications of the architectural definition. (See 10.1.)

Once a channel has performed initialization procedures for a control unit, including establishment of a logical path to a control-unit image, that control-unit image will be considered operational and capable of executing I/O operations over that channel path. A channel can permit one logical path to every control-unit image configured to it.

When frames are sent on a channel path, the link-level protocols create the necessary physical connections, detect when these connections are not available, and provide the appropriate responses. The appropriate link-control information to make any necessary connection in the dynamic switch will be included in the frame. If the channel path has a dynamic switch and the necessary connection cannot be made, the dynamic switch returns a link-control frame containing the reason the connection was not made.

Link-control protocols provide checking information in every frame to detect errors that may occur during frame transmission and to provide the appropriate response when errors are detected. For example, when a CRC error is detected in a frame that initiates a connection, a link-control-frame response called a link-level-reject frame will be sent to indicate that a transmission error has occurred.

4.4.2 Device-level

Device-level protocols allow for transfer of information specifically related to an I/O operation, transfer of status of an I/O device or control unit, and recovery when errors are detected by the channel or the control unit.

The types of device-level information transferred between a channel and control unit that use device-level protocols are: commands, data, control information, and status. Commands are provided by the channel program being executed. Data is the information associated with the command being executed that will be transferred to or from the I/O device. Control information includes functions that manage the transfer of data between the channel and control unit and functions that manage execution of the I/O operation. Status describes the results of the completion of an I/O operation (successful or unsuccessful) or provides information not associated with an I/O operation that will be reported to the channel.

Device-level information is transferred between a channel and a control unit in device frames. The formats of device frames and link-control frames are different. Device frames are transferred using both link-level and device-level functions and protocols. For example, when the channel receives initiative to start an I/O operation, the device-level functions and protocols obtain the command and other parameters from the current control block (or equivalent) and insert them into the appropriate fields within a device frame. When the device frame is ready for transmission, link-level functions and protocols will provide additional information within the device frame and then coordinate the actual transmission of the frame on the channel path.

The transfer of data will be initiated and paced using device-level protocols by the recipient of the data. The quantity of data in each device frame and the rate at which consecutive frames are sent are determined by parameters exchanged between both the link and the device-levels.

More detailed descriptions of the device-level functions and protocols are provided in clauses 12 and 13.

4.4.3 Addressing

Two levels of addressing are used for the link-level and device-level structure. All control units and channels can use both levels of addressing, link-level and device-level. Link-level addressing identifies the link-level facilities, which in turn identify the physical path within the channel-path configuration to be used for communication between a channel and a control unit. Link-level addressing also identifies the logical image (channel image or control-unit image) that shares the common facilities of the link-level facility.

Device-level addressing identifies the particular I/O device to the particular channel or control unit, once the physical path, as designated by the link-level addressing, has been determined. Link-level addressing is considered to be the first level of addressing because it determines the physical path and the sharing logical image; device-level addressing is considered to be the second level of addressing because it determines the particular I/O device once the physical path and logical image are determined. Both levels of addressing have specific address-assignment requirements.

Link-level addressing requires each link-level facility to be assigned an eight-bit address, called the link address. A link-level facility that does not have a link address assigned is unidentified; a link-level facility that has a link address assigned is identified. The assignment of a link address to a link-level facility occurs when the link-level facility performs initialization (see 10.1). When an unidentified link-level facility performs initialization, it acquires its link address through the procedures defined for link-address acquisition (see 9.1.14, 10.1.2.1 and 10.1.3.1). An address of all zeros or all ones cannot be assigned as a link address. The all-ones address will be reserved for use with certain link-control frames that are sent or received by a link-level facility, independent of whether the link-level facility is identified or unidentified. The all-zeros address will be reserved for use as a source link address with certain link-control frames sent by a link-level facility that is unidentified.

Link-level addressing also requires that each logical image be assigned a four-bit address, called the logical address. A control-unit image will be assigned a logical address when it is created. A logical address must be unique on a physical path but need not be unique on a channel path. The assignment of logical addresses and the method by which this assignment is performed are model-dependent.

When multiple control-unit images are created, they share a single link-level facility for each link, and therefore, the same respective link address. A maximum of 16 logical images can share the same link address. The combination of the link address and the logical address determines the logical image to which the device-level addressing applies.

When the recipient of a frame is unidentified, the action to be taken depends on the frame received. For an acquire-link-address frame, see 9.1.14, 10.1.2.1 and 10.1.3.1. For all other initiation frames, an acquire-link-address error will be detected (see 11.1.7).

If a frame is received with a destination address of all ones, and a destination address of all ones is permitted for the type of frame being received, the recipient of the frame accepts the frame and provides an appropriate response. If a frame is received with a destination address of all ones but a destination address of all ones is not permitted for the type of frame being received, the recipient of the frame discards the frame and detects a link-protocol error.

Device-level addressing depends on the assignment of an eight-bit address, called the device address, to every I/O device. A device address must be unique on a logical path on a control unit but need not be unique on a channel path because the resultant combination of the link address, control unit logical address, and the device address uniquely identifies an I/O device on a channel path. The assignment of device addresses and the method by which this assignment will be performed are model-dependent (see 12.1.2).

The dynamic-switch control unit has a link address assigned to it, along with one or more device addresses, and will be addressed as an I/O device on a channel path. The existence of a link address for the link-level facility of the dynamic switch does not affect normal dynamic-switching functions, other than by using one of the possible link addresses on the channel path. Every channel and control unit attached to links from the same dynamic switch addresses the dynamic-switch control unit with the same link address. All channels that are attached to links from the same dynamic switch and that address the I/O devices on the dynamic-switch control unit use the same set of device addresses to address those I/O devices.

Link addresses are not required to be unique across channel paths unless those channel paths attach to the same dynamic switch. Unique link addresses must be assigned to the respective link-level facilities of every channel or control unit that is attached through the same dynamic switch.

Control units that attach to more than one link provide a link-level facility per link interface, with each link-level facility having an assigned link address. The link-level facilities attached to links from the same dynamic switch are assigned different link addresses, while those link-level facilities attached to links from different dynamic switches or from different channels are assigned the same or different link addresses depending on the physical path and the addresses available within each channel-path configuration.

All frames sent on a channel path contain link-level-addressing information for determining the correct physical path to the required destination link-level facility and, for certain frame types, for determining the appropriate logical image associated with that link-level facility. If the frame sent is a link-control frame, then only link-level addressing will be required. Device frames require both link-level and device-level addressing. Device-level addressing identifies the I/O device that is the source or destination of the information in the frame.

Every frame sent contains either the link address assigned to the destination of the frame (destination link address) or the all-ones link address, and contains either the link address assigned to the source of the frame (source link address) or the all-zeros link address.

Link-control frames may use a destination logical address or a source logical address, or both depending on the link-control function indicated in the frame and the intended recipient (see clauses 8 and 9.)

Device frames use both a destination logical address and a source logical address (see clause 8). Device frames, with a few exceptions that depend on the device function specified in the frame, contain a device address (see clause 12.)

5 SBCON physical layer definition

5.1 Physical interface

The SBCON physical layer provides for two distinct physical interface implementations:

a) The multimode physical interface for use with 62.5/125 or 50/125-micrometer multimode fiber optic cable;

b) The single-mode physical interface for use with single-mode fiber optic cable.

Each physical interface provides a common, compatible I/O interface that products can use to communicate with each other through light pulses sent over optical transmission fibers.

The SBCON physical interface consists of an optical transmitter and optical receiver, collectively called the transmitter-receiver subassembly (TRS), and the optical duplex link consisting of one or more duplex optical cables, as shown in figure 8.

5.1.1 Fiber optic information transfer

Information is transmitted over an SBCON link by sending light pulses over optical transmission fibers. Each SBCON link consists of a pair of optical fibers; one for the transmission of light pulses and the other for the reception of light pulses as viewed from an individual TRS. Light pulses are sent and received simultaneously on a link.

A link typically has a main trunk cable, with a short duplex jumper cable at each end of the trunk for routing within the building or machine room, as shown in figure 8. A link is not required to be constructed with this configuration and may have alternate forms. For example, a short link may have only jumper cable and no trunk cable. Distribution panels provide a central location for the attachment of trunk and jumper cables. Distribution panels may be mounted in a rack, wiring closet, or on a wall.

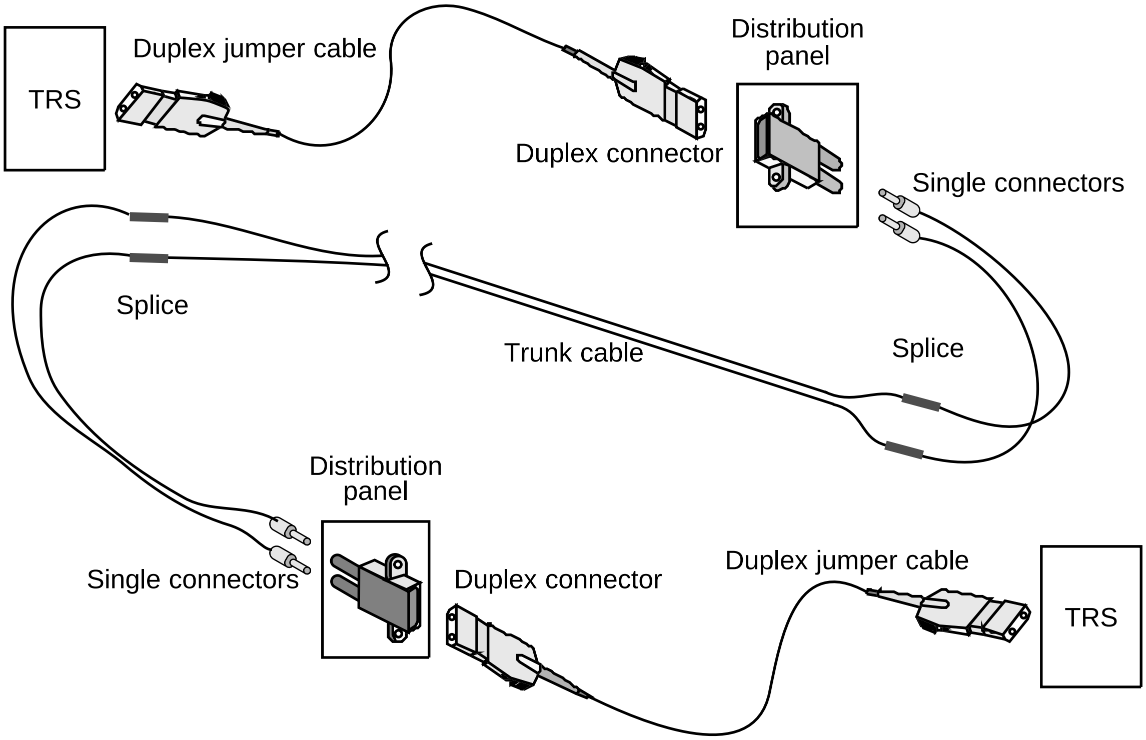


Figure 8 – A typical fiber optic channel

For attachment to an SBCON device, the end of the transmission cable is terminated in a duplex connector, which mates with an SBCON duplex receptacle on the device. When inserted into the receptacle, the connector is optically coupled to the transmitter and receiver, and the device can send and receive optical signals over the cable.

5.1.2 Data transmission

The data transmitted over a link shall be based on an 8-bit to 10-bit non-return-to-zero code. (See clause 6.) The idle function or one of the sequence functions shown in tables 15 or 16 shall be transmitted continuously when information is not being sent. The data-transmission rate shall be 200 MBaud with a tolerance of ±0.04 megabits per second.

The optical transceiver consists of both a transmitter and a receiver component. The transmitter component shall send a coded serial-optical signal over a fiber link to a receiver component that is capable of receiving the coded serial-optical signal. All specifications in this clause apply over the entire operating range including worst case condition and end-of-life effects.

5.1.3 Link, trunk cable, and jumper cable specifications

The specifications for the links, trunk cable, and jumper cable can be found in table 1 for multimode and table 2 for single-mode.

NOTE – It is recommended that for multimode the jumper cable jacket be orange and the duplex connector be black and for single-mode the jumper cable jacket be yellow and the duplex connector be gray.

Table 1 – Maximum multimode link loss (at 1,300 nm)

Maximum Link Length in km[1,2]	Maximum link loss (dB)	Trunk size (μm)	Minimum trunk modal bandwidth (MHz*km)
2.0	8.0	62.5	500
2.0	8.0	50.0	800
2.0 to 3.0	8.0	62.5	800

Notes

1 The maximum link length includes both jumper cables and trunk cables.

2 The maximum total jumper cable length shall not exceed 244 meters when using either 50/125 μm trunk fiber or when a 62.5/125 μm link exceeds 2 kilometers.

Table 2 – Maximum single-mode link loss (at 1,300 nm)

Maximum Link Length km[1,3,4]	Maximum Link Loss dB[2,5]	Fill Diameter μm
20	14.0	9 to 10

Notes

1 The maximum link length includes both jumper cables and trunk cables.

2 Single-mode connectors and splices shall meet a minimum return loss specification of 28 dB.

3 In a single-mode jumper cable, the minimum distance between the connectors or splices is 4 meters (13.1 ft.).

4 In a single-mode trunk cable, the distance between the connectors or splices shall be sufficient to ensure that only the lowest-order bound mode propagates.

5 In a single-mode link, the minimum return loss is 13.7 dB.

5.2 Multimode physical layer

The multimode physical layer allows links to extend up to 3 kilometers, without retransmission, using 62.5/125 μm multimode trunk fiber cable. Or, alternatively, up to 2 kilometers using 50/125 μm multimode trunk fiber cable.

5.2.1 Multimode output interface

Table 3 defines the serial optical signal at the multimode duplex receptacle when coupled into a multimode duplex jumper cable. The parameters specified in this section are based on the requirement that the bit error rate shall not exceed 10^{-15}, including operation at the minimum interface power level. The use of an *incoherent* light source, such as a light-emitting diode (LED), is required.

Table 3 – Multimode output interface optical signal

Parameter	Min.	Max.	Unit
Average power[1,2]	-20.5	-15	dBm
Center wavelength (λ)	1,280	1,380	nm
Spectral Width (FWHM)		175	nm
Rise time (Tr) (20-80%)[1,3]		1.7	ns
Fall time (Tf) (80-20%)[1,3]		1.7	ns
Eye-window[1]	3.4		ns
Extinction ratio[1,4]	8		dB
Tr, Tf at optical path output[3,5]		2.8	ns
Optical output jitter[6]		0.8	ns

Notes

1 Based on any valid 8B/10B code pattern. The length of jumper cable between the output interface and the instrumentation is 3 meters.

2 The output power shall be greater than -29 dBm through a worst-case link as specified in 5.2.3. Higher order mode loss (HOML) is the difference in link loss measured using the device transmitter as compared to the loss measured with a source conditioned to achieve an equilibrium mode distribution in the fiber. The transmitter shall compensate for any excess HOML occurring in the link. (e.g. HOML in excess of 1dB for a 62.5 µm link.)

3 The minimum frequency response bandwidth range of the optical waveform detector shall be 100 kHz to 1 GHz.

4 Measurement shall be made with a dc-coupled optical waveform detector that has a minimum bandwidth of 600 MHz and whose gain flatness and linearity over the range of optical power being measured provide an accurate measurement of the high and low optical power levels.

5 The maximum rise or fall time (from e.g., chromatic, modal dispersion, etc.) at the output of a worst-case link as specified in 5.2.3. The 0% and 100% levels are set where the optical signal has at least 10 ns to settle. The spectral width of the transmitter shall be controlled to meet this specification.

6 The optical output jitter includes both deterministic and random jitter. It is defined as the peak-to-peak time-histogram oscilloscope value (minimum of 3,000 samples) using a 2^7-1 pseudo random pattern or worst case 8B10B code pattern.

5.2.2 Multimode input interface

Table 4 specifies the input interface requirements.

To assist in fault isolation, the input interface shall activate a loss-of-signal (LOL) state when the optical data cannot be detected. The optical power threshold to activate the LOL state, the LOL optical power hysteresis, and the reaction time for the LOL state change to occur are also specified. The design of the machine receiving data from the interface determines how the state change is subsequently indicated.

5.2.3 Multimode link specifications

Table 1 lists the specifications for links using 62.5/125 µm or 50/125 µm fiber trunk cable. The trunk cable to which the jumper cables are connected shall have optical properties that conform to the specifications in the table.

5.2.4 62.5 µm multimode cable specifications

5.2.4.1 62.5 µm multimode trunk cable optical specifications

The specifications for multimode trunk cable support attachment of SBCON-capable devices are shown in table 5.

Table 4 – Multimode input interface characteristics

Parameter	Min.	Max.	Unit
Saturation level[1]	-14.0		dBm
Sensitivity[1,2]		-29	dBm
Acquisition time[3]		100	ms
LOL threshold[4]	-45	-36	dBm
LOL hysteresis[4,5]	0.5		dB
Reaction time for LOL state change	3	500	µs

Notes

1 Based on any valid 8B/10B code pattern measured at, or extrapolated to, 10^{-15} BER measured at center of eye. This specification shall be met with worst-case conditions as specified in table 3 for the output interface and 5.2.3 for the fiber optic link. This value allows for a 0.5 db retiming penalty

2 A minimum receiver output eye opening of 1.4 ns at 10^{-12} should be achieved with a penalty not exceeding 1 dB.

3 The time to reach synchronization after the removal of the condition that caused the loss of synchronization. The pattern sent for synchronization is either the idle character or an alternation of idle and data characters.

4 In direction of decreasing power:
If power > -36 dBm, LOL state is inactive; If power < -45 dBm, LOL state is active.
In direction of increasing power:
If power < -44.5 dBm, LOL state is active; If power > -35.5 dBm, LOL state is inactive.

5 Required to avoid random transitions between LOL being active and inactive when input power is near threshold level.

Table 5 – 62.5 µm multimode trunk cable specification

Type of fiber	Graded index with glass core and cladding
Operating wavelength	1,300 nm
Core diameter[1]	62.5 ±3.0 µm
Core concentricity error[2]	6% maximum
Cladding diameter[2]	125 ±3.0 µm
Cladding noncircularity[2]	2% maximum
Core and cladding offset	3 µm maximum
Numerical aperture[3]	0.275 ±0.015
Minimum modal bandwidth[4]	500 MHz*km at ≤2km; 800 MHz*km at >2km and ≤ 3km
Attenuation[5]	1.0 dB/km at 1,300 nm (typical)

Notes

1 Measured in accordance with EIA 455 FOTP 58, 164, 167, or equivalent.

2 Measured in accordance with EIA 455 FOTP 27, 45, 48, or equivalent.

3 Measured in accordance with EIA 455 FOTP 47 or equivalent.

4 Measured in accordance with EIA 455 FOTP 51 or equivalent.

5 The attenuation is a typical value rather than a specification. Table 1 is the specification.

5.2.4.2 62.5 μm multimode duplex jumper cable specifications

The specifications for multimode duplex jumper cable support attachment to SBCON-capable devices are shown in table 6.

Table 6 – 62.5 μm multimode duplex jumper cable specification

Type of fiber	Graded index with glass core and cladding
Operating wavelength	1,300 nm
Core diameter[1]	62.5 ±3.0 μm
Cladding diameter[2]	125 ±3.0 μm
Numerical aperture[3]	0.275 ±0.015
Minimum modal bandwidth[4]	500 MHz*km
Attenuation	1.75 dB/km at 1,300 nm (maximum)
Connector color	Black
Jacket color	Orange

Notes

1 Measured in accordance with EIA 455 FOTP 58, 164, 167, or equivalent.

2 Measured in accordance with EIA 455 FOTP 27, 45, 48, or equivalent.

3 Measured in accordance with EIA 455 FOTP 47 or equivalent.

4 Measured in accordance with EIA 455 FOTP 51 or equivalent.

5.2.5 50 μm multimode trunk cable optical specifications

The specifications for multimode trunk cable support attachment of SBCON-capable devices are shown in table 7.

Table 7 – 50 μm multimode trunk cable specification

Type of fiber	Graded index with glass core and cladding
Operating wavelength	1,300 nm
Core diameter[1]	50 ±3.0 μm
Core concentricity error[2]	6% maximum
Cladding diameter[2]	125 ±3.0 μm
Cladding noncircularity[2]	2% maximum
Core and cladding offset	3 μm maximum
Numerical aperture[3]	0.275 ±0.015
Minimum modal bandwidth[4]	800 MHz*km
Attenuation[5]	1.0 dB/km at 1,300 nm

Notes

1 Measured in accordance with EIA 455 FOTP 58, 164, 167, or equivalent.

2 Measured in accordance with EIA 455 FOTP 27, 45, 48, or equivalent.

3 Measured in accordance with EIA 455 FOTP 47 or equivalent.

4 Measured in accordance with EIA 455 FOTP 51 or equivalent.

5 The attenuation is a typical value rather than a specification. Table 1 is the specification.

5.2.6 Multimode duplex connector

The multimode duplex connector, shown in figure 9, contains both the receive and transmit fibers in one connector. The connector is keyed to provide correct orientation and uses release tabs to prevent accidental removal.

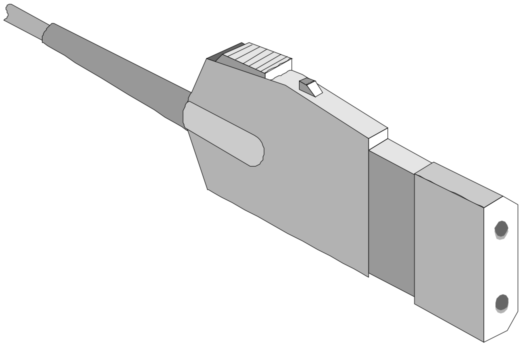


Figure 9 – Duplex multimode connector

5.2.7 Multimode duplex receptacle

A dimensional diagram of the multimode duplex receptacle and notes concerning the connector ferrule are shown in figure 10. Tx and Rx denote the transmitter and receiver ports respectively.

5.3 Single-mode physical layer

The single-mode physical layer allows links to extend up to 20 kilometers, without retransmission, using dispersion-unshifted, single-mode trunk fiber cable.

5.3.1 Single-mode output interface

Table 8 defines the serial optical signal at the single-mode duplex receptacle when coupled into a single-mode duplex jumper cable. The parameters specified in this section are based on the requirement that the bit error rate shall not exceed 10^{-15}, including operation at the minimum interface power level.

Table 8 – Single-mode output interface optical signal

Parameter	Min.	Max.	Unit
Average power into SMF[1]	-8	-4	dBm
Center wavelength (λ)[1]	1,261	1,360	nm
Rise time (Tr) (20-80%)[1,2]		1.5	ns
Fall time (Tf) (80-20%)[1,2]		1.5	ns
Eye-window[1]	3.5		ns
Extinction ratio[1,3]	8.2		dB
Relative intensity noise (RIN_{12})[4]		-112	dB/Hz
AC optical path penalty[5]		1.5	dB
Optical output jitter[6]		0.8	ns

Notes

1 Based on any valid 8B/10B code pattern. This measurement is made using a 4-meter single-mode duplex jumper cable and includes only the power in the fundamental mode of the single-mode fiber.

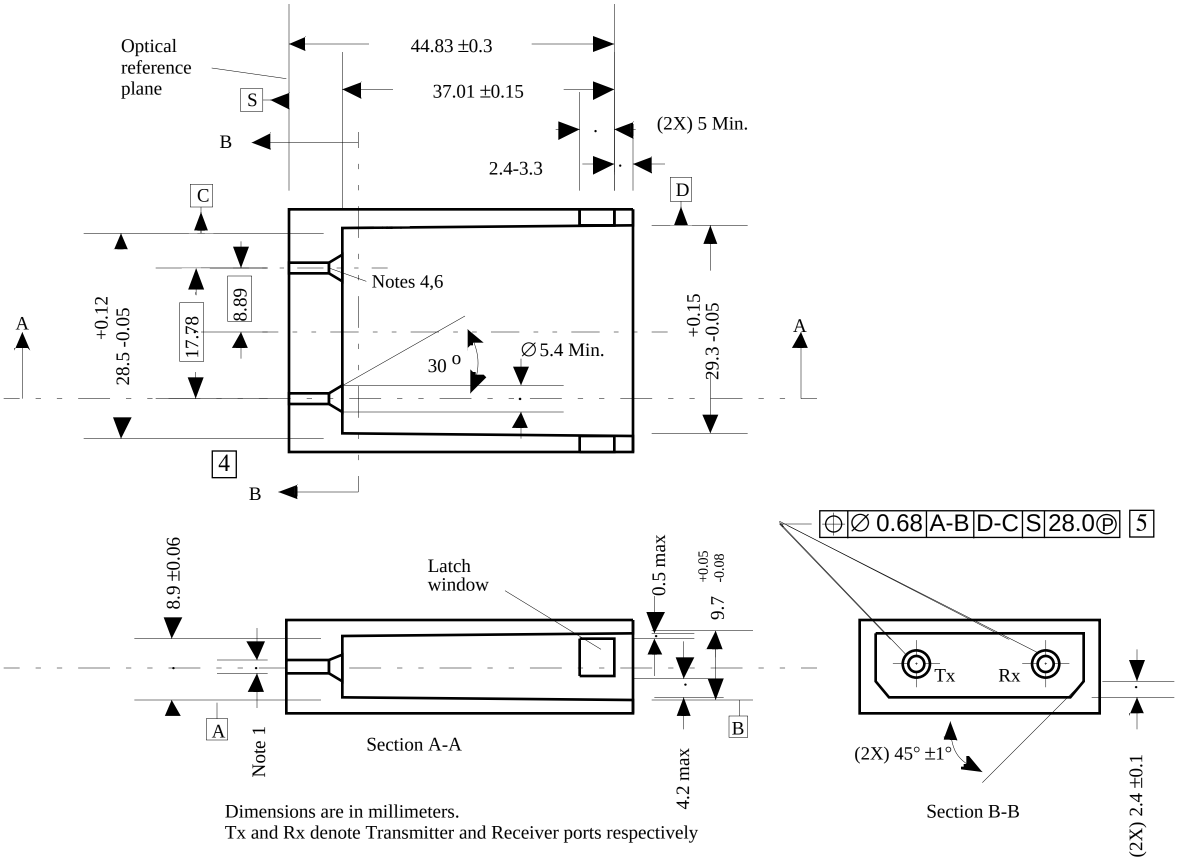
2 The minimum frequency response bandwidth range of the optical waveform detector is 100 kHz to 1 GHz.

3 Measurement shall be made with a dc-coupled optical waveform detector that has a minimum bandwidth of 600 MHz and whose gain flatness and linearity over the range of optical power being measured provide an accurate measurement of the high and low optical power levels.

4 The relative intensity noise is measured with a 12 dB optical return loss into the output interface.

5 The maximum degradation in input interface sensitivity (from e.g., jitter, mode hopping, intersymbol interference, etc.) that can occur by using a worst-case link as specified in 5.3.3. The spectral width of the transmitter shall be controlled to meet this specification.

6 The optical output jitter includes both deterministic and random jitter. It is defined as the peak-to-peak time-histogram oscilloscope value (minimum of 3,000 samples) using a 2^7-1 pseudo random pattern or worst case 8B10B code pattern.



Dimensions are in millimeters.
Tx and Rx denote Transmitter and Receiver ports respectively

Notes

1 Alignment bore to accept 2.4980/2.4990 [.098346/.098378] diameter ferrule.

2 (General) Plug ferrule end shall seat to the optical reference plane with a static force of 7.1 to 14 to 3.1 pounds) per ferrule.

3 (General) Ferrule material shall be zirconia with a maximum surface roughness of 0.2 µm.

4 The module bore must be able to withstand a force of 2.5 N [0.55 pounds] applied perpendicular opening at the outer most contact point of the sleeve.

5 As specified for a split module sleeve. for a solid sleeve. ⊕|∅ 0.55|A-B|D-C|S|28.0Ⓟ

6 Blend/taper bore entrance.

Figure 10 – Dimensional diagram of the multimode duplex receptacle

5.3.2 Single-mode input interface

Table 9 specifies the input interface requirements.

To assist in fault isolation, the input interface shall activate a loss-of-signal (LOL) state when the optical data cannot be detected. The optical power threshold to activate the LOL state, the LOL optical power hysteresis, and the reaction time for the LOL state change to occur are also specified. The design of the machine receiving data from the interface shall determine how the state change is subsequently indicated.

Table 9 – Single-mode input interface characteristics

Parameter	Min	Max	Unit
Saturation level[1]	-3		dBm
Sensitivity[1]		-28	dBm
Return loss[2]	12.5		dB
Acquisition time[3]		100	ms
LOL threshold	-40	-31	dBm
LOL hysteresis[4]	1.5		dB
Reaction time for LOL state change	0.25	5,000	ps

Notes

1 Based on any valid 8B/10B code pattern measured at, or extrapolated to, 10^{-15} BER shall meet this specification with worst-case conditions as specified in table 8 for the output interface and 5.3.3 for the fiber optic link.

2 This measurement is made using a 4-meter single-mode duplex jumper cable and includes only the power in the fundamental mode of the single-mode fiber.

3 The time to reach synchronization after the removal of the condition that caused the loss of synchronization. The pattern sent for synchronization is either the idle pattern or an alternation of idle and data characters. See clauses 6 and 7.

4 Required to avoid random transitions between LOL being active and inactive when input power is near threshold level.

5.3.3 Single-mode link specifications

Table 2 lists the specifications for links using 9/125 µm fiber cable. The trunk cable to which the jumper cables are connected shall have optical properties that conform to the specifications in table 2.

5.3.4 Single-mode cable specifications

5.3.4.1 Single-mode trunk cable optical specifications

The specifications for single-mode trunk cable support attachment of SBCON-capable devices are shown in table 10.

5.3.4.2 Single-mode duplex jumper cable specifications

The specifications for single-mode duplex jumper cable support attachment to SBCON-capable devices are shown in table 11.

5.3.5 Single-mode connector

SBCON specifies the use of the Fibre Channel single-mode connector as the connector for single-mode use (see FC-PH clause 10).

5.3.6 Class 1 laser safety

Meeting the requirements for a Class 1 classification is very important for an single-mode optical interconnect system due to the potential for exposure to laser radiation, and, therefore, the single-mode TRS shall comply with the local safety codes.

5.4 Bit-error-rate thresholding

The SBCON I/O interface is tolerant of bit errors that may be detected as code-violation, sequencing, or CRC errors. As the rate of errors

Table 10 – Single-mode trunk cable specification

Type of fiber	Dispersion unshifted
Operating wavelength	1,261 to 1,360 nm
Mode field diameter[1]	9 to 10 µm ± 10%
Core concentricity error[2]	1 µm maximum
Cladding diameter[2]	125 ± 2 µm
Cladding noncircularity[2]	2% maximum
Zero dispersion wavelength[3]	1,295-1,322 nm (nominal 1,310 nm)
Zero dispersion slope[3]	0.095 ps/(nm2*km) maximum
Cutoff wavelength (λc)[4]	1,280 nm maximum
Cutoff wavelength (λcc)[5]	1,260 nm maximum
Attenuation above nominal[6]	0.06 db/km maximum
Attenuation[7]	0.5 dB/km at 1,310 nm

Notes

1 Measured in accordance with EIA 455 FOTP 164, 167, or equivalent.

2 Measured in accordance with EIA 455 FOTP 45, 48, or equivalent.

3 Measured in accordance with EIA 455 FOTP 168 or equivalent.

4 Measured in accordance with EIA 455 FOTP 80 or equivalent.

5 Measured in accordance with EIA 455 FOTP 170 or equivalent.

6 The maximum attenuation for wavelengths from 1,261 to 1,360 nm shall not exceed the attenuation at 1,310 nm by more than 0.06 db/km. (Typically, this specification can be met by fiber with 1,383-nm water absorption peaks below 2 dB/km.)

7 The attenuation is a typical value, not a specification. Use the actual db/km attenuation value when calculating a link loss budget. The total link loss, however, shall not exceed 14.0 dB.

Table 11 – Single-mode duplex jumper cable specification

Type of fiber	Dispersion unshifted
Operating wavelength	1,261 to 1,360 nm
Mode field diameter[1]	9.0 ± 1.0 µm
Zero dispersion wavelength[3]	1,310 ± 10 nm
Dispersion (1,270-1,340 nm)[3]	6.0 ps/(nm-km) maximum
Cutoff wavelength[4]	1,260 nm maximum
Attenuation (1,270-1,340 nm)[6]	0.8 db/km max
Connector color	Gray
Jacket color	Yellow

Notes

1 Measured in accordance with EIA 455 FOTP 164, 167, or equivalent.

2 Measured in accordance with EIA 455 FOTP 45, 48, or equivalent.

3 Measured in accordance with EIA 455 FOTP 168 or equivalent.

4 Measured in accordance with EIA 455 FOTP 80 or equivalent.

5 Measured in accordance with EIA 455 FOTP 170 or equivalent.

6 Measured In accordance with EIA 455 FOTP 78 or equivalent.

increases, performance decreases. Beyond a certain point, the error rate may cause noticeable degradation.

The bit-error-rate thresholding process is designed to detect an increased error rate before performance degradation becomes serious. When the specified bit-error-rate threshold is reached, a report shall be made that a loss of signal may be impending. The system shall continue with degraded operation, but since the threshold is chosen at a point where minimal effect on system performance is expected, maintenance may be deferred until a more convenient time.

The bit-error-rate threshold shall be set at 10^{-10} (one bit error in ten to the tenth bits received). The bit-error rate shall be measured during frame, idle-function, and sequence reception. The bit-error rate shall not be calculated during times when character or signal synchronization has been lost, when in the offline state, or when in a link-failure state.

5.4.1 Types of link errors caused by bit errors

Bit errors are not detected directly. They usually cause one or more of the following types of errors:

a) Code-violation errors (see 11.1.3);

b) Sequencing errors (see 11.1.4);

c) CRC errors (see 11.1.4).

Only code-violation errors shall be counted toward the bit-error-rate threshold.

5.4.2 Error bursts

A single error may result in several related errors occurring closely together which in turn may result in multiple counts. A character might have a single bit error in it that caused a code-violation error. A disparity error may occur on a following character, caused by the same single error. To prevent multiple error counts from a single cause, the following concept of an error burst is introduced:

a) An error burst shall be a time period 1.5 ± 0.5 seconds long during which one or more code-violation errors occur;

b) Only one error in an error burst shall be counted toward the bit-error threshold.

5.4.3 Bit-error-rate-thresholding measurement

Measurement of bit-error-rate thresholding shall be accomplished by counting the number of error bursts that occur in a 5-minute period. When the count equals a specific number, the threshold is exceeded. This specific number is called the *threshold error count*.

Channels and control units shall detect a bit-error-rate threshold when 15 error bursts occur in a 5-minute period.

Dynamic-switch ports shall detect a bit-error-rate threshold when 12 error bursts occur in any 5-minute period.

The required accuracy of the 5-minute periods mentioned above shall be ± 0.3 seconds.

A dynamic-switch port shall detect a bit-error-rate threshold condition if the threshold error count occurs in any 5-minute period. The bit-error-counting process shall be reset and restarted when the offline state is exited and when a link failure is no longer recognized. When a dynamic-switch port detects a bit-error-rate-threshold condition, the bit-error-counting process shall also be reset and remain reset until it is restarted after a 5-minute period. The only other times when the dynamic switch may reset and restart the bit-error-counting process shall be when the dynamic switch is powered on or when a similar initialization procedure occurs.

Channels and control units shall reset and restart the bit-error-counting process when the offline state is exited, when a link failure is no longer recognized, and when a model-dependent amount of time has elapsed after a bit-error-rate threshold is detected. After a bit-error-rate threshold is detected, at least 15 additional error bursts shall occur before the next bit-error-rate threshold is detected. In addition, channels and control units may restart the bit-error-counting process whenever the 5-minute timer expires regardless of whether a bit-error-rate threshold is reached. The bit-error-counting process may also be reset and restarted when an initialization procedure occurs.

NOTE – The delay in restarting bit-error counting after a threshold is detected is to prevent threshold-exceeded conditions from being reported at an excessive frequency. A dynamic switch shall not report a threshold exceeded condition more frequently than once during any 5-minute period.

6 Transmission code

Information to be transmitted over a link is encoded eight bits at a time into a *10-bit transmission character* and then sent serially by bit. Information received over a link shall be collected ten bits at a time, and those transmission characters that are used for data, called *data characters*, shall be decoded into the correct eight-bit codes. The 10-bit transmission code provides for all 256 eight-bit combinations. Some of the remaining transmission characters, referred to as *special characters*, are used for functions which are to be distinguishable from the contents of a frame. Special characters do not have an eight-bit value from which they are encoded; instead, they have meaning only as transmission characters, which, like the transmission characters used for the contents of the frame, are transmitted serially by bit.

NOTE – The primary rational for the use of a transmission code is to improve the transmission characteristics of information to be transferred across the link. The encodings defined by the transmission code ensure that sufficient transitions are present in the serial-bit stream to make clock recovery possible at the receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information.

Ordered sets, sequences, and the idle-function provide for the delineation of a frame, control of dynamic connections, and synchronization between the transmitter and receiver circuits at opposite ends of a link. The following is a list of the functions performed by these transmission characters, ordered sets, or sequences:

- idle-function;
- not-operational (NOS) sequence;
- connect-start-of-frame (CSOF) delimiter;
- passive-start-of-frame (PSOF) delimiter;
- abort delimiter;
- disconnect-end-of-frame (DEOF) delimiter;
- passive-end-of-frame (PEOF) delimiter;
- unconditional-disconnect (UD) sequence;
- unconditional-disconnect-response (UDR) sequence;
- offline (OLS) sequence.

Sequences are uniquely identified by the particular ordered set used. Four ordered sets are defined for use in sequences:

- not operational;
- unconditional disconnect;
- unconditional-disconnect response;
- offline.

The respective sequences comprised of these defined ordered sets are called *defined sequences*. The ordered sets comprised of a special transmission character and a data character that are not defined for use in a particular sequence are reserved. A sequence comprised of reserved ordered sets is called a *reserved sequence* and, if recognized, causes a link error (sequencing error) to be detected.

6.1 Notation conventions

SBCON uses letter notation for describing the bits of an unencoded eight-bit byte. The following text describes the translation process between these notations and provides a translation example. It also describes the conventions used to name valid transmission characters. This text is provided for the purposes of terminology clarification only and is not intended to restrict the implementation of SBCON functions in any way.

The bit notation of A,B,C,D,E,F,G,H for an unencoded eight-bit byte is used in the description of the 8B/10B transmission code. The bits A,B,C,D,E,F,G,H are translated to bits a,b,c,d,e,i,f,g,h,j of 10-bit transmission characters; there is a correspondence between bit A and bit a, B and b, C and c, D and d, E and e, F and f, G and g, and H and h. Bits i and j are derived, respectively, from (A,B,C,D,E) and (F,G,H).

The notation for unencoded-SBCON information (bits 0,1,2,3,4,5,6,7) as described in this standard has bit 0 as the most significant bit (msb) and bit 7 as the least significant bit (lsb).

The bit labeled A in the description of the 8B/10B transmission code corresponds to the SBCON information bit labeled 7, B corresponds to bit 6, etc. The following shows the correspondence for the unencoded eight-bit SBCON information byte:

		msb							lsb
SBCON bit designation	-->	0	1	2	3	4	5	6	7
8B/10B bit designation	-->	H	G	F	E	D	C	B	A

The current documentation for the 8B/10B transmission code shows the byte in the order ABCDE FGH, which therefore corresponds to SBCON bits numbered 76543 210.

To clarify this correspondence, the following example shows the conversion from an unencoded-SBCON-information byte to a transmission character (using 8B/10B transmission-code notation).

SBCON byte notation:

x 'C5'

Bits:	0 1 2 3	4 5 6 7
	1 1 0 0	0 1 0 1

Converted to 8B/10B notation. Note that the order of bits is reversed:

Bits:	A B C D E	F G H
	1 0 1 0 0	0 1 1

Translated to a transmission character in the 8B/10B transmission code:

Bits:	a b c d e i	f g h j
	1 0 1 0 0 1	0 1 1 0

In table 12, the column “Data Byte Name” refers to values derived from the eight-bit representation of a character; for example, the x’C5’ is represented by the encoding character D5.6. “D” represents “data”; then the value of the EDCBA bits taken as a binary integer precedes the period, after which the value of the HGF bits taken as a binary integer follows the period. The following illustrates this naming convention:

(8-bit)

data byte name	H G F	E D C B A
D5.6	1 1 0	0 0 1 0 1
	(6)	(5)

In table 13, in the column “Special Code Name,” names are introduced by the letter “K” to distinguish them from data characters. The special code names were derived historically in a manner similar to data characters, but now, by convention, the special characters have no direct eight-bit counterpart.

6.2 8B/10B transmission code

The following information describes how the tables shall be used for both generating valid transmission characters (encoding) and checking the validity of received transmission characters (decoding).

6.2.1 Transmission order

Within the definition, the bit positions of the transmission characters are labeled a, b, c, d, e, i ,f, g, h, and j. Bit “a” shall be transmitted first, followed by bits “b,” “c,” “d,” “e,” “i,” “f,” “g,” “h,” and “j,” in that order. (Note that bit “i” shall be transmitted between bit “e” and bit “f,” rather than in the order that would be indicated by the letters of the alphabet.)

Characters within special-character-combinations (ordered sets and idle-function), as specified by 6.3, shall be transmitted sequentially beginning with the special character used to distinguish the ordered set or idle-function and proceeding character by character from left to right within the definition of the ordered set or idle-function until all characters of the ordered set or idle-function are transmitted.

The contents of a frame, as specified in clause 8 shall be transmitted sequentially beginning with the ordered set used to denote the start-of-frame (the SOF delimiter) and proceeding character by character from left to right within the definition of the frame until the ordered set used to denote the end-of-frame (the EOF delimiter) is transmitted.

6.2.2 Valid and invalid transmission characters

Tables 12 and 13 define the valid data characters (D characters) and valid special characters (K characters), respectively. The tables are used for both generating valid transmission characters (encoding) and checking the validity of received transmission characters (decoding). In the tables, each data-byte or special-code entry has two columns that represent two (not necessarily different) transmission characters. The two columns correspond to the valid transmission character based on the current value of the running disparity (“Current RD –” or “Current RD +”). Running disparity is a binary parameter with either the value negative (–) or the value positive (+).

The following rules for running disparity shall be used to calculate the new running-disparity value for transmission characters that have been transmitted (transmitter's running disparity) and that have been received (receiver's running disparity):

a) Running disparity for a transmission character shall be calculated on the basis of sub-blocks, where the first six bits (abcdei) form one sub-block (six-bit sub-block) and the second four bits (fghj) form the other sub-block (four-bit sub-block);

b) Running disparity at the beginning of the six-bit sub-block is the running disparity at the end of the last transmission character;

c) Running disparity at the beginning of the four-bit sub-block is the running disparity at the end of the six-bit sub-block;

d) Running disparity at the end of the transmission character is the running disparity at the end of the four-bit sub-block.

Running disparity for the sub-blocks shall be calculated as follows:

a) Running disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the six-bit sub-block is 000111, and it is positive at the end of the four-bit sub-block if the four-bit sub-block is 0011;

b) Running disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the six-bit sub-block if the six-bit sub-block is 111000, and it is negative at the end of the four-bit sub-block if the four-bit sub-block is 1100;

c) If neither of the above two conditions applies, running disparity at the end of the sub-block is the same as at the beginning of the sub-block.

After powering on, the receiver should assume either a positive or negative value for its initial running disparity. Upon reception of any transmission character, the receiver shall determine whether the transmission character is valid or invalid and shall calculate a new value for its running disparity based on the contents of the received character.

6.2.2.1 Generating transmission characters

The appropriate entry in the table shall be found for the data byte or the special code for which a transmission character is to be generated (encoded). The current value of the transmitter's running disparity shall be used to select the transmission character from its corresponding column. For each transmission character transmitted, a new value of the running disparity shall be calculated. This new value shall be used as the transmitter's current running disparity for the next data byte or special code to be encoded and transmitted.

6.2.2.2 Checking the validity of received transmission characters

The column corresponding to the current value of the receiver's running disparity shall be searched for the received transmission character. If the received transmission character is found in the proper column, according to the current-running disparity, then the transmission character shall be considered valid and the associated data byte or special code determined (decoded). If the received transmission character is not found in that column, then the transmission character shall be considered invalid. Independent of the transmission character's validity, the received transmission character shall be used to calculate a new value of running disparity. The new value shall be used as the receiver's current running disparity for the next received transmission character.

Table 12 – Valid data characters

Data Byte Name	Bits HGF EDCBA	Current RD –	Current RD +
		abcdei fghj	abcdei fghj
D0.0	000 00000	100111 0100	011000 1011
D1.0	000 00001	011101 0100	100010 1011
D2.0	000 00010	101101 0100	010010 1011
D3.0	000 00011	110001 1011	110001 0100
D4.0	000 00100	110101 0100	001010 1011
D5.0	000 00101	101001 1011	101001 0100
D6.0	000 00110	011001 1011	011001 0100
D7.0	000 00111	111000 1011	000111 0100
D8.0	000 01000	111001 0100	000110 1011
D9.0	000 01001	100101 1011	100101 0100
D10.0	000 01010	010101 1011	010101 0100
D11.0	000 01011	110100 1011	110100 0100
D12.0	000 01100	001101 1011	001101 0100
D13.0	000 01101	101100 1011	101100 0100
D14.0	000 01110	011100 1011	011100 0100
D15.0	000 01111	010111 0100	101000 1011
D16.0	000 10000	011011 0100	100100 1011
D17.0	000 10001	100011 1011	100011 0100
D18.0	000 10010	010011 1011	010011 0100
D19.0	000 10011	110010 1011	110010 0100
D20.0	000 10100	001011 1011	001011 0100
D21.0	000 10101	101010 1011	101010 0100
D22.0	000 10110	011010 1011	011010 0100
D23.0	000 10111	111010 0100	000101 1011
D24.0	000 11000	110011 0100	001100 1011
D25.0	000 11001	100110 1011	100110 0100
D26.0	000 11010	010110 1011	010110 0100
D27.0	000 11011	110110 0100	001001 1011
D28.0	000 11100	001110 1011	001110 0100
D29.0	000 11101	101110 0100	010001 1011
D30.0	000 11110	011110 0100	100001 1011
D31.0	000 11111	101011 0100	010100 1011
D0.1	001 00000	100111 1001	011000 1001
D1.1	001 00001	011101 1001	100010 1001
D2.1	001 00010	101101 1001	010010 1001
D3.1	001 00011	110001 1001	110001 1001
D4.1	001 00100	110101 1001	001010 1001
D5.1	001 00101	101001 1001	101001 1001
D6.1	001 00110	011001 1001	011001 1001
D7.1	001 00111	111000 1001	000111 1001
D8.1	001 01000	111001 1001	000110 1001
D9.1	001 01001	100101 1001	100101 1001
D10.1	001 01010	010101 1001	010101 1001
D11.1	001 01011	110100 1001	110100 1001
D12.1	001 01100	001101 1001	001101 1001
D13.1	001 01101	101100 1001	101100 1001
D14.1	001 01110	011100 1001	011100 1001
D15.1	001 01111	010111 1001	101000 1001
D16.1	001 10000	011011 1001	100100 1001
D17.1	001 10001	100011 1001	100011 1001
D18.1	001 10010	010011 1001	010011 1001
D19.1	001 10011	110010 1001	110010 1001
D20.1	001 10100	001011 1001	001011 1001
D21.1	001 10101	101010 1001	101010 1001
D22.1	001 10110	011010 1001	011010 1001
D23.1	001 10111	111010 1001	000101 1001
D24.1	001 11000	110011 1001	001100 1001
D25.1	001 11001	100110 1001	100110 1001
D26.1	001 11010	010110 1001	010110 1001
D27.1	001 11011	110110 1001	001001 1001
D28.1	001 11100	001110 1001	001110 1001
D29.1	001 11101	101110 1001	010001 1001
D30.1	001 11110	011110 1001	100001 1001
D31.1	001 11111	101011 1001	010100 1001
D0.2	010 00000	100111 0101	011000 0101
D1.2	010 00001	011101 0101	100010 0101
D2.2	010 00010	101101 0101	010010 0101
D3.2	010 00011	110001 0101	110001 0101
D4.2	010 00100	110101 0101	001010 0101
D5.2	010 00101	101001 0101	101001 0101
D6.2	010 00110	011001 0101	011001 0101
D7.2	010 00111	111000 0101	000111 0101
D8.2	010 01000	111001 0101	000110 0101
D9.2	010 01001	100101 0101	100101 0101
D10.2	010 01010	010101 0101	010101 0101
D11.2	010 01011	110100 0101	110100 0101
D12.2	010 01100	001101 0101	001101 0101
D13.2	010 01101	101100 0101	101100 0101
D14.2	010 01110	011100 0101	011100 0101
D15.2	010 01111	010111 0101	101000 0101
D16.2	010 10000	011011 0101	100100 0101
D17.2	010 10001	100011 0101	100011 0101
D18.2	010 10010	010011 0101	010011 0101
D19.2	010 10011	110010 0101	110010 0101
D20.2	010 10100	001011 0101	001011 0101
D21.2	010 10101	101010 0101	101010 0101
D22.2	010 10110	011010 0101	011010 0101
D23.2	010 10111	111010 0101	000101 0101
D24.2	010 11000	110011 0101	001100 0101
D25.2	010 11001	100110 0101	100110 0101
D26.2	010 11010	010110 0101	010110 0101
D27.2	010 11011	110110 0101	001001 0101
D28.2	010 11100	001110 0101	001110 0101
D29.2	010 11101	101110 0101	010001 0101
D30.2	010 11110	011110 0101	100001 0101
D31.2	010 11111	101011 0101	010100 0101

Table 12 *(continued)*

Data Byte Name	Bits HGF EDCBA	Current RD –	Current RD +
		abcdei fghj	abcdei fghj
D0.3	011 00000	100111 0011	011000 1100
D1.3	011 00001	011101 0011	100010 1100
D2.3	011 00010	101101 0011	010010 1100
D3.3	011 00011	110001 1100	110001 0011
D4.3	011 00100	110101 0011	001010 1100
D5.3	011 00101	101001 1100	101001 0011
D6.3	011 00110	011001 1100	011001 0011
D7.3	011 00111	111000 1100	000111 0011
D8.3	011 01000	111001 0011	000110 1100
D9.3	011 01001	100101 1100	100101 0011
D10.3	011 01010	010101 1100	010101 0011
D11.3	011 01011	110100 1100	110100 0011
D12.3	011 01100	001101 1100	001101 0011
D13.3	011 01101	101100 1100	101100 0011
D14.3	011 01110	011100 1100	011100 0011
D15.3	011 01111	010111 0011	101000 1100
D16.3	011 10000	011011 0011	100100 1100
D17.3	011 10001	100011 1100	100011 0011
D18.3	011 10010	010011 1100	010011 0011
D19.3	011 10011	110010 1100	110010 0011
D20.3	011 10100	001011 1100	001011 0011
D21.3	011 10101	101010 1100	101010 0011
D22.3	011 10110	011010 1100	011010 0011
D23.3	011 10111	111010 0011	000101 1100
D24.3	011 11000	110011 0011	001100 1100
D25.3	011 11001	100110 1100	100110 0011
D26.3	011 11010	010110 1100	010110 0011
D27.3	011 11011	110110 0011	001001 1100
D28.3	011 11100	001110 1100	001110 0011
D29.3	011 11101	101110 0011	010001 1100
D30.3	011 11110	011110 0011	100001 1100
D31.3	011 11111	101011 0011	010100 1100
D0.4	100 00000	100111 0010	011000 1101
D1.4	100 00001	011101 0010	100010 1101
D2.4	100 00010	101101 0010	010010 1101
D3.4	100 00011	110001 1101	110001 0010
D4.4	100 00100	110101 0010	001010 1101
D5.4	100 00101	101001 1101	101001 0010
D6.4	100 00110	011001 1101	011001 0010
D7.4	100 00111	111000 1101	000111 0010
D8.4	100 01000	111001 0010	000110 1101
D9.4	100 01001	100101 1101	100101 0010
D10.4	100 01010	010101 1101	010101 0010
D11.4	100 01011	110100 1101	110100 0010
D12.4	100 01100	001101 1101	001101 0010
D13.4	100 01101	101100 1101	101100 0010
D14.4	100 01110	011100 1101	011100 0010
D15.4	100 01111	010111 0010	101000 1101

Table 12 *(continued)*

Data Byte Name	Bits HGF EDCBA	Current RD –	Current RD +
		abcdei fghj	abcdei fghj
D16.4	100 10000	011011 0010	100100 1101
D17.4	100 10001	100011 1101	100011 0010
D18.4	100 10010	010011 1101	010011 0010
D19.4	100 10011	110010 1101	110010 0010
D20.4	100 10100	001011 1101	001011 0010
D21.4	100 10101	101010 1101	101010 0010
D22.4	100 10110	011010 1101	011010 0010
D23.4	100 10111	111010 0010	000101 1101
D24.4	100 11000	110011 0010	001100 1101
D25.4	100 11001	100110 1101	100110 0010
D26.4	100 11010	010110 1101	010110 0010
D27.4	100 11011	110110 0010	001001 1101
D28.4	100 11100	001110 1101	001110 0010
D29.4	100 11101	101110 0010	010001 1101
D30.4	100 11110	011110 0010	100001 1101
D31.4	100 11111	101011 0010	010100 1101
D0.5	101 00000	100111 1010	011000 1010
D1.5	101 00001	011101 1010	100010 1010
D2.5	101 00010	101101 1010	010010 1010
D3.5	101 00011	110001 1010	110001 1010
D4.5	101 00100	110101 1010	001010 1010
D5.5	101 00101	101001 1010	101001 1010
D6.5	101 00110	011001 1010	011001 1010
D7.5	101 00111	111000 1010	000111 1010
D8.5	101 01000	111001 1010	000110 1010
D9.5	101 01001	100101 1010	100101 1010
D10.5	101 01010	010101 1010	010101 1010
D11.5	101 01011	110100 1010	110100 1010
D12.5	101 01100	001101 1010	001101 1010
D13.5	101 01101	101100 1010	101100 1010
D14.5	101 01110	011100 1010	011100 1010
D15.5	101 01111	010111 1010	101000 1010
D16.5	101 10000	011011 1010	100100 1010
D17.5	101 10001	100011 1010	100011 1010
D18.5	101 10010	010011 1010	010011 1010
D19.5	101 10011	110010 1010	110010 1010
D20.5	101 10100	001011 1010	001011 1010
D21.5	101 10101	101010 1010	101010 1010
D22.5	101 10110	011010 1010	011010 1010
D23.5	101 10111	111010 1010	000101 1010
D24.5	101 11000	110011 1010	001100 1010
D25.5	101 11001	100110 1010	100110 1010
D26.5	101 11010	010110 1010	010110 1010
D27.5	101 11011	110110 1010	001001 1010
D28.5	101 11100	001110 1010	001110 1010
D29.5	101 11101	101110 1010	010001 1010
D30.5	101 11110	011110 1010	100001 1010
D31.5	101 11111	101011 1010	010100 1010

Table 12 *(continued)*

Data Byte Name	Bits	Current RD –	Current RD +
	HGF EDCBA	abcdei fghj	abcdei fghj
D0.6	110 00000	100111 0110	011000 0110
D1.6	110 00001	011101 0110	100010 0110
D2.6	110 00010	101101 0110	010010 0110
D3.6	110 00011	110001 0110	110001 0110
D4.6	110 00100	110101 0110	001010 0110
D5.6	110 00101	101001 0110	101001 0110
D6.6	110 00110	011001 0110	011001 0110
D7.6	110 00111	111000 0110	000111 0110
D8.6	110 01000	111001 0110	000110 0110
D9.6	110 01001	100101 0110	100101 0110
D10.6	110 01010	010101 0110	010101 0110
D11.6	110 01011	110100 0110	110100 0110
D12.6	110 01100	001101 0110	001101 0110
D13.6	110 01101	101100 0110	101100 0110
D14.6	110 01110	011100 0110	011100 0110
D15.6	110 01111	010111 0110	101000 0110
D16.6	110 10000	011011 0110	100100 0110
D17.6	110 10001	100011 0110	100011 0110
D18.6	110 10010	010011 0110	010011 0110
D19.6	110 10011	110010 0110	110010 0110
D20.6	110 10100	001011 0110	001011 0110
D21.6	110 10101	101010 0110	101010 0110
D22.6	110 10110	011010 0110	011010 0110
D23.6	110 10111	111010 0110	000101 0110
D24.6	110 11000	110011 0110	001100 0110
D25.6	110 11001	100110 0110	100110 0110
D26.6	110 11010	010110 0110	010110 0110
D27.6	110 11011	110110 0110	001001 0110
D28.6	110 11100	001110 0110	001110 0110
D29.6	110 11101	101110 0110	010001 0110
D30.6	110 11110	011110 0110	100001 0110
D31.6	110 11111	101011 0110	010100 0110

Table 12 *(concluded)*

Data Byte Name	Bits	Current RD –	Current RD +
	HGF EDCBA	abcdei fghj	abcdei fghj
D0.7	111 00000	100111 0001	011000 1110
D1.7	111 00001	011101 0001	100010 1110
D2.7	111 00010	101101 0001	010010 1110
D3.7	111 00011	110001 1110	110001 0001
D4.7	111 00100	110101 0001	001010 1110
D5.7	111 00101	101001 1110	101001 0001
D6.7	111 00110	011001 1110	011001 0001
D7.7	111 00111	111000 1110	000111 0001
D8.7	111 01000	111001 0001	000110 1110
D9.7	111 01001	100101 1110	100101 0001
D10.7	111 01010	010101 1110	010101 0001
D11.7	111 01011	110100 1110	110100 1000
D12.7	111 01100	001101 1110	001101 0001
D13.7	111 01101	101100 1110	101100 1000
D14.7	111 01110	011100 1110	011100 1000
D15.7	111 01111	010111 0001	101000 1110
D16.7	111 10000	011011 0001	100100 1110
D17.7	111 10001	100011 0111	100011 0001
D18.7	111 10010	010011 0111	010011 0001
D19.7	111 10011	110010 1110	110010 0001
D20.7	111 10100	001011 0111	001011 0001
D21.7	111 10101	101010 1110	101010 0001
D22.7	111 10110	011010 1110	011010 0001
D23.7	111 10111	111010 0001	000101 1110
D24.7	111 11000	110011 0001	001100 1110
D25.7	111 11001	100110 1110	100110 0001
D26.7	111 11010	010110 1110	010110 0001
D27.7	111 11011	110110 0001	001001 1110
D28.7	111 11100	001110 1110	001110 0001
D29.7	111 11101	101110 0001	010001 1110
D30.7	111 11110	011110 0001	100001 1110
D31.7	111 11111	101011 0001	010100 1110

Table 13 – Valid special characters

Special Code Name	Current RD –	Current RD +	Notes
	abcdei fghj	abcdei fghj	
K28.0	001111 0100	110000 1011	Reserved
K28.1	001111 1001	110000 0110	Comma
K28.2	001111 0101	110000 1010	
K28.3	001111 0011	110000 1100	Reserved
K28.4	001111 0010	110000 1101	
K28.5	001111 1010	110000 0101	Comma
K28.6	001111 0110	110000 1001	
K28.7	001111 1000	110000 0111	Comma
K23.7	111010 1000	000101 0111	Reserved
K27.7	110110 1000	001001 0111	Reserved
K29.7	101110 1000	010001 0111	Reserved
K30.7	011110 1000	100001 0111	Reserved

<u>Notes</u>

1 Reserved - valid transmission characters which are not defined for use by this standard

2 Comma - comma character

6.3 Special character combinations

The sender may transmit the following idle-function or ordered sets with any valid combination of encodings (see 6.2.2). The receiver shall recognize the following idle-function or ordered sets with any valid combination of encodings.

Tables 14 and 15 list the ordered sets that comprise the delimiter and sequence functions supported by SBCON:

Table 14 – Delimiter functions

Delimiter	Ordered Set
Connect-SOF[1]	K28.1 K28.7
Passive-SOF[1]	K28.5 K28.7
Disconnect-EOF[2]	K28.6 K28.1 K28.1
Passive-EOF[2]	K28.6 K28.2 K28.5
Abort	K28.6 K28.4 K28.4

Notes

1 Start-of-frame delimiter

2 End-of-frame delimiter

Table 15 – Sequence functions

Sequence	Ordered Set
Not-operational	K28.5 D0.2
Unconditional-disconnect	K28.5 D15.2
Unconditional-disconnect-response	K28.5 D16.2
Offline	K28.5 D24.2
Reserved	K28.5 DATA[1]

NOTE – 1: DATA is any valid data character that is not being used in a defined ordered set.

Table 16 lists the special character that comprises the idle-function supported by SBCON:

Table 16 – Idle-function

Idle-Function	Transmission Character
Idle character	K28.5

NOTE – The K28.5 special character is chosen as the first character of all sequence functions and the idle-function because the K28.5 character is used to identify the transmission-character boundary (character alignment) of the received bit stream.

For transmission of idle characters, a link-level facility shall send at least four consecutive idle characters (K28.5 characters). After this transmission requirement is met, idle characters can be continually transmitted as an even or odd number of idle characters.

For transmission of not-operational and offline sequences, the transmitter shall send the appropriate ordered set until the proper terminating conditions occur.

For transmission of unconditional-disconnect (UD) and unconditional-disconnect-response (UDR) sequences, the transmitter shall send the appropriate ordered set at least eight consecutive times and continue sending the ordered set until the proper terminating conditions occur.

After completion of a not-operational, UD, UDR, or offline sequence, the transmitter shall send a minimum of 20 consecutive idle characters before sending a frame.

An idle-function shall be recognized when a K28.5 character received does not appear as part of any ordered set. Even though one K28.5 character received can be recognized as an idle-function under certain conditions, recognition of the character as an idle-function is delayed until the next transmission character is received and recognized. If the next transmission character received after a K28.5 character is not a transmission character that completes an ordered set, then the previous transmission character (K28.5) shall be recognized as an idle-function. If the transmission character received after a K28.5 character is a K28.5 character, then the previous transmission character shall be recognized as an idle-function unless it was part of an ordered set. This second K28.5 character shall, in turn, not be recognized as an idle-function until the subsequent transmission character is received and recognized.

For recognition of not-operational, UD, UDR, offline, and reserved sequences, the receiver shall decode eight consecutive ordered sets of the appropriate sequence.

7 Link transmission and reception characteristics

This clause describes the characteristics and conditions that allow communication of information over the SBCON I/O interface.

7.1 Synchronization

Whenever a signal is present on a link, the receiver shall attempt to achieve and maintain synchronization on bit boundaries of the received encoded bit stream. Bit synchronization is achieved when the receiver has identified the start of each bit of the received signal. The method of achieving bit synchronization is model dependent.

> NOTE – The receiver contains model-dependent circuitry that provides transmission-character alignment. The circuitry will cause the received signal to be aligned on any occurrence of a K28.5 character.

Synchronization to transmission-character boundaries, hereafter referred to as *synchronization*, is achieved when the receiver has identified the transmission-character boundary as the one that is established by the input signal from the transmitter of the remote link-level facility or dynamic-switch port. When this condition is achieved, the receiver shall be in the *Synchronization-Acquired* state. When the transmission-character boundary identified by the receiver no longer matches the boundary established by the transmitter to which it is connected, the receiver shall be in the *Loss-Of-Synchronization* state. When the receiver is operational, it shall be in one of these two states. The determination of the receiver's state is based on the conditions described in 7.1.1 and 7.1.2.

Synchronization failures on either bit or transmission-character boundaries are not separately identifiable and cause loss-of-sync link-signal errors.

When the receiver is in the loss-of-synchronization state, the procedure described in 7.1.2 shall be used to regain synchronization.

If the receiver remains in the loss-of-synchronization state for longer than the link-interval duration, a link-failure condition shall be recognized (see 8.5 and 10.5).

When the loss-of-synchronization state is entered by a dynamic-switch port that is not in the static state, connection recovery shall be initiated within 100 milliseconds (see 10.4).

For a dynamically connected-switch-port, the transmission character received that caused the loss-of-synchronization state and the subsequent transmission characters shall not be propagated to the other switch port. The other switch port shall transmit idle characters until connection recovery is initiated.

For a dynamic-switch port that is in the static state, propagation of transmission characters is uninterrupted, and connection recovery shall not be initiated when the loss-of-synchronization state is entered.

7.1.1 Conditions that cause loss of synchronization

Any of the following three conditions shall cause a receiver to enter the loss-of-synchronization state:

a) Completion of the loss-of-synchronization procedure;

b) Transition to power on;

c) Detection of loss of signal.

7.1.1.1 Loss-of-synchronization procedure

The loss-of-synchronization procedure defines the method by which the receiver changes from the synchronization-acquired state to the loss-of-synchronization state. The procedure tests each received transmission character to determine whether the transmission character is valid as defined by tables 12 and 13.

Starting in the synchronization-acquired state, the receiver shall check each received transmission character to determine if the character is valid. The receiver shall continue to check the transmission characters and shall remain in the synchronization-acquired state until four invalid transmission characters are detected according to the following procedural rules. The following five detection states, also illustrated in figure 11 are defined as part of the loss-of-synchronization procedure:

a) *State 1:* No invalid transmission character has been detected;

b) *State 2:* The first invalid transmission character is detected;

c) *State 3:* The second invalid transmission character is detected;

d) *State 4:* The third invalid transmission character is detected;

e) *State 5:* The fourth invalid transmission character is detected, and the loss-of-synchronization state is entered.

When the procedure is in detection state 1, checking for an invalid transmission character shall be performed. After each invalid transmission character is detected, one of detection states 2 through 5 shall be entered.

When the procedure is in detection state 2, 3, or 4, checking for additional invalid transmission characters shall be performed in groups of 15 consecutive transmission characters. If 15 consecutive valid transmission characters are received, the indication of detection of the previous invalid transmission character shall be reset, and the previous detection state shall be entered. The loss-of-synchronization procedure is completed when detection state 5 is entered.

State 1 shall be entered on a transition to the synchronization-acquired state due to acquisition of synchronization, or after state 2 is reset.

State 2 shall be entered after the first invalid transmission character is detected or after the previous detection state (state 3) is reset. Subsequent transmission characters received shall be checked to determine whether an additional invalid transmission character is detected within the next consecutive 15 or fewer transmission characters received. If an additional invalid transmission character is detected within the next consecutive 15 or fewer transmission characters received, state 3 shall be entered. If 15 consecutive valid transmission characters are received, state 1 shall be entered.

State 3 shall be entered after the second invalid transmission character is detected or after the previous detection state (state 4) is reset. Subsequent transmission characters received shall be checked to determine whether an additional invalid transmission character is detected within the next consecutive 15 or fewer transmission characters received. If an additional invalid transmission character is detected within the next consecutive 15 or fewer transmission characters received, state 4 shall be entered. If 15 consecutive valid transmission characters are received, state 2 shall be entered.

State 4 shall be entered after the third invalid transmission character is detected. Subsequent transmission characters received shall be checked to determine whether an additional invalid transmission character is detected within the next consecutive 15 or fewer transmission characters received. If an additional invalid transmission character is detected within the next consecutive 15 or fewer transmission characters received, the loss-of-synchronization state (state 5) shall be entered. If 15 consecutive valid transmission characters are received, state 3 shall be entered.

State 5 shall be entered after the fourth invalid transmission character is detected. Subsequent transmission characters received shall not be checked by the loss-of-synchronization procedure, since this procedure is complete, and the loss-of-synchronization state is entered. The receiver shall remain in the loss-of-synchronization state until the receiver regains synchronization.

7.1.1.2 Transition to power on

When the receiver is operational after being powered on, the receiver shall enter the loss-of-synchronization state.

7.1.1.3 Detection of loss of signal

When a loss-of-signal condition is recognized, the loss-of-synchronization state shall be entered (if the receiver is not presently in that state), and the receiver shall remain in this state until the loss-of-signal condition is corrected. When the loss-of-signal condition is corrected, the receiver shall remain in the loss-of-synchronization state until the receiver regains synchronization.

7.1.2 Acquisition of synchronization

When a receiver that is in the loss-of-synchronization state and that does not have a loss-of-signal condition recognizes 15 or more K28.5 characters without intervening code-violation errors, the receiver shall enter the synchronization-acquired state. For purposes of acquisition of synchronization, the K28.5 characters recognized need not be part of an ordered set. The method used by the receiver to acquire synchronization is model dependent.

A receiver, having bit synchronization and interrogating the SBCON I/O interface, shall achieve the synchronization-acquired state within 100 milliseconds if a sequence or K28.5 characters are present at the receiver.

7.1.3 Synchronization state diagram

The loss-of-synchronization procedure and the acquisition of synchronization are illustrated in the state diagram in figure 11. States 1 through 5 are the five detection states described by the ordered list in 7.1.1.1. State 1 is the initial detection state entered by a receiver upon acquisition of synchronization. States 1 through 4 are detection states which are possible only when the receiver has achieved synchronization and is performing the loss-of-synchronization procedure (see 7.1.1.1). Entry into state 5 occurs when the receiver loses synchronization. State 5 and state-transition d illustrate the acquisition of synchronization (see 7.1.2).

State Transition Conditions: The following items represent the four state transitions used in the state diagram:

a) The first invalid transmission character is detected;

b) An additional invalid transmission character is detected in the next 15 or fewer transmission characters;

c) 15 consecutive valid transmission characters are received;

d) The receiver acquires synchronization.

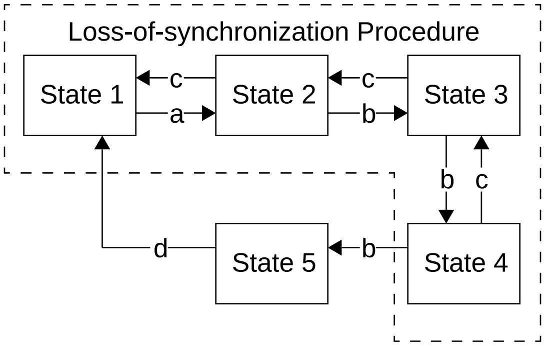


Figure 11 – Synchronization state diagram

7.2 Idle character

An *idle character* is a special character which constitutes a unique combination of bits that cannot normally occur within the contents of a frame (see table 16).

Idle characters shall be sent on a link to perform the *idle-function* whenever no other frame or sequence is to be sent on the link. If an idle character is recognized within a frame, a link error shall be detected.

An idle character is recognized as performing the idle-function only when the transmission character does not appear as part of an ordered set.

When it is necessary for the dynamic switch to adjust the rate of transmission characters being exchanged between two connected-switch-ports, the dynamic switch shall adjust the number of idle characters that are sent between frames. Either one or two idle characters may be added or removed from the idle-function between any two frames exchanged between the two connected-switch-ports (see 10.9 for other handling of idle characters by the dynamic switch).

To allow the dynamic switch to adjust the number of idle characters between frames, each link-level facility shall ensure that at least four idle characters are sent between frames. The maximum number is dependent on conditions at the channel and control unit. If a link-level facility sends fewer than four idle characters between frames, the adjustment of the number of idle characters between frames may cause link errors to occur, which, if observed by the receiving link-level facility, may cause one or more frames to be discarded.

7.3 Ordered-set reception

Ordered-set reception is the process of receiving and recognizing a group of transmission characters that satisfy a defined ordered set. Ordered-set reception can start any time after the receiver has achieved synchronization. When synchronization is achieved, the transmission-character boundaries of the serial bits being received are established, which permits the receiver to search for consecutive transmission characters that constitute an ordered set.

Ordered-set reception shall start when a transmission character received is the first transmission character of an ordered set (see 6.3).

A dynamic-switch port may not recognize every ordered set. (See 10.9 for actions taken by a dynamic-switch port in each switch-port state for each transmission character and ordered set received.)

Ordered-set reception, when started, shall continue until one of the following ordered-set-terminating conditions occurs:

a) The next transmission character received does not sustain any ordered set and is not a valid first transmission character of another ordered set;

b) A code-violation error is detected (see 11.1.3);

c) A loss-of-signal condition is detected (see 11.1.2);

d) A loss-of-sync (loss of transmission-character synchronization) condition is detected (see 11.1.2).

When ordered-set reception is ended before an ordered set has been recognized, that which has been received shall not be considered an ordered set; it shall be discarded, and a link error shall be detected. Whether the detected link error causes a link-error condition to be recognized depends on the conditions present (see 11.1). After ordered-set reception is ended by one of the ordered-set-terminating conditions, each subsequent transmission character received shall be checked to determine if another ordered set is starting. If a code-violation error or a loss-of-signal or loss-of-sync condition is detected or if the subsequent transmission character received does not cause either ordered-set reception or frame reception to start, then the transmission character received shall be discarded, and each subsequent transmission character received shall be checked to determine if another ordered set is starting. Additional link errors of the same type shall not cause additional link errors to be detected.

After an ordered set is recognized as a delimiter by the receiving link-level facility, the rules for the corresponding delimiter shall govern the action to be taken (see clause 8 and 8.4.2).

When an ordered set that is not a delimiter has been recognized, the rules for sequence reception shall govern the action to be taken (see 7.4.1).

7.4 Sequence transmission and reception

A sequence is used to perform recovery and signal states and transitions that either cannot adequately be signaled using frames or need to be signaled while transmission of a frame is not possible, such as recovery from a connection error, or transition from the offline state to the online state. A sequence is made up of a number of ordered sets defined for a particular sequence (see table 15).

7.4.1 Sequence reception

Sequence reception is the process of receiving and recognizing an ordered set that is not a delimiter the specified number of times. A sequence shall be recognized when the specified number of consecutive ordered sets have been received and recognized (see 6.3 for a definition of the required number of consecutive times an ordered set is received for the sequence to be recognized). Sequence reception shall start when an ordered set is recognized as an ordered set that comprises a sequence.

After a sequence is recognized, the rules that govern the action to be taken shall be determined by the particular procedure that is initiated by the sequence (see 10.3, 10.4 and 10.5).

Sequence reception, when started, shall continue until one of the following sequence-terminating conditions occurs:

a) The next ordered set received does not sustain the present sequence;

b) Any one of the ordered-set-terminating conditions occurs.

When sequence reception is ended, the rules for ordered-set reception shall govern the action to be taken (see 7.3).

When sequence reception is ended before a sequence has been recognized, that which has been received shall not be considered a sequence, it shall be discarded, and a link error shall be detected. Whether the detected link error causes a link-error condition to be recognized depends on the conditions present (see 11.1).

When there is a dynamic connection between two dynamic-switch ports and one of the switch ports detects an ordered set used for a defined sequence, the ordered set shall be discarded and the other switch port is caused to send idle characters. After a defined sequence is recognized, the dynamic connection shall be removed, and connection recovery shall be initiated by the switch port that did not receive the sequence (see 10.4).

When reception of a defined sequence is started because of an ordered set and sequence reception is ended before a sequence is recognized, the dynamic connection shall remain. The transmission characters which are not part of an ordered set shall be propagated through the dynamic connection. A link error may be detected at the receiver if frame reception is in progress.

If a dynamic-switch port receives an ordered set used for a reserved sequence, it is unpredictable whether the dynamic connection is removed.

7.4.2 Not-operational sequence

The *not-operational sequence* is composed of the proper number of consecutive ordered sets used for the not-operational sequence. The not-operational sequence shall be sent by a link-level facility, or by a dynamic-switch port which is not part of a static connection, when a link failure is recognized because of conditions other than the recognition of the not-operational sequence. The not-operational sequence shall also be sent by a link-level facility, or by a dynamic-switch port which is not part of a static connection, when performing an offline procedure and waiting to recognize the offline sequence (see 10.3.2). The not-operational sequence shall also be sent by a dynamic-switch port when that switch port has a static connection with another switch port on which the not-operational sequence is being received.

The not-operational sequence shall be transmitted as long as the link failure continues to exist.

If at least the minimum number of consecutive ordered sets required to recognize the not-operational sequence are received and recognized, a link-failure condition shall be recognized.

7.4.3 Unconditional-disconnect sequence

The *unconditional-disconnect (UD) sequence* is composed of the proper number of ordered sets used for the UD sequence. The UD sequence shall be sent when a dynamic connection between the link of a link-level facility or a dynamic-switch port and any other link is to be removed after certain abnormal conditions are recognized (see 10.4). The UD sequence shall also be sent when the offline sequence is recognized.

A link-level facility or dynamic-switch port that initiates transmission of the UD sequence shall send at least the minimum number of consecutive ordered sets that comprise the UD sequence to ensure that the recipient recognizes the sequence.

If at least the minimum number of consecutive ordered sets required to recognize the UD sequence are received and recognized, the recipient shall enter the UD-reception state (see 10.4.2.2).

7.4.4 Unconditional-disconnect-response sequence

The *unconditional-disconnect-response (UDR) sequence* is composed of the proper number of consecutive ordered sets used for the UDR sequence. The UDR sequence shall be sent when a link-level facility or dynamic-switch port receives and recognizes the UD sequence.

A link-level facility or dynamic-switch port that initiates transmission of the UDR sequence shall send at least the minimum number of consecutive ordered sets that comprise the UDR sequence to ensure that the recipient recognizes the sequence.

If at least the minimum number of consecutive ordered sets required to recognize the UDR sequence are received and recognized, the recipient shall enter the UDR-reception state (see 10.4.2.3).

7.4.5 Offline sequence

The *offline sequence* is composed of the proper number of consecutive ordered sets used for the offline sequence. The offline sequence shall be sent when a link-level facility or dynamic-switch port must inform the other end of the link to not recognize link errors or link failure. The offline sequence shall also be sent when the not-operational sequence is recognized (see 10.3 and 10.5).

A link-level facility or dynamic-switch port that initiates transmission of the offline sequence shall send at least the minimum number of consecutive ordered sets that comprise the offline sequence to ensure that the recipient recognizes the sequence (see 10.3.1).

If at least the minimum number of consecutive ordered sets required to recognize the offline sequence are received and recognized, the recipient shall enter the offline-reception state (see 10.3.2.2).

When the offline sequence is being sent as part of performing the offline procedure, then, depending on conditions that cause the transmission of the offline sequence, it may be necessary to transmit the offline sequence for a certain minimum amount of time (see 10.3.1).

8 Frame structure

Information is transferred on the SBCON I/O interface in frames. A frame is a unit of information that is sent or received according to a defined format. This format delineates the start and end of the unit of information and defines the placement of the information within these boundaries. The basic frame format consists of a fixed-length link-header field, a variable-length information field, and a fixed-length link-trailer field, as shown in figure 12.

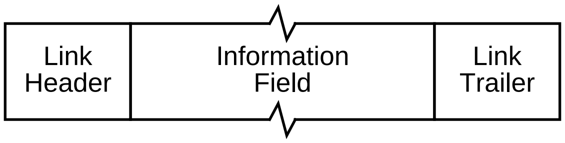


Figure 12 – Basic frame format

Two frame types are defined, the link-control frame and the device frame. Both frame types use this basic frame format. A link-control frame is used to transfer link-level information or specify link-level functions, and a device frame is used to transfer device-level information or specify device-level functions. Link-control frames that are used to specify certain link-level functions do not require an information field; they consist only of a link header and a link trailer as shown in figure 13.

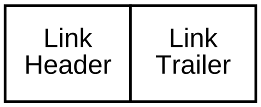


Figure 13 – Link control frame format

Device frames always require an information field. The information field may be further subdivided into two or more fields, for example, the device header and the device-information block (DIB), as shown in figure 14 (see clause 12 for the format of the information field for a device frame).

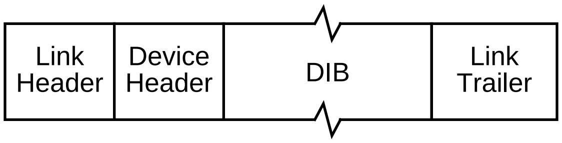


Figure 14 – Device frame format

Every frame is bounded by a start-of-frame (SOF) delimiter, which is found in the link header, and an end-of-frame (EOF) delimiter, which is found in the link trailer (see 8.1.1 and 8.3.2). Frame delimiters are unique combinations of special transmission characters. The information contained between frame delimiters, called the contents of the frame, consists of data characters (see 8.4). Therefore, the contents of the frame include the information contained in the link-header field, excluding the SOF delimiter, the information contained in the information field, and the information contained in the link-trailer field, excluding the EOF delimiter.

The link-header field contains, in addition to the SOF delimiter, the addressing information which is used to determine the physical path on which the frame is to be sent, the logical-addressing information which is used to determine the logical paths associated with the physical path, and the link-control information which is used to specify the frame type and the link-control function to be performed.

The link-trailer field contains, in addition to the EOF delimiter, the frame-checking information, called cyclic redundancy check (CRC). The CRC is used to verify that the integrity of the contents of the frame has been maintained during transmission.

The two types of frames, link-control frames and device frames, are distinguished by a bit in the link-header field. When a link-control frame is indicated, the link-control function is also specified in the link header. The link-control function specified determines the format of the information field, if required. The device-level function is specified in the information field. The format of a frame and consequently the location of the function specified cannot be determined until the integrity check on the frame is completed and no errors have been detected.

A frame is an integral number of transmission characters before frame reception and decoding occur. After decoding, the length of the contents of the frame is an integral number of bytes. Table 17 summarizes the maximum and minimum frame contents and information-field sizes. The values are shown in bytes.

NOTE – FC-SB frames transported over FC-PH are word-oriented, whereas SBCON frames are byte-oriented.

Table 17 – Frame sizes

Frame Type	Min. Frame Length	Max. Frame Length	Min. Info. Field	Max. Info. Field
Link Control	7	111	0	104
Device	12	1,035	5	1,028

The length of the information in the link header, not including the SOF delimiter, shall be five bytes, and the length of the information in the link trailer, not including the EOF delimiter, is two bytes. The minimum length of the information field is zero bytes for a link-control frame and five bytes for a device frame. The maximum length of the information field is 104 bytes for a link-control frame and 1,028 bytes for a device frame.

The minimum length of the contents of a frame is seven bytes for a link-control frame (five bytes in the link header, zero bytes in the information field, and two bytes in the link trailer) and 12 bytes for a device frame (five bytes in the link header, five bytes in the information field, and two bytes in the link trailer). The maximum length of the contents of a link-control frame is 111 bytes (five bytes in the link header, 104 bytes in the information field, and two bytes in the link trailer). The maximum length of the contents of a device frame is 1,035 bytes (five bytes in the link header, 1,028 bytes in the information field, and two bytes in the link trailer).

The actual number of bytes in the information field of a frame depends on the frame type and the function specified in the frame. For a link-control frame, the length of the information field is determined by the link-control function specified (see clause 9 for a description of the link-control functions and their information-field requirements). For a device frame, the length of the information field is determined by the device function specified and on conditions present at the channel or control unit (see clause 12 for a description of the device functions and their information-field requirements).

8.1 Link header

The link header consists of four fields: SOF delimiter, destination-address field, source-address field, and link-control field, as shown in figure 15.

SOF Delimiter	Destination Address	Source Address	Link Control

Figure 15 – Link header

Every valid frame sent has this link-header format.

8.1.1 Start-of-frame delimiter

The start-of-frame (SOF) delimiter is the first string of transmission characters of a frame. It is an ordered set of transmission characters that cannot appear in the contents of an error-free frame. The first field of the contents of the frame, the two-byte destination-address field, immediately follows the SOF delimiter.

There are two types of SOF delimiters, the connect-SOF delimiter, which is used to initiate the making of a dynamic connection, and the passive-SOF delimiter, which causes no action with respect to a dynamic connection.

The distinction between the two SOF delimiters may also be used by the recipient of the frame to determine what type of response or recovery action is appropriate for the frame. For example, a link-level-reject frame or port-reject frame is returned only in response to a frame having a connect-SOF delimiter.

8.1.2 Destination-address field

The destination-address field is a 16-bit field (before encoding). It is the first field of the contents of a frame and immediately follows the SOF delimiter. The destination-address field contains the destination link address and the destination logical address as shown in figure 16.

Destination Link Address	0 0 0 0	Destination Logical Address
0	8	12 15

Figure 16 – Destination address field

This format is referred to as a *full link address (FLA)*. Bits 8-11 of the destination address field shall be set to zeros; if any of bits 8-11 is not zero, a link-protocol error shall be detected.

8.1.2.1 Destination link address

The destination link address, bits 0-7 of the destination-address field, identifies the link-level facility of a channel or control unit that is the destination for the frame.

There are 255 valid destination link addresses, of which 254 may be assigned to a link-level facility of a channel or control unit. An address of all zeros is not assigned as a link address and is not valid as a destination link address in a frame. If an all-zero destination link address is used, a link-level facility receiving the frame shall detect a link-address error. If an all-zero destination link address is used in a frame that initiates a connection, a dynamic-switch port shall detect an address-invalid error. An address of all ones is not assigned as a link address but may be used as a destination link address in certain link-control frames. A destination link address of all ones is not permitted in a device frame. If a device frame is received with a destination link address of all ones, a link-protocol error shall be detected.

When it is permissible to use an address of all ones as a destination link address for a link-control frame, the recipient shall respond to that address as well as to its assigned link address. When it is not permissible, the recipient of a frame with a destination address of all ones shall detect a link-protocol error.

The destination-link-address field is used by the dynamic switch to route frames to their intended destination. If a dynamic switch is attached to the channel path, the frame starts with an initiate-connection control (connect-SOF delimiter followed by a valid character), and no dynamic connection exists (that is, the port is in the inactive state), then the destination link address is used by a dynamic switch to determine which dynamic connection is to be made in order to route the frame to the intended destination. If the destination link address is not all ones, the destination link address identifies the link address associated with the port with which the dynamic connection is to be made. If the destination link address is all ones, the dynamic connection is to be made with the dynamic-switch control unit (see 10.9).

8.1.2.2 Destination logical address

Bits 12-15 of the destination-address field contain the destination logical address. There are 16 possible destination logical addresses.

A destination logical address is provided and checked in all device frames and in some link-control frames, depending on the link control function specified. When a link control function is specified that does not require a destination logical address, the destination logical address is set to all zeros and checked for zeros by the recipient (see table 19).

8.1.3 Source-address field

The source-address field is a 16-bit field (before encoding) that immediately follows the destination-address field and is in the format of a full link address (FLA). The source-address field contains the source link address and the source logical address as shown in figure 17.

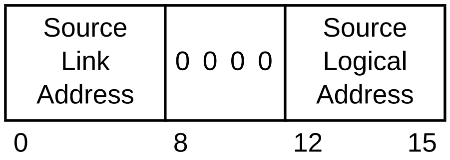


Figure 17 – Source address field

Bits 8-11 shall be set to zeros; if any of bits 8-11 are not zero, a link-protocol error shall be detected.

8.1.3.1 Source link address

The source link address, bits 0-7 of the source-address field, identifies the sending link-level facility.

A source link address cannot be all ones; if a frame with a source link address of all ones is received by a link-level facility, a link-protocol error shall be detected. If a dynamic-switch port detects a source link address of all ones in a frame that is initiating a connection, the dynamic-switch port shall detect an address-invalid error. It is permissible for a source link address of all zeros to be used for certain frames sent by an unidentified link-level facility. If a frame with a source link address of all zeros is received by a link-level facility when it is not permitted, a link-protocol error shall be detected.

An identified link-level facility shall provide its identity as the source of a frame by inserting its assigned link address in the source-address field of any frame that it sends. After a frame is received with a valid source address, the source address is used in most cases as the destination address in any subsequent response frame or future request frames to the same link-level facility.

8.1.3.2 Source logical address

Bits 12-15 of the source-address field contain the source logical address. There are 16 possible source logical addresses.

A source logical address shall be provided and checked in all device frames and in some link-control frames, depending on the link control function specified. When a link-control function is specified that does not require a source logical address, the source logical address shall be set to all zeros and checked for zeros by the recipient (see table 19).

8.1.4 Logical path identifier

The destination link address, destination logical address, source link address and source logical address together identify the logical path over which the frame is being sent.

The logical path determines the channel image and control-unit image sharing the link-level facilities that are identified by the link address. Whereas the link address determines the physical path and the attached facilities, the logical address determines the logical path and the logical image sharing these facilities.

8.1.5 Link-control field

The link-control field is an eight-bit field (before encoding), which indicates the type and format of the frame. The link-control field, which is the last field of the link header, immediately follows the source-address field.

The format of the link-control field depends on whether the frame type is a link-control frame or a device frame, as specified by the frame-type bit, bit 7 of the link-control field.

When bit 7 of the link-control field is set to one, a link-control frame is specified, and the link-control field has the format shown in figure 18.

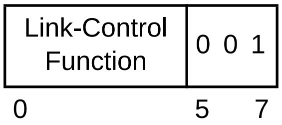


Figure 18 – Link control frame

The link-control function is specified by bits 0-4 of the link-control field. For a description of the link-control functions, see 9.1.

When the frame-type bit, bit 7 of the link-control field, is set to zero, a device frame is specified, and the link-control field has the format shown in figure 19.

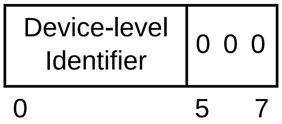


Figure 19 – Device frame

The device-level-identifier field specified by bits 0-4 identifies the device-level architecture that applies to the received device frame. When bits 0-4 are all zeros, the device-level architecture is that defined by the SBCON I/O interface architecture. If the device-level identifier field is set to a value other than all zeros, a link-protocol error shall be detected.

For both link-control and device frames, bits 5 and 6 are reserved and are set to zeros; if bits 5 and 6 are not zero, a link-protocol error shall be detected.

8.2 Information field

The information field is the first field following the link header. The size of the information field depends on the function performed by the particular frame. For a link-control frame, the length of the information field shall be from 0 to 104 bytes. For a device frame, the length of the information field shall be from 5 to 1,028 bytes (see 9.1 and clause 12).

8.3 Link trailer

The link trailer consists of a two-byte cyclic-redundancy-check (CRC) field followed by the EOF delimiter and has the format shown in figure 20.

Every valid frame has the link-trailer format described above. The presence of a link trailer is not known until the EOF delimiter is received and verified. Recognition of the EOF delimiter

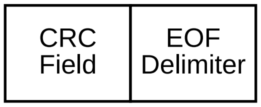


Figure 20 – Link trailer

during frame reception indicates that the preceding two transmission characters represent the two-byte cyclic-redundancy-check (CRC) field.

8.3.1 Cyclic-redundancy-check field

The cyclic-redundancy-check (CRC) field contains a 16-bit redundancy-check code that is used by the receiving link-level facility to detect most frame errors that occur during transmission of a frame from one link-level facility to another. The address, link-control, and information fields are used prior to encoding into transmission characters to generate the CRC and are therefore protected by the CRC. Bits of the CRC field are numbered 0-15. Bits 0-7 are in the byte which directly follows the information field.

At the sending link-level facility, each bit of the address, link-control, and information fields is treated as a coefficient of a polynomial D(x) of the order k, where k is 1 less than the total number of bits.

Coefficients of the polynomial D(x) are assigned starting at the x^k term in the following order:

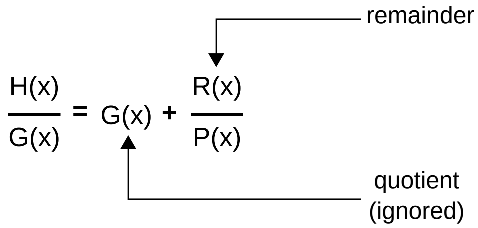
a) Bytes of the frame to be protected by the CRC, starting from the first byte of the destination-address field and continuing through the last byte of the information field (or last byte of the CRC field at the receiver);

b) Within a byte, starting at bit 7 of the byte and continuing through bit 0 (see 6.1).

Thus the coefficient of the x^k term of D(x) is the value of bit 7 of the first byte of the address field, $x^{(k-1)}$ is bit 6, etc. The coefficient of the x^0 term of the polynomial is the value of bit 0 of the byte in the frame just prior to the CRC field.

Polynomial H(x) is formed by multiplying D(x) by x^{16} and inverting 16 terms of the resulting polynomial starting at the $x^{(k+16)}$ term. H(x) is then divided by a reference polynomial P(x) of the order 16, using modulo-2 arithmetic. The remainder R(x) produced by dividing H(x) by P(x) is inverted, and the result is used as the 16-bit CRC value of the frame. The equations shown in figure 21 show the computation used.

$$H(x) = (x^{16})*D(x) + (x^k)*(x^{16} + x^{15} + ... + x)$$



$$CRC = R(x) + (x^{15} + x^{14} ... + 1) \quad [inversion]$$

Figure 21 – CRC formulas

The polynomial, P(x), to be used is referenced in ISO/IEC 9314-2.

$$x^{16} + x^{12} + x^5 +1$$

After the CRC value is determined, and before the CRC field of the frame has been encoded, the 16 bits of the CRC value are placed in the CRC field of the frame in the following order. The bits of the CRC value which correspond to the polynomial positions of x^8 through x^{15} are entered into bits 0-7 of the CRC field respectively; the bits of the CRC value which correspond to the polynomial positions of x^0 through x^7 are entered into bits 8 to 15 of the CRC field respectively.

After the frame is encoded, transmitted, and decoded, the receiving link-level facility uses a similar procedure to check the CRC of the frame that is received. The receiving link-level facility, however, treats the entire contents of the frame, including the CRC field, as coefficients of the polynomial D(x), and the remainder R(x) subsequently produced is not inverted, but is rather checked for a specific expected value. The specific expected value for a successfully transmitted frame is:

$$x^{12} + x^{11}+ x^{10} + x^8 + x^3 + x^2 + x^1 + 1 = (x'1D0F').$$

If the remainder does not match this expected value, a CRC error shall be detected.

8.3.2 End-of-frame delimiter

The end-of-frame (EOF) delimiter is an ordered set of transmission characters that shall terminate frame reception (see clause 6.) When the EOF delimiter is encountered during the reception of a frame, it signals the end of the frame and identifies the two transmission characters immediately preceding the EOF delimiter as representing the last two bytes of the contents of the frame, the two-byte CRC field. The EOF delimiter also indicates the extent of the frame for purposes of any applicable frame-length checks.

There are two types of EOF delimiters, the disconnect-EOF delimiter, which is used to initiate the removal of a dynamic connection, and the passive-EOF delimiter, which causes no action with respect to removing a dynamic connection.

The distinction between the two EOF delimiters may also be used by the receiving link-level facility to determine what type of response or recovery action is appropriate for the frame. For example, if a frame is ended by a disconnect-EOF delimiter, a link-level-reject frame cannot be returned.

8.3.3 Abort delimiter

The abort delimiter is an ordered set that may be used to prematurely terminate transmission of a frame. This delimiter is sent by a link-level facility if a link-abort condition is recognized while a frame is being sent. When an abort delimiter is received during frame reception, no CRC checking is performed, and the contents of the frame up to the abort delimiter are discarded (see 10.6).

8.4 Frame transmission and reception

This subclause describes the rules for sending and receiving a frame. These rules deal primarily with the transmission characters and the conditions that are satisfied to start and end frame transmission at both the sender and recipient. These rules are independent of frame type (link-control or device) and information content of a frame.

8.4.1 Frame transmission

The process of transmitting information serially bit by bit over a link such that the sequence of information transmitted conforms to the basic frame structure is called *frame transmission.*

Frame transmission by a link-level facility is permitted on a link when the link-level facility is in the inactive or the working state (see 10.8). Frame transmission by a dynamic-switch port is permitted on a link when the dynamic-switch port associated with that link is in the inactive, link-busy, connection, or static-pass-through state (see 10.9).

When a frame is transmitted on a link, the transmission of the idle-function shall be discontinued only for the time necessary to transmit the frame. The transition from idle-function transmission to frame transmission and the transition from frame transmission back to idle-function transmission occurs so that the appropriate transmission characters are always being transmitted on the link. When successive frames are sent, the link-level facility sending the frames shall ensure that at least four idle characters are sent between frames.

Frame transmission begins when the first transmission character of the start-of-frame (SOF) delimiter is sent and ends when the last transmission character of the end-of-frame (EOF) delimiter is sent. In the absence of errors, once transmission of a frame starts, the operation shall be synchronous and not interrupted until the last transmission character of the EOF delimiter is sent. After the EOF delimiter is sent, the idle-function shall be transmitted until the next frame or sequence is ready for transmission.

The information between the SOF delimiter and the EOF delimiter is called the contents of a frame. The contents of a frame are an integral multiple of eight bits before transmission and an integral multiple of 10 bits during transmission. Each group of eight bits to be sent represents a byte of information which is encoded into a data character (see clause 6 for a description of the bit order of sending transmission characters and for the encoding and decoding rules).

Special characters are not sent in the contents of a frame. If, after transmission of a frame is considered started, the synchronous transmission of the remainder of the frame cannot be sustained, frame transmission is terminated with the abort delimiter, if possible, followed by idle characters, if applicable.

The quantity of information sent as the contents of a frame depends on the type of frame, the conditions present at the channel or control unit, and model-dependent limits. Maximum and minimum values for the quantity of information that can be sent as the contents of a frame are defined in table 17. Errors are recognized if the quantity of information in the contents of a frame either exceeds the maximum value or is less than the minimum value (see 11.1).

8.4.2 Frame reception

A frame is received over a link by means of the frame-reception function. A link-level facility can receive a frame any time after the receiver has achieved synchronization. After synchronization is achieved, the transmission-character boundaries of the serial bits being received are established, which permits the receiver to search for an ordered set that represents an SOF delimiter (see 8.1.1).

A dynamic-switch port does not perform the frame-reception function. In certain port states, a connect-SOF delimiter followed by any valid transmission character is recognized and may cause a dynamic connection to be made. Depending on the port state, passive-SOF delimiters and disconnect-EOF delimiters are recognized (see 10.9 for information on how the dynamic-switch port processes the delimiter ordered sets and transmission characters that it receives).

Frame reception starts upon the recognition of an ordered set that comprises an SOF delimiter that is followed by any valid transmission character (see tables 12, 13 and 14.) An SOF delimiter received after frame reception is ended, followed by a reserved transmission character, allows frame reception to start; the use of the reserved transmission character shall be considered to be a reserved code received during frame reception, and a sequencing error shall be detected.

The transmission characters that are received as the contents of the frame shall be decoded to their corresponding eight-bit codes. The resultant eight-bit codes are then handled according to the format for a frame and the order in which these codes are received.

Frame reception, when started, shall continue until one of the following frame-terminating conditions occurs:

a) A delimiter ordered set is recognized;

b) A defined sequence is recognized. Additionally, a reserved sequence maybe treated as a frame-terminating condition, depending on the model. An error shall be detected after recognition of a reserved sequence even if frame reception is not terminated;

c) Two consecutive K28.5 characters are recognized;

d) The frame received has a quantity of transmission characters (including data characters, special characters that do not cause either frame-terminating condition a, b, or c, reserved transmission characters, and invalid transmission characters) in the contents of the frame that exceeds the maximum which can be accepted for the conditions present, called a maximum-frame-size error;

e) A link-signal-error condition is detected for loss of signal or loss of sync.

After frame reception is started, if special characters, combinations of special characters, ordered sets, reserved transmission characters, or code-violation errors occur that do not satisfy any of the frame-terminating conditions, a link error shall be detected. No immediate action is taken, and frame reception shall continue until one of the frame-terminating conditions occurs. After frame reception is ended, the frame shall be discarded and a link-error condition shall be recognized for this abnormal condition, provided no higher priority error is detected.

When frame reception is ended by frame-terminating condition a, b, or c, the type of delimiter or sequence recognized determines the subsequent action. If the delimiter is recognized as an abort delimiter or an EOF delimiter, the extent of the frame is established. If an EOF delimiter is recognized which is not valid for the link-level conditions present, or a sequence is recognized, a link error shall be detected. The link error is considered to be associated with the frame reception that just ended.

When frame reception is ended by other than an EOF delimiter, that which has been received is not considered to be a frame and is discarded. CRC checking is not performed on the partial frame received. For conditions other than the abort delimiter, a link-error condition shall be recognized for the abnormal condition that ended the frame (see 11.1).

If frame reception is ended by the detection of an SOF delimiter, frame reception may or may not be started for a new frame, depending on the conditions present and on the model.

If a data character that is not part of an ordered set is received after frame reception is ended, it shall be discarded, and a link error shall be detected.

If a special character that is not an idle character and that is not part of an ordered set is received after frame reception is ended, it shall be discarded, and a link error shall be detected.

If, before frame reception is started, an ordered set is received that is defined but not permitted for the conditions present (for example, an EOF delimiter is received before frame reception is started), it shall be discarded, and a link error shall be detected.

8.5 Link-interval duration

The link-interval duration is the time period used to measure the following:

a) A loss-of-signal or loss-of-sync condition that persists while not in certain offline states;

b) Transmission of the offline sequence after the not-operational sequence is no longer recognized (see 10.5);

c) The offline-transmission state and the off-line-reception state (see 10.3.2).

A link failure shall be recognized whenever the link-interval duration is exceeded while measuring items a and b above. For item c, when a link-level facility or dynamic-switch port is in an offline state for an attached link, it shall not recognize a link failure for that link.

The link-interval duration shall be 1 second, with a tolerance of +1.5 seconds and -0 second. For certain infrequent and unusual conditions, the link-interval duration may be exceeded.

8.6 Connection-recovery-interval duration

The connection-recovery-interval duration is used to measure only the states of the connection-recovery procedure (see 10.4).

A link failure shall be recognized whenever the connection-recovery-interval duration is exceeded while measuring one of the states of the connection-recovery procedure.

The connection-recovery-interval duration shall be 10 seconds, with a tolerance of +1.5 seconds and -0 seconds. For certain infrequent and unusual conditions the connection-recovery-interval duration may be exceeded.

9 Link-level-control functions

Link-level-control functions shall provide the means by which the physical paths that make up a channel path and the elements interconnected by the physical paths are brought to and maintained in the operational state. These functions shall be performed by means of link-control frames. Link-control frames shall be used to establish and maintain the physical and logical paths over which I/O operations or SBCON I/O interface functions are executed and provide information about conditions on the physical and logical paths that affect the transmission or reception of frames.

Link-level-control functions (or, for brevity, link-control functions) shall be performed during initialization or when conditions present prevent a frame that initiates a connection from either being routed to its intended destination or being accepted by the destination. These functions shall provide a level of communication capability that is consistent with the minimum functionality usually available when these functions are needed. For example, within any given channel path, during system initialization, the requirement to provide certain physical paths in the operational state exists, which requires the exchange of information over these physical paths before the system initialization for that path can proceed or complete.

The link control functions specified by the value of bits 0-4 of the link-control field are shown in table 18 .

Table 18 – Link-control functions and codes

Link-Control Function Bits 0 1 2 3 4	Link-Control Function
0 0 0 0 0	Test-initialization-result (TIR)
0 0 0 0 1	Test-initialization (TIN)
0 0 0 1 0	Link-level-reject (LRJ)
0 0 0 1 1	Port-reject (PRJ)
0 0 1 0 0	Link-level-busy (LBY)
0 0 1 0 1	Port-busy (PBY)
0 0 1 1 0	reserved
0 0 1 1 1	reserved
0 1 0 0 0	Establish-logical-path (ELP)
0 1 0 0 1	Remove-logical-path (RLP)
0 1 0 1 0	Logical-path-established (LPE)
0 1 0 1 1	Logical-path-removed (LPR)
0 1 1 0 0	Link-level-acknowledgment (ACK)
0 1 1 0 1	State-change-notification (SCN)
0 1 1 1 0	Acquire-link-address (ALA)
0 1 1 1 1	reserved
1 0 0 0 0	Identifier-response (IDR)
1 0 0 0 1	Request-node-identifier (RID)
1 0 0 1 0	Link-incident-notification (LIN)
1 0 0 1 1	Request-incident-record (RIR)
1 0 1 0 0	Link-incident-data (LID)
1 0 1 0 1 through 1 1 1 1 1	reserved

9.1 Link-control frames

When the frame-type bit, bit 7 in the link-control field of a frame, is set to one, a link-control frame is specified.

A summary of link-control frames, which shows the request frames, the response frames, the addressing for each frame type, and the contents of the information field is shown in table 19.

Link-control frames occur only in request-response pairs.

For brevity, a link-control frame shall be referred to by the link-control function indicated in it. For example, a link-control frame that indicates the ELP link-control function shall be referred to as an ELP frame.

The link-control function, specified by bits 0-4 of the link-control field, determines whether the associated link-control frame shall be sent as a request or a response. A link-control request frame has a connect-SOF delimiter and a passive-EOF delimiter. When a link-control request frame is sent, no other link-control request frame shall be sent until either an allowed response is received or the link timeout expires and connection recovery has been performed. A link-control request frame shall not be sent when a connection exists. A link-control response frame shall have a passive-SOF delimiter and a disconnect-EOF delimiter.

When the recipient of a frame is identified and the frame is received, the recipient shall compare the destination link address to its assigned link address, provided no link errors are detected that would affect this comparison. If they

Table 19 – Summary of link-control frames

Function	Sent By	Accepted By	Function-Specific Expected Response	Used In Response To	Source Address		Destination Address		Information Field		Subclause
					Link/	Logical	Link/	Logical	Length	Content	
Link-Control Request Frames (sent with connect-SOF and passive-EOF delimiters)											
TIN	CH,CU	CH,CU	TIR	N/A	S	SL	D	0's	0	N/A	9.1.6
ELP	CH	CU	LPE, LPR	N/A	S	SL	D	DL	8	Pacing parameters	9.1.1
RLP	CH	CU	LPR	N/A	S	SL	D	DL	0	N/A	9.1.2
SCN	HO, UO[1]	CH,CU	ACK	N/A	S	0's	D/1's	0's	2	Affected link address	9.1.13
ALA	CH,CU[2]	CH,SCU	ACK, LRJ	N/A	0's	0's	1's	0's	0	N/A	9.1.14
RID	CH,CU	CH,CU	IDR	N/A	S	0's	D/1's	0's	8	(See description)	9.1.15
LIN	CU	CH	ACK	N/A	S	0's	D	0's	0	N/A	9.1.17
RIR	CH	CU	LID	N/A	S	0's	D	0's	8	(See description)	9.1.18
Link-Control Response Frames (sent with passive-SOF and disconnect-EOF delimiters)											
TIR	CH,CU	CH,CU	N/A	TIN	S	0's	D	DL	3	Initialization status	9.1.7
LRJ	CH,CU	CH,CU	N/A	All Initiation frames[3]	S/0's	0's	1's	0's	1,2	Reject-reason code	9.1.8
PRJ	P	CH,CU	N/A	All Initiation frames[3]	SWCU	0's	1's	0's	1,2	Reject-reason code	9.1.9
LBY	CH,CU	CH,CU	N/A	All Initiation frames[3]	S/0's	0's	1's	0's	0	N/A	9.1.10
PBY	P	CH,CU	N/A	All Initiation frames[3]	SWCU	0's	1's	0's	1	Port-busy-reason code	9.1.11
LPE	CU	CH	N/A	ELP	S	SL	D	DL	0	N/A	9.1.3
LPR	CU	CH	N/A	ELP, RLP	S	SL	D	DL	1	LPR reason code	9.1.4
ACK[4]	CH,CU	CH,CU	N/A	ALA, SCN, LIN	S	0's	D	0's	0	N/A	9.1.5
IDR	CH,CU	CH,CU	N/A	RID	S	0's	D	0's	64	(See description)	9.1.16
LID	CU	CH	N/A	RIR	S	0's	D	0's	104	(See description)	9.1.19

Table 19 *(concluded)*

Explanation:

1	Required for dynamic-switch control unit, optional for other control units
2	Sent by all control units except dynamic-switch control unit
3	Initiation frames also include device frames that initiate a connection
4	Full abbreviation is link-level-ACK
0's	Set to all zeros
1's	Set to all ones
CH	Channel
CU	Control unit
D	Assigned link address of the intended recipient
D/1's	Assigned link address of the intended recipient or all ones
DL	Logical address of intended recipient image
HO	Optional for a channel
P	Dynamic-switch port
S	Assigned link address of sender
S/0's	Assigned link address of sender or all zeros
SL	Logical address of sending image
SCU	Dynamic-switch control unit
SWCU	Link address of dynamic-switch control unit; not all ones or zeros
UO	Optional for a control unit

match, or in some cases, if the destination link address is all ones, the frame shall be accepted, provided that no other conditions exist which prevent acceptance. If the addresses do not match and the destination link address is not all ones, the frame shall be discarded, and a destination-address-invalid error shall be detected. If the destination link address is all ones but a destination link address of all ones is not allowed for the frame that was received, a link-protocol error shall be detected.

If a frame is received with a source-logical-address field other than zero in a frame that does not permit a source logical address, the frame is discarded, and a link-protocol error shall be detected. If a frame is received with a destination-logical-address field other than zero in a frame that does not permit a destination logical address, the frame is discarded, and a link-protocol error shall be detected.

If a link-control frame is received with a destination link address not all ones for a link-control function that requires a destination link address of all ones, a link-protocol error shall be detected. If a link-control frame is received with a source link address of other than all zeros for a link-control function that requires a source link address of all zeros, a link-protocol error shall be detected. If a link-control frame is received with a source link address of all ones or a destination link address of all zeros, a link-protocol error shall be detected. If a link-control frame is received with a source link address of all zeros for a link-control function that does not permit a source link address of all zeros, a link-protocol error shall be detected.

If a link-control frame is received that contains an information field with a different length from that described in table 19, a link-protocol error shall be detected. In the case of link-level or port reject, an information field of either one or two bytes shall be accepted.

If a link-control response frame is received and no link-control request frame or device request frame was sent, a link-protocol error shall be detected. If a link-control request frame was sent and the link-control response frame is received but is not an allowed response, a link-protocol error shall be detected.

If a channel, control unit, or dynamic-switch control unit receives a frame specifying a link-control function that is reserved or not recognized, then a link-protocol error shall be detected for an unrecognized link-control function.

When the channel, control unit, or dynamic-switch control unit recognizes the specified link-control function, that function is accepted, and the normal response is sent, or, if the function specified cannot be performed because of a link-error or link-busy condition, the appropriate alternate response, if any, shall be sent. If a link timeout occurs and more than one link error has been detected, a link-error condition shall be recognized for the detected link error with the highest priority (see 11.1).

A frame that is defined to make a connection is called an *initiation frame*.

Two link-level facilities may simultaneously send initiation frames to each other. However, without the requirement that one link-level facility accede, possible endless repetition could occur if each link-level facility is in a state such that it cannot accept the received initiation frame and initiate the requested function or operation because it is waiting for a response to its sent initiation frame. Similarly, without the requirement that one link-level facility accedes, endless repetition could occur if a channel or control unit receives an initiation frame while it is preparing to send an initiation frame. The link-level facility of one accedes to the other by accepting and reacting to the received initiation frame as if it had not been waiting for a response to its sent initiation frame or had not been preparing to send an initiation frame.

The link-level facility that is not required to accede shall respond to an RID or an RIR initiation frame which it received in the following manner. It shall send an allowable alternate-response frame that contains a disconnect-EOF delimiter and that does not require any further SBCON I/O interface action on the part of the other link-level facility.

In cases other than those previously described, the link-level facility that is not required to accede shall respond to the initiation frame which it received in the following manner. It shall send an allowable normal-response or alternate-response frame that contains a disconnect-EOF delimiter and that does not require any further SBCON I/O interface action on the part of the other link-level facility.

The link-level facility that is required to accede may at first refuse the received frame, but, after several occurrences of the same frame sequences, the possibility of endless repetition is recognized; which link-level facility accedes shall be determined by the following rules for conflicting request frames:

NOTE – The preferred implementation for the link-level facility that is required to accede is to not refuse the received frame.

a) If simultaneously a channel and a control unit send device initiation frames, the control unit shall accede;

b) If simultaneously one link-level facility sends a link-control frame to a second link-level facility and the second sends a device frame to the first, the sender of the device frame shall accede;

c) If simultaneously one identified link-level facility sends a link-control frame to a second identified link-level facility, and the second also sends a link-control frame to the first, the link-level facility with the lower link address shall accede to the link-level facility with the higher link address (see 13.3.1.2 and 14.1.10). If either link-level facility is unidentified, special protocols shall be used (see 9.1.14, 10.1.2.1, and 10.1.3.1).

If a channel or control unit receives a frame that is not a reject or a busy frame and that does not match the expected response according to the rules for conflicting request frames defined above, a connection error shall be detected (see 11.1.1, 11.1.7 and 13.3.1).

NOTE – For example, some channels and control units may be capable of accepting an IDR frame or an LID frame as a normal response to their sent RID or RIR initiation frame when their link-level facility accedes. These channels and control units may not detect an error.

A channel or control unit can attempt to send an initiation frame to another channel or control unit and simultaneously receive a frame from yet another channel or control unit. This can occur only when a dynamic switch is present between the respective channels and control units. When this situation is encountered, the following shall apply:

a) If the initiation frame sent had an address other than the all-ones destination address, the frame received in response to the frame sent shall be a port-busy frame indicating source port busy monolog-D; otherwise, a connection error shall be detected;

b) The initiation frame received may be either honored or refused depending on its type and the conditions present at the channel or control unit. The appropriate response for the action taken on the initiation frame shall be sent. If the action taken is to refuse the initiation frame received, the channel or control unit may resend its initiation frame, but (to avoid the possibility of endless repetition) after a specified model-dependent number of occurrences of source port busy monolog-D, the initiation frame received shall be honored.

Some link-control functions shall be performed in a particular sequence; that is, certain functions can only be performed if other functions have already been completed. For example, link-address acquisition shall be performed before establishing logical paths. For those link-control functions which are performed in a particular sequence, a link-protocol error shall be detected if the sequence is not followed.

Each of the link-control requests and responses shall be described in the following subclauses.

Link-control-request frames shall not be sent during a connection. If initiative is generated to send a link-control-request frame during a connection, the connection shall first be removed by performing the connection-recovery procedure. If a link-control-request frame is received when a connection exists, a connection error shall be detected.

9.1.1 Establish-logical-path

The establish-logical-path (ELP) function shall be sent from a channel to a control unit to request the establishment of a logical path and supply the associated pacing parameters. A logical path, when established, shall identify a physical path and specify the pacing parameters to be used between a channel image and a control-unit image for the sending of device frames. The logical path being established is specified by the combination of the destination link address, the destination logical address, the source link address and the source logical address. The destination and source link addresses shall identify the physical path and the destination and source logical addresses shall identify the channel and control-unit image for which the logical path is being established. The pacing parameters shall be contained in the information field. Pacing parameters shall specify the number of requests beyond one that can be outstanding without a response for certain device-level operations, and the device-level-reception capability of the channel, called the channel's *pacing count*. For each device-information-block (DIB) size permitted by the channel, the channel's pacing count is specified as the minimum number of pairs of idle characters to be inserted between device frames of a particular DIB size for those certain device frames which require pacing control. The DIB size shall represent the length of data in a device-level data frame (see 12.2 and 13.1.2.4). Seven different DIB sizes shall be allowed: 16, 32, 64, 128, 256, 512, and 1,024 bytes.

The information field of the ELP frame shall contain eight bytes of pacing parameters. Byte 0 of the information field shall specify the number of requests beyond one that can be outstanding for certain device-level operations. When any of bytes 1-7 is zero, the DIB size corresponding to that byte shall not be permitted by the channel. When a DIB size is permitted, the corresponding byte of bytes 1-7 shall contain a value that indicates the minimum number of pairs of idle characters to be inserted between certain device frames for that DIB size (x’02’ to x’FF’). The format of the information field of the establish-logical-path frame is shown in figure 22.

If the control unit receives an ELP frame with either the x’01’ count for a DIB size or the x’00’ count for all of the DIB sizes, the control unit shall detect a link-protocol error, and a link-level reject shall be returned. If there is an existing logical path associated with the channel image and control-unit image specified by the erroneous ELP frame, that existing logical path shall not be removed, and the control unit shall have initiative to test whether that logical path exists. The existence of a logical path shall be tested by sending a TIN frame to the source link address in the ELP frame.

A channel shall permit all DIB sizes that are smaller than the indicated maximum DIB size which that channel permits. If the count of pairs of idle characters is shown as zero for any DIB size smaller than the maximum DIB size permitted, or if the count of pairs of idle characters is shown as zero for all DIB sizes, then the control unit shall detect a link-protocol error, and the logical path shall not be established. In order to establish the logical path, the control unit shall consider at least one of the DIB sizes to be permitted by both the control unit and the channel. In the event that a control unit does not accept any of the DIB sizes which are indicated as permitted by the channel in an ELP frame, the con-

Information-Field Bytes 0-7

NDR-W	IDLE16	IDLE32	IDLE64	IDLE128	IDLE256	IDLE512	IDLE1,024

Explanation:

NDR-W	Number of requests allowed, beyond one
IDLE16	Number of pairs of idle characters for 16-byte DIB size
IDLE32	Number of pairs of idle characters for 32-byte DIB size or zero, if 32-byte DIB size is not permitted
IDLE64	Number of pairs of idle characters for 64-byte DIB size or zero, if 64-byte DIB size is not permitted
IDLE128	Number of pairs of idle characters for 128-byte DIB size or zero, if 128-byte DIB size is not permitted
IDLE256	Number of pairs of idle characters for 256-byte DIB size or zero, if 256-byte DIB size is not permitted
IDLE512	Number of pairs of idle characters for 512-byte DIB size or zero, if 512-byte DIB size is not permitted
IDLE1,024	Number of pairs of idle characters for 1,024-byte DIB size or zero, if 1,024-byte DIB size is not permitted

Figure 22 – Information field of the ELP frame

trol unit shall not establish a logical path and shall send an LPR response containing the appropriate LPR reason code. This action shall be taken even if the requested logical path already exists based on previously acceptable pacing parameters; in this case, the existing logical path shall be removed, and an LPR response with the appropriate LPR reason code shall be sent.

The channel shall provide the same pacing parameters in all ELP frames sent on a particular channel path. If a control-unit image receives an ELP, the requested logical path shall become established or reestablished provided that the parameters for the requested logical path do not affect the parameters associated with other previously established logical paths. If the request to establish a logical path would affect parameters associated with other previously established logical paths, the requested logical path shall not become established, and an LPR shall be sent in response to the ELP with an appropriate LPR reason code (see 9.1.4).

The source-link-address field of the ELP frame shall identify the channel providing the pacing parameters. The pacing parameters shall be saved and associated with the logical paths established for that channel in such a way that all subsequent device frames for which those pacing parameters apply shall be transferred on that logical path using those parameters (see 12.2).

A channel shall attempt to establish logical paths to the control-unit images that are described in its configuration definition. This shall be done, for example, when a channel image is initialized, when configuration changes are made, or when the channel receives an indication that the logical path no longer exists (see 10.1.2.3). Each logical path shall be established with a separate exchange of frames, an ELP request and an LPE response. Failure to establish a logical path shall not affect other existing logical paths or the ability to establish other logical paths.

If an ELP frame is received with a request for a logical path that is already established, the frame shall be accepted, provided that no link errors are detected, and a system-reset (see 13.2.3) shall be performed with respect to that logical path. An LPE response shall be sent if the requested logical path remains established. If the existing logical path is removed as a result of the ELP, such as when the control-unit image does not accept any of the DIB sizes or when the parameters specified would affect other previously established logical paths over the same interface, an LPR response containing the appropriate LPR reason code shall be sent. The LPE or LPR response shall not indicate initiation or any degree of progress made for associated system-reset (see 13.2.3). The normal response to an ELP frame shall be an LPE frame. The channel shall not consider the logical path to be established until it receives an error-free LPE frame. The control unit shall not consider the logical path to be established until the necessary action is taken at the control unit to establish the logical path, and the sending of an error-free LPE frame is completed.

If a link error is detected when an ELP frame is received, the appropriate response, if any, shall be made, and the logical path shall not be established. If the ELP frame cannot be accepted for reasons other than a link-error or a link-busy condition, an LPR frame containing the appropriate LPR reason code shall be sent in response. If an LPR frame is received in response to an ELP frame, the logical path shall be considered not established. If an LPR is sent in response to an ELP received for a logical path that is already considered established, the existing logical path shall be removed, and the equivalent of a system-reset shall be performed for the affected logical path. If an LPR, link-level-busy, port-busy, link-level-reject, or port-reject frame is received, the ELP frame may be retried until the logical path is established. The number of retries shall be model-dependent. In the case of LPR, whether or not the ELP is retried shall depend on the reason code (see 9.1.4).

In the event that there is an error in the response to a channel request to establish a logical path, the channel shall not assume that the requested action has or has not taken place. If no valid response is received by the channel to the ELP request, the channel may retry the request. The number of retries shall be model-dependent.

9.1.2 Remove-logical-path

The remove-logical-path (RLP) function shall request the control unit to remove a logical path. The logical path to be removed is specified by the combination of the destination link address, the destination logical address, the source link address and the source logical address of the RLP frame. If the last remaining logical path to the channel for a specific interface on the control unit is to be removed, the pacing parameters for that channel shall also be removed. As long as one logical path remains to the channel over the interface on which the RLP frame is received, the pacing parameters for that channel shall be maintained. An RLP frame shall be sent when a change in the channel-path configuration requires a control unit to be either physically or logically removed.

The removal of a logical path shall cause the control-unit image and its associated I/O devices to be logically removed from the channel path. When an RLP frame is received and accepted, the logical path shall be removed, and the equivalent of a system-reset (see 13.2.3) shall be performed only for the affected logical path; that is, only the control-unit image associated with the logical path shall be affected, and only those operations within the control-unit image for this logical path shall be reset. The LPR frame shall be the normal response. Other logical paths associated with the same channel or different channels shall not be affected. After a logical path is removed, device frames shall not be sent or received using that logical path.

An RLP frame shall be received over the same physical path over which the logical path was established. If an RLP frame is received for a logical path that does not exist, the RLP shall be accepted, provided that no link errors are detected, and the LPR response shall be sent.

If a link error is detected when an RLP frame is received, the frame shall be discarded, the specified logical path shall not be removed, and the appropriate response, if any, for the link error recognized shall be sent. The channel shall not consider the logical path removed until it receives the LPR frame and no errors are detected. The control unit shall not consider the logical path removed until the LPR frame is completely sent without error.

A channel that receives a link-level-busy, port-busy, link-level-reject, or port-reject frame in response to an RLP may retry sending the RLP frame. The number of retries shall be model-dependent.

Unless the channel receives a valid response to a request to remove a logical path, the channel shall not assume that the requested action has or has not taken place. If an invalid response is received by the channel to the remove-logical-path request, the channel shall retry the request. The number of retries shall be model-dependent.

9.1.3 Logical-path-established

The logical-path-established (LPE) function shall confirm the successful completion of an ELP and the establishment of the logical path specified. The combination of the destination link address, the destination logical address, the source link address, and the source logical address in the LPE frame shall identify the logical path that was established.

When an ELP frame is accepted, the LPE frame shall be the normal response. A logical path shall not be considered to be established by the recipient of an ELP frame until the recipient has sent the LPE frame. A logical path shall not be considered to be established by the sender of an ELP frame until the LPE frame is received.

9.1.4 Logical-path-removed

The logical-path-removed (LPR) function shall confirm the successful completion of an RLP and the removal of the logical path specified. The combination of the destination link address, the destination logical address, the source link address, and the source logical address in the LPR frame shall identify the logical path that was removed. An LPR frame shall be the normal response to an RLP request. A logical path shall be considered not removed by the recipient of an RLP frame until it has sent the LPR frame in response and shall be considered not removed by the sender of the RLP frame until the LPR frame is received.

An LPR frame may also be sent in response to an ELP frame when no link-error or link-busy conditions are detected but the requested logical path is not established.

Bits 0-3 of the information field of the LPR frame shall be reserved; these bits shall be set to zeros by the sender of the LPR frame and shall be ignored by the recipient of the frame.

Bits 4-7 of the information field shall contain a reason code which indicates why the logical path was removed. The values (in binary) for the following reason codes shall have these meanings:

Value	*Meaning*
0000	This LPR frame is a response to RLP.
0001	The pacing requirements of the control unit are not met by the pacing parameters presented by the channel in the ELP frame, or the pacing parameters received would affect those previously accepted for already established logical paths with the same channel. The ELP shall not be retried for this condition.
0010	The control-unit image has no resources available for establishing new logical paths on this physical path. The channel may retry the ELP a model-dependent number of times.
0011	Device-level initialization is not complete; the control unit is not ready to perform device-level operations. The channel may retry the ELP a model-dependent number of times.
0100	A control-unit image corresponding to the destination logical address in the ELP does not exist. The channel shall not retry the ELP for this condition.
0101	The control-unit image has a predefined configuration which precludes establishing this new logical path on this physical path. The channel shall not retry the ELP for this condition.
0110-1111	Reserved.

A link-protocol error shall be detected by a link-level facility which receives an LPR frame containing a reserved reason-code value.

9.1.5 Link-level-acknowledgment

The link-level-acknowledgment (link-level-ACK) link-control function shall acknowledge receipt of certain link-control frames when they are received without a link error.

The link-level-ACK frame shall be sent only as a response to a link-control function that is sent as a request. In some cases the link-level-ACK frame shall indicate that the link-control frame for which it is a response was received without any errors and the function was completed, while in other cases the link-level-ACK frame shall only indicate that the link-control function for which it is a response was received without any link errors. (See the individual descriptions of the link-control requests for those that require a link-level-ACK frame and for the meaning of the link-level-ACK frame in each particular circumstance.)

Table 19 shows the link-control frames for which a link-level-ACK frame is normally the valid response. When a link-control frame requiring the link-level-ACK response is received and no link errors are detected, and when the necessary conditions for sending a link-level-ACK response are present, the link-level-ACK frame shall be sent. When the sender of a link-control frame requiring a link-level-ACK frame in response receives the link-level-ACK frame and no link errors are detected, the link timer shall be reset, and the link protocols for the frame requiring the link-level-ACK response shall be considered to be completed. The function shall either be completed or initiative to complete the function shall be accepted.

9.1.6 Test-initialization

The test-initialization (TIN) function shall provide a method for determining whether any logical paths are considered established for the source specified by the combination of the source link address and source logical address at the destination specified only by the destination link address. Upon accepting a TIN frame, the receiving link-level facility shall check whether it has logical paths with the source of the TIN frame. The normal response shall be a TIR frame.

9.1.7 Test-initialization-result

The test-initialization-result (TIR) function shall confirm the successful completion of the TIN function and indicate whether the logical paths are considered established for the source specified by the source link address.

The TIR frame shall be sent only as a response to a TIN frame that is accepted.

The destination logical address of the TIR shall match the source logical address of the TIN that initiated the test.

The format of the three-byte information field for the TIR frame is shown in figure 23**.**

The logical-path field, bits 8-23, of the information field, shall indicate whether the specified logical paths are considered established. There is a bit in the logical-path field for each possible logical path that may be established with the source of the TIN frame. Each bit corresponds to a logical address. Starting with bit 8, logical addresses 0 through 15 are assigned in ascending order. The bit for a logical address corresponding to a logical path shall be set to one if that logical path is established and shall be set to zero otherwise.

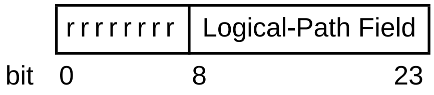


Figure 23 – TIR frame information field

Bits 0-7 of the information field shall be reserved; they shall be set to zeros by the source of a TIR frame and shall be ignored by the recipient of a TIR frame.

If the TIR is recognized by a channel or control unit, and one or more logical paths are indicated as not established when they were previously considered to be established, a test-initialization-result error shall be recognized. If one or more logical paths are indicated as being established when they were previously considered not established, a test-initialization-result error shall be recognized (see 14.1.9). If a test-initialization-result error is not recognized, no action shall be taken.

9.1.8 Link-level-reject

The link-level-reject (LRJ) function shall indicate that the frame received by a link-level facility was not accepted because of a link-error condition and that the frame received was discarded.

A link-level facility shall respond with a link-level-reject frame to certain errors that are detected without CRC checking, and all errors detected after CRC checking, provided the following conditions are satisfied:

a) Frame reception shall be started with the recognition of a connect-SOF delimiter followed by a valid transmission character;

b) Frame reception shall be ended with one of the following:

- recognition of a passive-EOF delimiter;

– recognition of two consecutive idle characters;

– detection of a maximum-frame-size error;

– detection of a link-signal error.

c) No other condition exists or has been recognized that requires the sending of either a link-level-busy frame or a sequence.

When a link-level-reject frame is received in response to a frame that initiates a connection, a link error shall be detected by the recipient of the link-level-reject frame. If a link-control request frame or device request frame has been sent, the type of link error shall be indicated by the reject-reason code. If the link-level-reject frame is received when no frame was sent, a link-error condition shall be recognized for a connection error or connection-protocol error (see 11.1.1 and 11.1.7). If a frame was sent that did not have a connect-SOF delimiter, a connection-protocol error shall be detected.

Each link-level-reject frame shall have a one- or two-byte information field. The first byte of the information field shall contain the reject-reason code. If the link-level-reject frame has a two-byte information field, the second byte of the information field shall be reserved and set to zeros.

The format of the one-byte information field of a link-level-reject frame is shown in figure 24**.**

	r	Reject-Reason Code	
bit	0	1	7

Figure 24 – LRJ frame information field

The reject-reason code shall appear in bits 1-7 of the information field of a link-level-reject frame. This field shall identify the link error or condition that caused the frame to be rejected. Table 20 is a summary of the reject-reason codes used for a link-level-reject frame.

See 11.1 for a description of the link errors that correspond to the reject-reason codes in table 20.

Bit 0 of the information field shall be reserved; the bit shall be set to zero by the link-level facility sending a link-level-reject frame and shall be ignored by a link-level facility receiving a link-level-reject frame.

Table 20 – LRJ reject-reason codes

Reject-Reason Code Bits 1 2 3 4 5 6 7	Reason Codes	Notes
0 0 0 0 0 0 0	reserved	
0 0 0 0 0 0 1	Transmission Error	1
0 0 0 0 0 1 0	reserved	
0 0 0 0 0 1 1	reserved	
0 0 0 0 1 0 0	reserved	
0 0 0 0 1 0 1	Destination-Address-Invalid Error	
0 0 0 0 1 1 0	Logical-Path-Not-Established Error	2
0 0 0 0 1 1 1	Reserved-Field Error	1
0 0 0 1 0 0 0	Unrecognized Link-Control Function	1,3
0 0 0 1 0 0 1	Protocol Error	1
0 0 0 1 0 1 0	Acquire-Link-Address Error	1
0 0 0 1 0 1 1	Unrecognized Device Level	1
0 0 0 1 1 0 0 through 1 1 1 1 1 1 1	reserved	

Notes

1 Special handling. See text.

2 Response to a device frame.

3 Response to a link-control frame only.

Channels and control units shall be capable of receiving a link-level-reject frame having a two-byte information field; the second byte shall be ignored. A link-protocol error shall be detected if the length of the information field of a link-level-reject frame is zero bytes or is more than two bytes.

A link-level-reject frame shall be sent with a destination link address of all ones. When an identified link-level facility sends a link-level-reject frame, the source link address shall be the assigned link address of the link-level facility. When an unidentified link-level facility sends a link-level-reject frame, the source link address of the link-level-reject frame shall be set to zeros.

Depending on the function being attempted at the time the link-level-reject frame is received and on the state of the receiving link-level facility, a retry of the function may be attempted (see clause 14).

A link-protocol error shall be detected by the link-level facility receiving a link-level-reject frame if the frame has any of the following:

– a reserved reject-reason-code value;

– a reject-reason-code value not appropriate for the conditions present.

The transmission-error reject-reason code shall be used to indicate that one of the following link errors was detected:

– link-signal error;

– code-violation error;

– sequencing error;

– CRC error.

Several specific reject-reason codes shall be used for certain errors defined within link-protocol error. These are:

– reserved-field error;

– unrecognized link-control function;

– acquire link-address error;

– unrecognized device level.

If there is no specific reject-reason code for a particular link-protocol error, or if a device-level protocol error results in a link-level reject frame, a reject-reason code of protocol error shall be used (see 14.1.10).

9.1.9 Port-reject

The port-reject (PRJ) function shall indicate that because of a link error or configuration restriction the dynamic-switch port did not accept a frame that was attempting to make a dynamic connection, and the frame received was discarded. A port-reject function shall be used to indicate that a dynamic connection for the frame received could not be made because of a reason other than a link-busy condition.

A dynamic-switch port shall respond with a port-reject frame to link errors that occur without CRC checking when all of the following conditions are satisfied:

a) The port is in the inactive state;

b) A connect-SOF delimiter followed by a valid transmission character is recognized;

c) No other condition exists or has been recognized that requires the sending of either a port-busy frame or a sequence;

d) The link error or configuration restriction detected prevents the dynamic switch from completing the specified dynamic connection.

When a port-reject frame is received in response to a frame that initiates a connection, a link error shall be detected by the recipient of the port-reject frame. The type of link error shall be indicated by the reject-reason code. Whether a link-error condition is recognized shall depend on the conditions present. A link-error condition shall be recognized if the port-reject frame is in response to a frame sent with a connect-SOF delimiter. If the port-reject frame is received when no frame was sent, a link-error condition shall be recognized, and the recipient of the port-reject frame shall perform connection recovery. If a frame was sent that did not have a connect-SOF delimiter, a link-protocol error shall be detected.

Each port-reject frame shall have a one- or two-byte information field. The first byte of the information field shall contain the reject-reason code. If the frame has a two-byte information field, the second byte of the information field shall be reserved and set to zeros.

The format of the one-byte information field of a port-reject frame is shown in figure 25**.**

	0	Reject Reason Code	
bit	0	1	7

Figure 25 – PRJ frame information field

The reject-reason code shall appear in bits 1-7 of the information field of a port-reject frame. This field shall identify the link error or condition that caused the frame to be rejected.

Bit 0 of the information field shall be set to zero by the dynamic-switch port and shall be ignored by a link-level facility receiving a port-reject frame.

Channels and control units shall be capable of receiving a port-reject frame having a two-byte information field; the second byte shall be ignored. A link-protocol error shall be detected if the length of the information field of a port-reject frame is zero bytes or is more than two bytes.

The reject-reason code in the port-reject frame further shall describe the cause of the dynamic-switch port reject. The values in the reject-reason-code field are defined in table 20 (see 11.2 for a description of the link errors that correspond to the reject-reason codes in table 20).

Table 21 – PRJ reject-reason codes

Reject-Reason Code Bits	
1 2 3 4 5 6 7	**Reason Codes**
0 0 0 0 0 0 0 through 0 0 0 1 1 1 1	reserved
0 0 1 0 0 0 0	Address-Invalid Error
0 0 1 0 0 0 1	Undefined-Destination-Address Error
0 0 1 0 0 1 0	Destination-Port Malfunction
0 0 1 0 0 1 1	Dynamic-Switch-Port Intervention Required
0 0 1 0 1 0 0 through 1 1 1 1 1 1 1	reserved

A link-protocol error shall be detected by the receiving link-level facility when a port-reject frame has a reserved reject-reason-code value.

The priority for port-busy and port-reject frames generated by dynamic-switch ports is described in 9.1.11.

9.1.10 Link-level-busy

The link-level-busy (LBY) function shall indicate that a frame was received and discarded by a channel or control unit because of a link-busy condition (see 10.11).

A link-level facility shall respond with a link-level-busy frame when a link-busy condition exists after frame reception is started with the recognition of a connect-SOF delimiter followed by a valid transmission character and then only if all of the following conditions are satisfied:

a) Frame reception is ended by one of the following:

- recognition of a passive-EOF delimiter;
- recognition of two consecutive idle characters;
- detection of a maximum-frame-size error;
- detection of a link-signal error;

b) No other condition exists or has been recognized that requires the sending of either a link-level-reject frame or a sequence.

(See 10.11.)

When a link-level-busy frame is received in response to a frame while a connection is pending, a temporary busy condition shall be recognized, and the link-level or device-level function may be immediately retried or retried at a later time. When a link-level-busy frame is received in response to a frame while a connection exists, a link-protocol error shall be detected.

When an identified link-level facility sends a link-level-busy frame, the source link address of the frame shall be the assigned link address. When an unidentified link-level facility sends a link-level-busy frame, the source link address of the frame shall be all zeros. The destination link address of the link-level-busy frame shall be all ones.

9.1.11 Port-busy

The port-busy (PBY) function shall indicate that a frame was discarded by a dynamic-switch port because of a link-busy condition (see 10.11).

A dynamic-switch port shall respond with a port-busy frame when a link-busy condition exists at the dynamic-switch port after recognition of a connect-SOF delimiter followed by a valid transmission character and then only if both of the following conditions are satisfied:

a) The switch port is in the inactive state, port-link-busy state, or monolog-D state;

b) No other condition exists or has been recognized that requires the sending of either a port-reject frame or a sequence.

(See 10.11.)

When a port-busy frame is received in response to a frame while a connection is pending, a temporary busy condition shall be recognized, and the link-level or device-level function may be immediately retried or retried at a later time. When a port-busy frame is received in response to a frame while a connection exists, a link-protocol error shall be detected.

The condition at a dynamic-switch port that caused the port-busy condition shall be described in byte 0, the port-busy reason-code field of the information field, which is shown in figure 26.

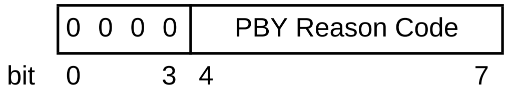


Figure 26 – PBY reason code field

Bits 0-3 of the port-busy reason-code field shall be set to zeros by the dynamic-switch port and shall be ignored by the recipient of a port-busy frame.

The port-busy reason codes, bits 4-7 of the port-busy reason-code field, shall identify the conditions which prevented a dynamic-switch port from making the requested dynamic connection. Table 20 lists the port-busy reason codes.

A link-protocol error shall be detected by the link-level facility receiving a port-busy frame if the frame has a reserved PBY reason-code value.

9.1.11.1 Source-port-busy monolog-D

A source-port-busy condition shall be recognized when a dynamic-switch port receives a frame with the SOF delimiter that requires a dynamic connection to be made and the receiving dynamic-switch port is in the monolog-D state and is connected to a dynamic-switch port other than the dynamic-switch port for which a connection is being requested (see 10.9).

Table 22 – PBY reason codes

PBY Reason Code Bits	
4 5 6 7	Reason Codes
0 0 0 0	reserved
0 0 0 1	Source port busy monolog-D
0 0 1 0	Source port busy unconnected
0 0 1 1	Destination port busy
0 1 0 0 through 1 1 1 1	reserved

The request for a connection shall be suppressed. The frame requesting the dynamic connection shall be discarded, and a port-busy frame shall be sent to the source of the frame requesting the dynamic connection. The port-busy frame shall be received by the channel or control unit after the frame which establishes the monolog-D state.

When a source-port-busy monolog-D condition is recognized, the dynamic-switch port-busy response frame shall be sent for that condition, even if other port-reject or port-busy conditions are recognized at the same time.

9.1.11.2 Source-port-busy-unconnected

A source-port-busy-unconnected condition shall be recognized when a dynamic-switch port receives a frame with the SOF delimiter that requires a dynamic connection to be made and the receiving dynamic-switch port is in the busy state (see 10.9). The request for a connection shall be suppressed. The frame requesting the dynamic connection shall be discarded, and a port-busy frame shall be sent to the source of the frame requesting the dynamic connection.

9.1.11.3 Destination-port-busy

A destination-port-busy condition shall be recognized when a source port in the inactive state requests a dynamic connection to a destination port which either:

– already has a dynamic connection with another dynamic-switch port;

– is in the port-link-busy state;

- is in the connection-recovery state.

(See 10.9.)

The request for a connection shall be suppressed. The frame requesting the dynamic connection shall be discarded, and a port-busy frame shall be sent to the source of the frame requesting the dynamic connection.

9.1.12 Multiple port-reject and port-busy conditions

When multiple port-reject and port-busy conditions exist at the same time and source port busy monolog-D is one of the conditions, the source-port-busy monolog-D condition shall be indicated; if source port busy monolog-D is not one of the conditions, the condition indicated shall be model-dependent.

9.1.13 State-change-notification

The state-change-notification (SCN) function shall report that a state change has occurred which may have affected logical paths at the specified link-level facility. The affected link-level facility is specified in the information field of the SCN frame. A link-level-ACK frame shall be the normal response to an SCN frame.

The SCN frame shall have a 16-bit information field. Bits 0-7 of the information field shall contain the link address of the link-level facility affected by the state change. Bits 8-15 shall be reserved and set to zeros. The information field of the SCN frame is shown in figure 27.

	Link Address Affected	00000000
bit	0 7	8 15

Figure 27 – SCN frame information field

The link address indicated by the information field shall not be all ones or all zeros nor shall the address match the destination link address of the SCN frame. If the information field is other than two bytes, or if the second byte of the information field is not zero, the frame shall be discarded, and a link-protocol error shall be detected. If an SCN frame is received and the indicated link address is all ones, all zeros, or matches the destination link address of the SCN frame, the frame shall be discarded, and a link-protocol error shall be detected.

The destination link address in the SCN frame may be either all ones or a specific link address, but all ones shall be used when it is intended that a dynamic switch, if present, report the state change to other link-level facilities.

When an SCN frame is received and no link errors are detected, a link-level-ACK frame shall be sent in response.

When an SCN frame is received by a channel, control unit, or dynamic-switch control unit, bits 0-7 of the information field shall be checked for the link address affected. If the link address affected corresponds to a link address to which one or more logical paths have been established, initiative to send a test-initialization frame shall be created. After the link-level-ACK frame is sent in response to the SCN, the test-initialization frame shall be sent to the link-level facility whose link address was provided in the information field of the SCN frame (see 9.1.6). If the link address affected corresponds to a link address to which no logical path has been established, initiative to send a test-initialization frame shall not be created, and no further action shall be needed. However a channel may have initiative to establish logical paths if its model-dependent configuration information indicates that a new control-unit image may have become available.

The following events shall cause the dynamic switch to create initiative to report a state change on behalf of a specific link-level facility:

a) The dynamic-switch control unit has received an SCN frame with a destination address of all ones from that link-level facility;

b) The dynamic-switch control unit has transmitted a link-level-ACK frame to that link-level facility in response to an ALA frame;

c) The dynamic-switch port to which that link-level facility is attached has entered the link-failure, offline, or static state at a time when that link-level facility was identified;

d) The capability of making dynamic connections between two dynamic-switch ports is altered.

For items a, b and c, the dynamic-switch control unit shall report the state change by sending an

SCN frame to each identified link-level facility attached to a port that is capable of having a dynamic connection to the port to which the link-level facility affected by the state change is attached.

For item d, the dynamic-switch control unit shall report the state change by sending an SCN frame to the link-level facilities attached to the two affected ports, provided that both link-level facilities are identified.

A channel or control unit may optionally report a state change that is caused by an event which affects existing logical paths or, for a control unit, an event which affects the ability to accept new logical paths. A control-unit image that shares link-level facilities common to a single link with other control-unit images shall be capable of generating the SCN frame. A channel image that shares link-level facilities common to a single link with other channel images shall be capable of generating the SCN frame.

The channel or control unit shall report the state change by sending an SCN frame with a destination link address of all ones or, optionally, with the destination link address of a specific channel or control unit. However a channel or control unit shall not issue an SCN frame if it is aware that it has already signaled a condition which would cause the dynamic switch, if present, to issue an SCN frame on behalf of the attached link.

If a dynamic-switch control unit receives an SCN frame with a destination link address that matches the assigned link address of the dynamic-switch control unit, the state of a logical path to the dynamic-switch control unit may have changed; this information shall be used directly by the dynamic-switch control unit, which shall not report it to any other link-level facility.

Once the dynamic-switch control unit receives initiative to send an SCN frame to a particular link-level facility concerning a specific affected link-level facility, subsequent state changes at the affected link-level facility or dynamic-switch port shall not create initiative to send another SCN frame to the particular link-level facility if the initiative to send the first SCN frame has not been discharged. When the initiative to send the first SCN frame has been discharged, a subsequent state change shall create a new initiative.

Once a link-level facility in a channel or control unit receives initiative to send an SCN frame, subsequent state changes in that channel or control unit shall not create initiative to send another SCN frame if the initiative to send the first SCN frame has not been discharged. Once the initiative to send the first SCN frame has been discharged, a subsequent state change shall create a new initiative. If the channel or control unit is sending SCN frames with specific destination link addresses, rather than with a destination link address of all ones, this rule shall apply separately to each destination link address.

Initiative to send an SCN frame shall be discharged when one of the following takes place:

– a link-level-ACK frame is received in response to the SCN frame;

– a condition is recognized for which retry of an SCN frame, if previously sent, is performed;

– for a condition that does not require retry to be performed indefinitely, the model-dependent number of retries is exhausted without success while attempting to send an SCN frame.

9.1.14 Acquire-link-address

The acquire-link-address (ALA) function shall provide a means by which an unidentified link-level facility acquires its assigned link address. The ALA frame shall be sent whenever a link-level facility is unidentified and is attempting to initialize (see 10.1).

The action taken by a link-level facility that receives an ALA frame shall depend on whether the recipient is identified or unidentified. If the recipient is identified, the action taken shall depend on whether the recipient is a channel, a dynamic-switch control unit, or a control unit other than a dynamic-switch control unit.

An ALA frame received by either an identified link-level facility of the channel, the link-level facility of a channel in a point-to-point configuration or the link-level facility of the dynamic-switch control unit (a dynamic-switch control unit is always identified) shall cause a link-level-ACK frame to be sent. The destination link address used in the link-level-ACK frame shall be the link address provided to the channel or dynamic switch for the unidentified link-level facility attached to the link over which the ALA frame was received. The source link address in the link-level-ACK frame shall be the link address assigned to the link-level facility of the channel or dynamic-switch control unit.

An ALA frame received by an identified link-level facility of a control unit shall cause a link-level-reject frame that indicates acquire-link-address error to be sent. The link-level-reject frame shall be sent using the destination link address of all ones and the source link address equal to the assigned link address of the sender of the link-level-reject frame.

An ALA frame received by an unidentified link-level facility of a control unit shall cause a link-level-reject frame that indicates an acquire-link-address error to be sent. The link-level-reject frame shall be sent using the destination link address of all ones and the source link address of all zeros.

The unidentified link-level facility that sends an ALA frame and receives a link-level-ACK frame in response shall adopt the link address contained in the destination-link-address field of the link-level-ACK frame as its assigned link address and becomes identified, provided no link errors are detected.

9.1.15 Request-node-identifier

The request-node-identifier (RID) function shall provide a method for acquiring the node identifier of the recipient. Upon accepting an RID frame, the receiving link-level facility shall check whether it has a valid node identifier. The normal response shall be an IDR frame.

The format of the eight-byte information field for the RID frame is shown in figure 28.

	x‘13 00 00 88 00 00 00 3C’	
byte	0	7

Figure 28 – RID frame information field

The format of the information field may or may not be checked by the recipient of the frame. When checking of the information field is performed, the contents of the information field shall be the same as the format defined in figure 28. If the information field is checked and found to be different from the defined format, a link-protocol error shall be detected.

NOTE – It is recommended that the recipient check the format of the information field.

9.1.16 Identifier-response

The identifier-response (IDR) function shall confirm the successful completion of the RID function, provide the unique node identifier of the responding link-level facility, and indicate the validity of the node identifier.

The IDR frame shall be sent only as a response to an RID frame that is accepted. The format of the 64-byte information field for the IDR frame is shown in figure 29.

Byte	
0	1000 0080 0000 0000 0000 0000 0000 0000 (hexadecimal)
16	0000 0000 0000 0000 0000 0000 0000 0000
32 48	Node Descriptor

Figure 29 – IDR frame information field

Portions of the information field may be checked by the recipient of the frame. If the recipient checks the contents of bytes 0-3 of the information field, these bytes shall be the same as the format defined in figure 29. If the information field is checked and found to be different from the defined format, a link-protocol error shall be detected.

NOTE – It is recommended that the recipient check the format of the information field.

Bytes 4-31 of the information field shall be set to zeros by the source of an IDR frame and shall be ignored by the recipient of an IDR frame. The optional checking of the node-descriptor portion of the information field and the resultant actions taken are described in clause 10.

See clause 16 for the format of a node descriptor.

When a link-level facility receives an RID frame, the link-level facility determines the node-identifier validity and sets the appropriate node-ID validity code in the flag field of the node descriptor. If, while obtaining its node identifier, the link-level facility detects an error or determines for some other reason that the node-descriptor information is suspect (that is, not valid), then the link-level facility sets the node-ID validity code to the value 2 in the flag field of the node descriptor being sent in the IDR frame.

In the response to an RID frame, the link-level facility sends a node-ID validity code of either 0 or 2 in the flag field of the node descriptor. Since the link-level facility’s node descriptor must be either valid or not valid, a node-ID validity code of 1 is never sent in an IDR frame.

The recipient of the IDR frame is not required to check the node-ID validity code in the flag field unless the link-level facility requires valid node-identifier information of its neighbor to continue the initialization process (see 10.1).

9.1.17 Link-incident-notification

The link-incident-notification (LIN) function shall provide a method for a control unit to notify a channel that a link-incident record has been generated and is available. The normal response to a LIN frame shall be an ACK frame.

Upon recognizing a LIN frame and sending an ACK frame, the channel shall prepare to request the link-incident record by means of a request-incident-record (RIR) frame.

9.1.18 Request-incident-record

The request-incident-record (RIR) function shall provide a method for a channel to request a link-incident record from a control unit. The normal response to an RIR frame shall be a LID frame.

Upon recognizing a RIR frame, the control unit shall prepare to send a link-incident-data (LID) frame.

The format of the eight-byte information field for the RIR frame is shown in figure 30.

	x‘13 00 00 88 00 00 00 64’	
byte	0	7

Figure 30 – RIR frame information field

The contents of the information field may or may not be checked by the control unit. When the information field is checked, the contents of the information field shall be the same as the format defined in figure 30. If the information field is checked and found to be different from the format defined in figure 30, a link-protocol error shall be detected.

NOTE – It is recommended that the recipient check the format of the information field.

9.1.19 Link-incident-data

The link-incident-data (LID) function shall provide a method for a control unit to pass a link-incident record to a channel. The format of the 104-byte information field for the LID frame is shown in figure 31.

byte	
0	x’10 00 00 80’
4 . . . 100	Link-Incident Record

Figure 31 – LID frame information field

The contents of bytes 0-3 of the information field may or may not be checked by the receiver of the frame. When the information field is checked, the contents of bytes 0-3 of the information field shall be the same as the format defined in figure 31. If the information field is checked and found to be different from the format defined in figure 31, a link-protocol error shall be detected.

NOTE – It is recommended that the recipient check the format of the information field.

The format of the 100-byte link-incident record is described in 10.2.1.

9.2 Functions and protocols for device frames

When the frame-type bit, bit 7 of the link-control field, is zero, a device frame is specified. If the frame type is a device frame, bits 0-4 shall contain a valid device-level identifier, and bits 5-6 shall be set to zeros.

The type of device frame shall determine the format and maximum length of the information field in a device frame. The format and length of the information field of a device frame are defined in clause 12.

Device-frame transmission requires that a logical path shall be established between the source and destination. Link-level protocols shall establish the necessary logical paths at initialization, based on configuration information (see 10.1.2.3 and 10.1.3.3). Removal of or failure to establish a logical path shall result in the associated control unit and its I/O devices being considered not operational on that channel path. A device frame shall not be accepted from a source to which the specified logical path is not considered established. When a frame is received for a logical path that is not established, the frame shall be discarded, and a logical-path-not-established error shall be detected.

Device frames shall be sent only with assigned link addresses. The destination link address of a device frame shall be the link address of the link-level facility of the channel or control unit to which the frame is being sent, and the source link address of a device frame shall be the link address assigned to the link-level facility of the channel or control unit sending the frame. A device frame shall not be sent with a destination link address of all ones. A device frame received with a destination link address of all ones shall cause a link-protocol error to be detected.

All device frames shall use both source and destination logical addresses. If a device frame is received with a destination address that does not match the full link address of the channel or control unit receiving the frame, or with a source address and destination address that do not correspond to the logical path over which the frame was received, a link-protocol error shall be detected.

A device frame with a connect-SOF delimiter shall not be sent during a connection. If a device frame with a connect-SOF delimiter is received during a connection, a connection error shall be recognized.

10 Link-level-facility and dynamic-switch-port states and protocols

This clause describes overall link and link-level protocols and describes states of both the link-level facility and the dynamic-switch port. Whether the function performed is link level or device level, link-level protocols shall be used. Link protocols are satisfied first, in order to verify the integrity of the information that forms the basis of subsequent action. Protocols such as CRC verification shall occur before the contents of a frame, the frame type, and the specified function can be considered as such. For example, a link-control frame with the RID function specified is considered neither a link-control frame nor a RID function until CRC has been checked and found to be correct.

10.1 Initialization

The initialization process shall establish the necessary conditions for elements of a channel path to be able to sustain both link-level and device-level communication. Under normal conditions, the initialization process occurs infrequently; for example, it is usually required only after a unit has been powered on or during a system-initialization procedure and is not part of the execution of an I/O operation.

For each channel link-level facility or control-unit link-level facility that is part of a channel path, the initialization process may be thought of as a series of hierarchical steps. At each step, a procedure shall be performed in the specified sequence until the procedure is completed.

NOTE – At some steps of the initialization process, it is necessary for the link-level facility performing the initialization process to perform the procedure relative to other link-level facilities. When this occurs and the initialization procedure completes for that other link-level facility, then the next step of the initialization process shall be reached relative to that other link-level facility, and the initialization process shall continue relative to that other link-level facility.

For each link that attaches to a dynamic switch there is one dynamic-switch port. For each dynamic-switch port, there is only one step of the initialization process; its associated procedure is the link-initialization procedure.

For the channel link-level facility, the steps of the initialization process and their associated procedures are:

a) Link-initialization procedure;

b) Channel link-address-acquisition initialization procedure;

c) Channel node-identifier-acquisition initialization procedure;

d) Channel logical-path-establishment initialization procedure.

The channel link-level facility shall perform the channel logical-path-establishment initialization procedure for each logical path required on the channel path. The control-unit link-level facility shall establish a logical path at the request of the channel link-level facility. As a procedure for the particular logical path is completed, the channel link-level facility shall complete the initialization process for that logical path.

For the control-unit link-level facility, the steps of the initialization process and their associated procedures are:

a) Link-initialization procedure;

b) Control-unit link-address-acquisition initialization procedure;

c) Control-unit node-identifier-acquisition initialization procedure;

d) Control-unit logical-path-establishment initialization procedure.

The control-unit link-level facility shall perform the control-unit logical-path-establishment initialization procedure for as many different logical paths as the control-unit link-level facility can maintain. The control-unit link-level facility shall establish a logical path at the request of the channel link-level facility. As each procedure is completed, the control-unit link-level facility shall complete the initialization process for that logical path.

If a procedure encounters a link-busy condition, the procedure shall be retried until either the link-busy condition no longer exists and the procedure is successful or a condition other than a link-busy condition is encountered, in which case, the protocols defined for that condition shall determine the action to be taken. The retry for a link-busy condition can be deferred until attempts to complete the procedure with other link-level facilities have been made.

If a procedure is not successful because of a link error (including receipt of a link-reject or port-reject frame), the procedure shall be retried. The number of retries performed beyond one because of a link error or port-reject frame is model dependent. If the procedure is not successful with the last retry, that portion of the initialization process shall be terminated and all or a portion of the channel link-level facility or control-unit link-level facility shall be considered uninitialized, depending on which initialization procedures were not successful. A subsequent operation or function that requires a previously unsuccessful initialization procedure to be completed shall cause the unsuccessful procedure to be retried. If the procedure is successful when retried, the remaining unperformed procedures shall be attempted. If the remaining procedures are successful, the operation or function shall be attempted on the channel path. If the procedure is not successful when retried, the remaining procedures shall not be attempted, and the operation or the function shall not be attempted on the channel path.

The hierarchy of initialization steps shall also be followed when there is an error or other event that causes regression within the initialization process. When an error or other event indicates that the results achieved at a previously completed step are no longer valid, information associated with that step and possibly subsequent steps in the hierarchy shall be discarded or may be considered no longer current for node-identifier information, and the initialization process for the affected link-level facility or dynamic-switch port shall be repeated for all affected steps (see 14.1). A no-longer-current node identifier is an identification of the last known node attached to the link-level facility, but as the result of some event, such as the loss-of-signal condition, may not reflect the current attached node, since a configuration change may have occurred. For example, if a control-unit link-level facility observes that a link which was previously considered operational is now not operational because of a loss-of-signal condition which has persisted for longer than the link-interval duration, then the control-unit link-level facility shall become unidentified, shall remove any logical paths established for that link, and shall either discard the neighboring-node-identifier or consider the neighboring-node-identifier to be no longer current. For this situation, the initialization process would start again with the link-initialization procedure when the loss-of-signal condition no longer exists.

A state-change-notification which contains the link address associated with a channel link-level facility or control-unit link-level facility for which the recipient of the state-change-notification considers the initialization process complete, may cause a test-initialization frame to be sent in order to verify the effect of the state change on previously established logical paths. If the test-initialization-result frame confirms that logical paths previously established by the initialization process are no longer considered established, the procedures shall be attempted, beginning at the appropriate step within the initialization-process hierarchy, to reestablish the logical paths.

A state-change-notification which contains the link address associated with a channel link-level facility or control-unit link-level facility for which the recipient of the state-change-notification considers the initialization process incomplete, may cause a test-initialization frame to be sent in order to verify the effect of the state change on logical paths previously established. If the test-initialization-response frame confirms that logical paths previously established by the initialization process are no longer considered established, the procedures shall be attempted, beginning at the appropriate step within the initialization-process hierarchy, to reestablish logical paths. If the test-initialization-result frame confirms that logical paths previously established are still valid, then the previously incomplete step shall be retried, and, if successful, the next step in the hierarchy shall be attempted for each logical path that is required on the channel path but that is not yet established. If the retry does not result in the successful completion of the initialization process, the recipient of the state-change-notification shall continue to consider the initialization process incomplete for that link-level facility.

When an error or other event causes regression to the node-identifier-acquisition step or to a previous step within the initialization process and the existing node identifier of its neighbor is valid, the link-level facility shall set the flag-field node-ID validity code to not-valid in the node descriptor for the affected neighboring node.

10.1.1 Link-initialization procedure

For a channel link-level facility, control-unit link-level facility, or dynamic-switch port, link initialization shall be started when a link-level facility or dynamic-switch port enters the offline-transmission state (see 10.3.1). Link initialization is complete when the link-level facility or the dynamic-switch port reaches the inactive state after having been in the offline-transmission state (see 10.8 and 10.9).

Once link initialization is complete for a link-level facility, a channel link-level facility or control-unit link-level facility shall consider the link attached to the link-level facility to be operational as long as the link-level facility remains in one of the following states:

a) Inactive;

b) Working;

c) One of the following connection-recovery states:

  1) UD-Transmission;

  2) UD-Reception;

  3) UDR-Reception.

When a link is considered operational, certain link-control frames which do not require that a link address be assigned to the sending channel link-level facility or control-unit link-level facility (that is, link-control frames that may be sent with a source link address of all zeros) can be sent provided that connection recovery is not being performed.

Once link initialization is complete for a dynamic-switch port, it shall consider the attached link to be operational as long as the dynamic-switch port remains in one of the following states:

a) Inactive;

b) Port link busy;

c) One of the following connection states:

  1) Monolog-S;

  2) Monolog-D;

  3) Dialog-1;

  4) Dialog-2.

d) One of the following connection-recovery states:

  1) UD-Transmission;

  2) UD-Reception;

  3) UDR-Reception.

When a link-level facility or dynamic-switch port enters any of the following states, the attached link shall be considered not operational, and the link-initialization procedure must be successfully completed in order to consider it operational:

a) One of the following offline states:

  1) Offline transmission;

  2) Offline reception;

  3) Wait for offline sequence;

b) One of the following link-failure states:

  1) Transmit OLS;

  2) Transmit NOS;

c) One of the following dynamic-switch-port static states:

  1) Static transmit OLS;

  2) Static pass-through.

10.1.2 Initialization for channel link-level facility

For a channel link-level facility, the initialization process consists of the following procedures:

a) Link initialization;

b) Channel link-address acquisition;

c) Channel node-identifier acquisition;

d) Channel logical-path establishment.

The channel link-address-acquisition procedure provides a means by which the channel link-level facility becomes identified. Once the channel link-level facility is identified, any of the link-level frames that are allowed for a channel link-level facility can be sent, unless information from the channel node-identifier-acquisition procedure is required.

The channel node-identifier-acquisition procedure provides a means by which the channel link-level facility can acquire the neighboring-node-identifier. The *neighboring-node-identifier* refers to the identification of the node attached to the other end of the link.

The channel-logical-path establishment procedure creates, at the channel link-level facility, information necessary for a particular channel image to communicate with a particular control-unit image to perform I/O operations.

When the logical path is established, the physical path, the link-level facilities at each end of the physical path, and the device-level facilities associated with those link-level facilities shall be considered operational and shall have the capability of performing their respective functions with respect to that logical path (see clause 12 for the device frames a channel and a control unit can send and receive).

10.1.2.1 Channel link-address acquisition

An unidentified channel-link-level facility (that is, a channel link-level facility that has no assigned link address) shall attempt to acquire its assigned link address as soon as it considers the link operational; the link address shall be acquired by means of the link-address-acquisition procedure which is initiated by sending an ALA frame.

If the unidentified channel-link-level facility sends an ALA frame and receives an error-free link-level-ACK frame in response, a dynamic switch is assumed to be attached to the other end of the link. If the channel is switched-point-to-point, the destination link address from the link-level-ACK frame is acquired as the assigned link address, and the channel link-level facility shall be become identified. If no dynamic switch and no link address is in the description of the channel-path configuration, then the initialization does not continue. If either no dynamic switch and one link address is in the description of the channel-path configuration, or a dynamic switch is in the description of the channel-path configuration, channel link-address acquisition is complete, and the channel link-level facility shall continue with its initialization process (see 10.1).

If the unidentified channel-link-level facility sends an ALA frame and receives a link-level-reject frame with a reject code of acquire-link-address error, a control unit is known to be attached to the other end of the link. The channel link-level facility shall proceed through model-dependent means to acquire its link address in accordance with the link-address definitions in 4.4.3. When the channel link-level facility acquires a link address, it shall become identified. The identified channel-link-level facility may then subsequently receive an ALA frame from the attached control unit, for which it determines through model-dependent means the link address for the link on which the ALA was received and shall use that link address as the destination link address to send a link-level-ACK frame.

If the unidentified channel-link-level facility sends an ALA frame and receives either a port-busy frame or a link-level-busy frame, it shall retry channel link-address acquisition by sending the ALA frame and awaiting the response.

If, while performing link-address acquisition, the unidentified channel-link-level facility receives a frame with a connect-SOF delimiter and a passive-EOF delimiter, and the frame is other than an ALA frame, the channel link-level facility shall respond with a link-level-reject frame containing an acquire-link-address-error reject-reason code.

If while performing channel link-address acquisition, the unidentified channel-link-level facility receives an ALA frame, a control unit is assumed to be attached to the other end of the link. The channel link-level facility shall proceed through model-dependent means to acquire its link address in accordance with the link-address definitions in 4.4.3. When the channel link-level facility acquires a link address, it becomes identified. The identified channel-link-level facility then determines through model-dependent means the link address for the link on which the ALA was received and shall use that link address as the destination link address to send a link-level-ACK frame.

If the channel link-level facility is unable to determine either the destination address or source address to be used in the link-level-ACK frame in time to avoid a link timeout at the control-unit link-level facility or if the time required to determine either of these addresses is unpredictable, the channel link-level facility shall respond to the ALA frame with a link-level-busy frame. After the channel link-level facility has acquired its link address by model-dependent means and determined what link address to associate with the control unit attached to the other end of the link, the channel link-level facility shall no longer respond with a link-level-busy frame when an ALA frame is received; instead the channel link-level facility shall respond with a link-level-ACK frame.

When it is determined or assumed that a control unit is attached to the other end of the link and when the channel link-level facility has acquired its link address and has sent a link-level-ACK frame in response to an ALA frame received from the attached control unit, channel link-address acquisition is complete. Alternatively, when it is determined or assumed that a control unit is attached to the other end of the link, the channel can treat link-address acquisition as complete as soon as the channel link-level facility becomes identified. If no dynamic switch is included in the description of the channel-path configuration, or a dynamic switch and one link address is included in the description of the channel-path configuration, then the channel link-level facility shall continue with the initialization process. If a dynamic switch and two or more link addresses are included in the description of the channel-path configuration, then the initialization process performed by this channel link-level facility does not continue. If the channel link-level facility is connected to a control-unit link-level facility by means of a static connection through a dynamic switch, the channel link-level facility is unable to determine the presence of that dynamic switch while the static connection exists.

While performing channel link-address acquisition, an unidentified channel-link-level facility may receive and send an ALA frame simultaneously. If this happens, the frame sent and the frame received represent two independent actions, each requiring a successful completion of the request/response protocols for link-control frames (see 9.1.14). In addition, as soon as the received ALA frame is recognized, a control unit is assumed to be attached to the other end of the link. At this point, the channel link-level facility shall proceed with channel link-address acquisition and shall acquire its link address by model-dependent means, while awaiting the response to the ALA that it sent. In this case, if the response to the ALA frame is a link-level-ACK frame, the destination link address of the link-level-ACK frame is not acquired as the assigned link address; channel link-address acquisition is terminated, and the initialization process performed by this channel link-level facility does not continue.

When a link address has been acquired, the link address is retained until the channel link-level facility has entered one of the offline states, entered the link-failure state, or an error or malfunction causes the link address to be reset (see 14.1, 14.1.8 and 14.2). When the link address is reset, the channel link-level facility shall become unidentified. When this occurs, any logical paths associated with that link address shall also be removed.

10.1.2.2 Channel node-identifier acquisition

A channel link-level facility shall attempt to acquire the neighboring-node-identifier by sending a request-node-identifier (RID) frame as soon as the channel has acquired its link address. The neighboring link-level facility shall respond with a link-control frame containing the node identifier (IDR) and a flag indicating the validity of the node identifier.

When a condition occurs that causes regression into the channel link-address-acquisition portion of the initialization process for a link, the channel link-level facility may discard the neighboring-node-identifier or may consider the neighboring-node-identifier to be no longer current. When conditions permit, the channel link-level facility shall attempt to acquire the current node identifier of the attached node.

When the channel does not require the information contained in the node identifier of its neighbor, the channel link-level facility shall be permitted to continue initialization of this and other control-unit link-level facilities, if appropriate, on the channel path even if the attempt to acquire the node identifier of its neighbor was unsuccessful. The lack of success in acquiring a valid and current node identifier need not prevent the establishment of logical paths or the execution of link-level and device-level functions. The retry attempt to acquire the node identifier may be deferred for a link error, a link-reject condition, a port-reject condition, a link-level-busy condition, or a port-busy condition.

When the channel requires the information contained in a valid and current node identifier of its neighbor and there is reason to expect that a retry will eventually be successful, the channel link-level facility shall retry channel node-identifier acquisition until a valid and current node identifier is acquired.

If the channel link-level facility sends the request-node-identifier frame and receives a link-level-reject frame with an acquire-link-address error, the channel link-level facility shall retry channel node-identifier acquisition by sending the request-node-identifier frame (see 14.1.8).

A channel link-level facility that checks the node-descriptor flag field in the received IDR frame shall perform the following actions:

a) When the node-ID validity code is zero, the channel link-level facility establishes the received 32-byte node descriptor as the node descriptor of its neighbor, and channel node-identifier acquisition is complete;

b) When the node-ID validity code is not zero, the channel link-level facility shall either:

  1) Check the existing node-descriptor node-ID-validity code if it had previously established the node descriptor of its neighbor and, if valid and not current, maintain the existing 32-byte node descriptor with an indication that the node identifier is no longer current, or maintain an indication that the node descriptor of its neighbor is not valid;

  2) Not check the existing node-descriptor node-ID-validity code and maintain an indication that the node descriptor of its neighbor is not valid.

NOTE – The preferred implementation is option b-1.

A channel that requires successful node-identifier acquisition shall perform the checks described above, and when the node-ID-validity code is not zero, the channel link-level facility shall retry channel node-identifier acquisition until a valid node descriptor is acquired or shall suspend the initialization process if there is reason to suspect that subsequent retries will not be successful. If the channel has to provide node descriptors in a function such as link-incident reporting prior to acquisition of a valid and current node descriptor, the channel shall send the node descriptor of its neighbor with a node-ID-validity code of 1 or 2, as appropriate.

10.1.2.3 Establishment of logical paths by the channel

The last initialization procedure performed at the channel link-level facility is the establishment of logical paths between the channel images sharing the link-level facility and the control-unit images configured to the channel images.

The channel shall initiate the establishment of the logical path by sending a link-control frame containing a request for the establishment of a logical path. If the control unit is able to form a logical path between the channel image and control-unit image, it shall respond with a link-control frame indicating that the request is accepted and the logical path has been established; otherwise, it shall indicate that the logical path is not accepted and the reason why. When the logical path is established, the channel then shall allow device-level communication on that logical path between the specified channel image and control-unit image (see 9.1.1).

The channel shall consider the logical path to be established upon receiving an error-free frame with the logical-path-established response from the control unit. When the channel considers the logical path to be established, the channel shall consider the initialization process to be complete for the combination of that channel image and that control-unit image. When the logical path is established, the physical path, the link-level facilities at each end of the physical path, and the device-level facilities associated with those link-level facilities shall be considered operational and have the capability of performing their respective functions (see clause 12 for the device frames a channel and a control unit can send and receive).

A logical path is no longer usable for the exchange of device frames when either end of the path no longer considers the logical path to be established. A channel link-level facility shall consider a logical path to be no longer established when any of the following events occurs:

- the channel link-level facility sends a remove-logical-path request to the control-unit link-level facility, and an error-free response is returned indicating logical path removed (see 9.1.2);

- a channel link-level facility detects an error for which the recovery action includes removing the logical path (see clause 14).

When a logical path is not established, the channel link-level facility shall not allow device-level functions and protocols with respect to the corresponding control-unit image. When the channel link-level facility is initialized, one or more attempts to establish a logical path with each control-unit image in the channel-path configuration shall be made. The number of times beyond one that the channel link-level facility attempts to establish the logical path for a control-unit image at initialization is model dependent.

A logical path may be removed by a control-unit link-level facility because of a link failure, the offline procedure, a link-error condition or a condition internal to the control unit; when this occurs, it may be asynchronous to the activity of the channel link-level facility (see 14.1). If the control-unit is not actively communicating with the channel image, the channel may not be immediately aware of the loss of the logical path. When the channel image later attempts to perform a device-level function using the logical path, abnormal conditions which affect the initialization of the channel link-level facility shall be encountered (see 14.1.8).

If the channel link-level facility sends the link-control frame containing a request for the establishment of a logical path and receives a link-level-reject frame with an acquire-link-address error, the channel link-level facility shall again attempt to initiate the establishment of the logical path by resending the link-control frame containing the request for the establishment of a logical path and awaiting the response.

During the initiation of a device-level function for an I/O device or control unit, if the channel determines that a logical path, which it considers to exist, is to be removed, then the channel shall consider that I/O device or control unit is not capable of performing device-level functions on this logical path. The channel also recognizes this condition for all other I/O devices for which an I/O operation was active or disconnected with respect to that logical path. All other I/O devices associated with the logical path that is removed are not affected.

During the initiation of a device-level function for an I/O device or control unit, if the channel determines that the logical path was previously not established, the channel link-level facility shall attempt, by means of the initialization process, to establish the logical path before performing the device-level function. If the logical path is successfully established, the channel shall not recognize an error condition, and the channel shall attempt to initiate the device-level function if the device-level function is still pending. If the initialization process is terminated during the initiation of the device-level function, the logical path remains not established and the channel shall consider the I/O device or control unit to be incapable of performing device-level functions with respect to that logical path.

10.1.3 Initialization process for control-unit link-level facility

For a control-unit link-level facility, the initialization process consists of the following procedures:

a) Link initialization;

b) Control-unit link-address acquisition;

c) Control-unit node-identifier acquisition;

d) Control-unit logical-path establishment.

The control-unit link-address acquisition procedure provides a means by which the control-unit link-level facility becomes identified. Once the control-unit link-level facility is identified, any of the link-level frames that are allowed for a control-unit link-level facility can be sent, unless information from the control-unit node-identifier-acquisition procedure is required.

The control-unit node-identifier-acquisition procedure provides a means by which the control-unit link-level facility can acquire the neighboring-node-identifier.

When the logical path is established, the physical path, the link-level facilities at each end of the physical path, and the device-level facilities associated with those link-level facilities shall be considered operational and shall have the capability of performing their respective functions (see clause 12 for the device frames a channel and a control unit can send and receive).

10.1.3.1 Control-unit link-address acquisition

An unidentified control-unit link-level facility shall attempt to acquire its assigned link address as soon as the control-unit link-level facility considers the link operational. Independent of the channel-path configuration, an unidentified control-unit link-level facility shall acquire its assigned link address by performing control-unit link-address acquisition. A control-unit link-level facility shall initiate the link-address acquisition procedure by sending the ALA frame.

If an unidentified control-unit link-level facility sends an ALA frame and receives a port-busy or link-level-busy frame, it shall continue to attempt to acquire its link address by sending the ALA frame and awaiting the response.

When an unidentified control-unit link-level facility sends the ALA frame and receives an error-free link-level-ACK frame in response, the destination link address of the link-level-ACK frame is adopted as the assigned link address, and the control-unit link-level facility shall become identified.

An unidentified control-unit link-level facility that receives a frame without any link errors and with a connect-SOF delimiter and a passive-EOF delimiter shall respond with a link-level-reject frame containing a reject-reason code of acquire-link-address error.

If an identified control-unit link-level facility which is not part of a dynamic-switch control unit receives an ALA frame, the control-unit link-level facility shall respond with a link-level-reject frame containing the acquire-link-address-error reject-reason code.

While performing control-unit link-address acquisition, an unidentified control-unit link-level facility may receive and send an ALA frame simultaneously. If this happens, the frame sent and the frame received represent two independent actions, each requiring a successful completion of the request/response protocols for link-control frames; that is, as soon as the received ALA frame is recognized, the link-level-reject frame is generated as a response, and the control-unit link-level facility shall continue to wait for the response to the ALA that it sent (see 9.1).

When a link address has been acquired, the link address is retained until either the control-unit link-level facility enters one of the offline or link-failure states, or an internal condition as the result of an error or malfunction causes the link address to be reset. When the link address is reset, the control-unit link-level facility shall become unidentified. When this occurs, any logical paths associated with that link address shall also be removed.

The control-unit link-level facility of a dynamic-switch control unit shall acquire its assigned link address through model-dependent means.

The ALA frame is always recognized by the dynamic-switch control unit. When an ALA frame is received, the dynamic-switch control unit, through model-dependent means, shall determine the link address associated with the switch port on which the ALA frame was received and shall use that link address as the destination link address to send a link-level-ACK frame. The source link address used in the link-level-ACK frame is the link address assigned to the dynamic-switch control unit. After transmitting the link-level-ACK frame, the dynamic-switch control unit shall have initiative to report a state change (see 9.1.13).

10.1.3.2 Control-unit node-identifier acquisition

A control-unit link-level facility shall attempt to acquire the neighboring-node-identifier by means of a request-node-identifier (RID) frame. The neighboring link-level facility shall respond with a link-control frame containing the node identifier (IDR) and a flag indicating the validity of the node identifier.

When a condition occurs that causes regression into the control-unit link-address-acquisition part of the initialization process, the control-unit link-level facility may discard the neighboring-node-identifier or may consider the neighboring-node-identifier to be no longer current. When conditions permit, the control-unit link-level facility shall attempt to acquire the current node identifier of the attached node.

A dynamic-switch control unit shall attempt to acquire the node identifier of the link-level facility attached to each switch port of the dynamic switch.

The link-level facility of a control unit other than a dynamic-switch control unit shall attempt to acquire the node identifier of the attached node.

When the control unit does not require the information contained in the node identifier of its neighbor, the control-unit link-level facility shall be permitted to continue initialization of this control-unit link-level facility even if the attempt to acquire the node identifier was unsuccessful. The lack of success in acquiring a valid and current node identifier need not prevent the establishment of logical paths or the execution of link-level and device-level functions.

When the control unit requires the information contained in a valid and current node identifier of its neighbor and there is reason to expect that a retry will eventually be successful, the control-unit link-level facility shall retry control-unit node-identifier acquisition until a valid and current node identifier is acquired. If the control unit receives and recognizes an ELP frame, the control unit sends in response an LPR frame with a reason code of device-level-initialization not complete (b’0011’).

A control-unit link-level facility that checks the node-descriptor flag field in the received IDR frame shall perform the following actions:

a) When the node-ID-validity code is zero, the control-unit link-level facility establishes the received 32-byte node descriptor as the node descriptor of its neighbor, and control-unit node-identifier acquisition is complete;

b) When the node-ID validity code is not zero, the control-unit link-level facility shall either:

  1) Check the existing node-descriptor node-ID-validity code if it had previously established the node descriptor of its neighbor and, if valid and not current, maintain the existing 32-byte node descriptor with an indication that the node identifier is no longer current, or maintain an indication that the node descriptor of its neighbor is not valid; or

  2) Not check the existing node-descriptor node-ID-validity code and maintain an indication that the node descriptor of its neighbor is not valid.

NOTE – The preferred implementation is option b-1.

A control unit that requires successful node-identifier acquisition shall perform the checks described above, and when the node-ID-validity code is not zero, the control-unit link-level facility shall retry control-unit node-identifier acquisition until a valid node descriptor is acquired or shall suspend the initialization process if there is reason to suspect that subsequent retries will not be successful. If the control unit has to provide node descriptors in a function such as link-incident reporting prior to acquisition of a valid and current node descriptor, the control unit shall send the node descriptor of its neighbor with a node-ID-validity code of 1 or 2, as appropriate.

10.1.3.3 Establishment of logical paths by the control unit

The control-unit link-level facility shall attempt to establish a logical path when a link-control frame containing a request for the establishment of a logical path between a specified channel image and a control-unit image is received. The control-unit logical-path-establishment procedure consists of either accepting or refusing the requested logical path. If the control-unit is able to accept the specified logical path, it shall respond with a link-control frame containing an indication that the request is accepted and the logical path has been established; otherwise, it shall indicate that the logical path is not accepted. When the logical path is established, the control-unit link-level facility then allows device-level communication on that logical path between the channel image and control-unit image (see 9.1.1).

The control unit shall consider the logical path to be established upon sending the logical-path-established response without errors. When the control-unit considers the logical path to be established, the control unit shall consider the initialization process to be complete for the combination of that channel image and that control-unit image.

A logical path is no longer usable for the exchange of device frames when either end of the path no longer considers the logical path to be established. A control unit shall consider a logical path to be no longer established when any of the following events occurs:

- a remove-logical-path request is received, and an error-free response is returned indicating logical path removed (see 9.1.2);
- the control-unit detects an error for which the recovery action includes removing the logical path (see 14.1);
- the control-unit link-level facility is powered off.

When a logical path is not established, the control-unit link-level facility shall not perform device-level functions and protocols with respect to the logical path.

10.2 Link-incident reporting

A *link incident* shall be recognized by a channel or control unit when a condition is detected for which an incident code is defined. When a link incident is recognized, a *link-incident record* that contains information related to the link incident shall be created. When a control unit recognizes a link incident, it shall use the link-incident-reporting procedure to pass a link-incident record to a channel (see 10.2.2). When a channel recognizes a link incident, a report shall be generated, but this link-incident-reporting procedure shall not be used.

10.2.1 Link-incident record

The format of the link-incident record is shown in figure 32:

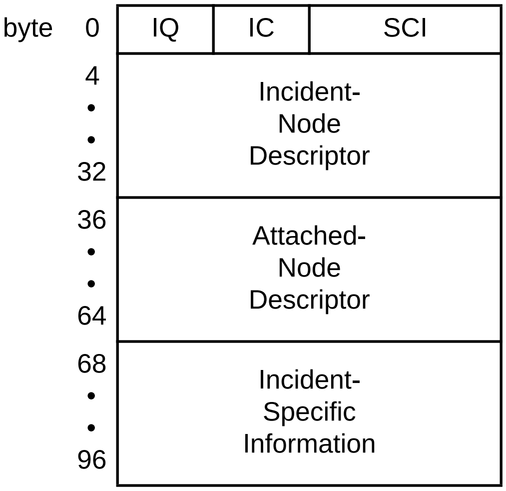


Figure 32 – Link-incident record

Incident Qualifier (IQ): Byte 0 describes the manner in which the contents of the link-incident record shall be interpreted. The meaning of bits 0-7 is as follows:

Bits	Meaning
0	*Null:* When one, bit 0 indicates a null link-incident record; the link-incident record does not contain valid incident data. When zero, bit 0 indicates that the link-incident record contains valid incident data.
1	*Resend:* When one, bit 1 indicates that the link-incident record has been previously sent to a channel in a LID frame. When zero, bit 1 indicates that the link-incident record has not been previously sent to a channel.
2	*Dynamic switch:* When one, bit 2 indicates that the incident node, identified by the incident-node descriptor, is a dynamic-switch port. When zero, bit 2 indicates that the incident node is not a dynamic-switch port.
3	*Static connection:* When bit 2 is one, indicating that the incident node is a dynamic-switch port, bit 3 indicates whether or not the switch port is in a static-connection state. When one, bit 3 indicates that the switch port is in a static-connection state. When zero, bit 3 indicates that the switch port is not in a static-connection state.
4-5	*Reporting class:* Bits 4 and 5 constitute a two-bit code which identifies the reporting class for the link incident. The codes and their meanings are as follows:

Value	Meaning
0	*Informational report:* All link incidents reported with incident-code bit 0 set to one use reporting class 0.
1	*Link degraded but operational:* Link incidents reported with incident-code bit 0 set to zero use reporting class 1 if the link-level facility or dynamic-switch port associated with the incident node is not in a link-failure or offline state as a result of the event which generated the link-incident report.

2 *Link not operational:* Link incidents reported with incident-code bit 0 set to zero use reporting class 2 if the link-level facility or dynamic-switch port associated with the incident node is in a link-failure or offline state as a result of the event which generated the link-incident report.

3 Reserved.

6 *Subassembly type:* When one, bit 6 specifies that the type of subassembly used for the link that is the subject of this link-level incident report is laser. When zero, bit 6 specifies that the type of subassembly used for the link that is the subject of this link-incident report is not laser.

7 *FRU identification:* When one, bit 7 specifies that the incident-specific-information field is in a format that provides field-replaceable-unit (FRU) identification. When zero, bit 7 specifies that the incident-specific-information field is model dependent.

Incident Code (IC): Byte 1 contains the incident code which describes the incident that was observed by the incident node.

Bit 0 of the incident code indicates whether the link-incident record is a primary or secondary report of the link incident. When bit 0 is set to zero, the link-incident record is a primary report. When bit 0 is set to one, the link-incident record is a secondary report.

Bits 1-7 of the incident code contain a value that specifies the type of incident which was observed. The values that can be specified and their meanings are as follows:

Value	Meaning
0	Reserved.
1	*Implicit incident:* A condition which has been caused by an event known to have occurred within the incident node has been recognized by the incident node. The condition affects the attached link in such a way that it may cause a link incident to be recognized by the attached node.
2	*Bit-error-rate threshold exceeded:* The number of code-violation errors recognized by the incident node has exceeded a threshold (see 5.4).
3	*Link failure - loss of signal or synchronization:* A loss-of-signal or loss-of-sync condition has been recognized by the incident node, and it persisted for more than the link-interval duration (see 10.5).
4	*Not-operational sequence (NOS) recognized:* The not-operational sequence has been recognized by the incident node (see 10.5).
5	*Link failure - sequence timeout:* The incident node has recognized either a connection-recovery timeout or a timeout when timing for the appropriate response while in the transmit-OLS state (see 10.5).
6	*Link failure - invalid sequence for link-level-facility state:* Either a UD or UDR sequence was recognized by the incident node while in the wait-for-offline-sequence state (see 10.5).
7-127	Reserved.

Statically Connected Switch Interface (SCI): When bits 2 and 3 of the IQ field are both ones, indicating that the incident node is a statically connected dynamic-switch port, bytes 2 and 3 of word 0 contain the physical-interface identifier of the other switch port participating in the static connection.

Incident-Node Descriptor: Bytes 4-35 contain the node descriptor of the incident node. The contents of a node descriptor are described in 16.1.

Attached-Node Descriptor: Bytes 36-67 contain the node descriptor of the node attached to the incident node at the time the link incident was detected. The contents of a node descriptor are described in 16.1.

Incident-Specific Information: When bit 7 of the incident-qualifier field is set to zero, bytes 68-99 contain node-dependent incident information, which may provide additional information related to the incident.

When bit 7 of the incident-qualifier field is set to one, bytes 68-99 contain field-replaceable-unit (FRU) identification information.

When the incident-specific-information field contains FRU identification information, the format of the incident-specific information is illustrated in figure 33.

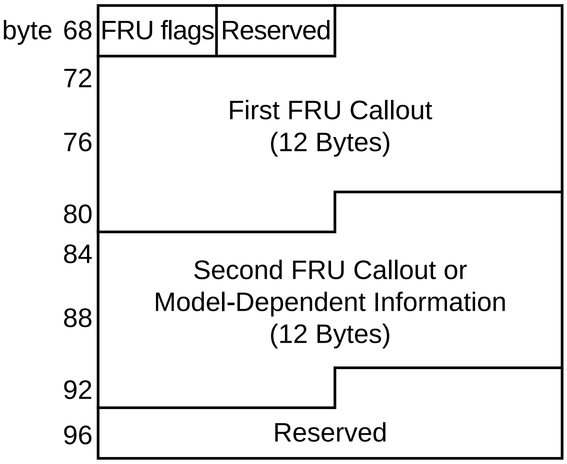


Figure 33 – Incident-specific information

Byte 68 contains the FRU-flags field. The meaning of bits 0-7 is as follows:

Bit	Meaning
0	Reserved.
1	*Format bit; FRU-callout-field format:* 0 = FRU-part-number format 1 = FRU-code format
2-5	Reserved.
6-7	*Validity code for FRU-callout fields:*

Value	Meaning
00	Reserved.
01	First-FRU-callout field valid; Second-FRU-callout field contains 12 bytes of model-dependent data.
10	First-FRU-callout and second-FRU-callout fields valid.
11	Reserved.

Byte 69 is reserved and set to zero.

Bytes 70-81 contain the first-FRU-callout identification information.

Bytes 82-93 contain either the second-FRU-callout identification information or 12 bytes of model-dependent information, depending on the value of bits 6-7 of the FRU-flags field.

Bytes 94-99 are reserved and set to zeros.

The format of a valid FRU-callout field depends on the value of bit 1 (format bit) of the FRU-flags field within the same link-incident record.

When the format bit is set to zero, the FRU-callout field is in EBCDIC, right justified, with either leading blanks (x'40') or leading EBCDIC zeros (x'F0').

When the format bit is set to one, the FRU-callout field is in hexadecimal, right justified, with leading zeros (x'00').

10.2.2 Link-incident-reporting procedure

The control unit shall generate one link-incident record per link incident. If the control unit attempts to generate a link-incident record but does not have resources to hold the record until it is read by a channel, the oldest link-incident record shall be discarded, and the new link-incident record shall be retained.

The control unit shall notify the channel of the availability of a link-incident record by sending a LIN frame to the channel. After the LIN frame is sent and acknowledged, the control unit shall not send another LIN frame on any path until it has received an RIR frame and has responded with a LID frame, or until a timeout (see text below) occurs. One LIN frame is sent per link-incident record. The control unit may select any channel with which it has an established logical path. If no logical paths are established, it is model dependent as to whether the control unit retains or discards the link-incident record. If the record is retained, the control unit shall retain initiative to send a LIN frame when a logical path is established.

A control unit shall not generate a control-unit or device busy condition as a result of sending a LIN frame and waiting for an RIR frame.

When the channel recognizes a LIN frame, it shall send a link-level-ACK frame in response. After recognizing a LIN frame and sending a link-level-ACK frame, the channel shall send an RIR frame to the control unit to request a link-incident record. The channel shall send the RIR frame on the same physical path on which the LIN frame was received. A channel shall send an RIR frame only after recognizing and acknowledging a LIN frame.

When the control unit receives the ACK frame from the channel in response to the LIN frame, the control unit shall wait a model-dependent interval, between 1 and 5 minutes, for a link-incident record to be read by means of an RIR frame. If any link-incident record remains unread for longer than the model-dependent time period, the control unit shall reestablish initiative to send another LIN frame to any channel with which it has an established logical path. The control unit shall send one LIN frame each time the period expires and shall begin timing again when the link-level-ACK frame is received in response.

A control unit shall accept an RIR frame on any physical interface, independent of the existence of any logical paths, and shall send a link-incident-data (LID) frame in response. After sending a LID frame containing a link-incident record, the control unit shall discard the link-incident record, unless it is the last link-incident record. The last link-incident record is retained and marked as *sent* to indicate that a LID frame has been previously sent for the link-incident record. A link-incident record that is not marked as sent is referred to as *unsent*, to indicate that no LID frame has been sent for that record.

The contents of the LID frame shall be determined according to the following rules:

a) If the control unit has one or more unsent link-incident records, the LID frame shall contain the record which was generated first among the set of unsent link-incident records. The null and resend bits in the link-incident record shall be set to zeros;

b) If the control unit has no unsent link-incident records, but has a sent link-incident record, the LID frame shall contain the sent record. The null bit shall be set to zero and the resend bit shall be set to one in the link-incident record;

c) If the control unit has no link-incident record, either sent or unsent, the LID frame shall contain a null link-incident record. The null bit shall be set to one and the contents of the other fields in the link-incident record are undefined.

10.3 Offline procedures

The offline procedures are used so that the link-level facility or dynamic-switch port at one end of a link can cause the link-level facility or dynamic-switch port at the other end of the link to avoid recognizing link errors or link failure.

For a channel, having a link offline shall cause the removal of logical paths, the information related to logical paths established over that link to be reset, the link address to be reset, the neighboring-node-identifier to be discarded or considered no longer current, and the channel path to be considered not operational. The I/O devices formerly accessible over that channel path shall be considered no longer capable of executing a device-level or link-control function over that channel path.

For a control unit, having a link offline shall cause the link address and logical paths for that link to be removed, and the neighboring-node-identifier to be discarded or considered no longer current. With the link offline, the control unit is unable to perform either device-level functions or link-control functions over that link.

When a link is no longer offline and when conditions permit, a channel or control unit shall start the initialization process as described in 10.1.

For a dynamic-switch port which is not in the static state, being in the offline state shall cause the dynamic-switch port to remove a dynamic connection if it exists and shall prevent the forming of dynamic connections with any other dynamic-switch port. The dynamic-switch port shall remain offline for as long as the conditions that cause the offline procedure exist. If the dynamic-switch port is in a static connection, the connection and switch-port state are not affected.

10.3.1 Conditions that cause the offline procedure

The offline procedure shall be performed in the following cases:

a) A link-level facility or dynamic-switch port intends to power off;

b) A link-level facility or dynamic-switch port is being powered on.

While a unit is being powered on, the link-level facility for that unit shall enter the offline-transmission state. When powering on is complete, the link-level facility shall be allowed to leave the offline-transmission state;

c) An event internal to the channel or control unit occurs which causes the link address related to that link within the channel or control unit to be reset; for example, power-on system-initialization procedure;

d) An event internal to the channel or control unit occurs which causes the logical-path information related to that link within the channel or control unit to be reset for all sharing channel or control-unit images and for which the option of sending state-change-notification is not used; for example, power-on system-initialization procedure;

e) A link-level facility or dynamic-switch port is being taken offline;

f) An event internal to the dynamic switch occurs which prevents the dynamic-switch port from performing its function of making and breaking connections with other switch ports;

g) The offline sequence is recognized.

NOTE – When a condition occurs that causes the offline procedure, the link-level facility or dynamic-switch control unit should consider the neighboring-node-identifier as no longer current. This requirement applies for all conditions which cause the offline procedure, except for powering off or system-initialization procedure, in which case, the link-level facility or the dynamic-switch control unit may discard the neighboring-node-identifier.

10.3.2 Offline states

A link-level facility or dynamic-switch port can be in one of three offline states for each attached link. A link-level facility or dynamic-switch port shall initiate the offline procedure for an attached link by entering the offline-transmission state and transmitting the offline sequence on the attached link. When conditions that cause the offline-transmission state no longer exist and the receiver recognizes a link failure because of a loss-of-signal or loss-of-sync condition, the link-level facility or dynamic-switch port shall enter the wait-for-offline-sequence state. A link-level facility or dynamic-switch port that recognizes the offline sequence shall enter the offline-reception state for that link and shall respond by transmitting the UD sequence on that link.

10.3.2.1 Offline-transmission state

When a link-level facility or dynamic-switch port is in the offline-transmission state for an attached link, it shall not recognize link errors or link failure for that link. When the link-level facility or dynamic-switch port enters the offline-transmission state, it shall send the offline sequence for a minimum of 1 millisecond. After entering the offline-transmission state and sending the offline sequence for a minimum of 1 millisecond, the link-level facility or dynamic-switch port may then do anything which will not cause the link-level facility or dynamic-switch port at the other end of the link to leave the offline-reception state or the wait-for-offline state unless intended to do so.

When a dynamic-switch port is in the offline-transmission state and a dynamic connection exists, the dynamic connection shall be removed, and the other dynamic-switch port is caused to enter the UD-transmission state and initiate transmission of the UD sequence.

If the conditions that initiate the offline procedure no longer exist, the link-level facility or the dynamic-switch port may leave the offline-transmission state. To leave the offline-transmission state, the link-level facility or the dynamic-switch port shall ensure the offline sequence has been sent for 1 second or more and then interrogates the information received on the link. The link-level facility or the dynamic-switch port shall continue the transmission of the offline sequence until one of the following conditions is satisfied:

a) The unconditional-disconnect (UD) sequence is received and recognized, causing the link-level facility or the dynamic-switch port to enter the UD-reception state;

b) The offline sequence is received and recognized, causing the link-level facility or the dynamic-switch port to enter the offline-reception state;

c) The not-operational sequence is received and recognized, causing the link-level facility or the dynamic-switch port to recognize a link failure and enter the transmit-OLS state;

d) A loss-of-signal or loss-of-sync condition that persists for longer than the link-interval duration is detected, causing the link-level facility or the dynamic-switch port to enter the wait-for-offline-sequence state;

e) At least the link-interval duration has elapsed since the link-level facility or dynamic-switch port started interrogating the information received on the link and before any of the preceding conditions were satisfied, causing the link-level facility or dynamic-switch port to enter the wait-for-offline-sequence state.

10.3.2.2 Offline-reception state

When a link-level facility or dynamic-switch port is in the offline-reception state for an attached link and is recognizing the offline sequence, it shall not recognize link errors or link failure for that link. The link-level facility or the dynamic-switch port shall continue the transmission of the UD sequence until one of the following conditions is satisfied:

a) The unconditional-disconnect-response (UDR) sequence is received and recognized, causing the link-level facility or the dynamic-switch port to enter the UDR-reception state;

b) The unconditional-disconnect (UD) sequence is received and recognized, causing the link-level facility or the dynamic-switch port to enter the UD-reception state;

c) The not-operational sequence is received and recognized, causing the link-level facility or the dynamic-switch port to recognize a link failure and enter the transmit-OLS state;

d) A loss-of-signal or loss-of-sync condition that persists for longer than the link-interval duration is detected, causing the link-level facility or the dynamic-switch port to enter the wait-for-offline-sequence state;

e) The link-interval duration has elapsed since the offline sequence was no longer recognized and none of the preceding conditions were satisfied, causing the link-level facility or the dynamic-switch port to enter the wait-for-offline-sequence state.

When a dynamic-switch port is in the offline-reception state and a dynamic connection exists, the dynamic connection shall be removed, and the other dynamic-switch port is caused to enter the UD-transmission state and initiate transmission of the UD sequence.

If conditions that cause entry into the offline-transmission state occur while the link-level facility or the dynamic-switch port is in the offline-reception state, the link-level facility or switch port shall enter the offline-transmission state.

10.3.2.3 Wait-for-offline-sequence state

When a link-level facility or dynamic-switch port is in the wait-for-offline-sequence state for an attached link, it shall not recognize link errors or link failure for that link. The link-level facility or the dynamic-switch port shall continue the transmission of the not-operational sequence until one of the following conditions is satisfied:

a) The offline sequence is received and recognized, causing the link-level facility or the dynamic-switch port to enter the offline-reception state;

b) The not-operational sequence is received and recognized, causing the link-level facility or the dynamic-switch port to recognize a link failure and enter the transmit-OLS state;

c) The UD or UDR sequence is received and recognized, causing the link-level facility or the dynamic-switch port to recognize a link failure and enter the transmit-NOS state.

If conditions that cause entry into the offline-transmission state occur while the link-level facility or the dynamic-switch port is in the wait-for-offline-sequence state, the link-level facility or switch port shall enter the offline-transmission state.

10.4 Connection recovery

Connection recovery is used by a link-level facility or by a dynamic-switch port to cause the removal of a dynamic connection between its link and any other link, if such a connection exists. Connection recovery uses the interlocked exchanges of the unconditional-disconnect (UD) and unconditional-disconnect-response (UDR) sequences.

A link-level facility or dynamic-switch port shall initiate connection recovery by transmitting the UD sequence. The link-level facility or dynamic-switch port that recognizes the UD sequence shall respond by transmitting the UDR sequence. When the link-level facility or dynamic-switch port that is transmitting the UD sequence recognizes the UDR sequence, it shall stop sending the UD sequence and shall start transmitting idle characters. When the link-level facility or dynamic-switch port that is transmitting the UDR sequence recognizes 16 or more consecutive idle characters, it shall stop sending the UDR sequence and shall start transmitting a string of at least 64 consecutive idle characters.

When the transmission of the UDR sequence is ended and the transmission of the string of 64 idle characters is started, the sending link-level facility shall consider connection recovery to be completed. When the link-level facility or dynamic-switch port that recognized the UDR sequence and is transmitting idle characters recognizes 16 or more consecutive idle characters or, optionally, start-of-frame reception, it shall consider connection recovery to be completed.

Once a link-level facility or dynamic-switch port considers connection recovery to be completed, it is permitted to accept frames and respond to those frames even though a minimum of 64 idle characters is not yet completely sent. When a frame allows connection recovery to be completed in this manner, that frame may be accepted, and the appropriate response for the conditions present sent.

If a link failure is recognized during these exchanges of sequences, then connection recovery shall be terminated, and the link-failure protocols shall be observed (see 10.5). If conditions that cause an offline procedure to be initiated are recognized during these exchanges of sequences, connection recovery shall be terminated, and the offline procedure shall be performed (see 10.3.1).

When the connection-recovery procedure is invoked by a switch port that is dynamically connected to another switch port, it causes removal of the dynamic connection of the two switch ports and causes the other switch port to perform the connection-recovery procedure. When the connection-recovery procedure is invoked to remove a dynamic connection, the two dynamic-switch ports shall perform the connection-recovery procedure separately, and each switch port becomes available for new connections when it successfully completes the procedure.

When the UD or UDR sequence is recognized by a switch port that is in the static pass-through state, the other switch port is caused to transmit the received sequence on that other switch port's link, to give the appearance of a point-to-point connection. The static connection is not removed, and the transfer of the transmission characters received from the link at one switch port through the switch connection to the link at the other switch port is unaffected by connection recovery.

The following abnormal conditions that occur during connection recovery are detected as sequencing errors (see 11.1.4).

- less than the required number of consecutive ordered sets needed for the recognition of the UD or UDR sequences are received;
- an unsolicited UDR sequence is received and recognized. (The UDR sequence is always solicited; that is, the UDR sequence is only transmitted in response to the reception and recognition of the UD sequence).

10.4.1 Conditions that cause connection recovery

A link-level facility or a dynamic-switch port shall perform the connection-recovery procedure when it recognizes the UD sequence.

A link-level facility shall initiate the connection-recovery procedure when it does not know the state of a connection or which destination link-level facility is connected, or when certain abnormal conditions are recognized. A dynamic-switch port shall initiate a connection-recovery procedure whenever it is caused to remove a dynamic connection, other than by recognition of a disconnect-EOF delimiter. A dynamic-switch port shall also initiate the connection-recovery procedure when certain abnormal conditions are recognized that cause the removal of a dynamic connection.

The following conditions cause the connection-recovery procedure at a channel or control unit:

a) A connection error (see 11.1.1);

b) A connection-protocol error (see 11.1.7);

c) A device-level-connection error (see 13.3.1);

d) Any error recognized in a frame with a disconnect-EOF delimiter;

e) Any error recognized in a frame with a passive-SOF delimiter received in response to an initiation frame;

f) Any error recognized in a frame with a passive-SOF delimiter received during a connection and the conditions present do not correspond to one of the following:

1) The error is recognized at the channel after an accept-command-response frame is sent but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame;

2) The error is recognized at a control unit after an accept-command-response frame is received in response to a command-response frame but before a status frame is sent;

3) The error is recognized at the channel after the channel has sent a command frame in response to a status frame but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame;

4) The error is recognized at the control unit after a command-response frame is sent in response to a command frame for other than the first command of a channel program but before a status frame for that command is sent.

10.4.2 Connection-recovery procedure

A link-level facility or dynamic-switch port can be in any one of the following three connection-recovery-procedure states:

UD-Transmission State: This state shall be entered when a link-level facility or a dynamic-switch port initiates transmission of the UD sequence.

UD-Reception State: This state shall be entered when the UD sequence is received and recognized by a link-level facility or a dynamic-switch port.

UDR-Reception State: This state shall be entered when the UDR sequence is received and recognized by a link-level facility or a dynamic-switch port.

A link-level facility or dynamic-switch port shall not initiate transmission of the UD sequence while it is in the UD-reception state or the UDR-reception state.

If a link-level facility or dynamic-switch port has been transmitting the UD sequence and also is ready to transmit the UDR sequence as the result of receiving the UD sequence, and if the UDR sequence is received before the UDR sequence is transmitted, then the facility or switch port may send idle characters instead of the UDR sequence.

10.4.2.1 UD-transmission state

When a link-level facility or dynamic-switch port (excluding a dynamic-switch port which is in the static state) is in the UD-transmission state, it shall transmit the UD sequence continuously until one of the following conditions is satisfied:

a) The UDR sequence is received and recognized, causing the link-level facility or the dynamic-switch port to enter the UDR-reception state;

b) The UD sequence is received and recognized, causing the link-level facility or the dynamic-switch port to enter the UD-reception state;

c) A link failure is recognized by a receiver when the not-operational sequence has been recognized, causing the transmitter at the end of the link recognizing the link failure to enter the transmit-OLS state;

d) A link failure is recognized due to a loss-of-signal or loss-of-sync condition that persists for longer than the link-interval duration, causing the transmitter at the end of the link recognizing the link failure to enter the transmit-NOS state;

e) The offline sequence is recognized, causing the link-level facility or the dynamic-switch port to go offline and enter the offline-reception state;

f) A link failure is recognized because the connection-recovery-interval duration has elapsed since transmission of the unconditional-disconnect (UD) sequence was initiated and none of the preceding conditions were satisfied. This timeout condition is called a connection-recovery timeout. The link-level facility or the dynamic-switch port shall enter the transmit-NOS state (see 10.5).

(For additional information on transmit-OLS and transmit-NOS states, see 10.5.)

If conditions for entering the offline-transmission state occur while the link-level facility or the dynamic-switch port is in the UD-transmission state, the link-level facility or switch port shall enter the offline-transmission state.

10.4.2.2 UD-reception state

When a link-level facility or dynamic-switch port (excluding a dynamic-switch port which is in the static state) is in the UD-reception state, it shall cause the transmission of the UDR sequence to be initiated. The UDR sequence shall be transmitted continuously until one of the following conditions is satisfied.

a) Sixteen or more consecutive idle characters are recognized, causing the link-level facility or the dynamic-switch port to start the transmission of a series of idle characters consisting of a minimum of 64 or more consecutive idle characters, upon which the connection-recovery procedure is considered to be completed by this link-level facility or dynamic-switch port, and the link-level facility or dynamic-switch port shall enter the inactive state;

b) The UDR sequence is received and recognized, causing the link-level facility or the dynamic-switch port to enter the UDR-reception state;

c) A link failure is recognized by a receiver when the not-operational sequence has been recognized, causing the transmitter at the end of the link recognizing the link failure to enter the transmit-OLS state;

d) A link failure is recognized due to a loss-of-signal or loss-of-sync condition that has persisted for longer than the link-interval duration, causing the transmitter at the end of the link recognizing the link failure to enter the transmit-NOS state;

e) The offline sequence is recognized, causing the link-level facility or the dynamic-switch port to go offline and enter the offline-reception state;

f) A link failure is recognized because the connection-recovery-interval duration has elapsed since transmission of the unconditional-disconnect-response (UDR) sequence was initiated and none of the preceding conditions were satisfied. This timeout condition is called a connection-recovery timeout. The link-level facility or the dynamic-switch port shall enter the transmit-NOS state.

(For additional information on transmit-OLS and transmit-NOS states, see 10.5.)

If conditions for entering the offline-transmission state occur while the link-level facility or the dynamic-switch port is in the UD-reception state, then the link-level facility or switch port shall enter the offline-transmission state.

10.4.2.3 UDR-reception state

When a link-level facility or dynamic-switch port (excluding a dynamic-switch port which is in the static state) is in the UDR-reception state, it shall cause the transmission of idle characters to be initiated. The idle characters shall be transmitted until one of the following conditions is satisfied:

a) Sixteen or more consecutive idle characters are recognized, upon which the connection-recovery procedure is considered to be completed by this link-level facility or dynamic-switch port, and the link-level facility or dynamic-switch port shall enter the inactive state;

b) The UD sequence is received and recognized, causing the link-level facility or dynamic-switch port to enter the UD-reception state;

c) A link failure is recognized by a receiver when the not-operational sequence has been recognized, causing the transmitter at the end of the link recognizing the link failure to enter the transmit-OLS state;

d) A link failure is recognized due to a loss-of-signal or loss-of-sync condition that has persisted for longer than the link-interval duration, causing the transmitter at the end of the link recognizing the link failure to enter the transmit-NOS state;

e) The offline sequence is recognized, causing the link-level facility or the dynamic-switch port to go offline and enter the offline-reception state;

f) A link failure is recognized because the connection-recovery-interval duration has elapsed since transmission of idle characters was initiated and none of the preceding conditions were satisfied. This timeout condition is called a connection-recovery timeout. The link-level facility or the dynamic-switch port shall enter the transmit-NOS state.

In addition to the above conditions, certain implementations may consider UDR-reception completed when start-of-frame reception is recognized.

(For additional information on transmit-OLS and transmit-NOS states, see 10.5.)

If conditions for entering the offline-transmission state occur while the link-level facility or the dynamic-switch port is in the UDR-reception state, then the link-level facility or switch port shall enter the offline-transmission state.

10.5 Link failure

A link failure is a condition which prevents a channel, control unit, or dynamic-switch port from receiving any information, other than possibly some specific sequences, from the link. The errors or conditions that result in a link failure being recognized may also result in a link error being detected elsewhere on a channel path.

10.5.1 Conditions that cause link failure

A link failure shall be recognized by a link-level facility or dynamic-switch port when:

a) A loss-of-signal or loss-of-sync condition has persisted for more than the link-interval duration;

b) The not-operational sequence is recognized;

c) A connection-recovery timeout occurs;

d) While in the transmit-OLS state, a timeout is recognized while awaiting the appropriate response;

e) While in the wait-for-offline-sequence state, either a UD or UDR sequence is recognized.

10.5.2 Link-failure procedure

A link-level facility or dynamic-switch port that is in other than the static state can enter one of two link-failure states: transmit-OLS state or transmit-NOS state. A link-level facility or dynamic-switch port that recognizes the not-operational sequence shall enter the transmit-OLS state and shall transmit the offline sequence. A link-level facility or dynamic-switch port shall enter the transmit-NOS state and shall transmit the not-operational sequence if one of the following occurs:

a) A valid response to a transmitted sequence is not recognized within the appropriate interval duration while in the transmit-OLS state or any of the connection-recovery states (see 8.5 and 8.6.);

b) A sequence is recognized which is not appropriate for the conditions present while in the wait-for-offline state;

c) A link-signal error has persisted for greater than the link-interval duration while in any state other than the transmit-NOS state or any of the offline states.

If conditions for entering the static state occur while the dynamic-switch port is in one of the link-failure states, then the switch port shall enter the static-transmit-offline-sequence (static-transmit-OLS) state.

10.5.2.1 Transmit-OLS state

When a link-level facility or dynamic-switch port is in the transmit-offline-sequence (transmit-OLS) state, it shall transmit the offline sequence continuously until one of the following conditions is satisfied:

a) The UD sequence is recognized, causing the link-level facility or the dynamic-switch port to enter the connection-recovery UD-reception state;

b) The offline sequence is recognized, causing the link-level facility or the dynamic-switch port to enter the offline-reception state;

c) A loss-of-signal or loss-of-sync condition is detected that has persisted for longer than the link-interval duration, causing the link-level facility or the dynamic-switch port to enter the transmit-NOS state;

d) The link-interval duration has elapsed since the not-operational sequence was no longer recognized and none of the preceding conditions were satisfied, causing the link-level facility or the dynamic-switch port to enter the transmit-NOS state.

If conditions for entering the offline-transmission state occur while the link-level facility or the dynamic-switch port is in the transmit-OLS state, then the link-level facility or switch port shall enter the offline-transmission state.

If conditions for entering the static state occur while the dynamic-switch port is in the transmit-OLS state, then the switch port shall enter the static-transmit-OLS state.

10.5.2.2 Transmit-NOS state

When a link-level facility or dynamic-switch port is in the transmit-not-operational-sequence (transmit-NOS) state, it shall transmit the not-operational sequence continuously until one of the following conditions is satisfied:

a) The offline sequence is recognized, causing the link-level facility or the dynamic-switch port to enter the offline-reception state;

b) The not-operational sequence is recognized, causing the link-level facility or the dynamic-switch port to enter the transmit-OLS state.

If conditions for entering the offline-transmission state occur while the link-level facility or the dynamic-switch port is in the transmit-NOS state, the link-level facility or switch port shall enter the offline-transmission state.

If conditions for entering the static state occur while the dynamic-switch port is in the transmit-NOS state, then the switch port shall enter the static-transmit-OLS state.

10.6 Link abort

Link abort shall be recognized when a malfunction or data-integrity problem occurs during the transmission of a frame. When link abort is recognized, the transmission of the frame shall be terminated, and the abort delimiter shall be sent, if possible (see 8.3.3). Link abort causes the transmission of a frame to be terminated, regardless of the number of remaining transmission characters to be sent. When the condition causing the link abort occurs, serialization and transmission of the current transmission character shall be completed, if possible, before the abort delimiter is sent. If the cause of link abort is such that transmission-character boundaries cannot be maintained on the link, the abort delimiter may or may not be sent, depending on the model.

When the abort delimiter is used to terminate transmission of a frame that started with a passive-SOF delimiter, the frame that was terminated can be resent no more than two additional times because of successive link aborts. When the abort delimiter is used to terminate transmission of a frame that had started with a connect-SOF delimiter, the link-level facility that aborted the frame shall initiate the connection-recovery procedure (see 11.1.1 and 14.1.3).

When the abort delimiter is detected during frame reception, the frame shall be discarded, and CRC checking shall be terminated. However, if the frame had started with a valid connect-SOF delimiter, then a dynamic connection may have been made as required for the frame; this connection would not have been removed as the result of the abort delimiter; if a frame is received that starts with a connect-SOF delimiter and ends with an abort delimiter, connection recovery shall be performed.

If the dynamic connection was not made because the dynamic switch determined that the dynamic connection was not permitted or the destination dynamic-switch port was not available for connection, the dynamic switch may send a port-reject frame even if the frame was terminated by the abort delimiter.

If frame reception is terminated with an abort delimiter, no link-level-reject frame shall be sent in response. The recipient of a frame terminated with the abort delimiter shall not necessarily be capable of accepting the immediate retransmission of the frame nor continuing any operation that was in progress when the abort delimiter was received. When a frame is received with an abort delimiter and the conditions present prevent the recipient from accepting a retransmission of the frame, the recipient shall wait for the retransmission of the frame, an appropriate frame that causes the operation to be terminated, or a timeout, before any action is taken. If the subsequent frame received does not cause the operation to be terminated, then the recipient can initiate termination of the operation. The action to be taken and the frames sent depend on the conditions present at the recipient.

10.7 Connection

A *connection* to a channel can be initiated by either the channel or control unit. A connection is initiated by sending an initiation frame, that is, a frame with a connect-SOF delimiter. After a connection is initiated, the sender of the initiation frame shall consider the connection to exist only when a response is received with a passive-SOF delimiter and a passive-EOF delimiter. During the interval between initiation of the connection and recognition of the response, the connection is considered *pending*. If the response frame contains a disconnect-EOF delimiter, a connection shall not exist, and the pending connection shall be considered removed.

When either a channel or control unit receives a device-level initiation frame, the recipient shall consider the connection pending until a response is sent. If the response uses a passive-EOF delimiter, a connection shall be considered to exist. If the response uses a disconnect-EOF delimiter, a connection shall not exist, and the pending connection shall be considered removed.

All link-control functions and some device-level functions do not require a connection and can be performed during a pending connection. These functions use frames that initiate a connection, but the required response to these initiation frames shall use a disconnect-EOF delimiter.

Some device-level operations and functions require a connection. When a connection does not already exist for the particular operation or function, frames that initiate a connection shall be used, and the normal response to these initiation frames shall use a passive-EOF delimiter.

If during a pending connection caused by an initiation frame having been sent or received, an initiation frame is either received or sent respectively, two pending connections are recognized: one for the initiation frame sent and one for the initiation frame received. Only one of the pending connections can result in a connection, and when that occurs, the response sent or received, as appropriate, for the other pending connection must remove that pending connection; if both of the connections result in connection, a connection error shall be recognized (see 9.1).

The receipt of two initiation frames without the recipient having sent an intervening response frame shall not cause two pending connections to be recognized. A pending connection shall be recognized for the first initiation frame received, and a connection error shall be recognized for the second initiation frame received (see 11.1.1).

When a connection or pending connection does not exist, the receipt of a frame other than an initiation frame shall not cause a connection or pending connection to be recognized but shall cause a connection error to be recognized (see 11.1.1).

10.8 Link-level-facility states

When performing the management functions of the attached link, a link-level facility can be in any one of the following mutually exclusive states:

a) Inactive;

b) Working;

c) Connection recovery:

  1) UD-transmission;

  2) UD-reception;

  3) UDR-reception;

d) Link failure:

  1) Transmit OLS;

  2) Transmit NOS;

e) Offline:

  1) Offline transmission;

  2) Offline reception;

  3) Wait-for-offline sequence.

The state of the link-level facility shall determine the actions to be taken when specific events occur. These events include information received from the link, information to be transmitted on the link, and link-level-facility-initiated operations.

The following subclauses provide a description of each link-level-facility state and the events that affect the state, including conditions for leaving that state.

10.8.1 Link-level facility inactive

The link-level facility is in the inactive state when it is not in any of the other states. When the link-level facility is in the inactive state, the link-level facility shall transmit idle characters on the link and can initiate monolog communication. The link-level facility can enter the inactive state from the working state or the connection-recovery UD-reception or UDR-reception state.

When the process of frame reception or frame transmission is completed and monolog communication is considered to exist, the link-level facility shall change from the inactive state to the working state.

When a link-level facility determines that it is temporarily unable to establish a two-way communication path, that is, the link-level facility is link busy, it shall change from the inactive state to the working state.

When the link-level facility determines that it is to be taken offline and initiates the offline operation, the link-level facility shall change from the inactive state to the offline-transmission state.

When the link-level facility determines that connection recovery is to be performed and initiates the connection-recovery operation, the link-level facility shall change from the inactive state to the connection-recovery UD-transmission state.

When the link-level facility recognizes the offline sequence, the link-level facility shall change from the inactive state to the offline-reception state.

When the link-level facility recognizes the not-operational sequence, the link-level facility shall change from the inactive state to the link-failure transmit-OLS state.

When the link-level facility recognizes the UD sequence, the link-level facility shall change from the inactive state to the connection-recovery UD-reception state.

When the link-level facility recognizes the UDR sequence, the link-level facility shall change from the inactive state to either the connection-recovery UDR-reception state or to the connection-recovery UD-transmission state.

When the link-level facility determines that a link-signal error has persisted for longer than the link-interval duration, the link-level facility shall change from the inactive state to the link-failure transmit-NOS state.

10.8.2 Link-level facility working

The link-level facility is in the working state when the link-level facility has performed the process of either frame reception or frame transmission and considers that monolog communication or dialog communication has occurred, or the link-level facility is link busy. Monolog communication is considered to exist when a two-way communication path is being established by the initial transmission or reception of an initiation frame. Dialog communication is considered to exist when a two-way communication path is established by the proper frame transmission or frame reception to a received or transmitted initiation frame, respectively. When the link-level facility considers the link to be in the working state, the link-level facility shall transmit idle characters on the link when it is not transmitting frames on the link. The link-level facility can enter the working state only from the inactive state.

When the link-level facility does not have a monolog-communication or dialog-communication condition and the link-level facility is not link busy, the link-level facility shall change from the working state to the inactive state.

When the link-level facility determines that connection recovery is to be performed and initiates the connection-recovery operation, the link-level facility shall change from the working state to the connection-recovery UD-transmission state.

When the link-level facility recognizes the offline sequence, the link-level facility shall change from the working state to the offline-reception state.

When the link-level facility recognizes the not-operational sequence, the link-level facility shall change from the working state to the link-failure transmit-OLS state.

When the link-level facility recognizes the UD sequence, the link-level facility shall change from the working state to the connection-recovery UD-reception state.

When the link-level facility recognizes the UDR sequence, the link-level facility shall change from the working state to either the connection-recovery UDR-reception state or to the connection-recovery UD-transmission state.

When the link-level facility determines that a link-signal error has persisted for longer than the link-interval duration, the link-level facility shall change from the working state to the link-failure transmit-NOS state.

10.8.3 Link-level-facility link failure

The link-level facility is in the link-failure state when the link-level facility recognizes an appropriate link-failure condition on the attached link. If the link-level facility is in the offline-transmission state and is not interrogating the information received on the link (that is, the link-level facility is not prepared to leave the offline-transmission state), then a link-failure condition is not recognized. There are two substates associated with the link-failure state:

a) Transmit OLS;

b) Transmit NOS (see 10.5).

The link-level facility can enter the link-failure state from any other state. A link-level facility shall leave the link-failure state when (1) no link-failure condition is recognized on its attached link, or (2) a condition is recognized which causes the link-level facility to enter the offline state.

10.8.4 Link-level-facility connection recovery

The link-level facility is in the connection-recovery state when the connection-recovery procedure is being performed (see 10.4).

A condition that causes connection recovery is not recognized by a link-level facility in:

a) The offline-transmission state and the link-level facility is not interrogating the information received on the link (that is, the link-level facility is not prepared to leave the offline-transmission state);

b) The offline wait-for-OLS state;

c) The link-failure transmit-NOS state.

There are three substates associated with the connection-recovery state:

a) UD-transmission state;

b) UD-reception state;

c) UDR-reception state (see 10.4.2).

The link-level facility can enter the connection-recovery state from any other state. A link-level facility shall leave the connection-recovery state when (1) the connection-recovery protocols described in 10.4.2 are satisfied, or (2) a condition is recognized which causes the link-level facility to enter either the offline or the link-failure state.

10.8.5 Link-level-facility offline

The link-level facility is in the offline state when the offline procedure is being performed (see 10.3). Specifically, the link-level facility shall enter the offline state when (1) the offline sequence is received and recognized from the link, or (2) the link-level facility has initiated an offline operation by causing the transmission of the offline sequence on the link. There are three substates associated with the offline state:

a) Offline-transmission;

b) Offline-reception;

c) Wait-for-offline-sequence (see 10.3.2).

The link-level facility can enter the offline state from any other state. A link-level facility shall leave the offline state when the offline protocols described in 10.3.2 are satisfied.

10.9 Dynamic-switch-port states

A dynamic-switch port can be in any of one of the following mutually exclusive states:

a) Inactive;

b) Port link busy;

c) Connection:

   1) Monolog-S;

   2) Monolog-D;

   3) Dialog-2;

   4) Dialog-1;

d) Connection recovery:

   1) UD-transmission;

   2) UD-reception;
   3) UDR-reception;
- e) Link failure:
   1) Transmit OLS;
   2) Transmit NOS;
- f) Offline:
   1) Offline transmission;
   2) Offline reception;
   3) Wait-for-offline sequence;
- g) Static:
   1) Transmit OLS;
   2) Pass-through.

The state of a dynamic-switch port shall determine the actions to be taken when specific events occur on the link or at other switch ports.

The following subclauses provide a brief description of each switch-port state and the conditions for entering or leaving that state.

Notes

1 The term *initiate-connection control*, as used in the following subclauses, refers to a connect-SOF delimiter followed by any valid transmission character.

2 The term *connected-switch-port*, as used in the following subclauses, refers to a switch port which is connected to the switch port under discussion.

10.9.1 Port inactive

The dynamic-switch port is in the inactive state when it is not in any of the other states. In the inactive state, a dynamic-switch port can accept or initiate the establishment of a dynamic connection with another dynamic-switch port.

A switch port may enter the inactive state only from the link-busy state unless another special condition is detected. The switch port shall transmit idle characters on the link in this state.

If a switch port in the inactive state receives an initiate-connection control and subsequent data from the link and (1) the data contains valid destination and source link addresses, (2) the switch port associated with the destination link address is in the inactive state, and (3) a dynamic connection with that switch port is permitted, then a dynamic connection with that switch port shall be established. If either link address in the received data is invalid or if the switch port associated with the destination link address is not in the inactive state or is otherwise unavailable for dynamic connection, a port-reject frame or port-busy response frame, as appropriate, shall be returned on the link.

A switch port shall leave the inactive state when a dynamic or static connection is established, a link error or link failure is recognized, a sequence is recognized, a port failure is detected, or the switch port is taken offline.

10.9.2 Port link busy

When a dynamic-switch port is in the link-busy state because of internal reasons, the switch port is unable to accept or initiate the establishment of a dynamic connection. The link-busy state is a normal occurrence at the dynamic-switch port and is considered to be temporary; the switch port is still considered to be usable. The link-busy state shall be used as a transition state before returning to the inactive state and may be entered for internal reasons, for example, to facilitate the recording of a detected error condition.

The link-busy state may be entered from any other state except the dialog-2 state, the UD-transmission state, the link-failure state, the offline state, and the static state. The switch port shall transmit idle characters on the link while in the link-busy state unless a frame with a connect-SOF delimiter is received from the link, in which case a port-busy frame shall be returned with a source-port-busy-unconnected reason code indicated.

A switch port shall leave the link-busy state when no busy condition exists, a static connection is established, a link-signal error or link failure is recognized, a sequence is recognized, a port failure is detected, or the switch port is taken offline.

10.9.3 Port connection

A dynamic-switch port may be caused to enter the connection state when a connect-SOF delimiter followed by a valid transmission character is received from a link. There are four substates associated with the connection state:

- a) Monolog-source (monolog-S);

b) Monolog-destination (monolog-D);

c) Dialog 2;

d) Dialog 1.

When a switch port is in the connection state, a *dynamic-switch-idle timeout* shall be recognized if a switch port receives nothing but idle characters for 30 seconds or more. When a dynamic-switch-idle timeout occurs, a connection error shall be detected (see 11.1.1).

10.9.3.1 Monolog-S

When a dynamic-switch port in the inactive state has received an initiate-connection control, causes a dynamic connection to be established accordingly, and has started the transfer of the initiate-connection control to the connected-switch-port, the inactive switch port shall enter the monolog-source (monolog-S) state.

The monolog-S state may be entered only from the inactive state. The switch port shall transmit idle characters on the link in the monolog-S state.

When a switch port in the monolog-S state receives a frame from a dynamically connected-switch-port, it shall enter the dialog-1 state if the frame has a passive-SOF delimiter or shall enter the dialog-2 state if the frame has a connect-SOF delimiter. When a switch port in the monolog-S state receives a frame from the link, the frame shall be passed to the dynamically connected-switch-port, and the switch port shall remain in the monolog-S state; if a disconnect-EOF delimiter is detected, the switch port shall enter the link-busy state.

A switch port shall leave the monolog-S state when any of the following occurs:

a) A frame is received from the dynamically connected-switch-port;

b) A disconnect-EOF delimiter is detected;

c) A static connection is established;

d) A link-signal error or link failure is recognized;

e) A sequence is recognized;

f) A dynamic-switch-idle timeout is recognized;

g) A port failure is detected;

h) The switch port is taken offline;

i) A condition associated with the connected-switch-port requires removal of the dynamic connection.

10.9.3.2 Monolog-D

When a dynamic-switch port in the inactive state determines that a dynamic connection with another switch port has been established at the initiative of the other switch port and the switch port starts receiving an initiate-connection control from the other switch port, it shall enter the monolog-destination (monolog-D) state.

The monolog-D state may be entered only from the inactive state. In this state, the switch port shall transmit idle characters on the link when it is not transmitting frames received from the connected-switch-port. Frames received from the connected-switch-port shall be transmitted on the link without a change of state. When a disconnect-EOF delimiter is detected in information passed to the link, the switch port shall enter the link-busy state. If a switch port in the monolog-D state receives an initiate-connection control with subsequent data from the link with a destination link address associated with a switch port other than that with which a dynamic connection exists, a port-busy frame with a source-port-busy monolog-D reason code shall be returned. If the destination link address matches the link address associated with the dynamically connected-switch-port, the switch port shall enter the dialog-2 state and shall pass the frame to the connected-switch-port. If a switch port in the monolog-D state receives a frame with a passive-SOF delimiter from the link, the switch port shall enter the dialog-1 state and shall transmit the frame to the dynamically connected-switch-port.

A switch port shall leave the monolog-D state when any of the following occurs:

a) A frame is received with a connect-SOF delimiter containing a destination link address corresponding to the connected-switch-port;

b) A frame containing a passive-SOF delimiter is received from the link and passed to the connected-switch-port;

c) A disconnect-EOF delimiter is received from the connected-switch-port;

d) A static connection is established;

e) A link-signal error or link failure is recognized;

f) A sequence is recognized;

g) A dynamic-switch-idle timeout is recognized;

h) A port failure is detected;

i) The switch port is taken offline;

j) A condition associated with the connected-switch-port requires removal of the dynamic connection.

A switch port in the monolog-D state that receives a frame with a connect-SOF delimiter followed by idle characters is permitted to discard these idle characters while servicing the request for a connection and determining the dynamic-switch-port state in which to transition. If the frame with the connect-SOF delimiter causes the switch port to enter the dialog-2 state and another frame immediately follows on the link at the same switch port, the switch port must ensure a minimum of two idle characters between these two frames is passed to the connected-switch-port.

NOTE – The preferred implementation is one that either avoids or minimizes the discarding of idle characters by employing design techniques which minimizes both the service time for a frame with a connect-SOF delimiter and the effect of abnormal conditions on the switch-port service rate.

10.9.3.3 Dialog-2

A switch port is caused to enter the dialog-2 state when a dynamic connection is established between two switch ports, each of which has received an initiate-connection control from the link which requires that a dynamic connection be established with the other switch port. Specifically, (1) when a dynamic-switch port in the monolog-D state receives an initiate-connection control and subsequent data from the link with a destination link address associated with the switch port with which it is connected, the initiate-connection control is passed to the connected-switch-port and the monolog-D switch port shall enter the dialog-2 state, or (2) when a dynamic-switch port in the monolog-S state receives an initiate-connection control passed through the dynamic connection, the monolog-S switch port shall enter the dialog-2 state and shall transmit the initiate-connection control on its link.

The dialog-2 state may be entered only from the monolog-S or monolog-D state. In this state, the switch port shall transmit idle characters on the link when it is not transmitting frames received from the connected-switch-port. Frames received from the connected-switch-port shall be transmitted on the link and frames received from the link shall be passed to the connected-switch-port without a change of state. When a disconnect-EOF delimiter is detected in information received from or passed to the link, the switch port shall enter the dialog-1 state.

A switch port shall leave the dialog-2 state when any of the following conditions occurs:

a) A disconnect-EOF delimiter is detected;

b) A static connection is established;

c) A link-signal error or link failure is recognized;

d) A sequence is recognized;

e) A dynamic-switch-idle timeout is recognized;

f) A port failure is detected;

g) The switch port is taken offline;

h) A condition associated with the connected-switch-port requires removal of the dynamic connection.

10.9.3.4 Dialog-1

A switch port with a dynamic connection to another switch port is caused to enter the dialog-1 state when a frame with a passive-SOF delimiter, in response to an initiate-connection control which caused a dynamic connection to be established, is received by the switch port from the link or is passed through the dynamic connection to the switch port. Specifically, (1) when a dynamic-switch port in the monolog-D state receives a frame with a passive-SOF delimiter from the link, the frame is passed to the connected-switch-port through the dynamic connection and the switch port shall enter the dialog-1 state or (2) when a dynamic-switch port in the monolog-S state receives a frame with a passive-SOF delimiter passed through the dynamic connection, the switch port shall enter the dialog-1 state and shall transmit the frame on its link. Each of two dynamically connected-switch-ports shall also be caused to enter the dialog-1 state if, while in the dialog-2 state, a frame with a disconnect-EOF delimiter is passed through the dynamic connection.

The dialog-1 state may be entered only from the monolog-S, monolog-D, or dialog-2 state. In the dialog-1 state, the switch port shall transmit idle characters on the link when it is not transmitting frames received from the connected-switch-port. Frames received from the connected-switch-port shall be transmitted on the link and frames received from the link shall be passed to the connected-switch-port without a change of state. When a disconnect-EOF delimiter is detected in information received from or passed to the link, the switch port shall enter the link-busy state.

A switch port shall leave the dialog-1 state when any of the following conditions occurs:

a) A disconnect-EOF delimiter is detected;

b) A static connection is established;

c) A link-signal error or link failure is recognized;

d) A sequence is recognized;

e) A dynamic-switch-idle timeout is recognized;

f) A port failure is detected;

g) The switch port is taken offline;

h) A condition associated with the connected-switch-port requires removal of the dynamic connection.

10.9.4 Port link failure

The dynamic-switch port is in the link-failure state when the dynamic-switch port recognizes an appropriate link-failure condition on the attached link (see 10.5).

A link failure is not recognized for a dynamic-switch port for the following conditions:

a) The switch port is in the static state;

b) The switch port is in the offline-transmission state and is not interrogating the information received on the link.

There are two substates associated with the link-failure state:

a) Transmit OLS;

b) Transmit NOS (see 10.5).

If the switch port has a dynamic connection with another switch port when it enters the link-failure state, the other switch port is caused to enter the connection-recovery state and shall initiate transmission of the UD sequence. The dynamic connection shall be removed.

The link-failure state may be entered from any other state. A dynamic-switch port shall leave the link-failure state when (1) no link-failure condition is recognized on that switch port’s link, or (2) a condition is recognized which causes the switch port to enter the offline or static state.

When a switch port to which an identified link-level facility is attached enters the link-failure state, the dynamic-switch control unit is given initiative to report the state change (see 9.1.13).

10.9.5 Port connection recovery

The dynamic-switch port shall be in the connection-recovery state when the connection-recovery procedure is being performed (see 10.4.2).

A condition that causes connection recovery is not recognized by a dynamic-switch port in:

a) The static state;

b) The offline-transmission state and the switch port is not interrogating the information received on the link (that is, the switch port is not prepared to leave the offline-transmission state);

c) The offline wait-for-OLS state;

d) The link-failure transmit-NOS state.

There are three substates associated with the connection-recovery state:

a) UD-transmission state;

b) UD-reception state;

c) UDR-reception state (see 10.4.2).

If the switch port has a dynamic connection with another switch port when it enters the connection-recovery state, the connected-switch-port is caused to enter the connection-recovery state and shall initiate transmission of the UD sequence. The dynamic connection shall be removed.

A dynamic-switch port shall leave the connection-recovery state when (1) the connection-recovery protocols described in 10.4.2 are satisfied, or (2) a condition is recognized which causes the switch port to enter the link-failure, offline, or static state.

If another dynamic-switch port receives a request to form a dynamic connection with a switch port that is in the connection-recovery state, a port-busy frame indicating the destination-port-busy reason code shall be returned in response to the request.

10.9.6 Port-offline state

A dynamic-switch port is caused to enter the offline state as a result of transmitting or recognizing the offline sequence unless the switch port is in the static state (see 10.3.1). There are three substates associated with the offline state:

a) Offline-transmission state;

b) Offline-reception state;

c) Wait-for-offline-sequence (see 10.3.2).

When a switch port to which an identified link-level facility is attached enters the offline state, the dynamic-switch control unit is given initiative to report the state change. (see 9.1.13).

If the switch port has a dynamic connection with another switch port when it enters the offline state, the other switch port is caused to enter the connection-recovery state and shall initiate transmission of the UD sequence. The dynamic connection shall be removed.

The offline state may be entered from any other state. A dynamic-switch port shall leave the offline state when (1) the offline protocols described in 10.3.2 are satisfied, or (2) a condition is recognized which causes the switch port to enter the static state.

10.9.7 Static state

A dynamic-switch port is caused to enter the static state when a static connection with that switch port is established using local or remote facilities provided by the dynamic-switch control unit. (A request to establish a static connection shall be denied only if one of the designated dynamic-switch ports is not implemented, not installed, or already in a static connection, or the switch port is associated with the dynamic-switch control unit; other switch-port states shall not prevent the establishment of a requested static connection.) There are two substates associated with the static state:

a) Static-transmit-OLS state;

b) Static-pass-through state.

If the switch port has a dynamic connection with another switch port when it enters the static state, the other switch port is caused to enter the connection-recovery state and shall initiate transmission of the UD sequence. The dynamic connection shall be removed.

The static state may be entered from any other state. When a switch port to which an identified link-level facility is attached enters the static state, the dynamic-switch control unit is given initiative to report the state change (see 9.1.13).

A switch port shall leave the static state when the static connection with that switch port is removed by using the local or remote facilities provided by the dynamic-switch control unit. (A request to remove an existing static connection shall always be honored.)

The static state of the dynamic-switch port shall not be affected by any information received from the link or from the statically connected dynamic-switch port. If a sequence is received by one of two statically connected-switch-ports, the received sequence shall normally be retransmitted on the connected-switch-port’s link. Frames may be received and transmitted simultaneously by statically connected-switch-ports. When a switch port is statically connected, it shall not generate dynamic-switch port-reject frames or dynamic-switch port-busy frames.

The definition of a static connection that appears in this document applies only when the static connection is used with the SBCON I/O-interface architecture; unpredictable results occur if a static connection is used with other protocols.

NOTE – When a static connection is established with a dynamic-switch port, any existing dynamic connection with that switch port shall be removed. Therefore, before a static connection is established, system activity with the affected dynamic-switch ports should be quiesced. Similarly, when a static connection is removed, activity at both switch ports involved in the static connection should be quiesced before the connection is removed.

10.9.7.1 Static-transmit-OLS state

The static-transmit-offline-sequence (static-transmit-OLS) state is used during the transition to the static-connection state.

A dynamic-switch port is caused to enter the static-transmit-OLS state when a static connection with that switch port is established, or when either that switch port or the other switch port participating in the static connection (which is also in the static-pass-through state) is taken offline. A switch port in the static-transmit-OLS state shall ignore anything received.

The static-transmit-OLS state may be entered from any other state. A switch port shall leave the static-transmit-OLS state and shall enter the static-pass-through state after the switch port has transmitted the offline sequence for at least 1 second, provided neither of the switch ports participating in the static connection has been taken offline. When the static connection is removed while the switch port is in the static transmit-OLS state, the switch port shall change to the offline-transmission state.

10.9.7.2 Static-pass-through state

The static-pass-through state provides information exchange between two switch ports that are connected through the static connection. The static-pass-through state may be entered only from the static-transmit-OLS state. A switch port in the static-pass-through state shall pass through anything received, including OLS or loss of signal, without a state change.

A switch port shall leave the static-pass-through state and shall enter the static-transmit-OLS state if either of the switch ports participating in the static connection is taken offline. When the static connection is removed while the switch port is in the static-pass-through state, the switch port shall change to the offline-transmission state.

10.10 Link timeout

The link timeout is used to detect errors associated with the protocols used for sending and receiving frames. The link timeout establishes both the duration of the waiting period and the instant that an error condition can be recognized. A control unit shall wait between 400 and 850 milliseconds for a channel response, and a channel shall wait at least 1 second for a control-unit response.

When an initiation frame is sent, the link timer shall be started. If the normal response or one of the alternate responses for the particular request is not received, one of the following shall occur:

a) If no response or sequence is received, the link timer expires, and a connection error is detected (see 11.1.1);

b) If a frame with a disconnect-EOF delimiter and an error is received, the timer is reset;

c) If a frame with a connect-SOF delimiter and an error other than a connection error is received, the link timer is not reset, and the recovery action for the frame received is started;

d) If a condition that causes a sequence to be sent occurs, the link timer is reset.

The duration of the link timeout is also the duration of the device-level timeout (see 13.3.5).

10.11 Link-busy condition

A link-busy condition is a temporary busy condition. When a link-busy condition exists at a channel, control unit, or dynamic-switch port it shall prevent the reception of a frame that is initiating a connection. When this condition exists at a channel or control unit, the link-busy condition may cause the frame to be discarded and a link-level-busy response to be sent (see 9.1.10). When this condition exists at a dynamic-switch port, the link-busy condition may cause the frame to be discarded and a port-busy response to be sent (see 9.1.11). The link-busy condition at a dynamic-switch port shall prevent the routing of a frame to its intended destination.

The following are examples of conditions for which a link-level-busy response is indicated:

a) A shared facility necessary for the reception of a frame is temporarily unavailable;

b) A self-initiated function is being performed.

The following are examples of conditions for which a port-busy response is indicated:

a) The required destination switch port is dynamically connected to another dynamic-switch port;

b) The source switch port is already connected to a dynamic-switch port other than the required destination switch port.

Link-busy conditions are not error conditions but can result in a link error being detected, depending on when they occur.

If a link-level-busy or port-busy response is received in response to a frame sent when a connection to the intended recipient of the frame was considered to already exist, or if these responses are received when a frame was sent which removed an existing connection, a link error shall be detected.

Frames initiating a connection that encounter a link-busy condition, other than source-port-busy monolog-D, may be retried until the frame is successfully transferred as indicated by the appropriate response. The number of retries is model dependent.

11 Link and dynamic-switch-port error conditions

This clause describes the types of link errors which can be detected by a link-level facility of a channel, a link-level facility of a control unit, or a dynamic-switch port.

Link errors may be *detected* any time that the link is operational. More than one link error or dynamic-switch-port error may be detected simultaneously.

If the link error causes action to be taken as a result of the error, the error is said to be *recognized*. An error that is detected may not always be recognized; it may be recognized based on conditions present when the error is detected.

11.1 Link errors

Link errors are abnormal conditions that affect the integrity of the link and the information being transmitted or received over that link. Link errors may be associated with the conditions that are necessary in order for the link to be considered operational, or they may be associated with the link-level functions and protocols which affect the integrity of the information being sent or received. These errors may affect the capability of a channel, control unit, or dynamic switch to route information to the appropriate destination, or they may prevent valid reconstruction of the information from the serial transmission.

Link errors are caused by transient noise or malfunction in the channel, control unit, or dynamic switch, or by external means (for example, powering off, system initialization, or removal of a cable). Link errors may also be caused by a failed or failing link.

If an abnormality is associated with one of the following, it shall be considered a link error:

- the link header or link trailer of a frame;
- the validity of transmission characters or the use of transmission characters;
- the integrity of the link or the signal sent over the link;
- the basic frame structure;
- the link address or logical address used;
- the link protocols for sending and receiving a link-control or device frame.

Link errors shall be identified according to the specific type of abnormality and what is affected by the abnormality. For example, an abnormality associated with a transmission character is called a code-violation error. See the following subclauses and 11.2 for a description of each link error.

When a link error is detected, it may or may not result in a link-error condition being recognized, depending on the type of link error and the conditions present at the time the error is detected. A link-error condition shall be recognized when the link error detected is associated with the contents of a frame or with the link protocols for performing a link-control function. For example, a code-violation error detected in the contents of a frame results in a link-error condition being recognized, but a code-violation error detected before frame reception is started does not result in a link-error condition being recognized (see the individual descriptions of the link errors in the following subclauses for additional information on when a link-error condition is recognized). If a link-error condition is recognized, the appropriate action shall be taken in order to provide for recovery. If a link-error condition is not recognized, no action shall be taken.

When a link-error condition is associated with a frame that is initiating a connection, a link-level-reject or port-reject frame shall be sent in response to the source of the frame, provided all of the conditions for sending the respective frames are satisfied. If a link-level-reject or port-reject frame is sent, a code in the frame called the reject-reason code shall identify the link error that caused the link-level-reject or port-reject frame to be sent (see 9.1.8 and 9.1.9). The conditions that determine whether a link-level-reject or port-reject frame is sent shall include all of the following:

a) the type of link error recognized;

b) the frame delimiters received;

c) the conditions present in the channel or control unit at the time of the error.

The link-level facility that receives a link-level-reject frame or port-reject frame as a response shall be responsible for determining what action or recovery to take.

When a link-level-reject frame or port-reject frame is received by a channel or control unit that is waiting for a response to a frame, a link timeout shall be prevented, and a link-error condition shall be recognized. The link-error condition recognized shall be identified by the information contained in the link-level-reject frame or port-reject frame.

A dynamic-switch port checks only for certain link errors. When making a dynamic connection, a dynamic-switch port shall check for certain link errors that affect the SOF delimiter, the destination link address, and the source link address. When the frame that caused the dynamic switch to make the dynamic connection is sent to its final destination, the recipient shall check for all link errors, including CRC check, independent of the checking performed by the dynamic-switch port. The integrity of the dynamic connection is therefore checked to some extent at the dynamic-switch port and completely checked by the recipient. After the connection is made, the dynamically connected dynamic-switch ports shall monitor for those link errors that affect the operational state of the links and their respective dynamic-switch ports. Errors that are detected by a dynamic-switch port are:

a) Connection error;

b) Link-signal error;

c) Code-violation error;

d) Sequencing error;

e) Address-invalid error;

f) Undefined-destination-address error;

g) Destination-port malfunction;

h) Dynamic-switch-port intervention required.

A link error, when it occurs, may be accompanied by other link errors. During frame reception, if a link error is detected by the recipient, checking for other link errors shall continue until frame reception is ended. When frame reception is ended, the frame shall be discarded and, if a link-level-reject frame is permitted, the highest priority link error detected shall be reported in the link-level-reject frame (see 9.1.8). In many cases the nature of a higher priority error becomes the cause for the presence of a lower priority error in the same frame.

For a channel or control unit, link errors shall fall in one of two categories, link errors received that are detected without CRC checking and link errors received that are detected only after CRC checking is performed for the frame. These categories shall be presented in descending order of priority, and, within a category, link errors shall be presented in descending priority; that is, a link error that is detected without CRC checking is considered a higher priority error than a link error that is detected after CRC checking is performed, and, within the latter category, a CRC error is the highest priority error.

a) Link errors detected without CRC checking;

  1) Connection error;

  2) Link-signal error;

  3) Code-violation error;

  4) Sequencing error;

b) Link errors detected only after CRC checking.

  1) CRC error;

  2) Destination-address-invalid error;

  3) Link-protocol error;

  4) Logical-path-not-established error.

The term *transmission error* shall be used when referring to a link error that can be caused by a link-signal error, code-violation error, sequencing error, or CRC error.

11.1.1 Connection error

A connection error shall be detected when a condition that affects the connection state maintained by the link-level facility or the dynamic-switch port for a link is detected. In most cases a connection error results from the improper use of delimiters. The following conditions cause a connection error to be detected by a link-level facility:

a) Recognition of an unconditional-disconnect sequence during a pending or existing connection;

b) Recognition of an unconditional-disconnect-response sequence, other than in response to either an unconditional-disconnect sequence or a not-operational sequence while in the wait-for-offline-sequence state;

c) Occurrence of a link timeout while a response to a link-level initiation frame is awaited;

d) Reception of an EOF delimiter or abort delimiter before frame reception is started;

e) Recognition of a frame with a passive-SOF delimiter when there is no pending or existing connection;

f) Recognition of a frame with a connect-SOF delimiter when a connection exists;

g) Recognition of a frame with a connect-SOF delimiter and either an abort delimiter or a disconnect-EOF delimiter;

h) Recognition of a frame with a connect-SOF delimiter followed by the recognition of a frame with a connect-SOF delimiter without having sent an intervening response;

i) Recognition of a frame with a connect-SOF delimiter followed by the recognition of a frame with a passive-SOF delimiter without having sent an intervening response, except for the case where a frame with a passive-SOF delimiter is expected in response to a previously sent initiation frame;

j) Recognition of a frame with a passive-SOF delimiter and passive-EOF delimiter in response to a link-control request frame;

k) Recognition of a frame-terminating condition other than an EOF delimiter, abort delimiter, or a defined sequence for a frame received with a passive-SOF delimiter in response to a frame sent with a connect-SOF delimiter;

l) Ending transmission of a frame that started with a connect-SOF delimiter by transmission of an abort delimiter, other than in reaction to recognition of a link failure or a defined sequence;

m) Recognition of an internal condition that causes information about the connection state to be lost.

The following conditions cause a connection error to be detected at a dynamic-switch port:

a) Recognition of an unconditional-disconnect sequence during a connection state;

b) Recognition of an unconditional-disconnect-response sequence other than in response to an unconditional-disconnect sequence or other than in response to a not-operational sequence while in the wait-for-offline-sequence state;

c) Occurrence of a timeout because of inactivity over two dynamically connected dynamic-switch ports. When a dynamic connection exists, timeout is recognized if a port receives nothing but idle sequences for 30 seconds or more;

d) Recognition of an internal condition that causes information about the connection state to be lost.

11.1.2 Link-signal error

A link-signal error shall be detected when one of the following occurs while not in the offline state:

- *loss of signal:* the receiver senses that the level of the signal on the link is below the value required for reliable communication (see clause 5);
- *loss of sync:* the receiver senses that either bit synchronization or transmission-character synchronization with the signal on the link does not exist (see 7.1).

These link errors are detected by both a dynamic-switch port and a link-level facility. A link-signal error can result in a link failure being recognized. If loss of signal or loss of sync is detected for the link-interval duration, the link-signal error shall become a link failure (see 10.5).

Until a link failure is recognized, the effect of the link-signal error on frame reception shall be treated in the same manner as a link error, and the appropriate action for the conditions present is performed. Any action taken as a result of a link-error condition being recognized shall not affect the checking for the link failure, which continues uninterrupted.

When a link-signal error is detected after frame reception is started, frame reception shall be terminated, a link-error condition shall be recognized, and the appropriate action for the conditions present shall be performed. If a link-level-reject frame is appropriate, the transmission-error reject-reason code shall be indicated. If a link-signal error is detected before frame reception is started, a link-error condition shall not be recognized, and no further action (other than the checking for a link failure) shall be performed, except that a dynamic-switch port that enters the loss-of-synchronization state shall perform connection recovery (see 7.1.1).

11.1.3 Code-violation error

A code-violation error shall occur when a received transmission character is recognized as invalid (see clause 6 for the definition of valid transmission characters).

When a code-violation error is detected after frame reception is started, frame reception shall continue until one of the valid conditions for ending frame reception is detected, at which time, a link-error condition shall be recognized for the code-violation error (see 8.4.2). If a link-level-reject frame is appropriate, the transmission-error reject-reason code shall be indicated. If the code-violation error is detected before frame reception is started, a link-error condition shall not be recognized, no recovery action shall be performed, and a link-level-reject frame shall not be sent.

The error which causes a transmission character to be invalid may have occurred previous to the character which is detected as invalid. Therefore, when a code-violation error is detected within a frame, all characters in the frame prior to the character on which the code-violation error is detected shall be assumed to be in error. However the code-violation error is always detected no later than the end of the end-of-frame delimiter of the frame in which the original error occurred.

11.1.4 Sequencing error

A sequencing error shall be detected by a link-level facility or by a dynamic-switch port when the special ordered sets used for a sequence are received, but without the minimum number of consecutive ordered sets required to recognize the sequence (see 7.4.1 and clause 6).

A sequencing error shall be detected by a link-level facility when one of the following occurs:

- a valid but reserved transmission character is detected in the encoded bit stream being received (see table 13);
- a valid transmission character is received which cannot be identified as an idle character, as part of a sequence, or as part of a frame, or a sequence is received in an order that is not permitted for the conditions present (see 7.4.1 and 8.4.2). The following are examples:
  - idle characters are not followed by either an SOF delimiter or a sequence;
  - an SOF delimiter is not followed by a data character;
  - during frame reception, a transmission character is received which is not a data character, is not part of an EOF delimiter, and is not part of an abort delimiter (see 8.4.2);
  - an EOF delimiter or abort delimiter is not followed by idle characters, a sequence, or an SOF delimiter.
- *minimum-frame-size error:* frame reception is ended by an EOF delimiter, and the contents of the frame are less than seven bytes (see 8.4.2 for the conditions that end frame reception).

  When a minimum-frame-size error is detected, a link-error condition shall be recognized, and the appropriate recovery action for the conditions present shall be performed;
- *maximum-frame-size error:* frame reception is ended when the quantity of bytes received for a frame exceeds the maximum frame size that the link-level facility is currently able to receive, and in any case, when the contents of a frame exceed 1,035 bytes.

  For example, if the link-level facility is currently able to receive only frames for which the contents of the frame are less than or equal to 32 bytes and the contents of the frame received exceed 32 bytes, a maximum-frame-size error is detected. A different link-level facility under the same conditions or the same link-level facility at a different instant might be capable of receiving any valid frame size.

When a maximum-frame-size error is detected, a link-error condition shall be recognized, frame reception shall be ended, and the appropriate recovery action for the conditions present shall be performed. Data characters received after frame reception is terminated for maximum-frame-size error shall be discarded with no additional link errors recognized;

- frame reception is ended by the recognition of an SOF delimiter, two consecutive K28.5 characters, or a sequence (see 7.4.1 and 8.4.2);

When frame reception is ended by an SOF delimiter, two consecutive K28.5 characters, or a sequence, a link-error condition shall be recognized, and the appropriate recovery action for the conditions present shall be performed. If reception of the current frame is ended by an SOF delimiter, frame reception for the frame associated with the new SOF delimiter may possibly be started, but only if the SOF delimiter is followed by a valid transmission character;

- a reserved sequence is recognized.

If a link-level-reject frame is sent for a sequencing error, the transmission-error reject-reason code shall be indicated.

11.1.5 CRC error

A CRC error shall be detected by a link-level facility when the CRC result obtained on the contents of the frame just received does not equal the expected value (see 8.3.1).

When a CRC error is detected, a link-error condition shall be recognized, and the appropriate recovery action for the conditions present shall be performed. If a link-level-reject response frame is appropriate, the transmission-error reject-reason code shall be indicated.

11.1.6 Destination-address-invalid error

A destination-address-invalid error shall be detected when an identified link-level facility receives a frame that contains a destination link address with a value other than all ones or the assigned link address. If the link-level facility is not identified, a destination-address-invalid error shall be detected only when a frame with a destination link address of all zeros is received (see 10.1.2.1 and 10.1.3.1).

When a destination-address-invalid error is detected, a link-error condition shall be recognized, and the appropriate recovery action for the conditions present shall be performed. If a link-level-reject frame is sent, the destination-link-address-error reject-reason code shall be indicated, or if a higher priority link error is detected, the reject-reason code for that link error shall be indicated.

11.1.7 Link-protocol error

A link-protocol error shall be detected by a link-level facility when one of the conditions in the list that follows occurs. This list of link-protocol errors, with the exception of connection-protocol error, is not exhaustive but is representative of link-protocol errors. Where a link function or protocol is described in clause 8 and clause 9 the abnormal conditions and whether the abnormal conditions cause a link-protocol error to be recognized are described.

a) *Destination-address error:* a frame is received with a destination link address or destination logical address that is either not permitted or inappropriate for the conditions present (other than as described above for a destination-address-invalid error). The following conditions are included:

1) A destination link address of all ones is used with a link-control function that is not permitted to be sent with the all-ones destination link address. Examples of such functions are establish logical path, logical path established, remove logical path, and logical path removed;

2) A destination link address other than all ones is used with a link-control function that can only be sent with the all-ones destination link address. Examples of such functions are link level busy and port busy;

3) A destination link address of all ones is used with a device frame;

4) The destination logical address in a link-control response frame that requires such an address does not correspond to the source logical address in the link-con-

trol request for which this response is received, as for functions such as:

i) A logical-path-established response to an establish logical path;

ii) A logical-path-removed response to an establish logical path or remove logical path;

5) The destination logical address used in a device frame received either in response to a device frame or during a connection does not correspond to the logical path being used for the existing or pending connection;

6) The destination-logical-address field is not zero for a device frame received by a channel that does not permit channel logical addressing;

b) *Source-address error:* a frame is received with a source link address or source logical address that is either not permitted or inappropriate for the conditions present. The following conditions are included:

1) A source link address of all ones is used in a link-control frame or device frame;

2) A source link address of all zeros is used in a device frame;

3) A source link address of all zeros is used with a link-control function that is not permitted to be sent with the all-zeros source link address. Examples of such functions are establish logical path, logical path established, remove logical path, and reset;

4) A source link address equal to the destination link address is used in either a link-control frame or device frame;

5) The source logical address in a link-control response frame that requires such an address does not correspond to the destination logical address in the link-control request for which this response is received. Examples of such functions are:

i) Logical path established in response to an establish logical path;

ii) Logical path removed in response to an establish logical path or remove logical path;

6) The source logical address in a device frame received either in response to a device frame or during a connection does not correspond to the logical path being used for the current device-level operation;

c) *Reserved-field error:* reserved bits in the link header of a frame or in the information field of a link-control frame are set to a value other than zero;

d) *Acquire-link-address error:* an initiation frame other than an ALA frame is received by an unidentified link-level facility, or an ALA frame is received by a link-level facility of a control unit other than a dynamic-switch control unit;

e) *Link-control-function error:* the link-control function specified in the frame received is inappropriate for the conditions present. The following conditions are included:

1) A link-control response is received when one is not expected;

2) The link-control response received is not appropriate for the link-control request sent;

3) A channel recognizes a link-control function that is not permitted to be specified in a frame sent to a channel, such as establish logical path or remove logical path;

f) *Unrecognized link-control function:* a link-control frame is received specifying a link-control function that is not recognized by the recipient;

g) *Frame-size error:* the link-control frame received is seven bytes or more, and either greater than or less than the number of bytes defined for the link-control function and extended-control bits indicated, or the contents of a device frame received are less than 12 bytes. The following conditions are included:

1) A link-level ACK frame is received with an information field;

2) The information field in an LPR frame contains other than one byte;

h) *Connection-protocol error:*

1) Recognition of a link-level-busy, dynamic-switch-port-busy, link-level-reject, or port-reject frame other than in response to a frame that initiates a connection;

2) Recognition of a port-busy frame with the source-port-busy monolog-D reason code in response to an initiation frame without the receipt of an intervening initiation frame;

3) Recognition of a frame with a passive-SOF delimiter and a source link address different from the link address used to create the existing connection;

4) Recognition of any frame other than a port-busy frame indicating source-port-busy monolog-D when conditions exist that require reception of that indication;

5) Recognition of a frame with a passive-SOF delimiter other than port busy, link busy, link reject, or port reject, and with a source link address different from an assigned link address used to create the existing pending connection;

i) *LPR error:* the reason-code field of a logical-path-removed (LPR) frame contains a reserved value;

j) *Unrecognized device level:* the received device-initiation frame has a device-level-identifier value which does not correspond to a device-level protocol permitted by the recipient;

k) *Mismatched device-level identifiers:* during an existing or pending device-level connection, a device frame is received, but the device-level identifier does not correspond to the device-level identifier being used for the current device-level operation.

When a link-protocol error is detected, a link-error condition shall be recognized, and the appropriate recovery action for the conditions present shall be performed. If a link-level reject frame is sent, the reject-reason code for a reserved-field error, unrecognized link-control function, acquire-link-address error, unrecognized device level, or protocol error shall be indicated; or, if a higher priority link error is detected, the reject-reason code for that link error shall be indicated. The protocol-error reject-reason code shall be used for all link-protocol errors which are not specifically covered by the aforementioned reject-reason codes. When a link-protocol error is recognized because a dynamic connection has been removed by inappropriate use of frame delimiters, no link-level-reject frame shall be sent.

11.1.8 Logical-path-not-established error

When either a channel or a control unit receives a device frame and the destination link address matches the assigned link address, then the combination of the source link address, the source logical address, the destination link address, and the destination logical address shall identify an established logical path; if no logical path is established with that combination of addresses, a logical-path-not-established error shall be detected.

When a logical-path-not-established error is detected, a link-error condition shall be recognized, and the appropriate recovery action for the conditions present shall be performed. If a link-level-reject frame is sent, the logical-path-not-established error reject-reason code shall be indicated, or if a higher-priority link error is recognized, the reject-reason code for that link error shall be indicated.

A logical-path-not-established error, if present, is usually detected when a device frame that initiates a connection is received. This error may also occur because of a malfunction which occurs when the control unit is considered connected, in which case, an error is detected when the next device frame is received, if it has not already been detected as a result of the conditions caused by the malfunction. In the case where the malfunction also causes the connection to be considered removed, other link errors of a higher priority may be detected.

11.2 Dynamic-switch-port errors

A dynamic-switch port recognizes errors different from those recognized by a link-level facility. A dynamic-switch port may also recognize errors based on the contents of a frame, especially the link-address portions of a frame, without recognizing the EOF delimiter and without validating the CRC of a frame. The following errors are only recognized by a dynamic-switch port in the inactive state.

11.2.1 Address-invalid error

An address-invalid error shall be detected when the dynamic switch receives a frame with a connect-SOF delimiter, the receiving dynamic-switch port is in the inactive state, and the frame contains either:

a) A source link address other than zero and the source link address does not correspond to the dynamic-switch port at which it was received;

b) A destination link address equal to zero or equal to the source address.

When an address-invalid error is detected, the requested connection shall be denied. The frame requesting the dynamic connection shall be discarded, and a port-reject frame with the address-invalid-error reject-reason code indicated shall be sent to the source of the discarded frame.

11.2.2 Undefined-destination-address error

An undefined-destination-address error shall be detected when the dynamic switch receives a frame with a connect-SOF delimiter, the receiving dynamic-switch port is in the inactive state, and the frame contains a destination link address which is not defined to the dynamic switch.

When an undefined-destination-address error is detected, the requested connection shall be denied. The frame requesting the dynamic connection shall be discarded, and a port-reject frame with undefined-destination-address-error reject-reason code indicated shall be sent to the source of the discarded frame.

11.2.3 Destination-port malfunction

A destination-port-malfunction condition shall be detected when the dynamic switch receives a frame with a connect-SOF delimiter, the receiving dynamic-switch port is in the inactive state, and the frame contains a destination link address that corresponds to a dynamic-switch port that is unavailable because of a link failure or a malfunction at the port.

When a destination-port-malfunction condition is detected, the requested connection shall be denied. The frame requesting the dynamic connection shall be discarded, and a port-reject frame with the destination-port-malfunction reject-reason code indicated shall be sent to the source of the discarded frame.

11.2.4 Dynamic-switch-port intervention required

A dynamic-switch-port intervention-required condition shall be detected when a dynamic switch receives a frame with a connect-SOF delimiter, the receiving dynamic-switch port is in the inactive state, and the frame contains a destination link address that corresponds to a dynamic-switch port that either:

a) Is inhibited from establishing a dynamic connection with the source port;

b) Is in the static state;

c) Is in the offline state, provided the offline state is not due to port malfunction.

When a dynamic-switch-port intervention-required condition is detected, the requested connection shall be denied. The frame requesting the dynamic connection shall be discarded, and a port-reject frame with the dynamic-switch-port intervention-required reject-reason code indicated shall be sent to the source of the discarded frame.

12 Device-frame format

Information associated with the execution of an I/O operation and the operation of an I/O-device is transferred between a channel and a control unit in the information field of a device frame. The information field of a device frame is used only for the transfer of commands, status, data, and control. The information field of every device frame consists of a device-header and a device-information block (DIB), as shown in figure 34.

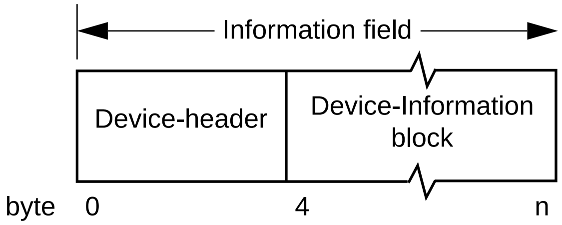


Figure 34 – Device frame information field

12.1 Device-header

The device-header is the first four bytes of the information field, which are assigned as follows: a one-byte information-field identifier (IFI), a two-byte device-address, and a one-byte device-header-flag (DHF) field. The format of the device-header is shown in figure 35

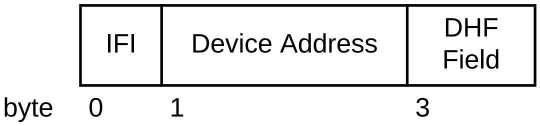


Figure 35 – Device-header

The device-header contains information-field-format identification, specifies device-level addressing, and designates the channel/control-unit flag bits common to more than one type of device frame. A summary of the IFI and DHF-field bit settings is provided in table 23.

12.1.1 Information-field identifier

The first byte of every device-header is the information-field identifier (IFI). The device frame type and the functions that affect field format or interpretation are identified by bits in the IFI. The format of the IFI is shown in figure 36.

Table 23 – Summary of IFI and DHF-field bit settings in the device-header

Type of Device Frame	IFI								Device-header flags							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
				AS	ES		T2	T1	E	EB	RDY	CN	DR			
Data	0	0	0	1	0	0	0	0	X	z	X	z	z	z	z	z
Command	0	0	0	1	U	0	0	1	X	z	z	X	X	z	z	z
Status	0	0	0	1	U	0	0	1	z	z	z	z	z	z	z	z
Control	0	0	0	X	X	0	1	1	X	X	X	X	X	z	z	z

Legend:

U Set to zero by the channel and causes unpredictable results if one
X May be set to either one or zero
z Set to zero by the sender and ignored by the recipient
0 Set to zero
1 Set to one

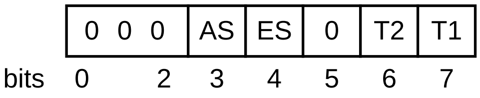


Figure 36 – Information field identifier (IFI)

Bits 0-2 and bit 5 of the IFI are reserved and are set to zeros in all device frame types; if any of these bits are not zero, a device-level error shall be detected (see 13.3.1).

12.1.1.1 Address-specific (AS)

The AS bit, bit 3 of the IFI, when set to one, indicates that the frame is associated with the specific I/O-device identified by the device-address field of the device-header. When the AS bit is set to zero, the frame shall not be associated with a specific I/O-device, and the device-address shall not be used.

For a data frame, the AS bit shall be set to one; otherwise, a device-level error shall be detected.

For a command frame, the AS bit is set to one; otherwise, a device-level error shall be detected.

For a status frame, the AS bit may be either one or zero, depending on whether the status is associated with the I/O-device or the control unit (see 12.2.3).

For a device-control frame, the AS bit may be either one or zero, depending on the particular control function (see 12.2.4 and table 23).

12.1.1.2 Supplemental-status (ES)

The ES bit, bit 4 of the IFI, may be set to either one or zero. The ES bit is used to control the transfer of supplemental-status. The meaning of the bit when set to one depends on the device frame type and on whether the frame was sent by the channel or the control unit.

For a data frame, the ES bit shall be set to zero; otherwise, a device-level error shall be detected.

For a command frame, the ES bit shall be set to zero; otherwise, unpredictable results occur.

For a status frame, the ES bit may be set to either one or zero. If the ES bit is set to one, supplemental-status shall be present in the DIB. If the ES bit is set to zero, supplemental-status shall not be present in the DIB. The ES bit may be set to one only when both the status frame is sent in response to a request-status frame having the ES bit set to one and the unit-check status bit of the status frame is set to one.

For a device-control frame sent by the control unit, the ES bit shall be set to zero by the control unit and shall be ignored by the channel. For a device-control frame sent by the channel, the ES bit may be set to either one or zero, depending on the particular control function and depending on conditions at the channel (see 12.2.4 and table 23).

12.1.1.3 Device-frame type (T2,T1)

Bits 6 and 7 of the IFI define the type of device frame. The device-frame type also determines the format of the device-information block. Table 24 summarizes the setting of the T2-T1 bits.

A data frame may be sent by either a channel or control unit.

A command frame shall only be sent by a channel. If a channel receives a command frame, a device-level error shall be detected.

A status frame shall only be sent by a control unit. If a control unit receives a status frame, a device-level error shall be detected.

Table 24 – Device-frame-type settings

T2	T1	Device-Frame Type	Device-Information Block
0	0	Data	Data
0	1	Command	Flags, CCW command, count
1	0	Status	Flags, status, count, supplemental-status
1	1	Control	Control function and parameters

A device-control frame may be sent by either a channel or control unit (see 12.2.4 and table 23).

12.1.2 Device-address

The 16-bit device-address is contained in the 16-bit field immediately following the IFI. When the AS bit is set to one, the device-address shall identify the I/O-device on the logical path that is to receive the contents of the information field in channel-to-control-unit frames or that is to send the contents of the information field in control-unit-to-channel frames.

Bits 0-7 of the device-address are reserved and are set to zeros by the source of a device frame. When the AS bit is set to one, if bits 0-7 of the device-address are not zeros, the recipient of a device frame shall detect a device-level error. When the AS bit is set to zero, the 16-bit device-address field shall be set to zero by the sender of the device frame and shall be ignored by the recipient of the device frame.

Only one device-address shall be used in a single connection. If more than one device-address is used with the AS bit set to one during the same connection, a device-level protocol error shall be detected.

If a control unit receives a device frame with the AS bit set to one and with a device-address of a not-ready device, it shall either perform the specified function and provide the appropriate response, if any, or, when the specified function requires a ready I/O-device, generates unit-check status, and the sense data associated with the unit-check indicates intervention required. If a control unit receives a device frame with the AS bit set to one and with the device-address of an uninstalled I/O-device, it shall generate either an address-exception condition or, optionally, unit-check status (see 12.2.4.9).

The manner in which device-addresses are assigned is model-dependent.

12.1.3 Device-header-flag field

The device-header-flag (DHF) field, which consists of the device-header flag bits, immediately follows the device-address. The device-header flag bits are used individually or collectively to invoke certain device-level protocols to control the execution of an I/O operation. The format of the device-header flag field is shown in figure 37. Bits 5-7 of the device-header flag field shall be set to zeros by the sender and ignored by the recipient.

	E	EB	RDY	CH	DR	0 0 0
bits	0	1	2	3	4	5 7

Figure 37 – Device-header flag field

12.1.3.1 End (E)

The E bit, bit 0, when set to one, shall be used to indicate a transfer of data or a request for data which exactly satisfies the remaining count for the current CCW being executed.

For a data frame, the E bit may be set to either one or zero. If the E bit is set to one, the data contained in the DIB exactly satisfies a prior data request which had the E bit set to one and exactly satisfies the current CCW count. If the E bit is set to zero, the data contained in the DIB does not completely satisfy a prior data request which had the E bit set to one (see 13.1.2).

For a command frame, the E bit may be set to either one or zero. If both the E bit and the DR bit are set to ones for a read operation, the data-request count exactly satisfies the count for the CCW. If the E bit is set to one and the DR bit is set to zero, no data request is being made because the count for the current CCW is zero. If the E bit is set to zero for a read operation, additional data requests are to be sent to satisfy the count for the CCW. The E bit shall be set to zero in a command frame for a write operation and shall be ignored by the recipient of a command frame for a write operation (see 13.1.1).

For a status frame, the E bit shall be set to zero by the control unit and shall be ignored by the channel receiving the status frame.

For a device-control frame, the E bit may be set to either one or zero (see 12.2.4, table 23, 13.1.2 and 13.1.2.1).

12.1.3.2 End-block (EB)

The EB bit, bit 1, when set to one, shall be used by a control unit to notify the channel of a disconnection. The EB bit shall be set to zero in frames sent by the channel and shall be ignored in frames received by the control unit.

For a data frame or status frame, the EB bit shall be set to zero by the control unit and shall be ignored by the channel.

For a multipurpose device-control frame from the control unit, the EB bit may be set to either one or zero. The EB bit shall be set to one with a rescind-connection frame to indicate that a connection granted by the channel in response to a request-connection frame is no longer needed by the control unit. The frame with the EB bit set to one has the disconnect-EOF delimiter; if the frame does not have a disconnect-EOF delimiter, a device-level error shall be detected. If the EB bit is set to zero, the control unit is not ending the connection with the channel.

For a device-level-ACK frame or a device-level-exception frame, when the EB bit is set to one, the control unit is ending the connection with the channel, and the frame has the disconnect-EOF delimiter; if the frame does not have a disconnect-EOF delimiter, a device-level error shall be detected. If a device-level-ACK frame or a device-level-exception frame is received by the channel with the EB bit set to zero, a device-level error shall be detected.

For a device-control frame other than a multipurpose device-control frame, device-level ACK, or device-exception, the EB bit shall be set to zero by the control unit and shall be ignored by the channel.

12.1.3.3 Ready (RDY)

The ready bit, bit 2, when set to one shall indicate one of the following conditions:

- the operation is considered in progress at the channel or the control unit;
- the sender of data is ready to transfer data;

- the sender of data is ready to accept a new data request;
- the channel grants a requested connection.

The meaning of the bit when set to one depends on the type of device frame, whether the frame was sent by the channel or control unit, and the conditions under which the bit is used.

For a data frame, the ready bit may be set to either one or zero. If the ready bit is set to one, the sender of data indicates that it may accept an additional request for data.

For a command frame, the ready bit shall be set to zero by the channel and shall be ignored by the control unit.

For a status frame, the ready bit shall be set to zero by the control unit and shall be ignored by the channel.

For a device-control frame, the ready bit may be set to either one or zero (see 12.2.4.1, 13.1.2 and 13.2.5).

12.1.3.4 Chaining (CH)

The CH bit, bit 3, when set to one, is used by the channel to signal its intention to chain or to confirm that chaining is continuing. The CH bit shall be set to zero in frames sent by the control unit and shall be ignored in frames received by the channel.

For a data frame, the CH bit shall be set to zero by the channel and shall be ignored by the control unit.

For a command frame from the channel, the CH bit may be set to either one or zero. If the CH bit is set to one, the command frame is a command update as a result of either command-chaining or data chaining, and the command-flag field and the count field contain information from the current CCW. Additionally, if the CH bit is set to one and the update is for command-chaining, the command field shall indicate the command from the new CCW. Whether the command update is for data chaining or command-chaining depends on the command-flag bits of the previous command frame and shall be indicated by the DU bit in the command-flag field. If the CH bit is set to zero, the command frame is not a command update. The CH bit shall be set to zero in all command frames initiating a channel program; if the CH bit is set to one for a command frame initiating a channel program, a device-level error shall be detected. The CH bit shall be set to one in all remaining command frames received during the execution of a channel program; if the CH bit is set to zero for a command frame after the first, chaining shall be terminated, and a device-level error shall be detected.

For a device-control frame from the channel, the CH bit may be set to one only for a status-accepted frame either during command-chaining and the status does not include device-end or when the status represents a retry request for an I/O operation and the status does not include device-end. The CH bit shall be set to one by the channel when chaining is still indicated for the I/O operation and set to zero when chaining is not or no longer indicated for the I/O operation or when a retry request is not accepted (see 13.1.4 and 13.3.2). The CH bit shall be set to zero for all control frames sent by the channel other than status accepted. The control unit shall ignore the CH bit for all device-control frames received other than status accepted.

12.1.3.5 Data-request (DR)

The DR bit, bit 4, when set to one, is used by the recipient of data, during an I/O operation that transfers data, to request a specific quantity of data.

For a data frame, the DR bit shall be set to zero by the sender of the data frame and shall be ignored by the recipient of a data frame.

For a command frame, the DR bit may be set to either one or zero. If data is to be transferred from the control unit to the channel and the data-request count is not zero, the DR bit shall be set to one; a device-level error shall be detected for a nonzero data-request count with the DR bit set to zero, or the DR bit set to one with a data-request count of zero. If data is to be transferred from the channel to the control unit, the DR bit shall be set to zero; if the DR bit is set to one, a device-level error shall be detected (see 13.1.2).

For a status frame, the DR bit shall be set to zero by the control unit and shall be ignored by a channel receiving a status frame.

For a device-control frame, the DR bit may be set to either zero or one. For a multipurpose device-control frame, the DR bit shall indicate that the device-control frame is being used to request data by the recipient of data. If the recipient of data receives a data request during data transfer, the recipient of data shall detect a device-level error. If a channel or control unit receives a data request other than during data transfer, the recipient of the data request shall detect a device-level error. For a command-response frame used for a write operation, the DR bit may be set to one by the control unit to indicate that data is requested. If the channel receives a command-response frame with the DR bit set to one when there is no write operation, the channel shall detect a device-level error. For a data request, a device-level error shall be detected if the DR bit is set to one and the data-request count is zero. For a command-response frame, if the DR bit is zero or if a read operation is indicated, the data-request count shall be ignored (see 13.1.2). For any device-control frame other than data request or command response, the DR bit shall be set to zero by the sender of the device-control frame and shall be ignored by the recipient.

12.2 Device-information block

The device-information block (DIB) is a variable-length field that immediately follows the device-header field. The length of the device-information block depends on the device frame type, the quantity of information to be sent, and the maximum DIB size in effect. The DIB used with a device-control frame, command frame, or status frame (without supplemental-status) is four bytes, and may be from 5 to 36 bytes for a status frame with supplemental-status. The DIB used with a data frame may be from 1 to 1,024 bytes. Figure 38 shows the device-information-block formats for each of the device frame types. If a device frame is received with a DIB size that is not permitted for the device frame type, a device-level error shall be detected. If a status frame is received with a DIB size that is not appropriate for the status frame considering whether supplemental-status is included, a device-level error shall be detected.

The set of DIB sizes allowed by the channel for data frames is determined when the logical path is established and remains in effect until the logical path is removed (see 9.1.1). When an I/O operation is initiated, the control unit shall select a maximum DIB size from the set of DIB sizes allowed by the channel, and this maximum DIB size is identified to the channel in the command response. This maximum DIB size is used for the duration of the I/O operation (see 12.2.4.2 for the possible choices of the maximum DIB size).

The DIB used for a command frame, device-control frame, or status frame is not affected by a choice of the maximum DIB size.

12.2.1 Data-frame DIB structure

Data is transferred between a channel and control unit in the device-information block (DIB) of the data frame. The format of the device-header and device-information block used for a data frame is shown in figure 38. The direction of data transfer (read or write) shall be determined by the current command. The number of bytes of data sent in the DIB used in a data frame shall be determined by parameters established at the start of execution of the current command and by conditions present when the data frame is sent. The number of data frames required for a data transfer depends on the quantity of data requested and the DIB size in effect.

12.2.2 Command-frame DIB structure

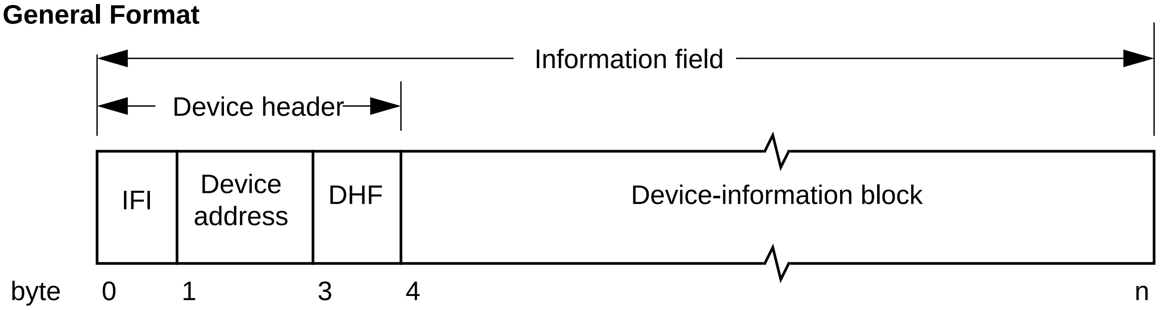
A command frame is used for the channel-to-control-unit transfer of the information associated with the current CCW being executed. At the beginning of an I/O operation, a command frame shall be used to initiate the operation with an I/O-device. When data chaining is performed, a command frame shall be used to update the information held about the current CCW at the control unit. The format of the device-header and DIB used for a command frame is shown in figure 38. The DIB used for command frame contains fields for the command flags, the command, and a 16-bit count field.

12.2.2.1 Command-flag field

The first byte of the DIB in a command frame is the command-flag field, which contains chaining flag bits from the current CCW and the data-chaining-update flag (DU). The format of the command-flag field is shown in figure 39.

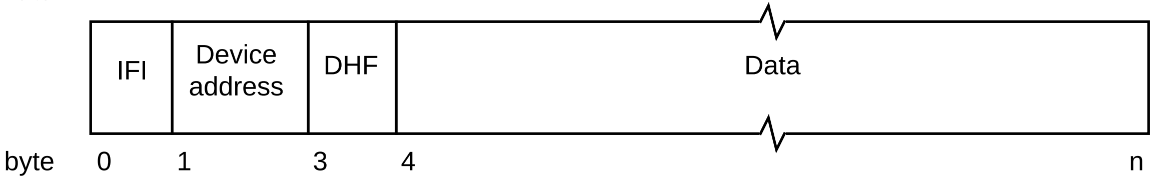
Bits 2-5 and bit 7 shall be reserved, set to zeros by the channel and ignored by a control unit receiving a command frame.

General Format



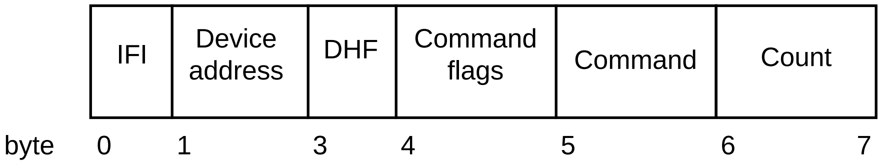
where n depends on the device frame type

Data

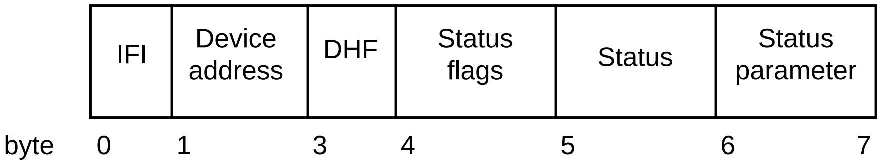


where n is less than or equal to 1,027

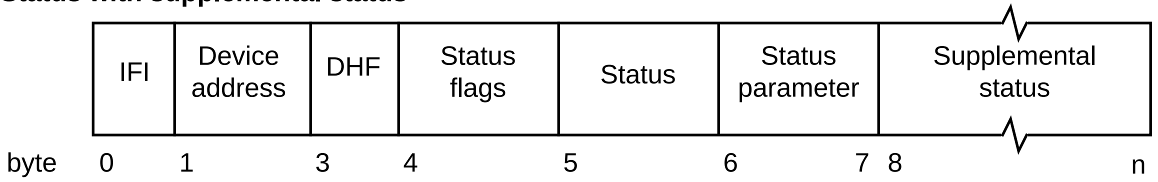
Command



Status



Status with supplemental status



Device-level control

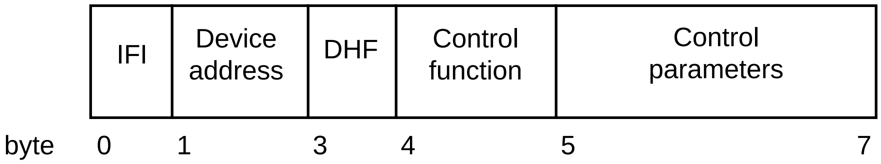


Figure 38 – Summary of device-frame information-field formats

	CD	CC	r	r	r	r	DU	r
bits	0	1	2			5	6	7

Figure 39 – Command-flag field

12.2.2.1.1 Chain-data (CD)

The CD flag, bit 0, when set to one, specifies an intent to perform chaining of data. It causes the command flags and count designated by the next command frame in which the DU flag and the CH bit are set to ones to be used with the current I/O operation. When the CD flag is set to one, the CC flag may or may not be ignored. If the CC flag is not ignored, the control unit is permitted to treat the CC and CD flags, both being set to one, as an indication that command chaining may be performed at the completion of the current CCW.

When data chaining occurs in the channel, a new set of command flags and a new count shall be transferred from the channel to the control unit in a command frame, and the CH bit and the DU flag shall be set to ones.

It is model-dependent whether the control unit permits data chaining for a particular command or for a particular I/O-device. If the CD flag is set to one in a command frame having the DU flag set to zero, the control unit may reject the command with unit-check status. If the command is rejected due to data chaining, the unit-check status shall be returned in a status frame sent either in direct response to the command frame or after sending a command-response frame but before sending either a data-request frame or a data frame. The sense data associated with the unit-check shall indicate command-reject along with a model-dependent indication that the unit-check occurred because data chaining is not permitted for the command.

12.2.2.1.2 Chain-command (CC)

The CC flag, bit 1, when set to one while the CD flag is set to zero, specifies an intent to perform chaining of commands. Upon normal completion of the current I/O operation in the channel, and after receiving device-end from the I/O-device, chaining of commands shall cause a new I/O operation to be initiated. When the CD flag is set to one, the CC flag may or may not be ignored. If the CC flag is not ignored, the control unit is permitted to treat the CC and CD flags, both being set to one, as an indication that command chaining may be performed at the completion of the current CCW.

When command-chaining occurs in the channel, a new set of command flags, a new command, and a new count shall be transferred from the channel to the control unit in a command frame; the CH bit shall be set to one in the device-header; and the DU flag shall be set to zero in the command-flag field. The CH bit set to one and the DU flag set to zero shall indicate that the status frame was accepted and that command-chaining is taking place.

12.2.2.1.3 Data-chaining-update flag (DU)

The data-chaining-update flag (DU flag), when set to one, shall indicate that the CC flag, the CD flag, and the count sent in that command frame are associated with a new CCW used during data-chaining. The CH bit in the device-header shall be set to one when the DU flag is set to one (see 13.1.2.3).

12.2.2.2 Command field

When the channel attempts to initiate an I/O operation with an I/O-device, the command field shall contain the command from the current CCW. The command shall specify the I/O operation to be executed. The basic operations are specified by the following commands: read, read backward, write, control, and sense. Table 25 lists the basic commands and the bit settings of the command field.

Table 25 – Contents of the command field

Command	Bit Position							
	0	1	2	3	4	5	6	7
reserved	M	M	M	M	0	0	0	0
Sense	M	M	M	M	0	1	0	0
reserved	M	M	M	M	1	0	0	0
Read backward	M	M	M	M	1	1	0	0
Write	M	M	M	M	M	M	0	1
Read	M	M	M	M	M	M	1	0
Control	M	M	M	M	M	M	1	1
NOTE – M represents a modifier bit								

The rightmost bit positions indicate the type of operation; the leftmost bit positions indicate a modification code which expands the meaning of the basic command that is to be performed. The modifier codes and the commands performed when they are decoded shall be model-dependent.

When the command frame is initiating an I/O operation, the command field may contain any value that is not reserved. If a reserved command code is received by an I/O-device, the I/O-device shall respond with unit-check status, and the sense data associated with the unit-check shall indicate command-reject.

There are some commands (particular combinations of eight bits in the command field) which have been specified to have special functions beyond those designated by the command type. These commands are specified in clause 15.

When a command frame is used to update the count and flags during data-chaining (the DU flag is set to one), the command field is set to zeros by the channel and is ignored by the control unit receiving the command frame.

Some commands, when executed, do not result in the transfer of data but cause the I/O-device to respond to the command with status which contains channel-end, with or without device-end (and without busy status), by sending a status frame. When this occurs, the I/O operation is called an immediate operation. There are other situations in which an I/O operation may result in no data being transferred, for example, when a CCW count of zero has been validly specified. However, the operation is not an immediate operation if the I/O-device responds to the command with a command-response frame. Each of the basic operations and specific I/O commands is described in one of the following subclauses.

12.2.2.2.1 Sense

The sense command is similar to a read command, except that the data is obtained from sense indicators rather than from a record source. Sense bytes are placed in a buffer in the same order as those transferred by the read command. Sense byte 0 will be the first byte transferred from the I/O-device to the channel (see 12.2.3.5).

12.2.2.2.2 Read-backward

A read-backward command initiates execution of an I/O operation in the same manner as the read command, except that bytes of data within a block are sent to the channel in an order which is the reverse of that used in writing. The bits within an eight-bit byte are in the same order as sent to the I/O-device on writing. The control unit may be designed to cause mechanical motion in the I/O-device in a direction opposite to that for a read command, or it may be designed to operate the I/O-device as it would for a read command.

Unless otherwise noted, any description that applies to read also applies to read backward

12.2.2.2.3 Read

A read command initiates execution of an I/O operation that performs device-to-channel data transfer. The bytes of data within a block are provided in the same sequence as those transferred by the write command.

12.2.2.2.4 Write

A write command initiates execution of an I/O operation that performs channel-to-I/O-device data transfer. The bytes of data within a block are provided in the same sequence as those transferred by the read command.

12.2.2.2.5 Control

A control command initiates execution of an I/O operation that performs channel-to-I/O-device data transfer. The I/O-device interprets the data as control information.

For many control functions, the entire operation is specified by the modifier bits in the command code, and the function is performed over the channel path as an immediate operation (see 12.2.2.2). A control command may be used to initiate at the I/O-device an I/O operation not involving transfer of data, such as backspacing or rewinding magnetic tape or positioning a disk-access mechanism.

12.2.2.3 Count field

The count field is a 16-bit field that specifies a count of data for the current CCW. The field is interpreted as a 16-bit unsigned binary integer. The value in the field may range from 0 to 65 535.

When the operation is a read and the data-request (DR) bit is set to one, the count field shall indicate the quantity of data requested by the channel. The end (E) bit shall indicate whether or not the count is equal to the count in the current CCW. When the end bit is set to one, the counts are equal, and when the end bit is set to zero, there are fewer bytes specified in the count field than in the current CCW.

When the operation is a write, the count in the count field shall be equal to the count in the current CCW, and the end bit shall be set to zero.

When the operation is treated as an immediate operation, the contents of the count field shall be ignored.

12.2.3 Status-frame-DIB structure

A status frame is used to transfer status, with or without supplemental-status, to a channel. Except for a valid system-reset frame, a status frame may be returned to any device frame that initiates a connection. Once a connection is made, a status frame may be sent as a result of a command frame or during data transfer. Additionally, a status frame may be sent after a connection is made either as the result of a request-connection or in response to a request-status frame as a part of retrieving supplemental-status.

If the control unit is disconnected during an I/O operation, a channel connection shall first be obtained by the control unit; the control unit shall request the connection by sending a request-connection frame. If the channel connection is granted, the status frame may be sent.

A status frame may be sent with the AS bit set to either zero or one depending on whether the status is associated with the control unit or an I/O-device. If the status frame contains one of the following status combinations, the status shall be associated with the control unit, and the AS bit shall be set to zero:

a) **Control-unit-busy:** If a control unit receives a device frame that initiates a connection and a busy condition exists which prevents the acceptance of any device frame with the exception of a valid system-reset frame, then the control unit sends a status frame with only the status-modifier and busy status bits set to ones and the AS bit set to zero. This status shall be referred to as control-unit-busy status (see 13.2.9.1);

b) **Control-unit-end-alone:** If a control unit does not recognize a stack-status frame sent in response to a status frame containing control-unit-busy status, the control unit will later send a status frame with the control-unit-end status bit set to one. When all other status bits are set to zeros, the AS bit shall be set to zero, and the status is referred to as control-unit-end-alone;

c) **Control-unit-end-with-busy:** When control-unit-end status is pending and the control unit recognizes a valid command frame that initiates a connection, the control unit may return a status frame with only the control-unit-end and busy status bits set to ones and the AS bit set to zero. This status combination is referred to as control-unit-end-with-busy (see 12.2.3.2.4);

d) **Control-unit-end-with-control-unit-busy:** If a control unit receives a device frame that initiates a connection and the frame is not a system-reset frame, the control unit may return a status frame with only the status-modifier, control-unit-end, and busy status bits set to ones, and the AS bit set to zero. This status combination is referred to as control-unit-end-with-control-unit-busy. When the channel recognizes this combination of status bits, it shall consider the status frame to be an indication that the control unit is not busy.

In all other cases, the status shall be associated with a device-address, and the AS bit shall be set to one; If the AS bit is set to zero with any status combination not listed above, the channel shall detect a device-level error.

A status frame with the AS bit set to zero may be sent only if the status is one of the preceding combinations and meets one of the following conditions:

a) The status frame is sent in response to an accept-connection frame, and the status is control-unit-end-alone;

b) The status frame is sent in response to a valid request-status frame that initiated a connection, and the status-byte is not control-unit-end-with-busy;

c) The status frame is sent in response to a valid command frame that initiated a connection, and the status is not control-unit-end-alone;

d) The status frame is sent in response to a valid device frame, other than a system-reset frame, that initiated a connection, and the status is control-unit-busy.

The channel shall detect a device-level error if a status frame is received that has the AS bit set to zero and does not meet one of the preceding conditions.

The format of the information field of a status frame, with and without supplemental-status, is shown in figure 38. The DIB of a status frame contains the status-flag field, status-byte field, status-parameter field, and, when the ES bit is set to one in the IFI, from 1 to 32 bytes of supplemental-status.

12.2.3.1 Status-flag field

The status-flag field is used to provide additional information to the channel concerning the conditions that were present in the control unit when status was generated and conditions that pertain to the status frame. These flags assist the channel in determining how to handle the status frame and what status to report. The format of the status-flag field is shown in figure 40.

	FFC	CI	SA	CR	LR	XV
bits	0 2	3	4	5	6	7

Figure 40 – Status-flag field

Flag-field code (FFC): The FFC, bits 0-2, is a three-bit encoded field that either in conjunction with or independent of the other status-flag bits further describes the status information contained in the status-byte or the status-parameter field or both. The flag-field code assignments are:

Value Meaning

000 - *No function:* The status-byte and status-parameter field are not affected by this FFC code.

001 - *Queuing-information-valid:* The status-parameter field contains control-unit queuing information. This code shall be set by the control unit and checked for by the channel only during the presentation of device-end status during a reconnection for an I/O operation.

010 - *Resetting-event:* A resetting-event condition exists for the logical path and the I/O-device for which unit-check status is being presented. This code is permitted only when unit-check status is presented for a resetting-event condition and then only as initial status in response to a command transfer for the first command of a channel program; if this code is indicated at any other time, the channel shall detect a device-level error.

Sense data indicating a resetting-event condition shall not be presented as supplemental-status.

011:111 - Reserved.

Channel-initiated (CI): The CI bit, bit 3, when set to one, indicates that the status frame resulted from the identified I/O-device having accepted a selective-reset frame, with either a request for retry or a request for unit-check (see 13.3.3.1 and 13.3.3.2). If the CI bit is set to zero, the status frame is not a direct response to a selective-reset frame with a request for retry or a request for unit-check.

Supplemental-status-available (SA): The SA bit, bit 4, may optionally be used by the control unit to indicate that supplemental-status associated with the current status frame is available. The SA bit shall only be used when unit-check status is indicated and only when the channel did not previously request supplemental-status. It shall be model-dependent whether the control unit sets the SA bit to one when it is permitted to do so. If both the SA bit and the unit-check status bit are set to ones, the channel may optionally retrieve the supplemental-status by using a request-status frame with the ES bit set to one (see 13.1.3.2 and 13.2.6).

Command-retry (CR): The CR bit, bit 5, when set to one, is used to request command-retry if the status-byte also contains retry status. If the CR bit is set to zero or the status-byte does not contain retry status, command-retry is not being requested. The status-byte contains retry status if it contains unit-check and status-modifier together with 1) channel-end alone (meaning the control unit or I/O-device is not yet ready to retry the command), or 2) channel-end and device-end (meaning the control unit and I/O-device are prepared for immediate command-retry). All other status combinations are not considered to be retry status; if the CR bit is set to one with any other status combination, a device-level error shall be detected (see 13.3.1).

Long-record (LR): The LR bit, bit 6, when set to one, indicates that a long-record condition was detected by the control unit; that is, additional data would have been sent or accepted beyond the data provided for the current command. The LR bit may be set to one by the control unit only when the channel-end status bit is set to one. If the channel-end status bit is set to zero in a status frame, the control unit shall set the LR bit to zero, and the LR bit is ignored by a channel receiving the status frame. When data-chaining is not used, the amount of data provided for the current command shall be equal to the count provided in the CCW. When data-chaining is used, the amount of data provided for the current command shall be equal to the sum of the counts of all CCWs used for the current command. If the XV bit is set to zero, the LR bit shall be ignored by the channel.

Transfer-count-valid (XV): The XV bit, bit 7, when set to one, indicates that the status-parameter field contains the transfer count; that is, the total number of bytes sent or received by the control unit for the current CCW. The XV bit may be set to one by the control unit only when the channel-end status bit is set to one. If the channel-end status bit is set to zero in a status frame, the control unit shall set the XV bit to zero, and the XV bit shall be ignored by a channel receiving the status frame. When the XV bit is set to zero, the status-parameter field shall not contain a transfer count (see 12.2.3.3).

12.2.3.2 Status-byte

The status-byte field indicates I/O-device and control-unit status. The format of the status-byte is shown in table 26.

Table 26 – Status-byte

Status Bit Position	Description
0	Attention
1	Status-modifier
2	Control-unit-end
3	Busy
4	Channel-end
5	Device-end
6	Unit-check
7	Unit-exception

If the channel receives a status frame with the status-byte set to zero, a device-level error shall be detected. The status frame shall contain a status-byte with a combination of status bits set to ones which is appropriate for the conditions existing when the status is presented. If the combination of status bits is not appropriate for the existing conditions, the channel may detect an error (see annex A).

The causes of these conditions for each type of I/O-device and the timing in presenting them are model dependent. When these conditions have been recognized, status shall be generated at the I/O-device or control unit, and shall be presented to the channel. The status shall be -maintained in the control unit or I/O-device until it is accepted by the channel, is withdrawn, or is reset. Status that is held in the control unit or I/O-device is called pending status. When the I/O-device presents status to the channel and the channel responds with stack status, the pending status is called stacked status. Forthcoming status is status which is owed to the path but is not yet pending, such as no-longer-busy status.

When the I/O-device is accessible from more than one channel, and when the dynamic-reconnection feature is not being used, status resulting from an I/O operation initiated by one of the channels shall be transferred to that channel on the appropriate logical path. The handling of conditions not associated with I/O operations, such as attention, unit-exception, and device-end because of a transition from the not-ready to the ready state, depends on the type of I/O-device and status condition and shall be model-dependent (see 12.2.3.2.6).

A channel shall indicate acceptance of the status by means of either a status-accepted frame, or, when retry or chaining is performed and the status includes device-end, by means of a command frame having the CH bit set to one.

12.2.3.2.1 Attention

The attention condition is generated when some asynchronous condition occurs in the control unit or I/O-device. The condition may be accompanied by other status. Attention shall not be associated with the initiation, execution, or termination of any I/O operation.

If an I/O operation is ended with attention status, then command-chaining, if any, shall not be performed, and the operation shall be terminated. When the attention condition occurs, the handling and sending of the status shall depend on the type of I/O-device or control unit. When an I/O-device is shared between more than one channel path, or more than one logical path, presentation of attention status to logical paths shall be model-dependent. Depending on the I/O-device, attention may or may not be presented until command-chaining is no longer indicated.

Attention shall be accompanied by device-end and unit-exception to indicate that a transition from the not-ready state to the ready state has occurred (see 12.2.3.2.6). Not-ready-to-ready state-transition status shall be sent to all established logical paths.The status combination shall not be presented as part of the status related to an I/O operation.

The I/O-device may signal the attention condition to the channel subsystem when no operation is in progress at the I/O-device. Attention may be indicated with device-end upon completion of an operation, and it may be presented to the channel subsystem during the initiation of a new I/O operation.

Attention accompanying device-end shall cause command-chaining and command-retry to be suppressed.

12.2.3.2.2 Status-modifier

Status-modifier shall be generated by the I/O-device when it cannot provide its current status in response to interrogation by the channel subsystem, when the control unit is busy, when the normal sequence of commands has to be modified, or when command-retry is to be initiated.

When the I/O-device is interrogated and the status-modifier condition signaled in the absence of any other status bit, this shall indicate that the I/O-device cannot provide its current status. The interruption condition, which may be pending at the I/O-device, shall not be cleared.

Presence of status-modifier and device-end indicates that the normal sequence of commands shall be modified. The handling of this set of bits by the channel subsystem depends on the operation. If command-chaining is specified in the current CCW and no unusual conditions have been detected, presence of status-modifier and device-end shall cause chaining to the CCW following the next CCW. If the I/O-device signals the status-modifier condition at a time when no command-chaining is specified, or when any unusual conditions have been detected, no action is taken on the channel subsystem on the SBCON I/O interface; however, the condition is signaled to the ULP.

Status-modifier shall be presented in combination with unit-check and channel-end to initiate the command-retry procedure.

Control units that recognize special conditions which must be brought to the attention of the ULP shall present status-modifier along with other status indications in order to modify the meaning of the status. The status presented is unrelated to the execution of an I/O operation.

When status-modifier is generated together with the busy status bit, it shall indicate that the busy condition pertains to the control unit associated with the addressed I/O-device. The control unit appears busy when it is executing a type of operation that precludes the acceptance and execution of any command and may appear busy when it contains status or sense information for an I/O-device other than the one addressed.

A control unit accessible from two or more channel paths appears busy to the other channel paths when it is communicating with any of the channel paths.

12.2.3.2.3 Control-unit-end

Control-unit end indicates that the control unit has become available for use for another operation.

Only control units that indicate a control-unit-busy condition may indicate a control-unit-end condition. Control-unit-end status shall be returned from the control unit to the channel after the channel accepts control-unit-busy status from the control unit. Control-unit-end status shall be returned after the control-unit-busy condition no longer exists. This is sometimes referred to as no-longer-busy status for the control unit.

Only one control-unit-end indication shall be returned on a logical path, regardless of the number of times the channel accepted control-unit-busy status on that logical path during the busy period. The control-unit-end condition shall be provided only by control units shared by I/O-devices or control units accessible by two or more channel paths, and only when one or both of the following conditions have occurred:

a) The channel subsystem had previously caused the control unit to be interrogated while the control unit was busy. The control unit is considered to have been interrogated in the busy state when a command has been transferred to an I/O-device on the control unit, and the control unit had responded with busy and status-modifier in the status-byte;

b) The control unit detected an unusual condition during the portion of the operation after channel-end had been signaled to the channel subsystem. The indication of the unusual condition shall accompany control-unit-end. However, the signaling of control-unit-end and device-end does not necessarily describe an unusual condition.

The two conditions described above shall be reset by the system reset and the selective reset signals. Therefore, if one of these resets occurs before control-unit-end is generated, no control-unit-end shall be generated. If control-unit-end has been generated but not presented to the channel subsystem by the time one of the resets occurs, the pending control-unit-end shall be reset.

The control unit shall not associate pending control-unit-end status with any device-address. A selective-reset shall not reset pending control-unit-end status. If the channel stacks or does not accept a status-byte that contains the control-unit-end status bit, the control-unit-end status shall not be held along with the status-byte, and shall remain pending and unstacked at the control unit. Control-unit-end status may be withdrawn by the control unit if the control unit becomes busy again before the status is accepted by the channel; in such a case, the control unit shall return control-unit-end status later, after the control-unit-busy condition no longer exists.

When control-unit-end is included with other status bits set, other than those status combination required to have the AS bit set to zero, the AS bit shall be set to one, and the device-address used is the device-address for which the other status is being sent (see 12.2.3). A pending control-unit-end shall not necessarily preclude initiation of new operations. Whether the control unit allows initiation of new operations shall be at the option of the control unit.

Control-unit-end shall not necessarily cause command-chaining to be suppressed. Control-unit-end shall not cause command-chaining to be suppressed when presented with the AS bit set to zero as described in 12.2.3.

When control-unit-end status is presented on a logical path along with status-modifier and busy, with no other status bits set to one, the combination shall be interpreted as control-unit-end status. When this status is accepted, the control unit shall no longer owe a control-unit-end status indication on that logical path. If the control unit was interrogated while it was in the busy state, and then system-reset is recognized by the control unit before control-unit-end status is accepted by the channel, the control unit shall not owe control-unit-end status.

A control unit shall only present control-unit-end status when a no-longer-busy condition is owed. However, the channel shall not detect an error if control-unit-end status is received when no control-unit-busy condition was indicated (that is, a no-longer-busy condition was not owed).

NOTE – Presentation of the control-unit-end status bit without any other status bits is the preferred implementation.

If the control unit remains busy with the execution of an operation after signaling channel-end but has not detected any unusual conditions and has not been interrogated by the channel subsystem, control-unit-end shall not be generated. Similarly, control-unit-end shall not be provided when the control unit has been interrogated and could perform the indicated function. The latter case is indicated by the absence of busy and status-modifier in the response to the interrogation.

When the busy condition of the control unit is temporary, control-unit-end may be included with busy and status-modifier in response to the interrogation even though the control unit has not yet been freed. The busy condition shall be considered as temporary if its duration is 2 milliseconds or less. If a temporary busy condition is indicated, the channel subsystem shall assume the responsibility to periodically reinterrogate the control unit until it is no longer busy.

The control-unit-end condition may be signaled with channel-end, with device-end, or between the two.

Control-unit end presented with channel-end is unusual status and shall cause the channel subsystem to suppress command-chaining, if specified.

12.2.3.2.4 Busy

The busy indication occurs only when conditions existing at the I/O-device or control unit preclude execution of the intended I/O operation because of one of the following four situations:

a) A previous I/O operation or chain of I/O operations is being initiated or being executed;

b) Stacked or pending status conditions exist, and the pending status conditions are returned in response to a command; busy shall be appended to the status returned;

c) The control unit is shared by channels or I/O-devices, or an I/O-device is shared by control units, and the shared facility is not available;

d) A self-initiated function (for example, microdiagnostics or data movement internal to the I/O-device) is being performed.

Status conditions for the addressed I/O-device, if any, shall accompany the busy indication.

For the first command of a command-chain, an I/O operation is being initiated by an I/O-device from the time the control unit returns a status response or command response that indicates the command is accepted, and an I/O operation is being executed by the control unit from the time the control unit receives acknowledgment that the status response or command response has been accepted by the channel. For other than the first command of a command-chain, an I/O operation is being executed by the I/O-device from the time the control unit returns a status response or command response that indicates the command is accepted. An I/O operation is being executed at the I/O-device until device-end status for the I/O operation is accepted by the channel or until the I/O operation is terminated by a reset.

If the busy condition applies to a control-unit facility, busy shall be accompanied by status-modifier. Depending on the design of the control unit, the control unit may or may not become busy during the time that an I/O operation is being executed by the I/O-device, and if it does become busy, the control unit shall not remain busy due to the I/O operation after the I/O-device is no longer busy for the I/O operation. Busy status shall cause command-chaining to be suppressed.

Whenever the I/O-device indicates that a busy condition exists and it is unable to execute an operation, the I/O-device shall signal the channel subsystem when it becomes no longer busy (see 12.2.3.2.6).

12.2.3.2.5 Channel-end

Channel-end status shall be presented when the data-transfer portion of an I/O operation is completed. Acceptance of channel-end status is considered to be the completion of the channel portion of the I/O operation.

Each I/O operation initiated at the I/O-device shall cause one and only one channel-end for an I/O operation. The channel-end condition shall not be generated when ULP errors or equipment malfunctions are detected during initiation of the operation. When command-chaining takes place, only the channel-end of the last operation of the chain shall be signaled to the ULP. The channel-end condition shall not be signaled to the ULP when a chain of commands is prematurely concluded because of an unusual condition indicated with device-end or during the initiation of a chained command.

The instant within an I/O operation when channel-end is generated depends on the operation and the type of I/O-device. For operations such as writing on magnetic tape, the channel-end condition shall occur when the block has been written on the media. On I/O-devices that verify the writing, channel-end may or may not be delayed until verification is performed, depending on the I/O-device. When magnetic tape is being read, the channel-end condition shall occur when the gap on tape reaches the read-write head. On I/O-devices equipped with data buffers, the channel-end condition shall occur upon completion of data transfer between the channel subsystem and the data buffer. During control operations, channel-end shall be generated when the control information has been transferred to the I/O-devices, although, for short operations, the condition may be delayed until completion of the operation. Operations that do not cause any data to be transferred may provide the channel-end condition during the initiation sequence.

Channel-end shall be presented in combination with status-modifier and unit-check by means of a special sequence to initiate the command-retry procedure.

12.2.3.2.6 Device-end

Device-end shall be presented to the channel for the following conditions:

- when the completion of an I/O operation is indicated by the I/O-device;
- when the I/O-device, having previously responded busy, indicates that a transition from the busy to the not-busy state has occurred;
- when the I/O-device indicates that a change from the not-ready to the ready state has occurred;
- when the control unit or I/O-device indicates that an asynchronous condition has been recognized.

Each I/O operation initiated at the I/O-device shall cause only one device-end indication to be generated. The device-end indication at the completion of an I/O operation shall be generated only if the command is accepted.

The device-end condition associated with an I/O operation shall be generated either simultaneously with the channel-end condition or later. For data-transfer operations on some I/O-devices, the operation is complete at the time channel-end is generated, and both device-end and channel-end occur together. The time at which device-end status is sent depends upon the I/O-device type and the command executed. For most I/O-devices, device-end status is sent when the I/O operation is completed. In some cases, for reasons of performance, device-end status is sent before the I/O operation has actually been completed at the I/O-device. However, when command-chaining was indicated and when device-end is sent in a status-byte, the I/O-device shall be considered available to start another I/O operation when the channel indicates its acceptance of the status.

When chaining is being performed, the channel normally makes available to the ULP only the device-end of the last operation of the chain. When only channel-end has been accepted by the channel, the next status sent by the I/O-device shall be either device-end or device-end and status-modifier; otherwise, command-chaining, if any, is suppressed. Device-end status, when received by the channel in the absence of any unusual conditions, shall cause the next command-chained I/O operation, if any, to be initiated. Device-end status, when received by the channel in the presence of unusual conditions, shall cause the termination of command-chaining, if any, and the unusual condition shall be reported to the ULP.

If an I/O-device previously responded busy (other than busy which is appended to pending status returned in response to a command and other than when control unit busy was indicated), then, when the I/O-device becomes not busy, status indicating device-end shall be sent on the logical path on which the busy was sent. The device-end status sent at this time is called no-longer-busy device-end. In this situation, device-end shall be sent only once, independent of the number of times the I/O-device responded with busy. A device-end shall not be returned as the result of busy status which is appended to pending status returned in response to a command (see 12.2.3.2.4).

Device-end shall be accompanied by attention and unit-exception when a transition from the not-ready state to the ready state occurs. The status combination of device-end, attention, and unit-exception shall be used only to indicate not-ready-to-ready state-transition status. The status combination may not be presented as part of the status related to an I/O operation.

An I/O-device shall be considered to be not ready when it is not installed or when operator intervention would be required to make the I/O-device ready.

An I/O-device for which a not-ready-to-ready state transition has occurred and which is not using the dynamic-reconnection feature shall send status of device-end, along with attention and unit-exception, on all paths.

Device-end shall be accompanied by other status when conditions are recognized that are unrelated to the execution of an I/O operation.

These conditions shall be brought to the attention of the ULP and are sent to the channel as they occur.

In the case of an I/O-device using the dynamic-reconnection feature, device-end status at the completion of an I/O operation may be presented to a logical path within the path group other than the logical path on which the I/O operation was started, provided there was no previous unit-check status for the same I/O operation. Similarly, an I/O-device using the dynamic-reconnection feature may present device-end to any logical path in the path group for the no-longer-busy indication or for the transition from the not-ready to the ready state. In the latter case, device-end shall be accompanied by attention and unit-exception.

Device-end normally indicates that the I/O-device has become available for use for another operation.

When command-chaining is specified and the suspend flag is zero in the next CCW, receipt of the device-end signal, in the absence of any unusual conditions, shall cause the channel subsystem to initiate transfer of the next command.

On some buffered I/O-devices, the device-end condition shall occur upon completion of the storage of the data on non-volatile media.

The ULP shall not request the channel to start a new channel program for a given I/O-device until the ULP has received the device-end signal for the prior channel program, if any, for that I/O-device.

For control operations, device-end shall be generated at the completion of the operation at the I/O-device. The operation may be completed at the time channel-end is generated or later.

12.2.3.2.7 Unit-check

Unit-check indicates that the I/O-device or control unit has detected an unusual condition that is detailed by sense information (see 12.2.3.5). The occurrence of unit-check may indicate that a ULP error or an equipment error has been detected, that the not-ready state of the I/O-device has affected the execution of the command, or that an exception condition other than the one identified by unit-exception has occurred. The unit-check bit provides a summary indication of the conditions identified by sense data.

An error condition shall cause the unit-check indication when it occurs during the execution of a command, during some activity associated with an I/O operation, or when an unusual condition is detected that is unrelated to execution of an I/O operation. Unless the error condition pertains to the activity initiated by a command then unit-check shall be presented together with attention, control-unit-end, or device-end as unsolicited alert status.

Unit-check shall be indicated when the existence of the not-ready state precludes a satisfactory execution of the command or when the command, by its nature, tests the state of the I/O-device. When no status is pending for the addressed I/O-device at the control unit, the control unit shall indicate unit-check when the no-operation control command is issued to a not-ready I/O-device. In the case of no-operation, the command shall not be accepted, and channel-end and device-end shall not accompany unit-check.

Unless the command is designed to cause unit-check, such as a command to rewind and unload magnetic tape, unit-check shall not be indicated if the command is properly executed, even though the I/O-device has become not ready during or because of the operation. Similarly, unit-check shall not be indicated if the command can be executed with the I/O-device not ready. A control unit could, for example, accept and execute a command which does not require the I/O-device to be in the ready state. A command received for an I/O-device in the not-ready state shall not cause a unit-check indication when the sense command is issued or when a status condition is pending for the addressed I/O-device at the control unit.

If, prior to acceptance of a command, the I/O-device detects that the command cannot be executed, a status-byte shall be returned containing unit-check with neither channel-end nor device-end. This status indicates that no action has been taken at the I/O-device in response to the command. If the condition that precludes normal execution of the operation occurs after the command has been accepted, unit-check shall be accompanied by channel-end, device-end, or both, depending on when the condition was detected. Errors detected after device-end is accepted may be indicated with a status-byte containing unit-check and attention, or unit-check and device-end.

Errors, such as a command-reject condition, shall not cause unit-check when the I/O-device is busy or contains pending status at the time a channel attempts to initiate an I/O operation. Under these circumstances, the I/O-device shall respond by indicating busy (busy bit set to one) along with any pending status.

Termination of an operation with the unit-check indication shall cause command-chaining to be suppressed, except when the status causes the initiation of command-retry.

Unit-check shall be presented in combination with channel-end and status-modifier to initiate the command-retry procedure. Otherwise, unit-check status presented either in the absence of or accompanied by other status indicates only that sense information is available to the basic sense command. Presentation of either channel-end and unit-check or channel-end, device-end, and unit-check does not provide any indication as to the kind of conditions encountered by the control unit, the state of the I/O-device, or whether execution of the I/O operation ever was initiated even though the command may have been accepted. Descriptions of these conditions shall be provided in the sense information.

12.2.3.2.8 Unit-exception

Unit-exception means that the I/O-device detected an unusual condition that needs to be reported to the ULP, such as recognition of a tape mark, and does not necessarily indicate an error.

During execution of an I/O operation, unit-exception has only one meaning for any particular command and type of I/O-device. A sense operation shall not be required as a result of unit-exception status.

A unit-exception condition may be generated when the I/O-device is executing an I/O operation or when the I/O-device is involved with some activity associated with an I/O operation and the condition is of immediate significance. If an I/O-device detects a unit-exception condition before a command has been accepted, status of unit-exception with neither channel-end nor device-end shall be returned to the channel. This status indicates that no action has been taken at the I/O-device in response to the command. If the condition that precludes normal execution of the operation occurs after the command has been accepted, unit-exception shall be accompanied by channel-end or device-end or both, depending on when the condition is detected. Any unusual condition associated with an operation but detected after device-end is accepted by the channel shall be indicated by a status-byte containing unit-exception and attention.

If the I/O-device responds with busy status to a command, the generation of unit-exception shall be suppressed even when execution of that command usually causes unit-exception to be indicated.

Concluding an operation with the unit-exception indication shall cause command-chaining and command-retry to be suppressed.

12.2.3.3 Status-parameter field

The status-parameter field is a 16-bit field that may contain either a transfer count or control-unit queuing information. If the queuing-information-valid code is not indicated in the FFC and the XV bit is set to zero, the control unit shall set the status-parameter field to zero, and a channel receiving the status transfer shall ignore the status-parameter field.

Transfer-count: The transfer-count is the total number of bytes sent or received by the control unit for the current CCW. The transfer count shall be meaningful only when the transfer-count-valid (XV) bit is one.

Queuing information: Queuing information represents the total time a command is queued at the control unit. When control-unit queuing is provided, I/O operations specified by a channel program are executed up to a point where I/O resources are unavailable to execute an I/O operation in that channel program (such resources would typically be in use executing I/O operations specified in other channel programs). Depending upon the control-unit design, queuing of a command may occur after receipt of the first command or after receipt of a subsequent command within a channel program. I/O operations executed up to and including the command to be queued may involve data transfer as a normal course of I/O operation execution. In all cases, queuing of a command for a channel program must not effect ongoing I/O operations or cause the sequential order of I/O execution to be altered. Queuing occurs only once within execution of a channel program and can occur only between execution of successive I/O operations; however, the point in execution chosen and the length of the queuing time are model independent. When a command is queued, it remains queued at the control unit until either the needed resources become available or when a system-reset or selective-reset function has been executed (see 13.2.3 and 13.2.4). In all cases, when queuing occurs, the control unit ends execution of the command to be queued by presenting status containing channel-end without device-end (this status may or may not include command retry status). At this time, execution of the command is completed at the channel. When the unavailable I/O resources become available, the command is dequeued, the specified I/O operation is executed, and is completed at the I/O-device when device-end status is presented to the channel, along with queuing information. The control unit retains the queuing information until it receives acknowledgment that the status transfer containing this information has been accepted (see 13.2.8). At this time additional commands may be transferred by the channel continuing execution of the channel program.

The queuing information shall be meaningful only when the flag-field code is set to the value b'001' and the device-end status bit is set to one in the accompanying status-byte.

When the status-parameter field contains queuing information the format of the queuing information is shown in figure 41.

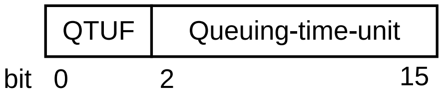


Figure 41 – Queuing information

Bits 0 and 1 provide a two-bit queuing-time-unit factor (QTUF) for the 14-bit value of the queuing time unit, as indicated in table 27.

Bits 2 through 15 of the queuing information contain the value of the queuing-time-unit to be adjusted by the QTUF to obtain the measured control-unit queuing time.

A queuing time of zero is indicated by setting the queuing-time-unit factor to b‘00’ and the queuing-time-unit to x’0000’.

Table 27 – Meaning of QTU bit 0

When QTUF is:	QTU bit 15 represents:
b’0 0’	128 microseconds
b’0 1’	512 microseconds
b’1 0’	2,048 microseconds
b’1 1’	32,768 microseconds

A control unit that performs queuing shall have a queuing-timing facility which is capable of measuring a time interval of up to its expected maximum queuing time, not exceeding 536.87 seconds. The control unit shall normally use the smallest queuing-time-unit factor in the status-parameter field that will yield the correct total queuing time.

When a control unit that performs queuing detects an error such that meaningful queuing time cannot be provided, the control unit shall set the queuing-information-valid code in the flag-field code, set the queuing information to the value x’8000’ in the status-parameter field, and send unit-check status in the status transfer that follows a request for reconnection. The control unit shall provide an indication of queuing-timing-facility failure in the sense information associated with the unit-check to allow for device-dependent ULP recovery. When unit-check status has been presented and accepted, the control unit may disable its queuing-timing facility and process subsequent I/O operations without queuing until this failure condition is corrected, or the control unit may continue queuing the subsequent I/O operations and present an indication of meaningless queuing time by setting the queueing-information-valid code in the flag-field code and setting the queuing information to the value x‘8000’ in the status-parameter field of the status transfer that follows a request for a reconnection. (In the latter case, unit-check status is not presented after the initial indication of queuing information not meaningful.)

When a control unit that performs queuing, detects an queuing-timing-facility overflow, the control unit shall set the queuing-information-valid code in the flag-field code and the queuing information to the value x’8000’ in the status-parameter field of the status transfer that follows a request for reconnection. (Unit-check is not indicated for this condition.)

12.2.3.4 Supplemental-status DIB structure

Supplemental-status provides additional information concerning conditions at the control unit or I/O-device for which status information is being provided. Supplemental-status exists only for unit-check status. When supplemental-status is included in a status frame, it shall be sent in the supplemental-status field. When supplemental-status is not included in the status frame with its associated status information, the supplemental-status shall be held by the control unit until one of the following occurs:

a) It is read by the appropriate sense command;

b) It is obtained by a status frame following notification by the channel that it is able to receive supplemental-status (see 13.1.3.2);

c) Its associated status is reset.

The supplemental-status field shall be valid when the ES bit, bit 4 of the information-field identifier, is set to one in a status frame. Up to 32 bytes of sense information may be sent in the supplemental-status field. The contents and exact number of bytes in the supplemental-status field shall be model-dependent (see 12.2.3.5).

When sense data is sent as supplemental-status, it shall be sent in ascending order starting with sense-data byte 0 as the first byte of the supplemental-status field; it shall be sent in a single status frame along with the status-byte. The size of the DIB used for the status frame with supplemental-status depends on the quantity of sense data for the particular I/O-device and shall be model-dependent. The length of sense data sent in a status frame with supplemental-status shall be from a minimum of one byte to a maximum of 32 bytes. The DIB size used for the status frame with supplemental-status shall be four bytes greater than the length of the sense data. A device-level error shall be detected for a status frame received with the ES bit set to one and a DIB size of less than five bytes or more than 36 bytes (see 12.2.3.5).

Supplemental-status shall not be used to report sense data associated with a resetting-event indication (see 13.4).

A control unit shall send supplemental-status in a single status frame. If the channel indicates the ability to accept supplemental-status, then the channel shall be capable of accepting a frame with a DIB size containing the maximum amount of supplemental-status (see 13.1.3.2).

12.2.3.5 Sense information

Sense information is the data associated with unit-check status (see 12.2.3.2.7). The sense data associated with a unit-check provides more detailed information than that supplied by the status-byte and may describe reasons for the unit-check indication. It may also indicate, for example, that the I/O-device is in the not-ready state, that a drive is in the write protected state, or to provide information to facilitate recovery. Sense data is transferred either during a basic sense operation, or, when permitted by the channel, as supplemental-status in the DIB of a status frame.

Sense byte 0 shall be the first sense byte transferred from the I/O-device to the channel.

Bits 0-5 of the first sense-data byte (sense byte 0) are common to all I/O-devices. The six bits are independent of each other and, when set to ones, specify the conditions listed in table 28.

Table 28 – Sense byte 0

Sense Bit	Description
0	Command-reject
1	Intervention-required
2	Bus-out-check
3	Equipment-check
4	Data-check
5	Overrun
6	model-dependent
7	model-dependent

Presentation of sense data as supplemental-status is described in 13.1.3.2.

All I/O-devices shall provide at least the first basic sense byte and may transfer up to 31 additional sense bytes during execution of the basic sense command. The amount and the meaning of the additional sense data and sense byte 0 bits 6 and 7 shall be model-dependent.

Sense data that pertains to a previous I/O operation or other unit action at an I/O-device may be reset any time after the completion of the basic sense command addressed to that I/O-device. Except for the no-operation and sense-ID command, any other command addressed to the control unit shall be allowed to reset the sense information, provided that the busy bit is not indicated in the status. The sense information may also be changed as a result of asynchronous actions, such as when not-ready-to-ready state-transition status is generated.

Sense information that results from more than one action at the control unit shall not be ORed when this condition would cause the ULP to misinterpret the original meaning and intent of the sense information.

12.2.3.5.1 Command-reject

The command-reject condition occurs when the I/O-device has detected a ULP error. The command-reject condition shall be generated when a command is received which:

a) The I/O-device is not designed to execute (such as a read backward issued to a direct-access storage I/O-device);

b) The I/O-device cannot execute because of its present state (such as a write issued to a write-protected tape unit).

In case b, the ULP may have required use of an uninstalled optional feature or may have specified invalid control data. Command-reject is also indicated when an invalid sequence of commands is recognized (such as a write to a direct-access storage I/O-device without the data block having previously been designated).

12.2.3.5.2 Intervention-required

The intervention-required condition occurs when the command could not be executed because of a condition that requires intervention at the I/O-device. It may indicate a condition such as a printer that is out of paper. The condition may also be generated when the addressed I/O-device is in the not-ready state, in test mode, or a required function is not provided on the control unit.

12.2.3.5.3 Bus-out-check

The bus-out-check condition occurs when the I/O-device or the control unit recognizes certain error conditions on the channel path.

A bus-out-check condition may indicate that the I/O-device has received a frame with invalid CRC over the channel path. During writing, bus-out-check shall indicate that incorrect data may have been recorded at the I/O-device, but the condition does not cause the operation to be terminated prematurely unless the operation is such that an error precludes meaningful continuation of the operation. Invalid CRC shall cause the operation to be immediately terminated and shall suppress checking for command-reject and intervention-required conditions.

For the SBCON I/O interface, bus-out-check may also indicate an error in the protocols defined for SBCON.

12.2.3.5.4 Equipment-check

During the last operation, the I/O-device detected equipment malfunctioning, such as a printer-buffer parity error. The equipment-check condition shall occur when an equipment malfunction has been detected between the I/O interface and the I/O medium. On write operations, this malfunction may have caused invalid data to be recorded. Detection of equipment-check shall stop data transmission and shall terminate the operation prematurely when the error prevents any meaningful continuation of the operation.

12.2.3.5.5 Data-check

The data-check condition occurs when invalid data has been detected by the control unit or I/O-device. Data-check identifies errors associated with the recording medium and includes conditions such as detecting invalid checking-block-code (CBC) on data recorded on magnetic tape.

Data errors on reading and writing shall cause the operation to be terminated prematurely only when the errors prevent meaningful continuation of the operation.

On an input operation, data-check indicates that incorrect data may have been placed in main storage. The I/O-device shall force correct CBC on data sent to the channel subsystem. On writing, this condition indicates that incorrect data may have been recorded at the I/O-device. Unless the operation is of a type where the error precludes meaningful continuation, data errors on reading and writing shall not cause the operation to be terminated prematurely.

12.2.3.5.6 Overrun

The overrun condition occurs when the channel subsystem fails to respond to the control unit in the anticipated time interval to a request for service from the I/O-device. When the total activity initiated by the ULP exceeds the capability of the channel subsystem, an overrun may occur when data is transferred to or from a control unit that is not buffered. An overrun condition also may occur when the I/O-device receives the new command too late during command-chaining.

12.2.4 Control-frame DIB structure

A device-control frame is used by a channel or control unit to provide device-level control for an I/O operation. The format of the information field of a device-control frame is shown in figure 38. The DIB used for a device-control frame consists of a one-byte control-function field and a three-byte control-parameter field.

The first byte immediately following the device-header of a device-control frame contains the control function, which is interpreted in conjunction with the bits in the device-header flag field to determine the device-level function to be performed and the format of the control-parameter field. A summary of the functions represented by coding of bits 0-4 of the control-function field is shown in table 29.

The setting of bits 5, 6, and 7 of the control-function field depends on the control function specified. For a command-response frame, the use of bits 5, 6, and 7 of the control-function field is described in 12.2.4.2. For all other device-control frames, bits 5, 6, and 7 of the control-function field shall be set to zeros by the sender of the device-control frame and are ignored by the recipient. Table 30 provides a definition, in summary form, of the IFI bits, device-header flag bits and frame delimiters used for the various control functions. The recipient of a device-control frame shall check the IFI bits, device-header flag bits, and delimiters according to the description contained in table 30.

If a reserved or non-supported vendor unique combination of bits 0-4 of the control-function field is recognized, a device-level error shall be detected.

NOTE – Support of vendor unique combinations of bits 0-4 of the control-function field is model-dependent. A particular model may support one or more of these combinations.

Table 29 – Summary of device-control functions

Bits[1]					
0	**1**	**2**	**3**	**4**	**Control Function**
0	0	0	0	0	Multipurpose
0	0	0	1	0	Command response
0	0	1	0	0	Stack-status
0	0	1	1	0	Cancel
0	1	0	0	0	System-reset
0	1	0	1	0	Selective-reset
0	1	1	0	0	Request-connection
0	1	1	1	0	Request-status
1	0	0	0	0	Device-level-exception
1	0	1	0	0	Status-accepted
1	0	1	1	0	Device-level-acknowledgment
1	1	0	0	0	
		through			vendor unique
1	1	1	1	1	

NOTE – 1: All combinations of bits 0-4 that are not listed above are reserved.

12.2.4.1 Multipurpose device-control function

The multipurpose device-control frame indicates that the device-control function to be performed is determined solely by the device-header flag bits, in conjunction with the context in which the frame is received. The following functions are indicated by a multipurpose device-control frame:

a) Accept command response;

b) Accept connection;

c) Deny connection;

d) Rescind connection;

e) Data request.

The control-parameter field for a multipurpose device-control frame is used only for a data request. For any multipurpose control function other than data request, the control-parameter field shall be set to zero by the sender of the frame

	IFI								Device-header flags								From	SOF	EOF	Control Parameter Field
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7				
Control Function				AS	ES		T2	T1	E	EB	RDY	CH	DR							
Multipurpose																				
Accept-command-response	0	0	0	1	z	0	1	1	z	z	1	z	z	z	z	z	CH	P	P	z
Accept-connection	0	0	0	X	U	0	1	1	z	z	1	z	z	z	z	z	CH	P	P	z
Deny-connection	0	0	0	X	z	0	1	1	z	z	0	z	z	z	z	z	CH	P	D	z
Rescind-connection	0	0	0	X	z	0	1	1	z	1	z	z	z	z	z	z	CU	P	D	z
Data-request	0	0	0	1	z	0	1	1	W	z	z	z	1	z	z	z	CC	P	P	#
Command-response	0	0	0	1	z	0	1	1	X	z	z	z	X	z	z	z	CU	P	P	@
Stack-status	0	0	0	X	z	0	1	1	z	z	z	z	z	z	z	z	CH	P	P	z
Cancel	0	0	0	1	z	0	1	1	z	z	z	z	z	z	z	z	CH	Y	P	z
System-reset	0	0	0	0	z	0	1	1	z	z	z	z	z	z	z	z	CH	Y	P	z
Selective-reset	0	0	0	1	z	0	1	1	z	z	z	z	z	z	z	z	CH	Y	P	¢
Request-connection	0	0	0	X	z	0	1	1	z	z	z	z	z	z	z	z	CU	C	P	z
Request-status	0	0	0	V	X	0	1	1	z	z	z	z	z	z	z	z	CH	Y	P	z
Device-level-exception																				
Address-exception	0	0	0	1	z	0	1	1	z	1	z	z	z	z	z	z	CU	P	D	$
Status-accepted	0	0	0	X	z	0	1	1	z	z	z	X	z	z	z	z	CH	P	P	z
Device-level-acknowledgment	0	0	0	X	z	0	1	1	z	1	z	z	z	z	z	z	CU	P	D	z

Table 30 – Summary of device-control frames

Explanation:

#	Includes data count in control-parameter field.
@	Includes DIB-size control in bits 5, 6, and 7 of the control-function field. For write operations, includes data-request count and pacing count in control-parameter field. For read operations, includes NDR-R count in control-parameter field.
¢	Includes modifier bits in control-parameter field.
$	Includes exception code in control-parameter field.
C	Connect-SOF delimiter.
CC	Either control unit or channel sends frame.
CH	Channel sends frame to control unit.
CU	Control unit sends frame to channel.
D	Disconnect-EOF delimiter.
P	Passive-SOF or passive-EOF delimiter.
U	Bit is set to zero; otherwise, unpredictable results
V	Bit is set to one; otherwise, unpredictable results
W	Bit is set to one if CCW count is satisfied by this data request; otherwise, the bit is set to zero.
X	Bit is set to one or zero as appropriate for conditions
Y	Connect-SOF delimiter when no connection exists; passive-SOF delimiter when a connection exists.
z	Bit or field is always set to zero by the sender and is ignored by the recipient.
0	Bit is always set to zero and is checked for zero.
1	Bit is always set to one and is checked for one.

and shall be ignored by the recipient of the frame.

The setting of the bits in the IFI field and the device-header flag field depend on the function being indicated by the multipurpose device-control frame and the context in which the frame was received. A device-level error shall be detected if the combination of bits in the IFI field and device-header flag field is inappropriate for the context in which the frame was received.

The following subclauses are descriptions of the functions performed by a multipurpose device-control frame:

12.2.4.1.1 Accept-command-response

When the channel uses a multipurpose device-control frame in response to a command response for the first command of a channel program, the channel is indicating that the command response is accepted and the control unit may proceed with execution of the I/O operation. For brevity, the frame used this way is called an accept-command-response frame (see 13.1.1).

12.2.4.1.2 Accept-connection

When the channel uses a multipurpose device-control frame with the ready bit set to one in response to a request-connection frame, the channel is indicating that the connection is accepted. For brevity, the frame used this way is called an accept-connection frame.

The channel shall set the AS bit with the accept connection to the same value as was received with the request-connection.

12.2.4.1.3 Deny-connection

When the channel uses a multipurpose device-control frame with the ready bit set to zero in response to a request-connection frame, the channel is indicating that the connection is denied. For brevity, the frame used this way is called a deny-connection frame. The channel shall set the AS bit with the deny-connection frame to the same value as was received with the request-connection frame.

12.2.4.1.4 Rescind-connection

When the control unit uses a multipurpose device-control frame with the EB bit set to one in response to an accept-connection frame, the control unit is indicating rescind connection. For brevity, the frame used this way is called a rescind-connection frame. A rescind-connection frame indicates to the channel that the connection previously requested by the control unit and granted by the channel is no longer needed, and the connection is removed. For the rescind-connection frame, the AS bit shall be set to the same value as the AS bit in the accept-connection frame; otherwise, a device-level error shall be detected.

12.2.4.1.5 Data-request

During data transfer, the recipient of data can use the multipurpose device-control frame with the DR bit set to one to request data from the sender of data. For brevity, the frame used this way is called a data-request frame. The E bit shall be set to one or zero to indicate whether the current data request includes the end of the count for the CCW. The control-parameter field contains one byte of zeros followed by a two-byte count field. The first byte of the control-parameter field shall be ignored by the recipient of a data-request frame. The two-byte count field indicates the number of bytes requested in the current data request (see 13.1.2).

12.2.4.2 Command-response frame

The command-response (CMR) frame indicates that the command received by the control unit was accepted for execution. A command-response frame shall only be sent by a control unit. If a control unit receives a command response, a device-level error shall be detected.

Table 31 summarizes the combinations of E and DR flag bits that may be used with the command-response frame. Note that a command-response frame shall not be sent for an immediate operation.

In a command-response frame, bits 5, 6, and 7 of the control-function field indicate the maximum DIB size to be sent in each data frame sent for the current I/O operation. If a value is specified in the DIB-size field of the command-response frame which exceeds the maximum DIB size specified by the channel when the logical path was established, a device-level error shall be detected by the channel. The meaning of the bit settings are explained in table 32.

The all-ones combination of bits 5, 6, and 7 is reserved; if a command-response frame is re-

Table 31 – Summary of device-header flags used with command response

Bit		
E	DR	Description
0	0	Command accepted (read), or command accepted with no data request yet (write)
0	1	Data request for less than CCW count (write operation)
1	0	End of operation with no data transfer (zero CCW count)
1	1	Data request for entire CCW count (write operation)

Table 32 – Maximum DIB size

Bits 5 6 7	DIB Size in bytes
0 0 0	16
0 0 1	32
0 1 0	64
0 1 1	128
1 0 0	256
1 0 1	512
1 1 0	1,024
1 1 1	reserved

ceived with this combination of bits, a device-level error shall be detected.

The format of the control-parameter field for a command-response frame depends on whether the I/O operation is a read operation or a write operation.

12.2.4.2.1 CMR frame for write operation

In a command-response frame used for a write operation, the format of the control-parameter field is shown in figure 42**.**

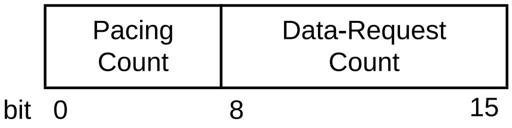


Figure 42 – CMR frame control-parameter field for write

12.2.4.2.2 CMR frame for read operation

For command responses used for read operations, the format of the control-parameter field is shown in figure 43.

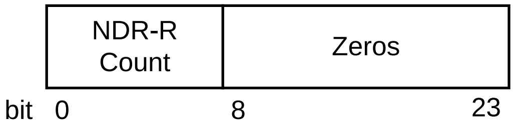


Figure 43 – CMR frame control-parameter field for read

For read operations, bits 8-23 are reserved, shall be set to zeros by the sender and shall be ignored by the recipient of the frame.

12.2.4.2.3 Pacing-count

Bits 0-7, specify the minimum number of pairs of idle characters to be used between successive data frames. The minimum number of pairs of idle characters that may be used between frames is two, which is represented by a count of x’02’. The occurrence of fewer idle characters than the minimum specified by the pacing count may result in link-level or device-level errors being detected. If the pacing count specified is less than two, a device-level error shall be detected (see 13.1.2.4).

12.2.4.2.4 Data-request-count

Bits 8-23 specify the number of bytes requested by an I/O-device during a write operation (see 13.1.2 and 13.1.2.1).

12.2.4.2.5 Number-of-data-requests (NDR-R-count)

For read operations, bits 0-7 of the control-parameter field specify the number of data requests beyond one that the control unit permits to be outstanding at one time. A data request shall be outstanding from the time it is sent until a ready indication is received in a data frame and there are no previous data requests outstanding for that CCW. A value of zero in this field indicates that one and only one data request may be sent at any one time and that a new data request may be sent only after a ready indication is received in a data frame.

12.2.4.3 Stack-status frame

The stack-status frame indicates that the channel does not accept the status. The control unit or I/O-device shall hold the status information (status-flags, status-byte, and status-parameter) and associated supplemental-status, if any, until the status is requested by a request-status frame, until the status is presented in a status frame as the response to a command frame, or until the status information is cleared or withdrawn (see 13.2.1).

A stack-status frame may be sent by the channel in response to a status frame. If the channel receives a stack-status frame, a device-level error shall be detected. If the control unit receives a stack-status frame in response to other than a status frame, a device-level error shall be detected.

The AS bit may be set to either one or zero. The setting of the AS bit in the stack-status frame is the same as the setting of the AS bit in the status frame for which the stack status is sent in response; otherwise, a device-level error shall be detected.

12.2.4.4 Cancel frame

The cancel frame causes the control unit to perform the cancel function for the specified I/O-device (see 13.2.2). A cancel frame shall be sent only by a channel; if a cancel frame is received by a channel, a device-level error shall be detected.

12.2.4.5 System-reset frame

The system-reset frame indicates that the control unit and associated I/O-devices shall be reset with respect to the logical path on which the function was received (see 13.2.3). A system-reset shall be sent only by a channel; if a system-reset is received by a channel, a device-level error shall be detected.

12.2.4.6 Selective-reset frame

The selective-reset frame causes the control unit to end execution of the current operation, if any, for the specified I/O-device, and, depending on the bits within the control-parameter field, to perform one of the following functions:

– selective-reset;

– channel-initiated-retry;

– channel-initiated-unit-check.

(See 13.2.4 and 13.3.3.)

The control-parameter field for a selective-reset frame is shown in figure 44.

	RC	0	0	RU	RO	Reserved
bit	0	1	2	3	4	5 23

Figure 44 – Selective-reset frame control-parameter field

Bits 1, 2, and 5-23 shall be set to zeros by the sender and ignored by the recipient.

When the RC bit, RU bit, and RO bit of the control-parameter field are all set to zeros, a selective-reset shall be performed.

When one or more of the RC, RU, and RO bits of the control-parameter field are set to ones and the requested function or functions cannot be performed, a selective-reset shall be performed.

When the RC bit of the control-parameter field is set to one, a channel-initiated retry shall be performed, if possible; if retry is not possible, the RU bit and RO bit of the control-parameter field shall determine the preferred method of terminating the I/O operation.

When either the RU bit or the RO bit of the control-parameter field is set to one and channel-initiated retry is either not requested or requested and not performed, a channel-initiated unit-check shall be performed, if possible; if channel-initiated unit-check is not performed, a selective-reset shall be performed.

A selective-reset frame shall be sent only by a channel; if a channel receives a selective-reset frame, a device-level error shall be detected.

Request-command-retry (RC): The RC bit, bit 0, when set to one, indicates that the channel is requesting the I/O-device to perform command-retry on behalf of the channel.

If the RC bit is set to one, the I/O-device shall perform command-retry if it can; if command-retry cannot be performed, the RU and RO bits shall determine the action to be taken. If the RC bit is set to zero, the RU and RO bits shall determine the action to be taken.

Request-unit-check (RU): Bit 3, when set to one, indicates that the channel is requesting the I/O-device to terminate the I/O operation on behalf of the channel with unit-check status.

Request-unit-check-with-overrun (RO): Bit 4, when set to one, indicates that the channel is requesting the I/O-device to recognize an overrun condition and terminate the I/O operation on behalf of the channel with unit-check status.

The channel may set to one either the RO bit or the RU bit depending on the error condition. When command-retry is either not requested or requested and not performed, the I/O-device interprets the RU and RO bits in the control-parameter field as indicated in table 33.

Table 33 – RU and RO bit meaning

RU	RO	Function Performed
0	0	A selective-reset shall be performed.
0	1	The I/O-device recognizes an overrun condition and shall generate unit-check status with the appropriate sense data. Retry may be requested.
1	0	The I/O-device recognizes a unit-check and shall generate unit-check status with the appropriate sense data.
1	1	*Error:* the I/O-device shall ignore the requested functions and shall perform a selective-reset.

12.2.4.7 Request-connection frame

The request-connection frame requests a connection to the channel in order to present status. The status may be either asynchronous status or status related to an I/O operation. A request-connection frame shall be sent only by a control unit; if a request-connection frame is received by a control unit, a device-level error shall be detected.

When a request-connection frame is specified, the AS bit is set to either one or zero. When the AS bit is set to one, the requested connection shall be only for the specified I/O-device. When the AS bit is set to zero, the requested connection shall be for an I/O-device that will be specified after the connection is granted.

12.2.4.8 Request-status frame

The request-status frame indicates that the channel is prepared to have status information presented for the addressed I/O-device (see 13.2.1 and 13.2.6). Request-status shall only be sent by the channel; if a request-status frame is received by the channel, a device-level error shall be detected.

The ES bit indicates whether the channel can accept supplemental-status in the response to the request-status frame. When the ES bit is set to one, the channel shall have the capability of accepting supplemental-status in a status frame sent in response to the request-status frame; otherwise, not.

When the channel receives a status frame with the supplemental-status-available (SA) status-flag bit set to one, the channel may retrieve the supplemental-status during the same connection by sending a request-status frame with the ES bit set to one (see 13.1.3.2). If a control unit receives a request-status frame during the same connection with the ES bit set to zero, the control unit shall detect a device-level error. The request-status frame in response to a status frame shall have a passive-SOF delimiter.

When the channel has not maintained the connection, a request-status frame shall have a connect-SOF delimiter.

12.2.4.9 Device-level-exception frame

The device-level-exception frame indicates that a certain abnormal condition was recognized in the device frame received for which a status frame is inappropriate or not permitted. The abnormal condition is indicated by the exception code present in the first byte of the control-parameter field.

A device-level exception has a control-parameter field with this format:

The format of the control-parameter field for a device-level exception is shown in figure 45.

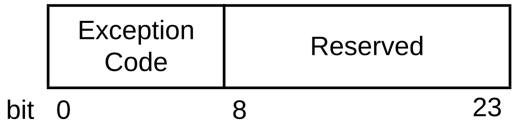


Figure 45 – Device-level exception frame control-parameter field

Bits 8-23 of the control-parameter field shall be set to zeros by the sender of the device-level-exception frame and shall be ignored by the recipient of the frame.

Exception code: Bits 0-7 specify the abnormal condition detected in the device frame for which this device-level-exception frame is the response. Table 34 shows the encoding of the exception codes.

Table 34 – Device-level-exception frame exception codes

Bits								Exception Code
0	1	2	3	4	5	6	7	
0	0	0	0	0	0	0	0	reserved
0	0	0	0	0	0	0	1	Address-exception
0	0	0	0	0	0	1	0 through 1 1 1 1 1 1 1 1	reserved

If a reserved exception code is used, a device-level error is detected by the recipient of the device-level-exception frame.

Address-exception: Address exception indicates that the device frame received contained the device-address of an uninstalled I/O-device or an I/O-device that is offline with respect to the logical path. A device-level-exception frame for an address exception shall be sent only by a control unit in response to certain device frames (see 13.3.4 for a description of the use of this exception code). If a device-level-exception frame for an address exception is received by a control unit, a device-level error shall be detected.

12.2.4.10 Status-accepted frame

The status-accepted frame indicates that the channel has accepted the status from a status frame. A status-accepted frame shall be sent only by a channel. If a status-accepted frame is received by a channel, a device-level error shall be detected.

The CH bit in the device-header flag field for a status-accepted frame may be set to either one or zero, depending on conditions at the channel (see 13.2.8).

The AS bit may be set to either one or zero. The setting of the AS bit in the status-accepted frame shall be the same as the setting of the AS bit in the status frame for which the status-accepted frame is sent in response; otherwise, a device-level error shall be detected.

12.2.4.11 Device-level-acknowledgment frame

Device-level acknowledgment (device-level ACK) is used for the control-unit response to stack status, cancel, system-reset, selective-reset, request status, or status accepted. Device-level ACK, when sent in response to a stack status, cancel, system-reset, selective-reset, or status accepted frame, indicates that the requested function was or will be performed. A device-level-ACK frame sent in response to a request-status frame which initiated a connection shall indicate that the requested function was or is to be performed. If a channel recognizes a device-level-ACK frame in response to a request-status frame which does not initiate a connection, the channel shall detect a device-level error.

A device-level-ACK frame shall be sent only by a control unit to a channel; if a control unit receives a device-level ACK, a device-level error shall be detected.

The AS bit may be set to either one or zero. The setting of the AS bit in the device-level-ACK frame shall be the same as the setting of the AS bit in the frame for which the device-level ACK is sent in response; otherwise, a device-level error shall be detected.

12.3 Device-frame delimiters

The delimiters used for a device frame are determined by the frame type, and additionally, in some cases, by the particular device-level function being performed or conditions at the channel when the device frame is used.

A data frame shall have a passive-SOF delimiter and a passive-EOF delimiter.

A command frame may have either a connect-SOF delimiter or a passive-SOF delimiter. A connect-SOF delimiter shall be used when the CH bit is set to zero. A passive-SOF delimiter shall be used when the CH bit is set to one. A command frame shall have a passive-EOF delimiter.

A status frame shall have a passive-SOF delimiter and a passive-EOF delimiter.

A device-control frame may have either a connect-SOF delimiter or a passive-SOF delimiter, and either a passive-EOF delimiter or a disconnect-EOF delimiter, depending on the particular control function being performed. The frame delimiters used for device-control frames are shown in table 30. A request-status frame, selective-reset frame, cancel frame, and system-reset frame may have either a connect-SOF delimiter or a passive-SOF delimiter, depending on whether a connection exists. If a connection does not exist, these frames shall be sent with a connect-SOF delimiter and if a connection exists, these frames shall be sent with a passive-SOF delimiter.

The frame delimiters used with the different device frames are summarized in table 35, and, for device-control frames, in table 30. If a device frame is received with delimiters other than as shown in these figures, a device-level error shall be detected.

Table 35 – Device-frame delimiter summary

Device-Frame Type	SOF Delimiter	SOF Delimiter	Used for
Data	Passive	Passive	Data transfer
Command	Connect	Passive	Channel program initiation (CH=0)
	Passive	Passive	Chaining (CH=1)
Status	Passive	Passive	Status transfer
Control	(See table 30)		

13 Device-level functions and protocols

This clause describes the functions and protocols necessary for the execution of an I/O operation, the exchange of control information, and device-level recovery. These functions depend on the successful completion of link-level protocols necessary to initiate device-level communication.

13.1 Device-level operations

Initiation of an I/O operation, the transfer of data, and the ending of an I/O operation are device-level operations which rely on device-level functions and protocols. The device-level functions discussed in this section have all been defined in clause 12. This subclause describes how these functions are used in order to perform device-level operations.

13.1.1 Initiating an I/O operation

An I/O operation is initiated with an I/O device when the channel transfers the command from the current CCW over the SBCON I/O interface to that I/O device.

When the control unit receives a command, at least the following conditions shall be satisfied in order for a response indicating acceptance of the command to be returned to the channel:

a) The AS bit is set to one in the IFI;

b) For those commands that require the I/O device to be installed and ready, the device address specifies an I/O device that is installed and ready;

c) For command-chaining, the device address designates the same I/O device that was designated for the command frame which initiated the channel program;

d) The chaining (CH) bit is set to zero for the first command of a channel program and set to one for all subsequent commands of the channel program;

e) For those I/O devices that provide commands executed as immediate operations, the command is checked to determine whether a command response or status is to be sent.

At least the following conditions shall be satisfied in order for the control unit to return busy status to the channel:

a) The status frame indicating device busy or control-unit-busy (status-modifier and busy) may be sent only in response to the first command of a channel program (CH bit set to zero in the command frame);

b) A response of control-unit-busy requires that the IFI of the command frame be checked. No other device-level checking is required;

c) A response of device busy requires that the IFI and device address be checked. No other device-level checking is required.

When the command is the first command of the channel program, the channel shall consider a connection to exist when the command response or status is received. The control unit shall consider a connection to exist when the command-response frame or status frame is sent.

For the first command of a command chain, an I/O operation is being initiated by an I/O device from the time the control unit returns a status response or command response that indicates the command is accepted, but the execution of the I/O operation shall not begin until the control unit receives acknowledgment that the status response or command response has been accepted by the channel.

When the command is the first command of the channel program, the channel shall consider the I/O operation to be in progress at the I/O device when the channel sends the appropriate response indicating the command response or status was accepted. The I/O device shall consider the I/O operation to be in progress and execution of the first command of the channel program to be permitted when the appropriate response indicating acceptance of the command response or status is received from the channel.

When the first command of a channel program is handled by the control unit as a nonimmediate operation, the control unit shall indicate acceptance of the command for execution by sending a command response to the channel. The channel shall indicate that the I/O operation is considered in progress, thus allowing execution of the command, by sending an accept-command-response frame to the control unit. If the operation is a read, data requested by the channel may be transferred after the control unit receives the accept-command-response frame. If the operation is a write, data requested by the control unit may be transferred after the channel sends the accept-command-response frame.

When the first command of a channel program is handled by the control unit as an immediate operation, the I/O device shall indicate acceptance of the command for execution by sending a status frame indicating channel-end status without busy status. If device-end status is not included with the channel-end status, the channel shall indicate that the status is accepted and that execution of the command is permitted by sending a status-accepted frame. The chaining bit (CH) shall be set to one in the status-accepted frame if chaining is to be indicated by the channel on acceptance of the status; if chaining is not to be indicated, the CH bit shall be set to zero on acceptance of the status.

If device-end status is included with the channel-end status for an immediate operation, the response from the channel indicating status is accepted and execution of the command is allowed depends on whether chaining is to be indicated by the channel. If chaining is not to be indicated, the channel shall send a status-accepted frame with the CH bit set to zero. If chaining is to be indicated by the channel, then the channel shall indicate status is accepted and execution of the command is permitted by sending the next command in a command frame with the CH bit set to one.

When the status frame received by the channel in response to the first command frame of a channel program indicates that supplemental-status is available, the status may be accepted as previously described, or the channel may request the control unit to return the supplemental-status by sending a request-status frame with the ES bit set to one. (Sending the request-status frame with the ES bit set to one does not indicate the channel has accepted status.) The channel shall indicate status is accepted by sending a status-accepted frame when the status frame is received in response to the request-status frame with the ES bit set to one (see 13.1.3.2).

When the status frame received by the channel in response to the first command of a channel program requests that command-retry be performed, the channel shall indicate acceptance of status and its intent to perform the retry by sending either a status-accepted frame or a command frame, depending on whether device-end status is present in the status frame and on whether the channel intends to perform the command-retry. The response from the channel shall depend on the following conditions:

a) If device-end status is not present, the channel shall indicate acceptance of the status by sending a status-accepted frame. If command-retry is to be performed, the CH bit shall also set be to one in the status-accepted frame. From the time when the status-accepted frame is sent, the channel shall consider the I/O operation in progress at the I/O device and the rules for command-chaining in effect (see 13.1.4). Conversely, if command-retry is not to be performed, the CH bit shall be set to zero in the status-accepted frame;

b) If device-end status is present and command-retry is to be performed, the channel shall indicate acceptance of the status and of the command-retry by sending a command frame with the command to be retried, along with the flags and parameters associated with the command. The CH bit shall be set to one in the command frame to identify this command as the one being retried. From the time the command frame is sent to the I/O device, the channel shall consider the I/O operation in progress at the I/O device and the rules for command-chaining in effect (see 13.3.2);

c) If device-end status is present and command-retry is not to be performed, the channel shall indicate acceptance of the status by sending a status-accepted frame. The CH bit shall be set to zero in the status-accepted frame to indicate that command-retry is not to be performed.

The following protocols shall be used for commands which are not executed as the first command of a channel program:

a) When a command which is not executed as the first command of the channel program is handled by the control unit as a nonimmediate operation, the control unit shall indicate acceptance of the command for execution by sending a command-response frame to the channel. Execution of the command may proceed as soon as the command frame is received and all error checking is completed. The I/O operation shall be considered in progress by the control unit when the control unit sends the command-response frame. The channel shall consider the I/O operation in progress when it receives the command-response frame from the control unit. If the operation is a read, the control unit may, immediately after sending the command-response frame, proceed to transfer the data that the channel requested without receiving a response from the channel to the control-unit command-response frame (as was done for the first command of the channel program). If the operation is a write, the channel may immediately proceed to transfer the data requested by the I/O device, providing the DR bit was set to one in the command-response frame. The channel shall consider the operation in progress at the I/O device when it receives the command-response frame from the control unit, even if the channel cannot send the requested data. The I/O device shall consider the operation in progress when it sends a command-response frame;

b) When a command that is not handled as the first command of the channel program is treated by the control unit as an immediate operation, the control unit indicates acceptance of the command for execution by sending to the channel a status frame showing channel-end status without busy status. Execution of the command may proceed as soon as the command frame is received and all error checking is completed. The I/O operation shall be considered in progress when the control unit sends channel-end status without busy status to the channel. Because execution of the command is permitted to start before the status frame is sent, the I/O operation may be completed at the I/O device at the time the status is to be sent, in which case, device-end status may be sent together with channel-end status.

c) When the channel receives channel-end and device-end status together without busy status, the I/O operation is considered to have started and completed without having been considered to be in progress at the channel.

13.1.2 Data-transfer protocol

Data is transferred between the channel and the control unit as part of the execution of an I/O operation. The transfer of data from the channel to the control unit is a write operation. The transfer of data from the control unit to the channel is a read operation. The operation, read or write, shall be determined by the current command. Read commands have bit 7 of the command byte set to zero, and write commands have bit 7 of the command byte set to one.

Some read and write commands do not result in the transfer of data. These commands are executed as immediate operations, are designed to be executed without transferring data, have counts of zero in the CCW, or have data-record lengths of zero (see 13.1.1 for the protocols used when the command is executed by the control unit as an immediate operation; see 13.1.2.5 for the protocols used when the CCW count is zero). When a read or write command has a nonzero count in the CCW and the data-record length at the control unit is zero, the command shall be accepted with a command-response frame, and status of channel-end with or without device-end shall be returned before data transfer is started; the transfer-count-valid bit shall be set to one in the status frame, and the transfer count shall be set to zero in the status-parameter field. The control unit shall not send a data request or a data frame.

A data transfer results in one or more data frames being sent between the channel and control unit. The quantity of data contained in the DIB of the data frame for both read and write operations shall be determined by the control unit and shall be based on the pacing parameters and the data-transfer requirements of the I/O device and control unit (see 12.2.1, 12.2.4.2, and 13.1.2.4).

The quantity of data to be transferred during an I/O operation shall be determined by either the counts for the CCWs (operation count) that are provided for the command or the data required by the I/O device for the current command, when less than the operation count. During data transfer, the recipient of data shall have the count provided with the current CCW. When the operation is a write, the control unit receives this count in a command frame from the channel.

Data transfer shall be initiated by the recipient of the data: the channel for a read operation and the control unit for a write operation. The recipient shall initiate data transfer by requesting data and indicating the quantity of data it is capable of receiving. A request for data shall be made by setting the data request (DR) bit to one in one of the following:

- a command frame for the first request made for the current CCW when the operation is a read;
- a command-response frame for the first request made for the current CCW when the operation is a write;
- a data-request frame for the current CCW when, for a write operation, the data-request bit was zero in the command response;
- a subsequent data-request frame, if any, sent for the current CCW when the operation is either a read or write.

When data transfer is initiated, the actual sending of data frames shall not occur until the I/O operation is considered in progress (see 13.1.1).

One or more data requests may be used to request the data for the current CCW. If the first data request for the current CCW specifies a quantity of data less than the count for the current CCW and additional data is required, then one or more additional data requests shall be made in order to transfer all of the data for the current CCW. For any CCW, a recipient of data shall not make further data requests after the first until it has received at least one data frame for that CCW. The amount of data requested shall always be less than or equal to the difference between the count for the current CCW and the quantity of data previously requested for the current CCW. For example, if the count for the current CCW is 512 bytes and the first data request for this CCW was 128 bytes, the second data request is less than or equal to 384 bytes. For each CCW, the number of data requests used to transfer the total quantity of data for that CCW shall depend on the amount of data specified with each data request, the amount being model dependent.

The NDR count shall specify one less than the maximum number of data requests that the sender of data allows to be outstanding at one time. A data request shall be outstanding from the time it is sent until a data frame is received with the RDY bit set to one and there are no previous data requests outstanding for that CCW.

A value of zero for the NDR count shall indicate that only one data request may be outstanding. If additional data requests are needed for the current CCW, the next data request shall not be sent until a data frame is received with the RDY bit set to one. This is called the single-data-request mode of data transfer. A nonzero value for the NDR count shall indicate that more than one data request may be outstanding at one time. Once NDR+1 data requests have been sent, no more shall be sent until a data frame is received with the RDY bit set to one. This is called the multiple-data-request mode of data transfer.

For read operations, the NDR count shall be provided by the control unit in the command-response frame as the NDR-R count (see 12.2.4.2). For write operations, the NDR count shall be provided by the channel in the ELP frame as the NDR-W count when the logical path is established (see 9.1.1). The NDR-W count, byte 0 of the information field for the ELP function, shall specify the number of data requests beyond one that the channel permits to be outstanding at one time. A value of zero in this field shall indicate that only one data request may be sent at any one time and that a new data request may be sent only after a ready indication is received in a data frame.

For read operations, the first data request made for the current CCW shall be made in the command frame for the CCW. For write operations, the first data request made for the current CCW may be made as part of the command response if data-chaining is not in effect. Other data requests for the current CCW, if required, may be made whenever (1) the sum of the counts for data requests already made for the CCW is less than the CCW count, and (2) the number of data requests sent for the current CCW less the number of ready indications received for the current CCW does not exceed the NDR count. Whenever these conditions exist, the rate at which data requests are sent shall depend on the design of the recipient of data. The ready indication shall be sent to the recipient of data in one of the data frames by setting the RDY bit to one. Data requests for a CCW subsequent to the first data request for a CCW shall be sent in a data-request frame.

The sender of data shall use the ready (RDY) bit to control the rate at which data requests may be received. The sender of data may indicate it is ready to accept a new data request (set the RDY bit to one in a data frame) whenever a data request has been received with the E bit set to zero. When the first data request for the current CCW is made, a ready indication may be received as early as the first data frame and as late as the last data frame for that data request.

If the amount of data already requested for the CCW is less than the CCW count and if the number of data requests sent for the current CCW less the number of ready indications received for the current CCW exceeds the NDR count by one, a ready indication shall be received in one of the subsequent data frames in order to continue the I/O operation; as soon as a ready indication is received, the next data request maybe made.

For any CCW, data requests and ready indications shall be exchanged by the sender of data and recipient of data as needed until all of the data required for the current CCW is requested. No ready indication shall be sent in response to a data request that is received with the E bit set to one.

The sender of data shall compare the number of data requests received to the number of ready indications sent and to the NDR count. During data transfer, if the number of data requests received for the current CCW minus the number of ready indications sent for the current CCW exceeds the NDR count by more than one, a device-level error shall be detected.

The recipient of data shall compare the count of ready indications received in data frames to the count of data requests sent. During data transfer, if the number of ready indications is greater than the number of data requests that were sent for the CCW, the requester of data may detect a device-level error. After the end of data transfer, if the number of ready indications for a CCW is greater than the number of data requests that were sent for the CCW, the requester of data shall detect a device-level error. No error shall be detected if the number of ready indications is smaller than or equal to the number of data requests sent.

The E bit shall be set to one in a data frame which exactly completes the CCW count. If the E bit is set to one in a data frame which does not exactly complete the CCW count, or if the E bit is set to zero in a data frame which does exactly complete the CCW count, then a device-level error shall be detected.

Except for the last data frame sent for a given data request, the size of the DIB in all data frames sent for a CCW shall be the size established by the control unit for the I/O operation. If a data frame is received with less than the maximum quantity of data allowed by the DIB size in effect and the next frame received is another data frame for the same data request, the recipient shall not directly detect an error due to the short DIB size but may possibly detect an error if a problem occurs due to the resulting difference in data-transfer rate.

The last data frame sent in response to a data request may have a DIB size less than the DIB size in effect for the I/O operation, depending on the conditions at and the design of the sender. If a new data request is not received before a channel or control unit has finished sending all of the data associated with a previous data request, the DIB size of the last data frame for the previous data request shall be sufficient to contain the remaining count for either the previous data request or the data record, whichever is less; the DIB size may be from one byte up to the DIB size in effect for the I/O operation. If a new data request is received before a channel or control unit has finished sending all of the data associated with all previous data requests, it shall be model-dependent whether the new data-request count is added to the counts remaining from previous requests or whether the new count is queued and held separately until the previous data-request counts have been satisfied.

During data transfer for an I/O operation, both the sender and recipient of data shall maintain a count of the number of bytes of data transferred over the interface for the current CCW. The status frame that is sent after the data transfer for a CCW shall normally include the transfer count accrued for that CCW. The channel shall compare the transfer count included in the status frame with the count maintained by the channel for the current CCW to verify that the amount of data transferred was equal to the amount of data sent or received (see 13.1.3.3).

13.1.2.1 Write operation

A write operation is the transfer of data from the channel to the control unit as part of the execution of a command that has bit 7 set to one.

When the control unit receives and accepts a command that specifies a write operation, a command-response frame shall be sent. If the control unit is immediately ready to accept data for the command, the DR bit shall be set to one in the command-response frame, and a quantity of data shall be specified in the count field. If the quantity of data requested equals the count for the current CCW, the E bit shall also be set to one; otherwise, the E bit shall be set to zero. If the CCW count is not equal to zero and either the control unit is unable to immediately accept data from the channel or the control unit does not intend to transfer data for the command, then both the DR bit and the E bit shall be set to zeros. If the CCW count is equal to zero, the DR bit shall be set to zero, and the E bit shall be set to one.

If the control unit sets the DR bit to one in the command-response frame, the channel may immediately send data unless the write command is executed as the first command of a channel program in which case, the channel may send data immediately after sending an accept-command-response frame. The channel may send a quantity of data equal to the count specified with the data request. If the E bit is set to one and the count for the data request is equal to the count for the current CCW, the channel shall not set the RDY bit to one in any of the data frames that are sent, and the E bit shall be set to one in the last data frame that exactly satisfies the count for the current CCW (see 13.1.2).

If the DR bit is set to zero in the command-response frame, data transfer shall not occur until the control unit sends a subsequent data-request frame. For commands that are executed as the first command of a channel program, a subsequent data-request frame shall not be sent until the accept-command-response frame is received from the channel. For commands other than those executed as the first command of a channel program, the first data request may be made whenever the control unit is ready to receive data (see 13.1.1).

When the channel sends data, it shall maintain a count of the number of bytes of data sent for the current CCW and, in a similar manner, when the control unit receives data, it shall maintain a count of the number of bytes of data received for the current CCW. This count is called the transfer count. When the last data frame for the current CCW is received and the E bit is set to one, the transfer count shall be equal to the number of bytes of data received from the channel for that CCW. If data-chaining is specified, the transfer count shall be held by the control unit until the command frame is received and accepted. When the channel sends a command frame for the next CCW, the transfer count at the channel shall be reset when the frame that contains the chaining indication and the new parameters from the current CCW is sent. When the control unit receives this command frame, a new CCW shall be considered to be in effect, and the transfer count at the control unit shall be reset (see 13.1.2.3).

If during a write operation the quantity of data required by the I/O device is less than the quantity of data specified for the command, the control unit shall set the E bit to zero in the data request that exactly satisfies the quantity of data needed, and no further data requests shall be made. When the quantity of data requested for the current command is received, a status frame shall be sent containing channel-end status and the transfer count.

If the quantity of data transferred is less than or equal to the total count associated with the current command and this quantity of data is greater than the quantity of data required by the I/O device, then unit-check status, channel-end status, and the transfer count shall be included in the status frame. The transfer count shall be equal to the number of bytes of data received, not the number of bytes of data used by the I/O device.

When the last data frame with the E bit set to one is received, data-chaining is not specified, and the quantity of data required by the I/O device for the current command is greater than the quantity of data specified by the count for the current CCW, then a status frame with channel-end status, the transfer count, and the long-record (LR) bit set to one shall be sent.

If, during data transfer, an error is detected by the channel on the data to be transferred, the error shall be held pending until the control unit requests at least one byte of the data in error. When this occurs, the error condition shall be recognized, and the I/O operation shall be terminated. The channel shall send all of the data up to but not including the data in error. The E bit shall not be set to one in the last data frame sent. After the last data frame is sent, the channel shall send a cancel frame. If the control unit does not request the data in error but ends the operation with status, the error detected on the data shall not be recognized; if the error is not recognized, a cancel shall not be sent to the control unit.

13.1.2.2 Read operation

A read operation is the transfer of data from the control unit to the channel as part of the execution of a command that has bit 7 set to zero.

When the channel sends a command frame for a read operation, the channel shall initiate the read operation by setting the DR bit to one and specifying the quantity of data requested in the count field of the frame containing the command. In the case of data-chaining, the channel shall continue the read operation by setting the DR bit to one in the command frame containing the chaining indication (CH set to one) and the parameters from the current CCW. If the quantity of data requested equals the count for the current CCW, the E bit shall be set to one; otherwise, the E bit shall be set to zero.

Data transfer shall not start until the I/O operation is considered in progress. For commands that are executed as the first command of a channel program, this shall not occur until the control unit receives an accept-command-response frame. For commands other than those executed as the first command of a channel program, the I/O operation shall be considered in progress immediately after the frame containing the command response is sent (see 13.1.1).

When data transfer starts, the control unit may send a quantity of data equal to the count that was specified with the data request.

When the control unit sends data to the channel, the control unit shall maintain a count of the number of bytes of data sent for the current CCW and, in a similar manner, when the channel receives data from the control unit, the channel shall maintain a count of the number of bytes of data received for the current CCW. This count is called the transfer count. When the last data frame is received for the current CCW and the E bit is set to one, the transfer count shall be equal to the number of bytes of data received from the control unit for that CCW.

If data-chaining is specified, when the transfer count equals the quantity of data specified for the current CCW, then the channel may perform data-chaining with the control unit. When the channel sends the command frame for the next CCW, the transfer count at the channel shall be reset. When the control unit receives this command frame, a new CCW shall be considered to be in effect, and the transfer count at the control unit shall be reset. During data-chaining, if the control unit prematurely terminates an I/O operation at exactly the point that the transfer count equals the quantity of data specified for the current CCW, the control unit does not normally return status for the I/O operation until it receives the command frame with the DU bit set to one for the next CCW. The control unit shall be permitted to send a status frame without waiting for the next frame from the channel only when an error detected by the control unit results in the transfer-count-valid bit being set to zero in the status frame.

If the quantity of data at the I/O device to be transferred for the current command is less than the quantity of data requested by the channel for the command, the control unit shall set the E bit to zero in the last data frame sent to the channel and shall send a status frame containing channel-end status and the transfer count.

If the quantity of data at the I/O device to be transferred for the current command is greater than the quantity of data requested by the data request with the E bit set to one from the channel, the control unit shall set the E bit to one in the last data frame that exactly satisfies the data request, and a status frame shall be sent containing channel-end status, the transfer count, and the LR bit set to one.

13.1.2.3 Data-chaining

When there are successive CCWs to be executed by the channel for a single I/O operation, the execution of these successive CCWs is referred to as data-chaining. When each CCW associated with a channel program becomes the current CCW being executed by the channel, the channel shall use a command frame to transfer the chain-data flag, count and other flags associated with this CCW to the control unit. In this manner, the control unit shall be informed that data-chaining is specified for the current CCW. When the execution of the current CCW is completed and the chain-data flag is set to one, data-chaining takes place, provided no abnormal conditions are detected and all other conditions for data-chaining are satisfied. Data-chaining may occur only during an I/O operation and only when the CD bit had been set to one in the last command frame. When data-chaining takes place at the channel and the command frame is used to update the flags and count held at the control unit, the chaining bit (CH bit) of the device-header flag field and the data-chaining-update (DU) flag of the command-flag field shall both set to ones to identify the command frame as being associated with data-chaining.

If the control unit receives a command frame with the DU flag and CH bit set to ones, if the control unit has the data-chaining condition set, and if the chain-data bit was set to one in the previous command frame, then the command flags and count from the current command frame shall be accepted and become the current flags and count.

The chain-data bit in the command-flag field of the command frame indicates there is a subsequent count and command-flag update for the I/O device that will immediately follow the execution of the current CCW, provided that no abnormal conditions are encountered.

For a data-chaining write operation, the control unit shall have the count for the current CCW. When the last data frame is received with the E bit set to one and the quantity of data received satisfies the data requested and the count for the CCW, data-chaining shall be expected. The next frame received shall either be the command frame with the flags and count from the next CCW or a device-control frame that terminates the I/O operation.

For a data-chaining read operation, the control unit shall receive either the count for the current CCW (E bit set to one) or a count less than the count for the current CCW (E bit set to zero). If the count is less than the count for the current CCW and all the data requested has been sent, the next frame received shall either be a data request or a device-control frame that terminates the I/O operation. Once the control unit receives a data request with the E bit set to one, the CCW count shall be known. When the last data frame is sent with the E bit set to one and the quantity of data sent satisfies the count for the CCW, data-chaining shall be expected. The next frame received shall either be the command frame with the flags and count from the next CCW or a frame that terminates the I/O operation. If the last data frame is sent with the E bit set to one, the quantity of data sent satisfies the count for the CCW, and conditions at the control unit or I/O device require the ending of the I/O operation, then the control unit shall normally wait for the next frame before sending a status frame (see 13.1.3.3). The control unit shall be permitted to send a status frame without waiting for the next frame from the channel only when an error detected by the control unit results in the transfer-count-valid bit being set to zero in the status frame. If the control unit terminates data transfer before all of the data for the CCW is transferred, then the E bit shall be set to zero in the last data frame sent by the control unit, and the control unit shall subsequently send a status frame to indicate the ending of the I/O operation (see 13.1.2).

The I/O device shall ensure that data-chaining is occurring at the proper times by recognizing a data-chaining condition. The data-chaining condition shall be recognized for each I/O device and shall be used to verify that successive CCWs are being performed by the channel. The data-chaining condition shall be set whenever the I/O device accepts a command frame and the chain-data bit is set to one in the command-flag field. The data-chaining condition shall be reset for any of the following conditions:

– system-reset or selective-reset is performed;

– the I/O device performs a cancel or stack-status function;

– the I/O device receives a command frame and the chain-data bit is set to zero;

– status containing device-end has been accepted by the channel.

If the data-chaining condition is set, if the control unit initiates the ending of the I/O operation by sending channel-end status, with or without device-end status, and if the quantity of data transferred is less than the count associated with the current CCW, then the data-chaining indication shall be reset, and chaining shall not performed.

Data-chaining shall occur whenever the data-chaining condition is set, the command-chaining condition is not set, and the chaining (CH) bit in the device-header flag field and the data-chaining-update flag (DU flag) are both set to ones in the command frame.

If the data-chaining condition is set in the control unit and if a command frame is received with the chaining bit (CH bit) or the data-chaining-update flag (DU flag) set to zero, then a device-level error shall be detected. If the data-chaining and command-chaining conditions are both not set and if a command frame is received with either the DU flag set to one or the CH bit set to one, then a device-level error shall be detected (see 13.1.4). The CH bit and DU flag shall be tested before the data-chaining condition is changed to conform to the new value of the chain-data bit in the command-flag field.

If the data-chaining condition is set, the control unit shall ensure that the path to the I/O device remains available when the current count is exhausted, until the next device frame is received or until the data-chaining condition is reset.

13.1.2.4 Pacing

Pacing is the method by which the channel or control unit adjusts the rate at which data is transferred. There are two methods by which the data rate may be adjusted:

a) Pacing parameters;

b) The ready-indication and data-request protocols.

Pacing parameters shall control the maximum amount of data that may be transferred in each data frame and the minimum amount of delay between successive data frames. The pacing parameters shall be established by the channel when the logical path is established and by the control unit in the command response. At the start of each I/O operation, the control unit shall indicate a valid DIB size and, for a write operation, the number of idle characters to be used in the command. Parameters established in the command response shall apply only for the current I/O operation (see 9.1.1 and 12.2.4.2).

Pacing parameters identify, for each DIB size allowed, the minimum number of pairs of idle characters to be used between two consecutive data frames (see 12.2). Pacing parameters and DIB sizes permitted apply only to data frames. If one or both of any two consecutive frames is not a data frame, the pacing parameters do not apply, and the minimum number of idle characters used between the frames is four.

The ready-indication and data-request protocols shall control the number of data frames that are to be received at any one time (see 13.1.2).

13.1.2.5 Zero CCW count

When the count in the current CCW is zero, the count field in the command frame shall be set to zero.

If the command specifies a read operation, the DR bit shall be set to zero, and the E bit shall be set to one. If the operation is not an immediate operation, the command response shall have the E bit set to one. The subsequent status frame shall have the transfer-count-valid bit set to one and the transfer count set to zero in the status-parameter field. If the control unit would have transferred data had the channel requested any, the LR bit shall be set to one; otherwise, it shall be set to zero.

If the command specifies a write operation, the E bit shall be set to zero in the command frame. If the operation is not an immediate operation, the command response shall have the DR bit set to zero, the data-request count set to zero, and the E bit set to one. The subsequent status frame shall have the transfer-count-valid bit set to one and the transfer count set to zero in the status-parameter field. When the control unit would have requested data if the count in the command were nonzero, the LR bit shall be set to one; otherwise, it shall be set to zero.

Whether a control unit considers a count of zero to be valid for a command shall be model-dependent.

13.1.3 Ending an I/O operation

The ending of an I/O operation shall be either channel initiated or control-unit initiated. The channel may initiate the ending of an I/O operation as the result of an abnormal condition or a non-error condition. The control unit may initiate the ending of an I/O operation as the result of the completion of the execution of the command or the transfer of all data associated with the command or as the result of an abnormal condition detected during the execution of the command.

When the channel initiates the ending of the I/O operation, it shall send a control frame indicating one of the following control functions:

- cancel (see 13.2.2);
- selective-reset (see 13.2.4);
- system-reset (see 13.2.3).

When the control unit initiates the ending of the I/O operation, it shall send a status frame.

The following subclauses describe the ending of an I/O operation using status frames.

13.1.3.1 Ending an I/O operation using a status frame

When the execution of the current command, including the transfer of data, if any, is completed, the control unit shall initiate the ending of the I/O operation with the channel by transferring channel-end status to the channel. When the execution of the current command, including the transfer of data, if any, encounters an abnormal condition, the control unit shall terminate the execution of the command and the transfer of data and initiate the ending of the I/O operation with the channel by transferring channel-end status and the appropriate status for the abnormal condition that was detected to the channel. When the status that initiates the ending of the I/O operation is transferred, device-end status may or may not be included. If device-end status is included, the operation shall be considered ended by the I/O device when this status is accepted by the channel. If device-end status is not included, the I/O operation shall be considered ended by the I/O device when the control unit later transfers device-end status and receives acknowledgment that the status has been accepted by the channel. If unit-check status is to be transferred, the control unit may indicate that supplemental-status is available by setting the supplemental-status-available (SA) bit to one in the status frame (see 13.1.3.2).

The control unit shall initiate the ending of the I/O operation with the channel by sending channel-end status with or without device-end status. In the absence of errors, when the current command is executed as a nonimmediate operation, the control unit shall initiate the ending of the I/O operation with the channel when one of the following conditions is satisfied:

- the quantity of data transferred exactly equals the total count associated with the current command and exactly equals the quantity of data required by the I/O device;
- the quantity of data transferred exactly satisfies the total count associated with the current command but does not satisfy the quantity of data required by the I/O device;
- the quantity of data transferred is less than the total count associated with the current command but exactly satisfies the quantity of data required by the I/O device.

If the quantity of data transferred is less than or equal to the total count associated with the current command and this quantity of data is greater than the quantity of data required by the I/O device, the control unit shall initiate the ending of the I/O operation with the channel by sending channel-end with unit-check status, with or without device-end status (see 13.1.2 for additional information on the data-transfer protocols).

When the current command is executed as an immediate operation, the control unit shall initiate the end of the I/O operation with the channel when channel-end status, with or without device-end status, is transferred to the channel and accepted. The status frame used to initiate the end of an immediate I/O operation shall have the transfer-count-valid bit set to one and the transfer-count field set to zero.

If the control unit is connected when it is ready to transfer status in order to initiate the ending of the I/O operation at the I/O device, a status frame with the appropriate status information including device-end status shall be sent, provided the conditions for sending a device frame are satisfied.

If the control unit is not connected when it is ready to transfer status in order to complete the ending of the I/O operation with the channel, as might be the case when device-end did not accompany channel-end, a request-connection frame shall be sent. When the connection is granted, a status frame with the appropriate status information including device-end status shall be sent, provided the conditions for sending a device frame are satisfied (see 13.2.5).

13.1.3.2 Supplemental-status

Supplemental-status provides sense information in a status frame to describe conditions at the control unit or I/O device for which status information containing unit-check is being provided.

The channel shall indicate whether it can accept a status frame that includes supplemental-status when the channel sends a request-status frame. The supplemental-status (ES) bit, when set to one, shall indicate that the channel will accept supplemental-status.

If the status frame has the supplemental-status-available (SA) status flag and unit-check status bit both set to ones, then instead of accepting the status, the channel may notify the control unit that it is capable of receiving supplemental-status by sending a request-status frame with the ES bit set to one, thus permitting the control unit to send the associated supplemental-status with the status information. The request-status frame in response to the status frame shall cause command-chaining and command-retry, if any, to be suppressed. If command-retry had been requested in the status frame having the SA bit set to one and the channel responded with a request-status frame, a retry request for the I/O operation shall not be honored by the channel, even if requested.

13.1.3.3 Transfer-count checking

During data transfer for an I/O operation, both the sender and recipient of data shall maintain a count of the number of bytes of data transferred over the interface for the current CCW. The status frame that is presented after the data transfer for an I/O operation is ended shall normally include the transfer count accrued for the last CCW executed. The channel shall compare the transfer count included in the status frame with the number of bytes sent or received by the channel for the last CCW executed to ensure that the number of bytes received or sent by the control unit matches the number of bytes sent or received by the channel respectively.

If data transfer occurs and the control unit does not detect an error during the data-transfer portion of the I/O operation for the CCW, the control unit shall set the transfer-count-valid bit to one and include the transfer count for the CCW in the status frame that also includes channel-end status.

For write operations, in order to report the correct transfer count, the control unit shall wait until it receives all the requested data before sending in status.

When the channel receives the status frame with the transfer-count-valid bit set to one after data transfer, the channel shall compare the transfer count from the status frame with the transfer count maintained by the channel for the current CCW. When both counts associated with data transfer for the current CCW are equal, the transfer count shall be considered valid. The control unit shall retain the transfer count until it receives acknowledgment that the status frame containing the transfer count has been accepted (see 13.2.8).

If the comparison shows that the transfer counts maintained by the channel and the control unit do not match, the transfer count associated with the data transfer for the current CCW shall be considered invalid, and a device-level error shall be detected; chaining, if specified, shall not be performed unless retry has been requested by the control unit and is to be honored by the channel (see 13.3.1).

If an error detected by the control unit does not affect the ability to present a valid transfer count, the transfer-count-valid bit shall be set to one in the status frame that also includes channel-end status. If the control unit detects an error such that it is unable to supply a valid transfer count, the control unit shall set the transfer-count-valid (XV) bit to zero in the status frame and include unit-check along with the channel-end status, or the I/O device shall set the transfer-count-valid bit to zero in the status frame and request retry. Any sense data associated with this unit-check status shall indicate that the data transfer is unpredictable.

If, after performing data transfer, a channel receives a status frame with the channel-end status bit set to one and the transfer-count-valid bit set to zero, the channel shall not compare the transfer count in the status frame to the transfer count held by the channel. The counts associated with the transfer of data for the current CCW shall be considered invalid. If, after performing data transfer, the channel receives a status frame with the channel-end status bit set to one, the unit-check bit set to zero, and the transfer-count-valid bit set to zero, the channel shall detect a device-level error.

If the transfer counts do not match or if the transfer-count-valid bit is zero when the channel-end status bit is set to one, and a request by the same status frame for command-retry is accepted by the channel, the error in the transfer count or the transfer-count-valid bit shall not be recognized, and retry shall proceed normally. If retry is not honored, the channel shall terminate the I/O operation and any command-chaining.

During data-chaining, the transfer count from the current CCW shall be reset when the next CCW takes effect. A new transfer count shall be started for the data transfer associated with the new CCW. During data-chaining at the channel, the transfer count shall be reset when the next CCW takes effect and the command frame shall be sent. During data-chaining at the control unit, the transfer count shall be reset when the command frame for the new CCW is received and accepted.

If data-chaining is specified for a write operation and an error condition or early-end condition is detected by the control unit after the control unit sends a data request with the E bit set to one, the control unit shall wait for all of the data requested and the next command frame to be received before sending the status frame. The transfer-count-valid bit shall then set to one with a transfer count of zero.

If data-chaining is specified for a write operation and an error or early-end condition is detected by the control unit before the control unit sends a data request with the E bit set to one, the control unit shall wait for all of the data requested to be received before sending the status frame. A valid transfer count, if available, shall be included with the status frame.

If data-chaining is specified for a read operation and either an error condition or an early-end condition is detected by the control unit before the control unit has sent a data frame with the E bit set to one, the control unit shall send a status frame after the last data frame is sent. A valid transfer count, if available, shall be included with the status frame.

If data-chaining is specified for a read operation and either an error condition or an early-end condition is detected by the control unit after the control unit sends the data frame with the E bit set to one, then the control unit shall normally wait for the command frame to be received from the channel before sending the status frame, and the status frame shall normally have the transfer-count-valid bit set to one with a transfer count of zero. The control unit shall be permitted to send a status frame without waiting for the next frame from the channel only when an error detected by the control unit results in the transfer-count-valid bit being set to zero in the status frame.

13.1.4 Command-chaining

When there are successive I/O operations to be executed by the channel for a single channel program, the execution of these successive I/O operations is referred to as command-chaining. When an I/O operation is executed, the channel shall use a command frame to transfer the command, chain-command flag, count, and other flags associated with this CCW to the control unit. In this manner, the control unit shall be informed at the start of the I/O operation whether command-chaining is specified for the current CCW. When the execution of the I/O operation is completed and the chain-command and chain-data flags were previously set to one and zero, respectively, command-chaining shall take place, provided no abnormal conditions are detected and all other conditions for chaining are satisfied. When command-chaining takes place at the channel, the command frame used to update the command, flags, and count shall have the chaining bit (CH bit) set to one and the data-chaining-update flag (DU flag) set to zero to identify the command frame as being associated with command-chaining.

When the control unit receives a command frame with the CH bit set to one (except when data-chaining takes place at the channel and the command frame is used to update the count and flags) and command-chaining is indicated at the control unit, the command, flags and count from the current command frame shall be accepted, and a new I/O operation shall be initiated (see 13.1.1 and 13.1.2.3).

Command-chaining shall also be indicated by the channel when the channel accepts status for an I/O operation and the status does not contain device-end. The channel shall accept the status with a status-accepted frame. The channel shall indicate chaining by setting the chaining (CH) bit to one in the status-accepted frame. If the chaining bit is set to zero in the status-accepted frame, command-chaining shall be reset by the control unit.

The combination of the chain-command bit set to one and the chain-data bit set to zero in the command-flag field of the command frame shall indicate that there is a subsequent I/O operation for the I/O device which will immediately follow the execution of the current CCW, provided that no abnormal conditions are encountered.

The control unit shall ensure that command-chaining is occurring at the proper times by recognizing a command-chaining condition. The command-chaining condition shall be recognized for each I/O device to verify that successive I/O operations are being performed by the channel.

The command-chaining condition shall be set either as the result of command-chaining or as the result of command-retry. During command-chaining, the command-chaining condition shall be set whenever the I/O device accepts a command and the chain-command flag (CC flag) is set to one in the command-flag field, and the chain-data flag (CD flag) shall be set to zero in the command-flag field. The command-chaining condition shall be set for command-retry when the CH bit is set to one in the frame used by the channel to accept the status that requests a command-retry (see 13.3.2).

The command-chaining condition shall be reset for any of the following conditions:

- system-reset or selective-reset is performed;
- the I/O device receives a cancel frame or stack-status frame;
- status containing channel-end but not device-end for an I/O operation has been accepted by the channel and the chaining bit is set to zero in the status-accepted frame;
- status containing device-end for an I/O operation has been accepted by the channel using an accept-status frame;
- an address-exception condition is recognized and a device-level-exception frame for this condition is sent;
- the channel accepts a valid status frame with the AS bit set to one and the status contains unit-check (except when command-retry is requested and the retry requested is to be honored by the channel), unit-exception, busy, attention, status-modifier without any other bits set, or control-unit-end, provided it is not control-unit-end alone. A valid status frame with the AS bit set to zero shall not suppress command-chaining.

Command-chaining shall occur whenever the command-chaining condition is set in the I/O device, and the CH bit and the DU bit in the device-header flag field of the command frame are set to one and zero, respectively.

If the command-chaining condition is set in the I/O device, the control unit shall ensure that the path to the I/O device remains available when device-end status is presented until the next device frame is received or until the command-chaining condition is reset. If the command-chaining condition is set in the control unit and if a command frame is received with the CH bit set to zero or the DU flag set to one, a device-level error shall be detected. If the command-chaining and the data-chaining conditions are both not set and if a command frame is received with the CH bit set to one or the DU flag set to one, then a device-level error shall be detected.

13.2 Device-level controls

Device-level controls are used to modify or control the execution of an I/O operation, to provide for the transfer of status when an I/O operation is not being executed, or to place the control unit and I/O device in a known state. Most device-level controls are specified by combinations of bits in the device-header flag field and the IFI field in the device header. For device-control frames, device-level controls are also specified in the DIB. For other frame types, device-level controls are specified only by the device-header flag field and IFI field.

13.2.1 Stack-status function

The stack-status control function, indicates that the channel does not accept the status; the control unit or I/O device shall hold the status information (status flags, status byte, and status parameter) and associated supplemental-status, if any, until requested by the channel or until status information is cleared, withdrawn, or unstacked. In addition, a control unit may unstack any status which has remained stacked for at least 1 minute. In certain unusual control-unit-recovery situations, such as when the control unit has lost the indication of whether status should or should not be unstacked for a particular I/O device, the control-unit-recovery action may possibly cause some stacked status to become not stacked without a request-status frame, and subsequently that status may be presented by the control unit through request connection. A channel may stack any status except for a status frame with the AS bit set to one that is received in response to the first command frame of a channel program. If a stack-status frame is received in response to a status frame with the AS bit set to one, which in turn was sent in response to the first command frame of a channel program, then a device-level error shall be detected. Stacking of status other than control-unit-end or control-unit-busy causes the suppression of chaining when chaining is in progress.

The busy bit shall not be considered part of the status to be held by the control unit or I/O device when the busy bit has been set to one in order to indicate that the status being presented to a command was already pending.

The CI bit of the status-flag field shall not be held by the control unit or device as part of stacking status. The CI bit shall be used to indicate that the status was generated in direct response to a selective-reset frame; this would not apply after the status was stacked.

The control unit shall reset the data-chaining or command-chaining condition when it receives a stack-status frame in response to a status frame and the status byte does not contain either:

a) The control-unit-end status bit set to one and all other status bits set to zeros;

b) The busy and status-modifier bits set to ones and all other status bits set to zeros.

(See 13.1.2.3 and 13.1.4.)

Status shall be stacked and unstacked with respect to a specific logical path. Status that is not associated exclusively with the current logical path may be presented on another path on which it either has been unstacked or has not been stacked.

The channel may request pending status or request the unstacking of status by using the request-status control function (see 13.2.6).

If conditions change at the control unit or I/O device which affect status information that is being held as a result of receiving stack-status, the most current information shall be sent in a status frame (when permitted by receipt of a request-status frame) or in response to a command frame.

A control unit shall indicate to the channel that it recognized the stack-status by sending a device-level-ACK frame. The disconnect-EOF delimiter on the device-level-ACK frame shall cause the existing connection to be removed. The channel shall consider the stack-status function to have been performed when it receives the device-level-ACK frame.

If a control unit recognizes a stack-status frame in response to a status frame with the AS bit set to zero, the control unit shall perform one of the following actions:

- if the status was control-unit-busy, the status shall be withdrawn. In addition, control-unit-end status shall not be owed unless control-unit-busy status had been previously accepted by the channel;
- if the status was control-unit-end alone, the status shall remain pending and unstacked;
- if the status was control-unit-end with busy, the busy bit shall not be retained, and the control-unit-end shall remain pending and unstacked;
- if the status contained control-unit-busy along with control-unit-end, the status shall be withdrawn. In addition, control-unit-end status shall not be owed unless control-unit-busy status had been previously accepted by the channel.

When status information is caused to be stacked for one I/O device and the control unit is no longer connected to the channel, a request for a connection to the channel in order to send pending not-stacked status for a different I/O device shall be permitted.

13.2.2 Cancel function

The cancel function shall cause the designated I/O device to terminate execution or perform nullification of the current operation, if any. When an I/O operation is terminated by the cancel function, no further data shall be transferred by either the control unit or the channel, the I/O device shall proceed to its normal ending point (including mechanical motion) and, as a result of having gone to its normal ending point, the I/O device shall generate channel-end and device-end status, as appropriate, for the I/O operation. When an I/O operation is nullified for the cancel function, the I/O device shall consider the initial command to have not started; that is, no action shall be taken on the initial command, and no status shall be owed for the initial command. If an I/O operation is neither being initiated nor in progress, the cancel function shall cause no action at the I/O device.

A cancel function shall not be initiated by the channel during the time between when a command frame is sent to initiate an I/O operation and when the response to the command frame is received. (System-reset or selective-reset would ordinarily be permitted in the same interval if the CH bit in the command frame was set to one.)

When the cancel frame is received and the device address specified corresponds to the I/O device for which the control unit is connected to the channel, the device-level-ACK frame shall be sent, and the cancel function shall be performed for the specified I/O device. When the cancel frame is received and no connection to the channel is considered to exist, the device-level-ACK frame shall be sent, and the cancel function shall be performed for the specified I/O device. When the cancel frame is received and the device address specified does not correspond to the I/O device for which the control unit is connected to the channel, the device-level-ACK frame shall not be sent, and a device-level error shall be detected (see 13.3.1).

The channel shall consider the cancel function to be performed when it receives the device-level-ACK frame in response.

The cancel function shall not cause any pending or stacked status to be cleared. If the cancel function causes an operation to be nullified, the initial status, if present, shall be withdrawn. If the cancel function is performed after the sending of status indicating status-modifier and busy but before the receipt of the status-accepted frame, then the status shall be withdrawn.

When the cancel function is performed, chaining shall be suppressed for the affected I/O operation.

When the cancel function is performed by an I/O device before the I/O device considers an I/O operation in progress or as being initiated, no action is taken by the I/O device. Stacked or pending status for that I/O device shall not be affected. The I/O device shall not become busy as a result of performing the cancel function. The following conditions are included:

- the control unit is not connected to the channel, and no operation is considered as either being in progress or being initiated with the specified I/O device;

- the control unit is connected to the channel, has sent a status frame for asynchronous status, and has not received a status-accepted response.

When the cancel function is performed by an I/O device after the I/O device considers an I/O operation that is not the result of chaining to be initiated but before the I/O operation is considered in progress by the I/O device, the I/O operation shall be nullified (see 13.1.1). No action for the current command shall be taken by the I/O device, and no I/O operation shall be considered to be in progress. Any status associated with the nullified I/O operation shall be discarded. The I/O device shall not become busy as a result of performing the cancel function.

When the cancel function is performed after an I/O operation is considered to be in progress at the I/O device, the I/O operation shall be terminated; status containing channel-end, device-end, or both, as appropriate shall be generated as a result of the I/O device proceeding to its normal ending point (see 13.1.1 and 13.1.3). In this case, the presentation of the channel-end or device-end status shall require a new connection to the channel, which shall be initiated by sending a request-connection frame. If channel-end is included in this status, then the transfer count shall also be included in the status frame with the transfer-count-valid bit set to one, unless conditions at the control unit prevent transfer of a valid transfer count, in which case the transfer-count-valid bit shall be set to zero, and unit-check shall be included in the status. If the long-record (LR) bit is set to one with the channel-end status, the long-record condition shall be ignored by the channel. The I/O device shall remain busy until all status for the I/O operation is accepted by the channel. Any abnormal I/O-device operation shall be indicated by unit-check in the status, and the sense information provides additional details concerning the operation. The cancel function shall cause an I/O operation to be terminated at the control unit for the following conditions:

- the control unit is connected to the channel executing an I/O operation, and status containing channel-end status has not been sent or, if sent, has not yet been accepted by the channel;
- the control unit is connected to the channel as a result of a request status, and status containing channel-end or device-end status for an I/O operation has been sent but not yet accepted by the channel;
- the control unit is connected to the channel as a result of a request connection, and status containing channel-end or device-end status for an I/O operation has been sent but not yet accepted by the channel.

If the cancel function is performed while the I/O device is in the process of sending a frame, the I/O device shall finish sending that frame with the correct CRC and delimiters; however, any additional data associated with the I/O operation in progress shall not be sent.

When the channel sends a cancel frame during an existing connection for an I/O operation, there may be one or more device frames already in transit from the control unit to the channel that normally would have been part of the I/O operation had the cancel not been sent. Otherwise-valid frames that are received after the cancel frame is sent but before the device-level ACK in response to the cancel frame is received shall be ignored by the channel.

Except when dynamic reconnection is being used, performing the cancel function shall not affect I/O operations or status associated with a logical path other than the logical path over which the cancel frame was received (for I/O devices using dynamic reconnection, see 13.5.2).

13.2.3 System-reset function

The system-reset function shall be performed by the control unit with respect to a logical path whenever any of the following conditions is recognized by the control unit:

- a valid device-level system-reset frame is received on the logical path. A device-level system-reset frame shall always be sent with the AS bit set to zero;
- the logical path is removed or established;
- certain internal recovery procedures occur within the control unit.

In addition, whenever a logical path does not exist, the control unit and all attached I/O devices shall remain in the reset state with respect to the logical path that could be established between the control unit and the channel.

After a valid system-reset frame is received from the channel, a control unit shall indicate to the channel that it recognizes the system-reset by sending a device-level-ACK frame. The disconnect-EOF delimiter on the device-level-ACK frame shall cause the existing connection to be removed. When system-reset is initiated in this way, the channel shall consider that the system-reset function was performed when it receives the device-level-ACK frame.

When the channel sends a system-reset frame during an existing connection for an I/O operation, there may be one or more device frames already in transit from the control unit to the channel that normally would have been part of the I/O operation had the system-reset not been sent. Otherwise-valid frames that are received after the system-reset frame is sent but before the device-level ACK in response to the system-reset frame is received shall be ignored by the channel.

A system-reset function, when performed by the control unit, shall cause the control unit to reset all I/O devices, I/O operations, queued commands, if any, and pending status with respect to a particular logical path. The interpretation of the reset state of an I/O device shall be model-dependent. A system-reset may possibly reset forthcoming status which is owed to the path signaling reset but not yet pending, such as no-longer-busy status. Any I/O operation in progress shall be terminated, and the I/O device shall proceed to a normal mechanical stopping point, if applicable. If the control unit is in the process of sending a frame when it recognizes a system-reset, it shall finish sending that frame with the correct CRC and delimiters, if possible. No further frames shall be sent pertaining to the I/O operation that was reset.

While the system-reset is either pending or being performed by the control unit, busy status may be returned in response to any command frame from the logical or physical path with which the system-reset is associated. While system-reset is being performed, busy status may be returned in response to any command frame from any path. If busy status is returned in response to a command frame during one of these intervals, then appropriate no-longer-busy status shall be made pending for the logical path when the busy condition no longer exists.

The system-reset associated with a logical path shall not directly affect I/O operations or status associated with any other logical path. However, if the status for another logical path has not been presented because of an inhibiting condition and the reset clears the inhibiting condition, then previously owed or pending status associated with that other logical path may be presented along that other logical path after the reset.

The ready or not-ready state of the control unit or I/O device shall generally not be changed by a system-reset. When, however, the online/offline switch was changed before the reset but is not yet effective because of required inhibiting conditions, the ready or not-ready state may change if the reset clears those inhibiting conditions.

A system-reset shall not reset activity that occurs logically subsequent to the moment that system-reset is recognized by the control unit.

If a system-reset terminates an active I/O operation, but leaves the device busy until the end of mechanical motion, then device-end status or device-end and unit-check status at the end of mechanical motion may be presented. If presented, this status may be presented using any logical path which is established with respect to the I/O device.

When a system-reset is performed, a resetting-event condition shall be generated for each affected I/O device (see 13.4).

13.2.4 Selective-reset function

The selective-reset function shall cause a device and its status to be reset with respect to the particular logical path. Only the device and certain allegiances associated with the designated logical path shall be reset.

NOTE – Examples of certain allegiances are a control unit resetting owed “no-longer-busy” status such as device-end or control-unit-end or a control unit resetting sense data owed for previously accepted unit-check status for other than resetting-event notification.

The definition of the reset state of the I/O device shall be model-dependent.

The selective-reset frame shall pass information to the I/O device which results in one of two types of action:

a) When one or more of the RC, RU, and RO bits is set to one, the channel is requesting that command-retry be requested or that the I/O device reply with status containing unit-check. The device shall signal that it is capable of complying with the request by returning a status frame with the CI bit set to one (see 13.3.3);

b) When the I/O device neither requests the indicated retry nor returns the indicated status, the I/O device shall return a device-level-ACK frame and perform the selective-reset function.

A selective-reset function, when performed by an I/O device, shall cause the I/O device to reset all queued commands, if any, and status with respect to a particular logical path. Any I/O operation in progress for the I/O device on that logical path shall proceed to a normal ending point, if applicable, with no further data transfer. If the control unit is in the process of sending a frame when it performs the selective-reset function, the control unit shall finish sending that frame with the correct CRC and delimiters. No further frames shall be sent pertaining to the I/O operation that was reset.

Performing the selective-reset function may reset forthcoming status which is owed to the path signaling reset but not yet pending, such as no-longer-busy status.

A control unit shall indicate to the channel that selective-reset has been or will be performed by sending a device-level-ACK frame. The disconnect-EOF delimiter on the device-level-ACK frame shall cause the existing connection to be removed. The channel shall consider the selective-reset function to be performed when it receives the device-level-ACK frame.

When the channel sends a selective-reset frame during an existing connection for an I/O operation, there may be one or more device frames already in transit from the control unit to the channel that normally would have been part of the I/O operation had the selective-reset frame not been sent. Otherwise-valid frames that are received after the selective-reset frame is sent but before the device-level-ACK or status frame in response to the selective-reset frame is received shall be ignored by the channel. A status frame received in response to the selective-reset frame shall have the CI bit set to one.

While the selective-reset function is pending or is being performed by the control unit, busy status may be returned in response to any command frame from the logical path which initiated the selective-reset function. If the selective-reset function is being performed by the control unit, busy status may be returned in response to any command frame from any path. If busy status is returned in response to a command frame during one of these intervals, then appropriate no-longer-busy status shall be made pending for the logical path when the busy condition no longer exists.

If the selective-reset function is performed by a control unit during initiation of an I/O operation which is not the result of command-chaining, and the control unit receives the selective-reset frame as the next frame after making a command response, or as the next frame after presenting status which indicates that a command is accepted, then the I/O operation shall be nullified, and selective-reset shall be performed. When the channel generates the selective-reset frame for these cases, the RC bit shall be set to zero; otherwise, a device-level error shall be detected by the I/O device.

If performing the selective-reset function results in the termination of an active I/O operation, but leaves the I/O device busy until the end of mechanical motion, then device-end status or device-end and unit-check status at the end of mechanical motion may be presented. If presented, this status may be presented to any logical path which is established with respect to the I/O device.

The ready or not-ready state of the control unit or I/O device shall generally not be changed when performing the selective-reset function. When, however, the online/offline switch was changed before performing the selective-reset function, but is not yet effective because of certain inhibiting conditions, the ready or not-ready state may change if performing the selective-reset function clears those inhibiting conditions.

Performing the selective-reset function shall not reset any activity that occurs logically subsequent to the moment that selective-reset is recognized by the control unit.

13.2.5 Request-connection function

The request-connection control function shall be used to transfer status for an I/O operation which was previously disconnected or for which cancel was received or to transfer asynchronous status. Depending on the status and the conditions when status is presented, transferring status for an I/O operation may result in the continuation of a channel program.

The address-specific (AS) bit may be set to either one or zero with a request-connection frame. If the AS bit is set to one, the control unit is requesting a connection to the channel for a specific I/O device. If the AS bit is set to zero, the control unit is deferring the identification of the specific device address until the connection is granted; the device address associated with the connection, if any, shall later be identified by the control unit in a status frame sent during the resulting connection.

A channel shall grant a connection in response to a request-connection frame by sending an accept-connection frame. The value of the AS bit in the accept-connection frame shall be equal to the value of the AS bit received with the request-connection frame. When the accept-connection frame is sent, the control unit shall be considered connected by the channel and when the accept-connection frame is received by the control unit, the control unit shall consider itself connected. Once the control unit is connected, status may be transferred to the channel.

When the channel does not grant a connection, a deny-connection frame shall be sent in response. The value of the AS bit in the deny-connection frame shall be equal to the value of the AS bit received with the request connection. When the control unit receives this response, a new request-connection frame may be sent. The control unit shall continue to send request-connection frames until the request is granted or until conditions at the control unit no longer require a connection. The minimum time between any two requests on the same logical path shall depend on the transmission delay, delay caused by the dynamic switch, if present, and delays at the channel and control unit. However, in no case, shall the minimum time between requests be less than 50 microseconds.

The control unit or I/O device may cause a disconnection from the channel at the completion of the channel portion of each I/O operation, after the transfer of channel-end status without device-end status. Later, when the control unit and I/O device are ready to transfer the device-end status for this I/O operation, a request-connection frame shall be sent to the channel for the purpose of transferring the status. The AS bit shall be set to one or zero, depending on whether the control unit identifies the specific device address at the time of the request connection or at the time of the status frame, respectively. If the connection is granted, the device-end status may be sent in a status frame with the AS bit set to one. Command-chaining may or may not be indicated in response to the status frame, the same as if there had been no disconnection. If the channel responds with the cancel frame, the status shall remain pending, and command-chaining shall be terminated at the control unit. The device-level-ACK frame sent in response to the cancel frame shall have a disconnect-EOF delimiter, which makes the physical path to the channel available for other operations (see 13.1.4 and 13.2.2).

A link-busy condition may be encountered when the control unit attempts to reconnect, but once the connection is made, a link-busy condition shall not be encountered. If a link-busy condition occurs during the interval between acceptance of a command and disconnection from the channel, then a link-error condition shall be recognized, and the operation in progress shall be terminated (see 10.11).

When a control unit has asynchronous status to transfer to the channel, the control unit shall send a request-connection frame to the channel. When the connection is granted, the status frame shall be sent. If the status is accepted by the channel, a status-accepted frame shall be sent. If the channel responds with a stack-status frame, the protocols for stacking status shall be in effect (see 13.2.1). If the channel responds with a cancel frame, the status shall remain pending (see 13.2.2).

After a request-connection frame is sent, if conditions at the control unit change such that the connection to the channel is no longer needed, the control unit shall wait for the accept-connection frame and then send a rescind-connection frame. The value of the AS bit used in the rescind-connection frame shall be equal to the value of the AS bit sent in the request-connection frame. Because of the disconnect-EOF delimiter on the rescind-connection frame, the physical path to the channel shall become available for other operations after this frame is sent.

When the control unit has status for more than one interface, a request for a connection may be sent over each interface for which status may be presented. (The requests may be for presentation of multiple status bytes, or for presentation of a single status byte which may be presented for more than one interface, or a combination of the two.) If, at the time the request is granted on any given interface, the control unit determines that a connection is no longer required or the connection cannot be honored under the existing conditions, the control unit shall respond with a rescind-connection frame.

13.2.6 Request-status function

The request-status function shall indicate that the channel is prepared to have pending status information presented for the addressed I/O device. The status shall be associated with the logical path on which the request-status frame was received. Request status shall have no effect on status associated with logical paths other than that on which the request status was received.

The request-status control function may be used to obtain stacked status or to obtain supplemental-status when the control unit indicates supplemental-status is available. If the channel uses a request-status frame in response to a status frame having the supplemental-status-available status flag set to one, then the device address shall be the same as the device address from the status frame.

The ES bit may be set to either one or zero, depending on the conditions present at the channel when the request-status frame is sent. The ES bit shall be set to one when the request-status frame is using an existing connection; that is, the channel is subsequently notifying the control unit that it is capable of receiving supplemental-status. When an existing connection is used and the ES bit is set to zero in the request-status frame, a device-level error shall be detected. The ES bit may be set to either one or zero when the request-status frame is used to establish a connection. When a connection is being established, the value of the ES bit is independent of whether any status has been stacked for which supplemental-status might exist. The ES bit shall be set to one if the channel is currently capable of accepting supplemental-status; otherwise the ES bit shall be set to zero.

When the control unit accepts a request-status frame, it shall unstack any status previously stacked for the specified device address. If the request-status frame initiated a connection, the control unit may send a status frame for either the specified I/O device or for the control unit. If no status frame is sent, the control unit shall send a device-level-ACK frame. If the request-status frame did not initiate a connection, the control unit shall send a status frame for the specified I/O device.

When the control unit accepts a request-status frame with the ES bit set to one, the control unit may present supplemental-status, if present, together with the status information in the status frame. If there is no supplemental-status or if the control unit elects not to present it, only the status information shall be sent in the status frame, with the ES bit and supplemental-status-available (SA) status flag set to zeros. If the ES bit is set to zero in the request-status frame and the status information being returned has supplemental-status, only the status information shall be returned in the status frame with the ES bit set to zero and the SA bit set to one; the supplemental-status shall be held by the I/O device. The supplemental-status shall be held until retrieved by the channel as supplemental-status, transferred as sense information by the appropriate sense command, or cleared (see 12.2.3.5).

If a control unit receives a request-status frame during an existing connection in response to a status frame having the supplemental-status-available (SA) status flag set to zero, a device-level error shall be detected.

When the control unit receives a request-status frame which initiated a connection and no status is returned in direct response to the request status, the control unit shall send a device-level-ACK frame. The AS bit shall be set to one.

If the control unit receives a request-status frame but is unable to honor it, the control unit shall send a status frame with the AS bit set to zero, the status-modifier status bit set to one, and the busy status bit set to one, indicating a control-unit-busy condition. If the channel accepts status indicating a control-unit-busy condition, control-unit-end shall be owed. If the channel responds with a stack-status frame, control-unit-busy shall be withdrawn, and control-unit-end shall not be owed unless it was previously owed.

If the control unit receives a request-status frame and the designated I/O device is not installed, one of the following shall occur:

- if pending status exists for the designated I/O device, the control unit shall unstack the status if it had been previously stacked and may send the pending status information. If status is not returned in direct response to the request status, the control unit shall send a device-level-ACK frame. The status shall later be presented through the request-connection protocols;
- if pending status does not exist for the designated I/O device, the control unit may either recognize an address-exception condition and send a device-level-exception frame, or send a device-level-ACK frame (see 13.3.4 and 13.2.9).

A channel may use a request-status function to solicit stacked status from a control unit provided a control-unit-busy (CUB) response to a previous request-status frame to the same control unit was not received and accepted by that channel. A CUB received and accepted by the channel in response to other than a request-status frame does not prevent the channel from issuing a request-status frame. If a CUB is received in response to a request-status frame, the channel may either accept or refuse the CUB status. If the channel accepts the CUB status, the request-status function is not issued to the control unit until a control-unit-end (CUE) “no-longer CU busy” is received. If the channel refuses the CUB status, the stack-status function is performed and the channel may, without initiative from the control unit, again attempt the request-status function.

NOTE – Channel designers should avoid immediately attempting to redrive the request-status function with a control unit that previously responded to a request-status frame with a CUB which the channel refused. CUB conditions may last for prolonged periods of time and may depend on the control unit being able to connect to a channel to present status. A channel that attempts to immediately and persistently attempt the request-status function with such a control unit, could prevent the control unit from clearing the CUB status condition and ultimately creating timeout errors at the ULP.

13.2.7 Device-level-exception function

The device-level-exception control function shall indicate certain abnormal conditions. For the use of the device-level-exception control, see 13.3.4.

13.2.8 Status-acceptance function

The channel shall indicate that status is accepted in several ways, depending on whether the status is for an I/O operation, whether the channel intends to perform command-chaining, and, if command-chaining is to be performed, whether the status contains device-end.

If the status is considered by the channel to be not related to an I/O operation or if the channel does not intend to perform command-chaining, the channel shall indicate that status is accepted by sending a status-accepted frame with the chaining (CH) bit in the device-header flag field set to zero.

If the channel intends to perform command-chaining and the status frame has the channel-end status bit set to one and the device-end status bit set to zero, the channel shall indicate that status is accepted by a sending a status-accepted frame with the CH bit set to one.

If the channel intends to perform command-chaining and the status frame has the device-end status bit set to one, the channel shall indicate implicitly that the status is accepted by sending a command frame with the CH bit set to one (see 13.1.4).

A control unit shall indicate to the channel that it recognizes the status-accepted frame by sending a device-level-ACK frame. The disconnect-EOF delimiter on the device-level-ACK frame shall cause the existing connection to be removed. The channel shall consider the status-accepted function to have been performed when it receives the device-level-ACK frame.

If the control unit considers that command-chaining is suppressed or not indicated, or the control unit has presented status not related to an I/O operation and then detects the CH bit set to one in either a command frame or a status-accepted frame, the control unit shall detect a device-level error and shall not allow command-chaining to occur. If the control unit receives a command frame with the CH bit set to one in response to a status frame that had the device-end status bit set to zero, the control unit shall detect a device-level error and shall not allow command-chaining to occur. If the control unit receives a status-accepted frame with the CH bit set to one in response to a status frame that had the device-end status bit set to one, the control unit shall detect a device-level error and shall not allow command-chaining to occur.

13.2.9 Device-level-acknowledgment function

The device-level-acknowledgment (device-level ACK) function shall indicate that the control unit has recognized a corresponding stack-status, cancel, system-reset, selective-reset, request-status, or status-accepted frame. For the use of the device-level-ACK device-control function, refer to 13.2.1, 13.2.2, 13.2.3, 13.2.4, 13.2.6, and 13.2.8.

13.2.9.1 Control-unit-busy condition

Once a logical path is established, all device frames that initiate a connection which are received by a control unit over that logical path may encounter a busy condition that prevents the acceptance of any device-level function except for a valid device-level system-reset. When the device-level function is not accepted due to a control-unit-busy condition, a status frame shall be returned with the AS bit set to zero and only the status-modifier and busy status bits set to ones; this status is called control-unit-busy.

The control unit may return control-unit-busy status when all of the following conditions are met:

a) There are no link errors;

b) The control unit considers no prior connection to exist;

c) A device frame is received that initiates a connection;

d) The control unit is incapable of recognizing the function contained in the device frame due to a control-unit-busy condition;

e) The frame is not a valid system-reset frame.

If the response to the status frame is a status-accepted frame, then when the busy condition no longer exists, a status byte containing the control-unit-end status bit set to one shall be returned to the destination that accepted the status byte with the control-unit-busy indication. If the response to the status frame is a stack-status frame, then the status shall be withdrawn and the control unit shall proceed as though the status were never sent. If the response to the status frame is anything else, then either a link-level error or a device-level error shall be detected.

13.3 Error handling at the device level

The following subclauses describe device-level errors and several procedures used by the control unit and channel to report unusual conditions and to recover from I/O-device errors and device-level errors. I/O-device errors are associated with the execution of an I/O operation or with an I/O-device state. Device-level errors are associated with the device-level protocols described in this chapter. Certain recovery actions, such as command-retry, may be requested by either the control unit or the channel and are applicable to either I/O-device errors or device-level errors.

13.3.1 Device-level errors

Device-level errors are abnormal conditions that affect the integrity of the information contained in the device header and device-information block (DIB) of a device frame. Device-level errors are also abnormal conditions associated with the protocols for sending and receiving device frames. When a device-level error is detected, no irreversible action shall be taken based on that error until it is determined that no link errors exist. If a link error is detected, the device-level error shall be discarded and the action taken shall be based on the link error.

13.3.1.1 Device-level-connection errors

Device-level-connection errors are abnormal conditions that affect the device-level-connection protocols for creating a connection or pending connection, maintaining an existing connection, or removing an existing connection or pending connection. If one of the following abnormal conditions occurs, a device-level-connection error shall be recognized:

- recognition of a frame at a time when the channel or control unit has initiative to remove the existing connection, for example, when it has initiative to send a device-level-ACK frame which removes the connection;
- recognition of a frame that does not remove the existing connection when the device-level protocols require the frame to remove the existing connection. For example, the response to a system-reset frame is received with a passive-EOF delimiter;
- recognition of an unpredictable connection state as a result of a frame being received with an error and the device-level protocols in effect allow for the removal of the existing connection. For example, the channel recognizes a frame in response to an accept-connection frame and the response has an abnormal frame-terminating condition (see 8.4.2);
- recognition of a frame that removes an existing connection when the device-level protocols in effect do not allow for the removal of the connection. For example, the channel recognizes a frame during data transfer that removes the connection;
- recognition of a device-level-initiation timeout or a device-level-inactivity timeout (see 13.3.5).

13.3.1.2 Device-level-protocol errors

If an abnormal condition is associated with one of the following, a device-level-protocol error shall be recognized:

- the device header;
- the DIB of a device-control frame;
- the command-flag field or count field of a command frame;
- the status-flag field, status field, status-parameter field of a status frame, with or without supplemental-status;
- the quantity of data required in the DIB for the device frame is not met;
- the device-level protocols for sending and receiving device frames.

The following are a few examples of device-level protocol errors:

- reserved bits in the IFI field of the device header are set to ones;
- the AS bit is set to zero in a command frame;
- supplemental-status is included in a status frame with the ES bit set to one when the channel does not permit supplemental-status;
- the channel receives a data frame in response to a command frame that has the DU bit set to zero;
- the device address specified in the device frame does not specify the same I/O device for which the existing connection was created.

See 14.1.10 for the link-level recovery performed and 14.2.2 for the device-level recovery performed as the result of a device-level error.

13.3.2 Command-retry

An I/O device may request command-retry in order to recover from a transient error detected by the I/O device when existing conditions prevented execution of the command at the time it was previously issued or when the I/O device accepts a request from the channel to perform command-retry.

When command-retry is performed by the channel, the previous command frame containing a command shall be resent. The command frame shall have the CH bit set to one and the DU bit set to zero and shall be received like any other command frame that results from command-chaining.

When command-retry is initiated by the I/O device because of conditions detected by the I/O device, the retry request may be made only after the I/O device has received a command frame from the channel that the I/O device intends to accept, and before the I/O device sends a device-level-ACK frame in response to a status-accepted frame. A channel may receive a request to perform command-retry any time after the channel sends a command frame that is initiating an I/O operation until the channel receives either status that indicates the command was not accepted or a device-level-ACK frame in response to a status-accepted frame.

To request command-retry, an I/O device shall send a status frame with the status byte containing retry status and with the CR bit set to one. The status byte shall contain retry status if it contains unit-check and status-modifier together with either of the following:

a) Channel-end alone (meaning the control unit or I/O device is not yet ready to retry the command);

b) Channel-end and device-end (meaning the control unit and I/O device are prepared for immediate command-retry).

Once status with channel-end has been presented and accepted for a command, command-retry shall not be performed for that command (see 12.2.3.1).

The channel may or may not accept the request for command-retry. When the channel does not accept the command-retry request, the channel shall accept the status frame that requests command-retry by means of a status-accepted frame with the CH bit set to zero. The command-chaining condition is reset whenever command-retry-request status is stacked for single path mode of operation (also see 13.1.4 and 13.5.2).

Command-retry may be immediate or deferred, depending on whether device-end is part of the retry status in the status byte. If device-end is part of the retry status, then the I/O device shall immediately be ready to perform command-retry. The channel shall accept the immediate-retry request by responding with a command frame with the CH bit set to one and the DU bit set to zero. If the channel responds to the status frame for an immediate-retry request with a status-accepted frame, retry shall not be performed, and command-chaining shall be reset.

If device-end is not part of the retry status, then the retry shall be deferred until the I/O device sends in device-end, with or without status-modifier, in a subsequent status frame. If the channel accepts the request for deferred command-retry, the channel shall respond with a status-accepted frame that has the CH bit set to one. The device-level-ACK frame in response to the status-accepted frame shall cause disconnection. Later, when the I/O device is ready to perform the command-retry, the I/O device shall send a request for connection to present device-end status. The CR bit shall be set to zero in the subsequent status frame indicating device-end; otherwise, a device-level error shall be detected. The channel shall continue the command-retry process by responding to the status frame with a command frame having the CH bit set to one and the DU bit set to zero. If conditions in the channel have changed so that the channel may no longer retry the command, or if the status presented contains anything besides device-end or device-end and status-modifier, the channel shall accept the status by means of a status-accepted frame with the CH bit set to zero. If the channel responds to the status frame with the status-accepted frame, command-retry shall not be performed, and command-chaining shall be reset.

13.3.3 Channel-initiated recovery procedures

During error recovery, the channel may initiate a device-level recovery procedure by sending a selective-reset frame which specifies the permitted recovery procedures. The channel may prompt the control unit to send a request for command-retry, unit-check without overrun, and unit-check with overrun. Each of these is described in the following subclauses. The intended recovery procedure shall be specified by one or more of the RC, RU, and RO bits of the control-parameter field of the selective-reset frame.

When the channel sends a selective-reset frame at a time when the control unit is allowed to send status, the channel shall ignore any status frame received with the CI bit set to zero and continue waiting for one of the permitted responses as detailed below.

If the channel receives a status frame that has the CI bit set to one but does not contain unit-check status in response to a channel-initiated-retry request or a channel-initiated-unit-check request, a device-level error shall be recognized.

Table 36 summarizes the permitted responses to the selective-reset frame for all combinations of RC, RO, and RU bits (see 12.2.4.6). If a response is received that is not permitted for the combination of RC, RU, and RO bits in the selective-reset frame sent (see table 36), the channel shall detect a device-level error. If a selective-reset frame is received with the RU and RO bits both set to ones, the control unit shall detect a device-level error.

Table 36 – Permitted responses to a selective-reset frame

RC	RU	RO	CU Response
0	0	0	Device-level ACK
0	0	1	Device-level ACK or status with/without retry
0	1	0	Device-level ACK or status without retry
0	1	1	Device-level ACK
1	0	0	Device-level ACK or status with retry
1	0	1	Device-level ACK or status with/without retry
1	1	0	Device-level ACK or status with/without retry
1	1	1	Device-level ACK or status with retry

13.3.3.1 Channel-initiated-retry request

A channel may prompt an I/O device to request the retry of a command in order to recover from a transient error or when conditions at the channel prevent the execution of the command.

The channel shall not make a retry request between the time it sends a command frame to initiate an I/O operation and the time it receives either a command-response frame or a status frame containing appropriate status indicating the command is accepted. For the first command of a channel program, the retry request shall only be sent after the channel sends an acknowledgment to the response from the I/O device that indicates a command has been accepted. A retry request may be made anytime thereafter until the channel accepts channel-end status for that command. For subsequent commands of the channel program, the retry request may be made only after the channel has received a response from the I/O device that indicates a command has been accepted and before the channel accepts channel-end status for that command.

To request retry, the channel shall send a selective-reset frame with the RC bit in the control-parameter field set to one. This is called a channel-initiated-retry request.

Except for those cases where the I/O operation is nullified, an I/O device may receive a channel-initiated-retry request anytime after the I/O device sends a response to the command frame that indicates the command has been accepted and until the I/O device receives an indication that the channel accepts a status frame containing channel-end status for that command.

The I/O device may or may not accept the request for retry. If the I/O device determines that it is capable of performing the requested retry, the I/O device shall respond to the channel-initiated-retry request with a status frame indicating retry and the CI bit set to one (see 13.3.2). The form of retry (either immediate or deferred) is at the discretion of the I/O device. If the I/O device, at the time of receiving the channel-initiated-retry request, is unable to perform the requested retry, the action taken shall depend on the value of the RU and RO bits in the control-parameter field of the selective-reset frame and the state of the I/O device (see 13.3.3.2). If the RU and RO bits are both set to zeros, a selective-reset shall be performed for the I/O device.

If the channel-initiated-retry request is not accepted by the I/O device, or the I/O device requests retry in response to the channel-initiated-retry request, but the channel is unable to complete the retry procedure, then the channel shall terminate the I/O operation and provides the appropriate channel-error indication in the ending status.

When the control unit recognizes a selective-reset frame with a channel-initiated-retry request and no I/O operation is in progress for the specified I/O device, the control unit shall respond with a device-level-ACK frame and performs a selective-reset function.

13.3.3.2 Channel-initiated-unit-check request

The channel may prompt an I/O device to terminate the current operation and recognize a unit-check in order to recover from an error detected by the channel either that resulted from an overrun condition caused during data transfer for a read operation or that resulted from a link-level or device-level error being recognized.

The channel may make a unit-check request for the first command of a channel program anytime after the I/O operation is considered to be in progress by the channel; that is, anytime after the accept-command response is sent and before the accept status for the channel-end status is sent.

The channel may make a unit-check request for a command other than the first command of a channel program anytime after the command is sent and before the status-accepted frame to accept channel-end status is sent.

The channel may make a unit-check request for an overrun condition only after the I/O operation is considered to be in progress and then only for data received for a read operation.

To request unit-check, the channel shall send a selective-reset frame with either the RU or the RO bit set to one. This is called a channel-initiated-unit-check request.

An I/O device may receive a channel-initiated-unit-check request anytime after the I/O device sends a command response or status in response to the first command of a channel program and before the I/O device has received a status-accepted frame in response to channel-end status. If command-chaining is expected, the I/O device may receive a channel-initiated-unit-check request anytime after the status frame having device-end status is sent and if chaining occurs, anytime before the I/O device has received a status-accepted frame in response to device-end status.

If the I/O device receives a channel-initiated-unit-check request when no I/O operation is considered to be in progress, for example after an I/O operation is nullified, the I/O device shall respond to the selective-reset with a device-level-ACK frame and perform a selective-reset.

If a retry request accompanies a unit-check request, the I/O device shall perform retry, if possible, and ignores the unit-check request. The I/O device accepts a unit-check request only if the channel either:

a) Does not request retry;

b) Requests retry but the I/O device does not accept retry.

Depending on conditions at the I/O device a channel-initiated-unit-check request may or may not be performed. If the I/O device determines it is able to provide the requested unit-check, the I/O device shall respond to the selective-reset with a status frame indicating unit-check and the CI bit set to one. If the I/O device determines it is unable to provide the requested unit-check or the unit-check request is received when no I/O operation is considered to be in progress, then the I/O device shall respond to the selective-reset frame with a device-level-ACK frame and perform a selective-reset.

If the channel receives a device-level-ACK frame in response to a channel-initiated-unit-check request, the channel shall terminate the I/O operation and provide the appropriate channel-error indication.

In certain instances the control unit is permitted to attempt recovery beyond that which was requested in a selective-reset frame. When this is done, the control unit shall return status, with or without retry, and the CI bit set to one in response to a selective-reset frame that either requests channel-initiated retry but does not request channel-initiated unit-check or does not request channel-initiated retry but does request channel-initiated unit-check for an overrun (RO bit set to one). If the selective-reset frame has the RC bit set to one and the RU and RO bits both set to zeros, the control unit shall be permitted to respond with unit-check status without a request for retry. If the selective-reset frame has the RC and RU bits both set to zeros and the RO bit set to one, the control unit shall be permitted to respond with unit-check status along with a request for retry.

If the channel receives a retry request in response to a channel-initiated-unit-check request for other than an overrun (RU bit set to one) that was made without a channel-initiated-retry request (RC bit set to zero), the channel shall detect a device-level error, shall not accept the status, and shall send the selective-reset frame with the RC, RO, and RU bits set to zeros.

13.3.3.2.1 Channel request for unit-check without overrun

If the channel recognizes a link-level or device-level error either during the execution of an I/O operation for the first command of a channel program or after a command other than the first command of a channel program is sent during chaining, the channel may request the control unit to recognize a unit-check in order to allow for the transfer of sense data that contains information necessary for recovery. The channel shall make a unit-check request for these conditions by sending a selective-reset frame with the RU bit set to one.

An I/O device shall accept a unit-check request only if it can provide appropriate sense data for the unit-check that will allow for device-dependent program recovery. When the sense data is generated, byte 0 shall indicate bus-out check. If the I/O device does not accept the unit-check request, a selective-reset shall be performed.

An I/O device that receives a selective-reset frame with the RU bit set to one and the RC bit set to zero shall not request retry. If the channel receives a status response with a request for retry and the CI bit is set to one, a device-level error shall be detected.

13.3.3.2.2 Channel request for unit-check with overrun

When during the execution of an I/O operation for a read, the channel recognizes an overrun on the data received, the channel shall request the control unit to recognize a unit-check for an overrun condition in order to allow for the transfer of the appropriate sense data that contains information necessary for recovery. The channel shall make a unit-check request for this condition by sending a selective-reset frame having the RO bit set to one.

An I/O device shall accept a unit-check request for overrun, unless prevented from doing so by an abnormal condition, and generate the appropriate sense data for an overrun. When sense data is generated, byte 0 shall indicate overrun. If an abnormal condition prevents the I/O device from accepting the unit-check request, a selective-reset shall be performed.

An I/O device that receives a selective-reset frame with the RO bit set to one and the RC bit set to zero may request retry. If a retry is requested, the CI bit shall be set to one. In this case the channel may or may not honor the retry request depending on conditions present.

NOTE – When an I/O device accepts a channel-initiated-unit-check request, the information presented to the ULP, if any, indicates the point of execution termination as viewed by the channel even though, in some cases, it is not known whether the current command was accepted by the I/O device. For those cases where the channel makes a channel-initiated-unit-check request instead performing a selective-reset, an I/O device accepting the request for a unit-check provides the appropriate sense data describing the point of execution termination as viewed by the I/O device. The sense data generated for the unit-check both indicates that the operation was terminated at the request of the channel and provides the information necessary for recovery.

13.3.4 Address-exception condition

An address-exception condition shall be recognized and a device-level-exception frame shall be sent when one of the following device frames is received with a device address for an uninstalled I/O device by a control unit that is not designed to respond to this condition with either unit-check status or a device-level-ACK frame.

- a command frame that starts an I/O operation either for the first command frame of a channel program or for a command frame for command-chaining;
- a request status with a connect-SOF delimiter;
- a cancel or a selective-reset.

The exception code indicated in the device-level-exception frame shall be address exception (see 12.2.4.9).

The device address returned in the device-level-exception frame for an address-exception condition shall be the same as the device address in the device frame which caused the exception. A device-level error shall be detected by the channel if the addresses do not match.

When an address-exception condition is recognized during command-chaining, the command-chaining condition shall be reset for the I/O device for which the current connection was created.

13.3.5 Device-level timeout

Device-level timeout shall be used to detect errors associated with the protocols used for device frames and device-level functions. There are two distinct modes of device-level timeout. When a device-level initiation frame is sent, the device-level timeout shall be used to ensure that a valid response to that initiation frame is received within the timeout interval. This is called device-level-initiation timeout. When a connection for a device-level function exists and the device-level timeout is not being used for a device-level initiation frame, then the device-level timeout shall be used to ensure that device-level activity continues to occur on the interface at least as often as the timeout interval. This is called device-level-inactivity timeout.

The device-level timeouts shall be performed using a device-level timer. The duration of the waiting period for the device-level timeout for both modes shall be the same as the duration of the link timeout (see 10.10).

For purposes of device-level timeout, the following shall be treated as initiation frames, even though some of these may be sent during a connection with a passive-SOF delimiter:

– cancel frame;

– command frame with the CH bit set to zero;

– request-connection frame;

– request-status frame which initiates a connection;

– selective-reset frame;

– status frame;

– system-reset frame.

For device-level-initiation timeout, when the initiation frame is sent, the device-level timer shall be started. If the normal response or a valid alternate link response or device-level response is not received before the device-level timer expires, a device-level-connection error shall be detected. Each device-level initiation frame shall have a different set of link-level and device-level responses which are recognized as responses to the initiation frame. If a link frame or device frame is received that is recognized as a response to the device-level initiation frame, the device-level timer shall be reset, and device-level-initiation timeout shall not be recognized. After this response, if a connection continues to exist for a device-level function, then the device-level timer shall be used for the device-level-inactivity timeout.

For a command frame with the CH bit set to zero, the following shall be recognized as responses:

– command-response frame;

– device-level-exception frame indicating address exception;

– link-level busy frame;

– link-level-reject frame;

– port-busy frame;

– port-reject frame;

– status frame.

For a request-status frame with a connect-SOF delimiter, the following shall be recognized as responses:

– device-level-ACK frame;

– device-level-exception frame indicating address exception;

– link-level-busy frame;

– link-level-reject frame;

– port-busy frame;

– port-reject frame;

– status frame.

For a cancel frame or a selective-reset frame, the following shall be recognized as responses:

– device-level-ACK frame;

– device-level-exception frame indicating address exception;

- link-level-busy frame;
- link-level-reject frame;
- port-busy frame;
- port-reject frame;
- status frame indicating control-unit-busy;
- status frame with the CI bit set to one (for selective-reset only).

For a system-reset frame, the following shall be recognized as responses:

- device-level-ACK frame;
- link-level-busy frame;
- link-level-reject frame;
- port-busy frame;
- port-reject frame.

For a request-connection frame, the following shall be recognized as responses:

- accept-connection frame;
- deny-connection frame;
- link-level-busy frame;
- link-level-reject frame;
- port-busy frame;
- port-reject frame.

For a status frame, the following shall be recognized as responses:

- cancel frame;
- command frame;
- request-status frame with the ES bit set to one when the status frame had the SA bit set to one;
- selective-reset frame;
- stack-status frame;
- status-accepted frame;
- system-reset frame.

While a connection exists for a device-level function and device-level-initiation timeout is not being performed, device-level-inactivity timeout shall be performed. If a control unit or channel fails to detect device-level activity on the interface for the duration of the device-level timeout period, a device-level-connection error shall be detected. Device-level activity on the interface shall consist of sending or receiving a device frame.

If a device-level initiation frame is sent while the device-level timer is being used for device-level-inactivity timeout, the device-level-inactivity timeout shall no longer be performed, and instead the device-level timer shall be started for device-level-initiation timeout for the initiation frame that is sent.

If a sequence is sent or received, any connection or pending connection that existed for a device-level function shall no longer be considered to exist; device-level-initiation timeout or device-level-inactivity timeout shall no longer be performed, and no device-level timeout shall be detected.

13.4 Resetting event

When an event occurs that causes the control unit or device to perform the system-reset function or its equivalent, both a condition and an indication shall be generated for each channel-path-and-device pair for which the reset was applicable. The condition, which is called the resetting-event condition, shall be generated when one of the following conditions occurs:

- either the control unit or device performs a system-reset;
- either the control unit or device performs a power-on reset;
- either the control unit or device makes the transition from offline to online;
- the channel path or control-unit path makes the transition from disabled to enabled;
- the control unit performs an initialization function;
- an error condition for a control unit or an I/O device causes system-reset or its equivalent to be performed.

If the resetting-event condition exists for the addressed device, it shall cause the control unit to reject any read-class command, other than the

IPL-read command, and some additional model-dependentcommands, by presenting ending status containing unit-check to the first command received that may be rejected after the condition occurs. Each device implementation may choose to allow certain additional commands not to be rejected by the resetting-event condition. However, the execution of these additional commands shall not cause the resetting-event condition to be reset. The following commands are always rejected when the resetting-event condition exists:

- reserve;
- release;
- unconditional reserve;
- assign;
- unassign;
- any command used to maintain path groups;
- any read-class command;
- any write-class command.

The resetting-event condition shall be cleared either when any command other than an IPL-read command is received and rejected with unit-check status or when an IPL-read command is received.

The resetting-event indication shall be cleared when either of the following conditions occurs:

- an IPL-read command is received;
- any command other than a basic sense command is received after the acceptance of unit-check status which resulted because of the resetting-event condition.

If a basic sense command is received, the resetting-event indication shall not be reset until the ending status for a basic sense command, which shall contain no errors or exception indications, is accepted by the channel.

A control unit shall not request a connection to signal unit-check status for a resetting-event condition alone, but, if unit-check is to be reported on the same channel path for another condition for which a control-unit request for connection may be made, then the sense data which is subsequently returned to the program may contain both the data for the other condition and the data that indicates a resetting-event condition existed.

The control unit shall be able to permit the execution of operations to devices for which no resetting-event condition exists when one or more resetting-event conditions are pending for other devices. The existence of the resetting-event condition and indication shall be treated similarly to the status and indication of a device that has been partitioned from the control unit; that is, the resetting-event condition shall be signaled first and cleared before any status reflecting the state of the actual device may be signaled. However, once unit-check status has been accepted by the channel path in order to report the resetting-event condition, the control unit may, depending on control-unit type, use the control-unit-busy status as a means of serializing the use of the control unit until the associated resetting-event indication has been cleared.

The existence of a resetting-event condition or a resetting-event indication or both on one channel path shall not be affected by any interface activity for that device from another channel path.

The sense data that is used to indicate a resetting event may be merged with other sense data when reported to the program; if this is done, the resetting-event indication shall be suitably distinct to allow the program to recognize the resetting-event condition along with the other conditions reported by the use of sense data.

Selective-reset, cancel, or system-reset shall not clear a resetting-event condition regardless of when they are signaled. If selective-reset, cancel, or system-reset is received when sense data exists in the control unit for the purpose of reporting a resetting-event indication, then the sense data shall be cleared, the resetting-event indication shall not be cleared, and a resetting-event condition shall become pending again.

13.5 Special functions

13.5.1 Path groups

When multiple logical paths exist between the channel subsystem and the I/O device, they may be logically arranged into a group of logical paths, all of which are associated with a single system or logical partition. The method for establishing a group of logical paths shall be model dependent for the specific I/O device.

An I/O device implementing the reserve or assign command shall also provide the reservation or assignment function, respectively, applying to a path group instead of to an individual path. The reservation or assignment performed for any path of a path group shall provide a single reservation or assignment which is applicable to all paths of the path group. If one path of the group is reset, the reservation or assignment shall be retained for the remaining paths of the path group.

A path group shall also be used to identify the logical paths which may be used for reconnection when dynamic reconnection is used.

13.5.2 Dynamic reconnection

Dynamic reconnection provides a device with the ability to select any available logical path in a group of logical paths for transferring information to a system. Specifically, a device that has disconnected from a logical path during the execution of a command may reconnect to any one of a group of logical paths to continue the execution of the current command. In addition, whenever the control unit or one of its attached I/O devices recognizes an asynchronous condition, the status may be presented to the system over any logical path in the group.

During execution of successive I/O operations, use of dynamic reconnection occurs only when both:

a) The I/O device is capable;

b) The channel subsystem is also capable.

The capabilities of the channel subsystem when using dynamic reconnection shall be model-dependent.

When dynamic reconnection is used, the sequences described elsewhere in this standard shall be unmodified. However, in some cases, the device-level protocols shall be modified. These changes shall be described in the following paragraphs and apply whenever a group of one or more logical paths has been formed.

When the I/O device presents status, except control-unit-end alone, on any logical path in the group after the I/O device has initiated an I/O operation, it may be interpreted by the channel as status resulting from the I/O operation in progress. When control-unit-end alone is presented on any logical path in the group after the I/O device has initiated an I/O operation, it shall not be interpreted as status resulting from the I/O operation in progress. When status is stacked on one of the logical paths, the stacked status may be presented on any other logical path in the group. If the channel signals stack-status, command-chaining may or may not be indicated when that status is subsequently accepted by the channel on any logical path in the group. If the channel signals stack-status, the command-chaining condition at the device may or may not be reset.

When system-reset is received and an I/O operation has been initiated but is not actively in progress on that logical path (that is, a connection does not already exist for the I/O operation on that logical path), performance of the system-reset function shall have no effect on the I/O operation. When system-reset is received over any logical path of the group and the system-reset function is performed while status is pending, the status shall remain pending and shall be presented on any logical path in the group. When system-reset is received and the system-reset function is performed over all logical paths in the group while an I/O operation is in progress or when status is pending, the corresponding I/O operation or pending status, as applicable, shall be reset.

When selective-reset is received on one logical path of the group and the selective-reset function is performed while an I/O operation for the same device is actively being executed (a connection is pending or exists for the I/O operation) on another logical path of the group, then the performance of the selective-reset function shall have no effect on the I/O operation. At any other time while an I/O operation is in progress, the performance of the selective-reset function on any logical path of the group shall reset the I/O operation. If status is stacked or pending when the selective-reset function is performed, the status shall be reset.

When cancel is received on one logical path of the group and the cancel function is performed while an I/O operation for the same device is actively being executed (a connection is pending or exists for the I/O operation) on another logical path of the group, then the performance of the cancel function shall have no effect on the I/O operation.

If the cancel function is performed after the I/O device has initiated an I/O operation but before the channel has accepted status containing channel-end, the device shall go to its normal ending point, and then status information shall be generated, if appropriate, and presented on any logical path in the group. If the cancel function is performed during this period, then the command-chaining condition at the device shall be reset, and command-chaining shall not be indicated when the status is subsequently accepted by the channel on any logical path of the group. If the cancel function is performed after the channel accepted status containing channel-end with command-chaining indicated but before the channel accepts status containing device-end, then the command-chaining condition at the device may or may not be reset, and command-chaining may or may not be signaled when that status is subsequently accepted by the channel on any logical path in the group.

If a device responds to a command with busy status, it shall subsequently present device-end to indicate the end of the busy period. The device shall present only one device-end indication per group of logical paths, regardless of the number of times it responded busy over the collection of logical paths in the group. The device-end may be presented on any logical path in the group, regardless of whether the associated device-busy indication had been signaled on that logical path. This device-end shall be accepted or reset before any new I/O operation may be accepted by the device from any logical path in the group.

When the control-unit-busy period no longer exists, a control unit shall present a control-unit-end status indication over each logical path over which it previously returned status of control-unit-busy. The control-unit-end status indication may be included with other status.

Whenever an error condition that is described by sense information has been recognized, prior to the resetting or acceptance of the sense information, any selection attempts for that device on any other logical path of the group shall receive either a control-unit-busy or device-busy indication. If the device presents device-end status separate from the presentation of channel-end and unit-check, the device-end status shall be presented over the logical path on which the unit-check was presented.

If a group of logical paths exists and the device is selected on a logical path other than the one for which the device requires selection, status of status-modifier, busy, and control-unit-end shall be returned as an indication that the device needs to be selected on a different logical path in the group.

14 Recovery actions for link-level and device-level errors

This clause describes recovery actions for link-level and device-level errors. When multiple errors are present, the link-level and dynamic-switch-port recovery performed shall be determined by the relative severity of the error, with recovery performed first for the most severe error detected.

For link-level recovery, the following shall be the order of severity, from most severe condition to least severe condition:

a) Link failure;

b) Offline condition;

c) Connection error;

d) Transmission error;

e) Destination-address-invalid error;

f) Link-protocol error;

g) Logical-path-not-established error;

h) Link-level-reject and port reject;

i) Test-initialization-result error;

j) Device-level error.

For recovery at a dynamic-switch port, the following shall be the order of severity, from most severe condition to least severe condition:

a) Link failure;

b) Offline condition;

c) Connection error;

d) Transmission error;

e) Dynamic-switch-port error.

Whenever a frame is received with an error by either a link-level facility or dynamic-switch port, that frame shall be discarded as part of the recovery performed.

Figures 46 and 47 contain flow diagrams of the recovery action taken for each error recognized. Recovery actions shown in the flow diagrams are described in later subclauses.

14.1 Link-level recovery

When an error is detected, link-level recovery, if appropriate, shall be the first recovery performed. Link-level recovery shall attempt to remove any existing connection by transmitting the appropriate sequence or sending the appropriate frame. The link-level recovery for certain error conditions shall resend a frame and may reset the internal indicators for completed initialization procedures or may invoke the initialization procedures.

When the internal indicator for an established logical path is reset at a control unit as a result of a recovery action, the pacing information previously provided by the channel for that logical path shall be removed.

14.1.1 Recovery for a link failure

When a link failure is recognized, the link-level facility shall enter the link-failure state for the affected link and shall perform the appropriate link-failure procedure for the type of link failure recognized (see 10.5).

When a link failure is recognized, the internal indicators for the logical paths previously considered established over the affected link shall be reset; that is, the link-level facility considers these logical paths to no longer exist.

If a frame is in the process of being sent at the time that a link failure is recognized, frame transmission shall be terminated by an abort delimiter, if possible, before the not-operational sequence or offline sequence is started.

When the channel recognizes a link failure, the link-level facility shall become unidentified. Furthermore, when the channel recognizes a link failure, initiative to perform initialization procedures shall not be generated as a result of the link-level facility becoming unidentified and the internal indicators for established logical paths being reset (see 10.1).

When a control unit other than a dynamic-switch control unit recognizes a link failure, the link-level facility shall become unidentified.

For the device-level recovery that is performed for a link failure, see 14.2.1.

14.1.2 Recovery for an offline condition

When an offline condition, that is, a condition which causes a link-level facility to enter the of-

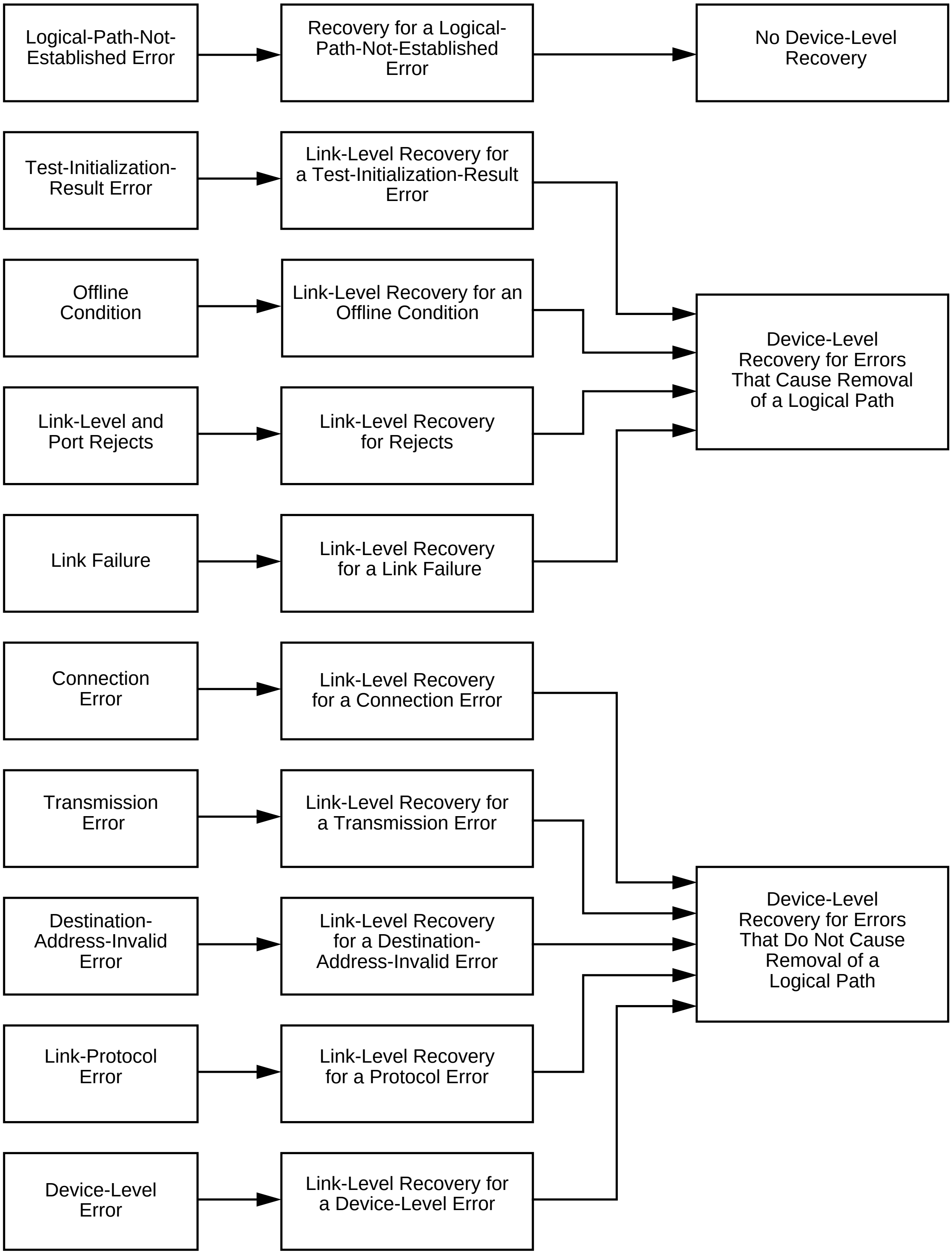


Figure 46 – Recovery actions for channels and control units

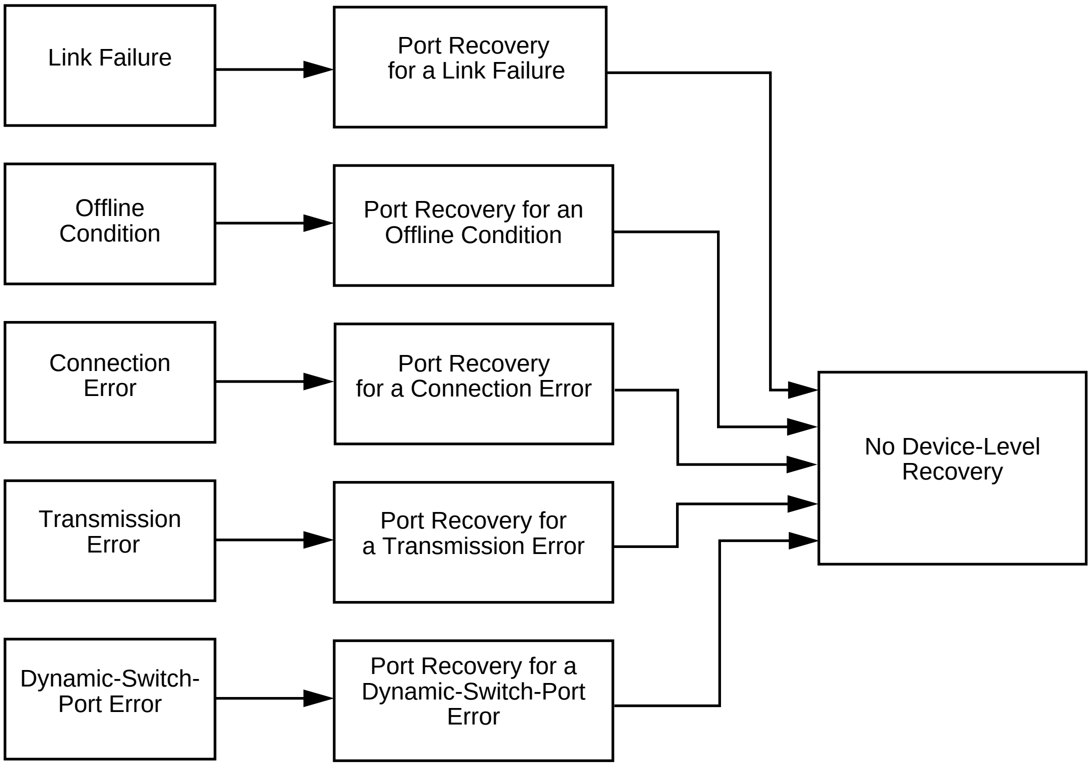


Figure 47 – Recovery actions for a dynamic-switch port

fline state, is recognized, the link-level facility shall enter the offline state and perform the appropriate offline procedure (see 10.3).

When an offline condition is recognized, the internal indicators for the logical paths previously considered established over the affected link shall be reset; that is, the link-level facility considers these logical paths to no longer exist.

If a frame is in the process of being sent at the time that an offline condition is recognized, frame transmission shall be terminated by an abort delimiter, if possible, before the offline sequence or unconditional-disconnect sequence is started.

When the channel recognizes an offline condition, the link-level facility shall become unidentified. Furthermore, when the channel recognizes an offline condition, initiative to perform initialization procedures shall not be generated as a result of the link-level facility becoming unidentified and the internal indicators for established logical paths being reset.

When a control unit other than a dynamic-switch control unit recognizes an offline condition, the link-level facility shall become unidentified.

For the device-level recovery performed for an offline condition, see 14.2.1.

14.1.3 Recovery for a connection error

A connection error is an error that violates the protocols which control the connection states (see 11.1.1). When a connection error is recognized, the link-level facility shall enter the connection-recovery state for the affected link and perform the appropriate connection-recovery procedure for the conditions present. The following describes the link-level recovery performed after the connection-recovery procedure is completed:

a) If, at the time the connection error was recognized, there was no pending or existing connection, no other link-level recovery shall be performed;

b) If, at the time the connection error was recognized, a link-control request frame was either partially sent (prior to the sending of the passive-EOF delimiter) or completely sent, one of the following recovery actions shall be performed after the connection-recovery procedure is completed:

1) If the connection-recovery procedure was performed before recognition of an error-free link-control response to the link-control request frame and the initiative to send the link-control request frame exists, the link-control request frame shall be resent;

2) If the connection-recovery procedure was performed before recognition of an error-free link-control response to the link-control request frame and the initiative to send the link-control request frame no longer exists, the link-control request frame shall not be resent. If the link-control request frame was other than an ELP or RLP, no other link-level recovery shall be performed. If the frame was an ELP or RLP, the channel shall recognize that a logical path may have been established or removed at the control unit, and one of the following recovery actions shall be performed:

i) If the frame sent was an ELP, an RLP shall be sent;

ii) If the frame sent was an RLP, an ELP shall be sent;

3) If the connection-recovery procedure is performed after recognition of an error-free link-control response to the link-control request frame, that is, after the pending connection is removed, no other link-level recovery shall be performed for a connection error;

c) If, at the time the connection error was recognized, a device frame was either partially sent (prior to the sending of the passive-EOF delimiter) or completely sent, no other link-level recovery shall be performed after the connection-recovery procedure is completed;

d) If, at the time the connection error was recognized, a connection was pending as the result of having received an initiation frame and recognition of the frame had not yet occurred, no other link-level recovery shall be performed after the connection-recovery procedure is completed;

e) If, at the time the connection error was recognized, a connection was pending as the result of having received an initiation frame and recognition of the frame had occurred, one of the following recovery actions shall be performed after the connection-recovery procedure is completed:

1) If the frame received was an error-free link-control request frame and a connection error that resulted in the connection-recovery procedure being performed was not caused by this frame, the link-level facility either may take no action or may perform the specified function. In either case, no response frame shall be sent. If the link-level facility performs the specified function and that function requires the sending of one or more link-control request frames, the link-level facility shall send those frames as appropriate;

2) If the frame received was the cause of a connection error that resulted in the connection-recovery procedure being performed, the link-level facility shall neither perform the function nor send a response. For example, if the connection-recovery procedure is performed as a result of having received two successive initiation frames without having sent an intervening response to the first frame, the link-level facility neither performs the specified function in the second frame nor sends a response;

f) If, at the time the connection error was recognized, a connection was pending or existed as the result of having received a device frame, no other link-level recovery shall be performed after the connection-recovery procedure is completed.

The number of times an initiation frame is resent as a result of an error that results in connection recovery shall be model-dependent. When the model-dependent number of times the frame can be resent is exhausted, initiative to perform the link-control function shall be suspended for the affected physical path.

When an alternate physical path may be selected to perform the link-control function that is being attempted (for example, the link-incident-notification function), initiative to perform that function shall be terminated on the physical path over which it was attempted and established on another physical path, provided the selected physical path is operational and the conditions associated with that path satisfy the conditions for performing the link-control function.

See 14.2.2 for the device-level recovery for a connection error.

14.1.4 Recovery for a transmission error

When a transmission error is recognized before frame reception is started, no link-level recovery shall be performed. When the transmission error is caused by a link-signal error, the link timer shall be initialized to the link-interval duration, and the timer shall be started.

When a transmission error is recognized after frame reception is started, the recovery action performed depends on the SOF and EOF delimiters used and the state of the connection at the time the frame is received. The following describes the recovery performed based on these conditions:

a) If the transmission error is recognized after frame reception is started with a connect-SOF delimiter, a link-level-reject with the transmission-error reject-reason code shall be sent, provided that all of the conditions for sending a link-level-reject are satisfied. (See 9.1.8);

b) If the transmission error is recognized after frame reception is started with a passive-SOF delimiter and a connection is pending, the recovery performed shall be the same as that performed for a connection error. (See 14.1.3);

c) If the transmission error is recognized after frame reception is started with a passive-SOF delimiter and a connection exists, one of the following recovery actions shall be performed:

1) If the error is recognized at the channel after an accept-command-response frame is sent but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame, no link-level recovery shall be performed;

2) If the error is recognized at a control unit after an accept-command-response frame is received in response to a command-response frame but before a status frame is sent, no link-level recovery shall be performed;

3) If the error is recognized at the channel after the channel has sent a command frame in response to a status frame but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame, no link-level recovery shall be performed;

4) If the error is recognized at the control unit after a command-response frame is sent in response to a command frame for other than the first command of a channel program but before a status frame for that command is sent, no link-level recovery shall be performed;

5) If the error is recognized under conditions other than those described in the preceding items, the link-level recovery performed shall be the same as that performed for a connection error;

d) If the transmission error is recognized in a frame with a disconnect-EOF delimiter, the recovery performed shall be the same as that performed for a connection error.

See 14.2.2 for the device-level recovery for a transmission error.

14.1.5 Recovery for a destination-address-invalid error

When a destination-address-invalid error is recognized, one of the following recovery actions shall be performed:

a) If the destination-address-invalid error is recognized in a frame with a connect-SOF delimiter and a passive-EOF delimiter, a link-level-reject with the destination-address-invalid reject-reason code shall be sent, provided that all of the conditions for sending a link-level-reject are satisfied (see 9.1.8);

b) If the destination-address-invalid error is recognized in a frame with a passive-SOF delimiter and a passive-EOF delimiter and a connection is pending, the recovery performed shall be the same as that performed for a connection error (see 14.1.3);

c) If the destination-address-invalid error is recognized in a frame with a passive-SOF delimiter and passive-EOF delimiter and a connection exists, one of the following recovery actions shall be performed:

  1) If the error is recognized at the channel after an accept-command-response frame is sent but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame, no link-level recovery shall be performed;

  2) If the error is recognized at a control unit after an accept-command-response frame is received in response to a command-response frame but before a status frame is sent, no link-level recovery shall be performed;

  3) If the error is recognized at the channel after the channel has sent a command frame in response to a status frame but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame, no link-level recovery shall be performed;

  4) If the error is recognized at the control unit after a command-response frame is sent in response to a command frame for other than the first command of a channel program but before a status frame for that command is sent, no link-level recovery shall be performed;

  5) If the error is recognized under conditions other than those described in the preceding items, the link-level recovery performed shall be the same as that performed for a connection error;

d) If the destination-address-invalid error is recognized in a frame with a disconnect-EOF delimiter, the recovery performed shall be the same as that performed for a connection error.

See 14.2.2 for the device-level recovery for a destination-address-invalid error.

14.1.6 Recovery for a link-protocol error

When a link-protocol error is recognized, one of the following link-level recovery actions shall be performed:

a) If the link-protocol error recognized is a connection-protocol error, the recovery performed shall be the same as that performed for a connection error (see 14.1.3);

b) If the link-protocol error recognized is other than a connection-protocol error and the frame has a connect-SOF delimiter and a passive-EOF delimiter, a link-level-reject with the appropriate reject-reason code shall be sent, provided all of the conditions for sending a link-level-reject are satisfied (see 9.1.8);

c) If (a) the link-protocol error recognized is other than a connection-protocol error, (b) the frame has a passive-SOF delimiter and a passive-EOF delimiter, and (c) connection is pending, the recovery performed is the same as that performed for a connection error;

d) If (a) the link-protocol error recognized is other than a connection-protocol error, (b) the frame has a passive-SOF delimiter and passive-EOF delimiter, and (c) a connection exists, one of the following recovery actions is performed:

  1) If the error is recognized at the channel after an accept-command-response frame is sent but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame, no link-level recovery shall be performed;

  2) If the error is recognized at a control unit after an accept-command-response frame is received in response to a command-response frame but before a status frame is sent, no link-level recovery shall be performed;

  3) If the error is recognized at the channel after the channel has sent a command frame in response to a status frame but before a status-accepted, stack-status, or request-status frame is sent in response to a

subsequent status frame, no link-level recovery shall be performed;

4) If the error is recognized at the control unit after a command-response frame is sent in response to a command frame for other than the first command of a channel program but before a status frame for that command is sent, no link-level recovery shall be performed;

5) If the error is recognized under conditions other than those described in the preceding items, the link-level recovery performed shall be the same as that performed for a connection error;

e) If the link-protocol error recognized is other than a connection-protocol error and the frame has a disconnect-EOF delimiter, the recovery performed shall be the same as that performed for a connection error.

See 14.2.2 for the device-level recovery for a link-protocol error.

14.1.7 Recovery for a logical-path-not-established error

When a logical-path-not-established error is recognized, a link-level-reject with the logical-path-not-established reject-reason code shall be sent, provided that all of the conditions for sending a link-level-reject are satisfied (see 9.1.8).

No device-level recovery shall be performed for a logical-path-not-established error.

14.1.8 Recovery for link-level-reject and port-reject

When either a link-level-reject or port-reject is received in response to an initiation frame, the link timer shall be reset and, depending on the reject-reason code and the number of attempts already made to resend the frame, the frame may or may not be resent. If the frame is not resent, initiative to perform the operation or function shall either suspended or removed. The frame shall be resent at least once on the same physical path if one of the following link-level or port reject-reason codes is indicated:

- transmission error;
- acquire-link-address error;
- address-invalid error;
- undefined-destination-address error;
- dynamic-switch-port intervention required;
- destination-port malfunction.

The number of times, beyond one, that the initiation frame can be resent shall be model-dependent. An SCN frame sent by a dynamic-switch control unit because of a change in state at a dynamic-switch port shall be resent indefinitely for a transmission-error reject-reason code.

If one of the following reject-reason codes is indicated, the initiation frame shall either be not resent or, optionally, resent a model-dependent number of times:

- destination-address-invalid error;
- reserved-field error;
- unrecognized-link-control-function error;
- protocol error;
- logical-path-not-established;
- unrecognized-device-level.

When the model-dependent number of times the frame can be resent is exhausted, the link-level recovery performed shall depend on the reject-reason code indicated and the operation or function being performed. The following summarizes the link-level recovery performed for the particular reject-reason code indicated:

- If the reject-reason code indicated is an address-invalid error, then the internal indicators for established logical paths held for the link-level facility to which the initiation frame was sent shall be reset. If the link-level facility that receives the reject has no logical path established with any link-level facility either as a result of the recovery performed for the reject or because it had no logical path established with any link-level facility when the reject was received, it shall become unidentified (except that a dynamic-switch control unit would not become unidentified);
- If the reject-reason code indicated is logical-path-not-established and the logical path was previously considered estab-

lished, the internal indicator for the logical path shall be reset;

- If the reject-reason code indicated is other than those described in the preceding items and the function being performed was indicated in a frame with a destination address other than all ones, the internal indicators for established logical paths held for the link-level facility to which the initiation frame was sent shall be reset.

The following summarizes the link-level recovery performed when the model-dependent number of times a frame can be resent is exhausted for the particular operation or function being attempted:

- If a link-control function is being attempted, initiative to perform that function for the physical path over which it was being attempted shall be suspended. The link-control function is not attempted for that physical path again until initiative to perform that function is reestablished; for example, when an SCN frame is received that specifies the affected link. When an alternate physical path may be selected to perform the link-control function that is being attempted (for example, the link-incident-notification function), initiative to perform that function shall be terminated on the physical path over which it was attempted and established on another physical path, provided the selected physical path is operational and the conditions associated with that path satisfy the conditions for performing the link-control function;
- If a device-level operation or function is being attempted and the reject-reason code was other than logical-path-not-established, initiative to perform that operation or function shall be removed for the logical path over which it was being attempted;
- If a device-level operation or function is being attempted and the reject-reason code was logical-path-not-established, then initiative to perform the appropriate initialization functions shall be generated. If successful, the device-level operation or function shall be retried; if not successful, the initiative to perform the device-level operation or function shall be removed for the logical path over which it was being attempted.

See 14.2.1 for the device-level recovery when the number of times a frame can be resent as a result of receiving a reject is exhausted.

14.1.9 Recovery for a test-initialization-result error

When a test-initialization-result error is recognized because the logical paths that are indicated in the test-initialization-result frame as no longer being established were previously considered to be established, the internal indicators shall be reset for those logical paths. At the channel, initiative shall be generated to perform the initialization procedures to reestablish the affected logical paths (see 10.1).

When a test-initialization-result error is recognized because the logical paths that are indicated in the TIR as being established were previously considered not established, there shall be no change to the internal indicators for those logical paths. If the control unit is defined in the configuration for the channel, an RLP frame shall be sent in an attempt to remove the logical path. Initiative shall not be generated to perform the initialization procedures to reestablish the affected logical paths. If the control unit is not defined in the configuration for the channel, the sending of the RLP frame shall be optional, and if not sent, the channel shall take no further recovery action (see 10.1).

See 14.2.1 for the device-level recovery performed when a test-initialization-result error causes the internal indicators for a logical path to be reset.

14.1.10 Recovery for device-level errors

When a device-level error is recognized, one of the following link-level recovery actions shall be performed:

a) If a device-level-connection error is recognized, the link-level facility shall be given initiative to perform the same recovery as that performed for a connection error (see 14.1.3);

b) If a device-level-protocol error is recognized in a frame that initiates a connection and a connection does not already exist, the link-level facility shall be given initiative to send a reject with the protocol-error reject-reason code indicated;

c) If a device-level-protocol error is recognized in a frame that neither initiates nor removes a connection and a connection is already pending, the link-level facility shall be given initiative to perform the same recovery as that performed for a connection error;

d) If a device-level-protocol error is recognized in a frame that neither initiates nor removes a connection and a connection already exists, the link-level facility shall be given initiative to perform one of the following recovery actions:

1) If the error is recognized at the channel after an accept-command-response frame is sent but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame, no link-level recovery shall be performed;

2) If the error is recognized at a control unit after an accept-command-response frame is received in response to a command-response frame but before a status frame is sent, no link-level recovery shall be performed;

3) If the error is recognized at the channel after the channel has sent a command frame in response to a status frame but before a status-accepted, stack-status, or request-status frame is sent in response to a subsequent status frame, no link-level recovery shall be performed;

4) If the error is recognized at the control unit after a command-response frame is sent in response to a command frame for other than the first command of a channel program but before a status frame for that command is sent, no link-level recovery shall be performed;

5) If the error is recognized under conditions other than those described in the preceding items, the link-level recovery performed shall be the same as that performed for a connection error;

e) If a device-level-protocol error is recognized in a frame that removes a connection, the link-level facility shall be given initiative to perform the same recovery as that performed for a connection error.

14.2 Device-level recovery

Device-level recovery shall be performed for error conditions that affect only device-level functions or device-level operations that are being attempted or performed over the SBCON I/O interface. Internal errors detected at the channel or control unit or errors at the I/O device shall not be included in device-level recovery.

When a link-level error is detected in a frame during a pending connection or existing connection for the performance of a device-level function or operation, device-level recovery shall be performed on the basis of link-level recovery either having been completed or not having been performed. If link-level recovery is performed, device-level recovery shall be performed on the basis of whether or not the internal indicator for a logical path was reset, or, when the internal indicators for logical paths are not reset, on the basis of whether or not a pending or existing connection was removed. If link-level recovery is not performed, device-level recovery shall be performed on the basis of an abnormal condition having occurred rather than the type of error recognized.

14.2.1 Errors that cause the removal of a logical path

The removal of a logical path for the error cases that initiate the recovery described in 14.2.1.1 and 14.2.1.2 involves only the resetting of the internal indicators for a logical path and not the transmission of an RLP frame. It is the resetting of the internal indicator for a logical path that causes the following device-level recovery. The errors that cause the internal indicator for a logical path to be reset are:

- certain test-initialization-response errors (see 9.1.7);
- errors caused by the recognition of the offline sequence;
- receipt of certain rejects (see 9.1.8 and 9.1.9);
- exhausted retry counts for rejects;
- a link failure.

14.2.1.1 Recovery at the channel

When the internal indicator of an established logical path is reset by the channel and no device-level operation is being performed or attempted over the affected logical path, no device-level recovery shall be performed.

When the internal indicator for an established logical path is reset by the channel and a device-level operation is being attempted over the affected logical path, one of the following device-level recovery actions shall be performed:

a) If the device-level operation being attempted is initiation of the first command of a channel program, the I/O device shall be considered not operational on that logical path, and no further device-level recovery shall be performed;

b) If the operation being attempted is a cancel or selective-reset that resulted from an error being detected during the execution of an I/O operation over the interface, the I/O operation, if not already terminated, shall be terminated;

c) If the operation being attempted is a cancel or selective-reset that resulted from program initiative, the I/O device shall be considered not operational on that logical path, and no further device-level recovery shall be performed;

d) If the operation being attempted is a device-level system-reset, no further device-level recovery shall be performed;

e) If the operation being performed is a request status, any operation associated with the specified I/O device shall be terminated. Initiative to retrieve previously stacked status from that I/O device shall be removed.

When the internal indicator for a logical path is reset by the channel and an I/O operation is being executed over the affected logical path, the I/O operation shall be terminated.

14.2.1.2 Recovery at the control unit

When the internal indicator for a logical path is reset, the equivalent of a system-reset shall be performed with respect to the affected logical path or paths, and no further device-level recovery shall be performed.

14.2.2 Errors that do not cause the removal of a logical path

The errors that do not cause the removal of a logical path are:

- connection error;
- transmission error;
- destination-address-invalid error;
- link-protocol error;
- device-level error.

These errors result in one of the following link-level recovery actions during a device-level operation:

a) The link-level recovery performed shall remove the pending or existing connection by either the performance of the connection-recovery procedure or the sending of a reject;

b) No link-level recovery is performed, and the pending or existing connection was removed by the error. The link error or device-level error occurred under conditions for which link-level recovery shall not be performed;

c) No link-level recovery is performed, and the pending or existing connection is unaffected. The link error or device-level error occurred under conditions for which link-level recovery shall not be performed.

The number of times a device-level operation or function can be retried as the result of an error, other than the receipt of a reject in response to an initiation frame, shall be model-dependent. When the model-dependent number of times the operation or function can be retried is exhausted, initiative to perform the operation or function on the affected logical path shall be removed. If conditions permit, the operation or function may be attempted on another channel path.

For brevity, the removal of a connection during link-level recovery or the errors that do not result in any link-level recovery are referred to as abnormal conditions in 14.2.2.1 and 14.2.2.2.

See also 13.2.2**,** 13.2.3 and 13.2.4 for a description of the post-recovery effect of the cancel, selective-reset, and system-reset functions on device-level operations.

14.2.2.1 Recovery at the channel

When, for the first command of a channel program, the channel recognizes an abnormal condition after the command frame is sent but before a status-accepted frame with the CH bit set to one or command frame with the CH bit set to one is sent in response to a status frame that is received in direct response to the command frame, the channel shall recognize that nullification of the command at the I/O device occurs. Subsequent to recovery from the abnormal condition, the channel may resend the command frame for the first command of the channel program.

When, for the first command of a channel program, the channel recognizes an abnormal condition after the command frame is sent but before a status-accepted frame with CH bit set to zero is sent in response to a status frame that is received as a response to the command frame, the channel shall recognize that nullification of the command, if it was accepted by the I/O device, occurs. In this case, if the status frame indicated that supplemental-status was available, the channel has sent the request-status frame with the ES bit set to one but has not sent the status-accepted frame, and the channel recognizes an abnormal condition, then the channel shall recognize that nullification of the command, if it was accepted by the I/O device, occurs.

When, for the first command of a channel program, the channel recognizes an abnormal condition after the command frame is sent but before an accept-command-response frame is sent in response to the command-response frame, the channel shall recognize that nullification of the command at the I/O device occurs. Subsequent to recovery from the abnormal condition, the channel may resend the first command of the channel program.

When, for a command other than the first command of a channel program, the channel recognizes an abnormal condition after the channel has sent a command frame in response to a status frame but before a response to the command frame is recognized, the channel shall send a selective-reset frame that requests a unit-check but does not request a command-retry.

When the channel recognizes an abnormal condition after an accept-command-response frame has been sent in response to the command-response frame for the first command of a channel program or after a command-response frame is received in response to a command frame other than the first command of a channel program, the channel shall send a selective-reset frame. If a status-accepted frame or a command frame has not yet been sent in response to a subsequent status frame, the selective-reset frame shall contain a request for command-retry and, optionally, a request for unit-check.

If the abnormal condition was the removal of the connection or if the link-level recovery performed for the abnormal condition removed the connection, then the channel may communicate with other control units, or with the same control unit for other I/O devices, prior to sending the selective-reset frame with the request for command-retry. When the selective-reset frame with the request for command-retry is sent, it shall use the same logical path as the failed operation for which this retry is being attempted.

When the channel recognizes an abnormal condition after a status-accepted frame is sent but before a device-level-ACK frame is recognized, the channel shall terminate the operation, if any, discards the status, and subsequently send a selective-reset frame that does not request command-retry or recognition of unit-check.

When the channel recognizes an abnormal condition after a stack-status frame is sent but before a response to that frame is recognized, no device-level recovery shall be performed, and the channel shall consider the stack-status frame as being accepted and the status to be stacked at the control unit.

When the channel recognizes an abnormal condition after a request-status frame is sent during a connection in order to retrieve supplemental-status but before a status-accepted frame is sent in response to the subsequent status frame, and an I/O operation is in progress, the channel shall send a selective-reset frame with a request for command-retry and, optionally, a request for unit-check. If the abnormal condition was the removal of the connection or if the link-level recovery performed for the abnormal condition removed the connection, then the channel may communicate with other control units, or with the same control unit for other I/O devices, prior to sending the selective-reset frame with the request for command-retry. When the selective-reset frame with the request for command-retry is sent, it shall use the same logical path as the failed operation for which this retry is being attempted.

When the channel recognizes an abnormal condition after a request-connection frame is recognized but before a status-accepted frame or a command frame is sent in response to a subsequent status frame, no device-level recovery shall be performed.

When the channel recognizes an abnormal condition after one of the following frames is sent but before a response to the frame is recognized, no device-level recovery shall be performed, and the frame is resent:

– a cancel frame;

– a selective-reset frame with or without a request for command-retry;

– a selective-reset frame with or without a request for unit-check;

– a request-status used other than to request supplemental-status during a connection;

– a system-reset frame.

14.2.2.2 Recovery at the control unit

When, for the first command of a channel program, a control unit recognizes an abnormal condition after a command frame is recognized but before a status-accepted frame or command frame is recognized in response to a status frame that was sent as a response to the first command frame, the control unit shall nullify an I/O operation being initiated, if any, and perform no further device-level recovery. In this case, if the control unit sent a status frame indicating that supplemental-status is available and an abnormal condition is detected before a status-accepted frame is recognized, the control unit shall nullify the I/O operation being initiated, if any, and perform no further device-level recovery. The state of the control unit and I/O device is such that no further recovery shall be required by the channel or program.

When, for the first command of a channel program, a control unit recognizes an abnormal condition after a command frame is recognized but before an accept-command-response frame is recognized in response to a command-response frame, the control unit shall nullify the pending I/O operation and perform no further device-level recovery. The state of the control unit and I/O device is such that no further recovery shall be required by the channel or program.

When a control unit recognizes an abnormal condition after the control unit recognizes a command frame in response to a status frame, but before the control unit has sent a response to the command frame, the control unit shall recognize a unit-check. The control unit shall request command-retry, if appropriate, when the unit-check status is sent to the channel.

When a control unit recognizes an abnormal condition after the control unit has either recognized an accept-command-response frame in response to a command-response frame for the first command of a chain or after a command-response frame is sent in response to a command frame other than for the first command of a chain but before the control unit has sent a subsequent status frame, the control unit shall recognize a unit-check. The control unit shall request a retry, if appropriate, when the unit-check status is sent to the channel.

When the control unit recognizes an abnormal condition after a status frame other than in response to a command frame for the first command of a chain is sent but before an error-free response to the status frame is recognized, the status shall be considered to be not accepted by the channel and one of the following recovery actions shall be performed:

a) If the status frame was sent in order to present status that includes channel-end status, the control unit shall recognize a unit-check. The control unit may request command-retry;

b) If the status frame was sent in response to a selective-reset frame that was sent during a connection, the status shall remain pending at the control unit;

c) If the status frame contains busy status with status-modifier, the status shall be withdrawn by the control unit;

d) If the status frame was sent in response to an accept-connection frame that was received in response to a request-connection frame, the status shall remain pending at the control unit;

e) If the status frame was sent in response to a request-status frame that was sent during a connection in order to retrieve supplemental-status, the status shall remain pending at the control unit;

f) If the status frame was sent in response to a request-status frame that initiates a connection, the status shall remain pending.

When the control unit recognizes an abnormal condition after having recognized a status-accepted frame in response to a status frame but before the device-level ACK is sent, the status sent in the status frame shall remain pending.

When the control unit recognizes an abnormal condition after a stack-status frame is received but before a response to that frame is sent, no device-level recovery shall be performed, and the control unit shall consider the stack-status frame as being accepted and the status to be stacked.

When a control unit recognizes an abnormal condition after recognition of a cancel frame, a selective-reset frame without a request for retry and without a request for unit-check, or a system-reset frame, but before a response to that frame is sent, the control unit may perform or nullify the indicated function.

When a control unit recognizes a command frame in response to a status frame and then recognizes a cancel frame before a response to the command frame is made, a unit-check shall be recognized.

When a control unit recognizes an abnormal condition after recognition of a selective-reset frame with either a request for retry or a request for unit-check but before a response to that frame is sent, the control unit may nullify or perform one of the indicated functions. If the control unit elects to perform the selective-reset function rather than the request for retry or request for unit-check, no further device-level recovery shall be performed. If the control unit elects to perform either the request for retry or request for unit-check, the status generated shall become pending status. If this status is presented during a subsequent control-unit-initiated connection (request-connection), the CI bit shall be set to zero. If, before the control unit can present this status to the channel, another selective-reset frame with a channel-initiated-retry request is received for the I/O device, the status, if sent in response, shall be sent with the CI bit set to one.

When the control unit recognizes an abnormal condition after recognition of a request-status frame but before a response to that frame is sent, the control unit may nullify or perform the indicated function; that is, if the status requested was previously considered to be stacked, that status shall now be considered unstacked. If the request status is being performed for the purpose of retrieving supplemental-status, the status shall remains pending.

When the control unit recognizes an abnormal condition after a request-connection frame is sent but before a response is received, no device-level recovery shall be performed.

When the control unit receives a selective-reset frame with a channel-initiated-retry request, but an abnormal condition causes the connection to be removed, and yet the control unit elects to perform the requested retry, then the status generated shall become pending status and, if presented during a control-unit-initiated connection, the CI bit shall be set to zero. The CI bit shall be set to one only in status frames that are sent in response to a channel-initiated-retry request or a channel-initiated-unit-check request.

14.3 Dynamic-switch-port recovery

This subclause describes recovery at a dynamic-switch port for errors recognized. The recovery performed depends on the error condition and the dynamic-switch-port state. A dynamic-switch port shall not perform device-level recovery.

14.3.1 Recovery for a link failure

When a link failure is detected by a dynamic-switch port in the static state, no recovery shall be performed, and no port state change shall occur.

When a link failure is recognized by a dynamic-switch port in the offline state, the dynamic-switch port shall enter the link-failure state. If the link failure is caused by the recognition of the not-operational sequence, the offline sequence shall be transmitted. If the link failure is caused by either the recognition of the unconditional-disconnect or unconditional-disconnect-response sequence while in the wait-for-offline-sequence state, the dynamic-switch port shall enter the link-failure state and transmits the not-operational sequence.

When a link failure is recognized by a dynamic-switch port in any state other than the static or offline state, a dynamic connection, if present, shall be removed, and the port shall enter the link-failure state if it is not already in the link-failure state. If the dynamic-switch port was not already in the link-failure state when it recognized the link failure, initiative to perform state-change notification shall be generated. If the link failure is caused by one of the following, the not-operational sequence shall be transmitted:

a) Either loss of signal or loss of sync for greater than the link-interval duration;

b) Failure to recognize an expected valid response to a transmitted sequence before the appropriate interval duration has elapsed.

If the link failure is caused by the recognition of the not-operational sequence, the offline sequence shall be transmitted.

14.3.2 Recovery for an offline condition

When a condition that causes a dynamic-switch port to go offline is recognized by a dynamic-switch port in a connection state, the existing connection shall be removed, and the dynamic-switch port shall enter the offline state; initiative to perform state-change notification shall be generated.

When a condition that causes a dynamic-switch port to go offline is recognized by a dynamic-switch port in a state other than a connection state, no recovery action shall be performed (see 10.3 for additional information).

14.3.3 Recovery for a connection error

When a connection error is recognized by a dynamic-switch port in a connection state, the existing connection shall be removed, and both ports shall enter the connection-recovery state and perform the connection-recovery procedure.

When a connection error that is caused by the recognition of the unconditional-disconnect-response sequence is recognized by a dynamic-switch port in the busy or inactive state, the dynamic-switch port shall enters the connection-recovery state and perform the appropriate connection-recovery procedure.

A connection error shall not be recognized in any of the other states of the dynamic-switch port.

14.3.4 Recovery for a transmission error

When a transmission error is detected by a dynamic-switch port in the link-failure, connection-recovery, offline, or static state, no recovery shall be performed, and no port state change shall occur.

When a link-signal error is recognized by a dynamic-switch port in the connection state, the existing connection shall be removed, and, as a result of the removal of the connection, the affected dynamic-switch ports shall perform the connection-recovery procedure.

When a link-signal error is recognized by a dynamic-switch port in either the inactive or busy state, the dynamic-switch port shall perform connection recovery.

When a transmission error other than a link-signal error or a code-violation error in a frame attempting to initiate a connection is detected by a dynamic-switch port in the inactive, busy, or connection state, no recovery shall be performed, and no port state change shall occur.

When a dynamic-switch port in the inactive state recognizes a code-violation error in a frame that is attempting to initiate a connection, the dynamic-switch port shall enter the busy state and a port-busy frame with a source-port-busy-unconnected reason code shall be sent. Once a dynamic connection is made, code-violation errors detected by the connected dynamic-switch ports shall not cause a port-reject frame to be sent and shall not cause any frames to be discarded.

14.3.5 Recovery for a dynamic-switch-port error

When a port error is recognized by a dynamic-switch port in the inactive state, a port-reject frame with the appropriate reject-reason code shall be sent, provided all of the conditions for sending a port reject are satisfied (see 9.1.9).

A dynamic-switch-port error shall not be recognized in any of the other dynamic-switch-port states.

15 Specific I/O-device commands

15.1 Basic sense command

The basic sense command shall initiate a sense operation at the addressed I/O-device and cause the retrieval of up to 32 bytes of sense data. The basic sense command shall be available on all I/O-devices and shall be accepted by an available control unit even when the addressed I/O-device is in a not-ready state. If the control unit detects an error during the sense operation, unit-check shall be indicated with the channel-end status condition.

The purpose of the basic sense command is to provide sense information detailed enough to ascertain the state of the I/O-device and identify any unusual conditions associated with the execution of an I/O operation during which an error was detected.

The basic sense command shall initiate a sense operation at all I/O-DEVICEs and shall not cause the command-reject, intervention-required, data-check, or overrun bit to be set to one. If the control unit detects an equipment malfunction the equipment-check bit shall be set to one, and unit-check shall be indicated in the device-status-byte.

15.2 No-operation control command

The no-operation control command shall perform no operation at the I/O-device, except for satisfying any previously indicated chaining operations and allowing certain I/O-devices to wait for conditions of checking (or any device-synchronizing indications) before disconnecting from the channel. If the command is accepted, the no-operation order shall cause the addressed device to respond with channel-end and device-end without causing any action at the I/O-device. Depending on the I/O-device, the order may be executed as an immediate operation. A no-operation control command shall not reset pending sense information.

If a no-operation control command is sent to an I/O-device in the not-ready state and no status is pending for the addressed I/O-device at the control unit, then the command shall not be accepted, and the control unit shall return unit-check status in response to the command (see 12.2.3.2.7.)

15.3 Read-configuration-data command

The read-configuration-data command shall initiate an operation to read a configuration record at the I/O-device. A configuration record contains information that describes the internal configuration of the I/O-device. Execution of the read-configuration-data command proceeds exactly as for a read command.

The command code for the read-configuration-data command shall be I/O-device dependent. It may be provided by the I/O-device in a *command-information-word* (CIW) (see 15.6.2).

When a control unit receives the read-configuration-data command and the addressed I/O-device is absent or not ready, the command may or may not be executed. If the command is executed under these conditions, the configuration record shall contain the same information as it would contain if the I/O-device were present and ready. If the command is not executed under these conditions, the command shall be rejected with unit-check status.

The configuration record consists of an integral number of 32-byte fields. The number of 32-byte fields contained in a particular configuration record shall be determined by dividing 32 into the value that is provided in the count field of the read-configuration-data CIW that is transferred by issuing the sense-ID command to the same I/O-device.

Each 32-byte field is identified as either a *node-element descriptor (NED)* or a *node-element qualifier (NEQ)*, or it is unused (see 16.3, 16.4 and 16.5). The term *node element* refers to any identifiable entity within the I/O-device.

A configuration record may contain any combination of NEDs, NEQs and unused fields, except that the record shall contain only one general NEQ but at least two NEDs, one of which is designated as a *token NED*.

If the record contains multiple NEDs (other than the NED designated as a token NED), the order in which they are arranged shall be the reverse of the order in which the corresponding node elements are accessed when an I/O operation is performed.

NOTE – For example, if a configuration record contains NEDs that describe a control unit and an I/O-device, the NED that describes the I/O-device is first in the configuration record. It shall not be required that NEDs be placed in contiguous 32-byte fields.

Two or more similar node elements that have no hierarchical relationship with respect to each other shall be described by consecutive NEDs, but they shall not be placed in any particular order.

When one node element emulates another, and both are described by NEDs, the NED that describes the emulated node element shall precede the NED that describes the actual node element. Multiple NEDs that describe emulated node elements may precede a single NED that describes the actual node element.

An NEQ that contains device-dependent information about a specific node element is called a *specific NEQ*. Specific NEQs shall be placed in contiguous 32-byte fields, immediately following the NED that describes the node element to which the NEQ information applies. Multiple specific NEQs may apply to a single node element.

An NEQ that contains device-dependent information that is not related to a particular node element is called a *general NEQ*.

A configuration record that is obtained by addressing an I/O-device on one interface may contain different information than a configuration record that is obtained by addressing the same I/O-device on a different interface.

Between initializations of the I/O-device, the format and contents of the configuration record for a particular device shall remain constant when requested on a particular I/O interface, unless the I/O-device indicates to the program, by means of an I/O-device-dependent procedure, that the contents of the record may no longer be valid. In this case, the format or contents of a newly obtained configuration record may be different from the format or contents of a previously obtained configuration record, even though both records pertain to the same I/O-device and are obtained on the same I/O interface.

Figure 48 illustrates the sequence of NEDs and NEQs within a configuration record. Although figure 48 shows an 80-word record with five NEDs and three NEQs, a configuration record may be as small as 24 words and may not include more than two NEDs, or an emulation NED, or specific NEQs.

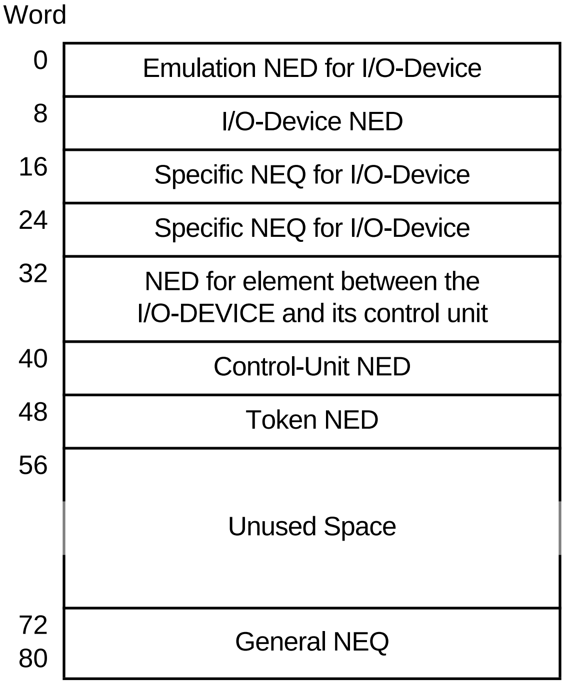


Figure 48 – Sequence of NEDs and NEQs within a configuration record

15.4 Read-IPL command

The initial-program-loading (IPL) read command shall initiate an operation to read an IPL record from the I/O-DEVICE. The IPL function shall cause a basic read command (read command with all modifier bits set to zeros) to be transferred to the designated I/O-DEVICE. I/O-DEVICEs that perform special functions in order to provide an IPL record shall recognize the basic read command as the read-IPL command only if it is the first command received after a system-reset function. Other I/O-DEVICEs may just provide the next available record or reject the command.

15.5 Read-node-identifier command

The read-node-identifier command shall initiate an operation to read a node-identification record.

The command code for the read-node-identifier command shall be I/O-DEVICE dependent. It may be provided by the I/O-DEVICE in a CIW (see 15.6.2).

Normally, a CCW that specifies the read-node-identifier command shall be immediately preceded by a CCW that specifies the set-interface-identifier command and command-chaining. In this case, the node-selector and interface ID that are provided to the I/O-device by the set-interface-identifier command shall be used to select the node that is to be identified in response to the read-node-identifier command.

When the node-selector provided by a preceding set-interface-identifier command contains a value of zero and the flag field of the node descriptor (ND) specifies that the requested node ID is valid, the I/O-device shall transfer a 32-byte node descriptor followed by one or more node qualifiers (NQs) (see 16.1 and 16.2). There shall be at least as many node qualifiers as are required to provide all of the interface IDs that are recognized by the I/O-device when they are specified by set-interface-identifier commands using the same interface that was used to receive the read-node-identifier command. Depending on the I/O-device, the node qualifiers containing interface IDs may be followed by additional node qualifiers containing model-dependent information. The value contained in the count field of the CIW that describes the read-node-identifier command shall be a multiple of 32 and shall specify the amount of data that is transferred by the I/O-device when the data includes node qualifiers. The number of node qualifiers that are transferred shall be determined by dividing the CIW-count value by 32 and subtracting one from the result.

When the node-selector provided by a preceding set-interface-identifier command contains a value of zero and the flag field of the node descriptor specifies that the requested node ID is not valid, the I/O-device shall transfer only a 32-byte node descriptor.

When the node-selector provided by a preceding set-interface-identifier command contains a value of 1 or 2, the I/O-device shall transfer only a 32-byte node descriptor.

When a control unit receives the read-node-identifier command and the addressed I/O-device is not installed or not ready, the command may or may not be executed. If the command is executed under these conditions, the node-identification record shall contain the same information as it would contain if the I/O-device were present and ready.

When a CCW that specifies the read-node-identifier command is not immediately preceded by a CCW that specifies the set-interface-identifier command and command-chaining, the read-node-identifier command shall be terminated with status containing unit-check, and the command-reject sense bit shall be set to one in the corresponding sense data.

15.6 Sense-ID command

The sense-ID command shall initiate a sense-type operation to provide the type/model number and, depending on the model, additional information about the I/O-device. If the control unit or I/O-device is available and not busy, then the sense-ID command shall be executed, and up to 256 bytes of identification data shall be transferred.

The sense-ID command is similar to a read command, except that the data is obtained from model-dependent indicators rather than from a record source. Sense-ID bytes are placed in a buffer in the same order as those transferred by the read command. Sense-ID byte 0 will be the first byte transferred from the device to the channel

The data sent in response to sense-ID are shown in table 37.

Table 37 – Sense-ID response data

Sense-ID Bytes	Contents
0	x’FF’
1-2	Control-unit type number
3	Control-unit model number
4-5	I/O-DEVICE type number
6	I/O-DEVICE model number
7-255	Variable-length extended-identification information

I/O-devices that attach to SBCON I/O interfaces shall not reset basic sense information as a result of executing the sense-ID command.

The sense-ID command shall not initiate any operations other than the sensing of identification information. If the control unit and I/O-device are available, then the sense-ID command shall be executed even if the I/O-device is absent or not ready.

Basic sense data shall not be reset as a result of an I/O-DEVICE executing the sense-ID command.

15.6.1 Basic identification information

Bytes 1 and 2 shall contain the four-decimal-digit control-unit type number that corresponds directly with the control-unit type number on the tag attached to the control unit.

Byte 3 shall contain the control-unit model number, if applicable. If not applicable, byte 3 is a byte of all zeros.

Bytes 4 and 5 shall contain the four-decimal-digit I/O-DEVICE type number that corresponds directly with the I/O-DEVICE type number on the tag attached to the I/O-device.

Byte 6 shall contain the I/O-DEVICE model number, if applicable. If not applicable, byte 6 is a byte of all zeros.

The seven bytes of basic-identification information may be followed by variable-length extended-identification information beginning with byte 7 (see 15.6.2).

Some older control units and I/O-devices may properly execute the sense-ID command, may execute the command as the basic sense command, or may reject the sense-ID command with unit-check status, depending on the control-unit and I/O-DEVICE model.

If a control unit can be addressed separately from the attached I/O-device or I/O-devices, then the response to the sense-ID command shall depend on the unit addressed. If the control unit is addressed, the response to the sense-ID command shall be as follows:

Bytes	*Contents*
0	x’FF’
1-2	Control-unit type number
3	Control-unit model number

The response shall consist of the control-unit type and model number, with normal ending status presented after byte 3.

For cases where the control unit and device are not distinct, the identification information shall be the same as if a control unit were being addressed.

In the case where an older device may be addressed separately, the response to the sense-ID command shall be as follows:

Bytes	*Contents*
0	x’FF’
1-2	I/O-device type number
3	I/O-device model number

The response shall consist of the I/O-DEVICE type and model number, with normal ending status presented after byte 3.

All unused sense bytes shall be set to zeros.

When the first byte of the data that is transferred by the sense-ID command contains x’FF’, it shall specify that the addressed I/O-device has properly executed the sense-ID command and the data contains basic-identification information.

The type and model information returned in the basic-identification information shall not necessarily match the data on the serial-number plate attached to the surface of the unit. I/O-Devices which provide extended-identification information accurately describe their manufacturer-specified nomenclature (and may also describe the type and model of an I/O-device they emulate) by providing node-element descriptors (see 15.6.2 and 16.3).

15.6.2 Extended-identification information

All SBCON I/O-devices return extended-identification information that describes additional characteristics of the I/O-device. Extended-identification information consists of one byte of zeros, byte 7, and beginning with byte 8, a variable number of command-information-words (CIWs). Basic and extended-identification information together shall not exceed 256 bytes.

Each command-information-word (CIW) describes a command that may be used by the program to cause the I/O-device to perform a specific operation. The format of a command-information-word is shown in figure 49.

	ET	R	CT	Command	Count	
bit	0	2	4	8	16	31

Figure 49 – Command-information-word (CIW) format

ET (entry type): Bits 0-1 of byte 0 shall be zero and one, respectively.

R: Bits 2-3 of byte 0 shall be reserved.

CT (command type): The command-type field, bits 4-7 of byte 0, contains a code that specifies the operation to be performed by means of the command that is specified in this CIW. The command-type values that may be specified and their meanings are as follows:

Value	*Meaning*
0	*Read-configuration-data:* The command specified in this CIW shall be used to obtain data, arranged in a device-independent format, that describes the internal configuration of the I/O-device (see 15.3).
1	*Set-interface-identifier:* The command specified in this CIW shall be used to identify an interface which is used to determine the node ID interface that is used to determine the node ID. This interface identification shall determine the node-identification information that is obtained by a subsequent read-node-identifier command (see 15.7).
2	*Read-node-identifier:* The command specified in this CIW is used to obtain node-identification information (see 15.5).
3-15	Reserved.

Command: Byte 1 shall contain the command code that is used to cause the I/O-device to perform the operation specified by the code that is contained in the CT field.

Count: Bytes 2-3 shall specify the number of bytes that are transferred when the command specified in the command field is issued to the same I/O-device that was addressed to obtain this count field.

All devices designed to attach to SBCON I/O interfaces shall provide extended-identification information that includes at least a CIW for each of the following commands:

- read-configuration-data;
- set-interface-identifier;
- read-node-identifier.

15.7 Set-interface-identifier command

The set-interface-identifier command shall initiate an operation to write a four-byte interface-identifier record to an I/O-device. Execution of the set-interface-identifier command proceeds exactly as for a write command.

The command code for the set-interface-identifier command shall be I/O-device dependent. It may be provided by the I/O-device in a CIW (see 15.6.2).

Normally, a CCW that specifies the set-interface-identifier command shall also specify command-chaining and shall be followed by a CCW that specifies the read-node-identifier command. In this case, the node-selector and interface ID that are provided to the I/O-device by the set-interface-identifier command shall be used to select the node ID that is provided in response to the chained read-node-identifier command.

When a control unit receives the set-interface-identifier command and the addressed I/O-device is absent or not ready, the command may or may not be executed. If the command is executed under these conditions, the same operations shall be performed as would be performed if the I/O-device were present and ready.

If command-chaining is not specified in the set-interface-identifier CCW, or command-chaining is specified but the next CCW does not specify the read-node-identifier command, no other operations shall be performed for the set-interface-identifier command, and the execution of this command shall have no effect on any subsequent read-node-identifier command.

The format of the interface-identifier-record is shown in figure 50.

	Order	r r r r r r	NS	Interface ID	
bit	0	8	14	16	31

Figure 50 – Interface-identifier-record format

Order: Byte 0 shall contain the value x'B0' hex.

Reserved (r): Bits 8-13 shall be reserved and set to zeros.

NS (Node-Selector): Bits 14-15 shall specify which type of node is to be identified in response to the read-node-identifier command specified in a CCW which in turn is chained to the CCW that specifies this set-interface-identifier command. The meaning of each value is as follows:

Value	*Meaning*
0	The node determined by the interface that is used to receive the read-node-identifier command is to be identified. The contents of the interface-ID field shall have no meaning in this case and are ignored.
1	The node determined by the interface associated with the specified interface ID is to be identified.
2	The node attached (by means of a link) to the interface associated with the specified interface ID is to be identified.
3	Reserved.

Interface ID: Bytes 2-3 shall contain the interface ID of an interface for which node-identification information is requested.

When a control unit receives an interface-identifier record with invalid contents, the set-interface-identifier command shall be terminated with status containing unit-check, and the command-reject sense bit shall be set to one in the corresponding sense data. The interface-identifier record shall have invalid contents when byte 0 contains a value of x’B0’ and one or more of the following conditions exist:

- bits 8-13 are not all zeros;
- the NS field contains a value of 3;
- the NS field of the interface-identifier record contains a value of 1 or 2, and the I/O-device does not recognize the interface ID;
- the NS field contains a value of 2, and the requested information is not available for the specified interface and will not be available if requested again.

If byte 0 contains a value other than x’B0’, the I/O-device may recognize that value as an order to perform some function other than set-interface-identifier, or the I/O-device may terminate the command. The action taken shall be I/O-device dependent.

16 I/O device self-description

This clause describes the formats for self-description information. Self-description is the ability of a product to provide identifying and other information on request at any of its interfaces. Self-description is a basic capability of all devices that attach to the SBCON I/O interface.

The ability of the components of a configuration to supply self-description information serves a number of purposes, including, but not limited to, the following:

- an inventory of the components of a configuration can be assembled and maintained under program control;
- the interconnections of the configuration can be determined;

Specific uses of self-description information include:

- determining whether the actual configuration at initialization matches a predefined configuration;
- determining, after a malfunction such as loss of light on a SBCON I/O interface, whether the restored configuration matches the configuration that existed before the malfunction;
- providing unique identification of failing components under program control.

The term *self-describing component* (SDC) shall be used in this publication to mean a product that is attached to, and can provide self-description information on, one or more channel paths, and that has a unique identification based on information provided on the manufacturer's serial-number plate attached to the external surface of its enclosure. The physical interfaces to which the channel paths are attached are called *SDC interfaces*. Each such point of attachment shall determine a node. Therefore, an SDC shall contain as many nodes as it contains SDC interfaces.

The self-description information that is obtained using a particular SDC interface shall always includes the identity of the node that is determined by that SDC interface; it shall also include the identities of all other nodes contained in the SDC.

The components of a node are all the entities that can be accessed using the SDC interface that determines the node. Within a node, certain of its components may be identified as node elements. It shall be model-dependent which components of a node are identified as node elements, except that I/O devices shall be identified as node elements. Only those components of a node that are identified as node elements shall be represented in self-description information for that node.

The formats in which self-description information is provided include node descriptors, node qualifiers, node-element descriptors, and node-element qualifiers. These formats are described in the following subclauses.

16.1 Node descriptor

A node descriptor (ND) is a 32-byte field that describes a node. The node descriptor is returned in the first 32 bytes of a node-identification record provided in response to a read-node-identifier command.

The node ID is composed of two parts:

a) *SDC ID:* The first 26 bytes of the node ID shall identify the SDC containing the interface that determines the node. The SDC ID shall correspond to the information provided on a serial-number plate attached to the external surface of the structure that contains the self-describing component.

b) *Interface ID (Tag):* The last two bytes of the node ID contain an interface identifier (ID) that shall uniquely identify the physical location of the associated SDC interface.

Node IDs with the same SDC ID shall not use the same interface ID.

Collectively, the 25 bytes of information contained in word 1, bytes 0 and 1 of word 2, bytes 1-3 of word 3, and words 4-7 of the node descriptor shall provide a worldwide-unique node identifier.

The format of a node descriptor is shown in figure 51.

Flags: Byte 0 of word 0 describes the manner in which selected fields of the node descriptor are to be interpreted. The meaning of bits 0-7 is as follows:

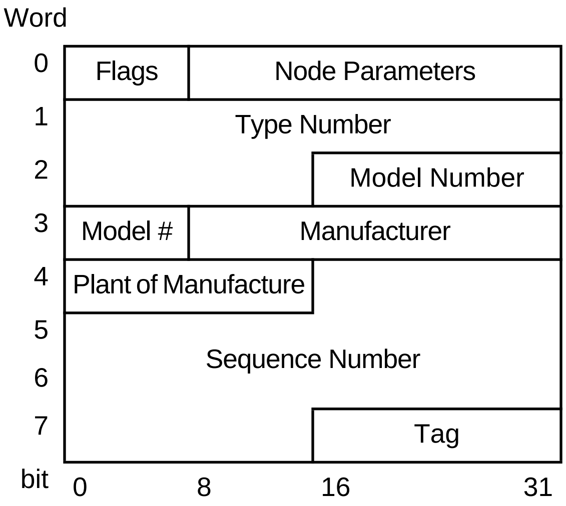


Figure 51 – Node descriptor

Bits *Description*

0-2 *Node-ID-validity:* Bits 0-2 contain a three-bit code that shall describe the validity of the node ID contained in words 1-7. The codes and their meanings are as follows:

0 Node-ID is valid.

1 Node ID is valid, however, it may not be current. This value shall be used when the SDC has obtained the requested node ID but subsequently has observed some event (such as the loss of signal on a link) which may have resulted in a configuration change. The SDC has been unable to obtain the node ID again.

2 Node ID is not valid. The SDC is unable to obtain the requested node ID. Except for the node-ID-validity field, the contents of the node descriptor shall have no meaning.

4-7 Reserved.

Node-parameters: Bytes 1-3 of word 0 contain additional information about the node. When bit 3 of the flag field is zero, indicating that this is a device-type node, the contents of bytes 1-3 of word 0 are as follows:

Byte *Description*

1 Reserved and set to zero.

2 *Class:* Byte 2 of word 0 contains an eight-bit code that specifies the class to which the device belongs. The codes and their meanings are as follows:

0	Unspecified class
1	Direct access storage (DASD)
2	Magnetic tape
3	Unit record (input)
4	Unit record (output)
5	Printer
6	Communications controller
7	Terminal (full screen)
8	Terminal (line mode)
9	Standalone channel-to-channel adapter (CTCA)
10	Switch
11	Protocol converter
12-255	Reserved

3 When code 11 is specified in the class field, indicating that this is a protocol converter, and when the link address of the device with which the protocol converter communicates has been determined by the protocol converter, byte 3 contains the link address of that device. Otherwise, byte 3 contains zero.

When bit 3 of the flag field is one, indicating that this is a channel-subsystem-type (CSS-type) node, the contents of bytes 1-3 of word 0 shall be as follows:

Byte *Description*

1 *Type:* When the class field contains a value other than 4, this field shall be reserved and set to zeros. When the class field contains a value of 4, byte 1 of word 0 shall contain host-related side and partitioning-type information for the specified interface. The bits and their meanings are as follows:

– Bits 0-2 shall be reserved and set to zeros.

– Bit 3 shall contain a bit that identifies the machine side.

The value of the bit shall directly reflect a numeric identifier of a host side. For example, a value of b’0’ designates host-side 0. Likewise, a value of b'1' designates host-side 1.

All nodes which are part of the same host side shall have the same value.

– Bit 4, when zero, shall indicate that the host cannot be physically partitioned into more than one host side. Bit 4, when one, shall indicate that the host is capable of being physically partitioned into more than one side.

– Bit 5, when zero, shall indicate that the host has not been physically partitioned into more than one side; bit 5 shall always be zero if bit 4 is zero. Bit 5, when one, shall indicate that the host has been physically partitioned into more than one side.

– Bits 6-7 shall be reserved and are set to zeros.

2 *Class:* Byte 2 of word 0 contains a code that specifies the class to which the interface belongs. The codes and their meanings are as follows:

0	Unspecified class
1	Channel path
2	Integrated channel-to-channel adapter (CTCA)
3-255	Reserved

3 *Identification:* Byte 3 of word 0 contains the *channel-path-identifier* (CHPID) of the channel path that contains the specified interface.

The contents of the following fields correspond to the information provided on a serial-number plate attached to the external surface of the SDC.

Type-number: Word 1 and bytes 0-1 of word 2 contain the six-character (0-9) EBCDIC type number of the SDC. The type number shall be right justified with leading EBCDIC zeros if necessary.

Model-number: Bytes 2-3 of word 2 and byte 0 of word 3 contain, if applicable, the three-character (0-9 or uppercase A-Z) EBCDIC model number of the SDC. The model number shall be right justified with leading EBCDIC zeros if necessary.

Manufacturer: Bytes 1-3 of word 3 contain a three-character (0-9 or uppercase A-Z) EBCDIC code that identifies the manufacturer of the SDC.

Plant-of-manufacture: Bytes 0-1 of word 4 contain a two-character (0-9 or uppercase A-Z) EBCDIC plant code that identifies the plant of manufacture for the SDC.

Sequence-number: Bytes 2-3 of word 4, words 5-6, and bytes 0-1 of word 7 contain the 12-character (0-9 or uppercase A-Z) EBCDIC sequence number of the SDC. The sequence number shall be right justified with leading EBCDIC zeros if necessary.

A serial number consists of the concatenation of the plant-of-manufacture designation with the sequence-number designation.

Tag: Bytes 2-3 of word 7 contain the physical identifier for the SDC interface that is identified by the preceding 26 bytes of the node descriptor.

Notes

1 The information in the node ID should only be used to determine the identity of the component. The information in the node ID should not be used to determine the functional characteristics of the component. The information in the configuration record, that is, the emulation node-element descriptor, or, in its absence, the node-element descriptor, may be used to infer the programming characteristics of the device (see 16.3).

2 Model numbers may change as the result of concurrent upgrade of the SDC. Programs should be written to allow the model number to change without affecting operations.

16.2 Node-qualifier

A node qualifier (NQ) is a 32-byte field containing SDC information about a node. NQs, when provided, shall follow the node descriptor (ND) in the node-identification record returned in response to a read-node-identifier command (see 15.5).

The contents of a node qualifier shall be dependent on the setting of the flag field of the node qualifier. The format of a node qualifier is shown in figure 52.

Flags: Byte 0 of word 0 describes the manner in which the contents of the node-qualifier-infor-

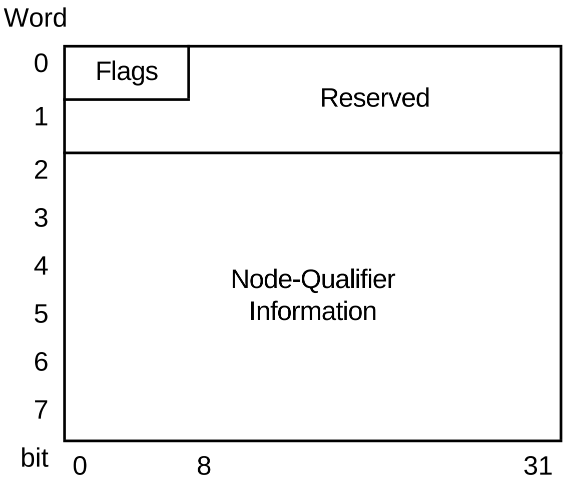


Figure 52 – Node qualifier

mation field are to be interpreted. The meaning of bits 0-7 is as follows:

Bits	*Description*
0-2	Node-qualifier contents. Bits 0-2 contain a three-bit code that specifies the contents of the node-qualifier-information field contained in words 2-7 as follows: 0 The node-qualifier-information field contains a list of the interface IDs that shall be recognized when they are presented using the same interface that was used to request this information. This list may or may not be followed by model-dependent information. 1 The node-qualifier-information field contains model-dependent information. 2-7 Reserved.
3-7	Reserved.

Reserved: Bytes 1-3 of word 0 and word 1 shall be reserved.

Node-qualifier-information: When bits 0-2 of the flag byte contain a value of zero, the node-qualifier-information field contains a list of interface IDs. The list is contained in a set of contiguous words beginning with the first word of the node-qualifier-information field. The format and contents of each word that describes interface IDs is shown in figure 53:

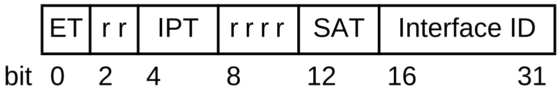


Figure 53 – Interface ID format

ET (entry-type): Bits 0-1 contain a two-bit code that specifies the contents of the corresponding word. The ET values and their meanings are as follows:

0	The corresponding word (and the remainder of the node-qualifier-information field) contains no meaningful information.
1	The corresponding word contains an interface ID.
2	The corresponding word contains an interface ID that is the first or last of a range of interface IDs. The first word in the node-qualifier-information field that has an ET field with a value of 2 shall specify the beginning of an interface-ID range, and shall be immediately followed by another word with an ET value of 2 that specifies the end of the interface-ID range. The node-qualifier-information field may contain subsequent pairs of contiguous words with ET values of 2 which specify additional interface-ID ranges.
3	The corresponding word (and the remainder of the node-qualifier-information field) contains model-dependent information.

Reserved (r): Bits 2-3 are reserved.

IPT (Interface-protocol type): Bits 4-7 contain a code that specifies the type of interface protocol used by the link or links indicated by the interface ID or the interface-ID range, respectively. The codes and their meanings are as follows:

0	Unspecified
1	Parallel I/O
2	SBCON I/O
3	Fiber-extended channel

4	Fiber-extended control unit
5-15	Reserved.

The IPT field shall be valid when the entry-type field has the value 1 or when the entry-type field has the value 2 and the entry is specifying the beginning of the interface-ID range.

Reserved (r): Bits 8-11 are reserved.

SAT (Subassembly type): Bits 12-15 contain a code that specifies the type of subassembly used on the link or links indicated by the interface ID or the interface-ID range, respectively. The codes and their meanings are as follows:

0	Unspecified
1	LED fiber optic
2	Laser fiber optic
3-15	Reserved.

The subassembly-type field shall be valid when the entry-type field has the value 1 or when the entry-type field has the value 2 and the entry is specifying the beginning of the interface-ID range.

Interface-ID: Bytes 2-3 contain an SDC interface ID.

Model-dependent-information: When bits 0-2 of the flag field contain a value of 1, the node-qualifier-information field contains model-dependent information.

16.3 Node-element-descriptor

A node-element descriptor (NED) is a 32-byte field that describes a node element. At least two NEDs shall be returned in a configuration record provided in response to a read-configuration-data command (see 15.3).

Collectively, the 28 bytes of information contained in words 1-7 of the NED shall provide a worldwide-unique node-element identifier (ID). The node-element ID maybe used to distinguish a node element from every other node element, independent of the SDC that contains it and the address, if any, that is used to access it. The format of a NED is shown in figure 54.

Flags: Byte 0 of word 0 describes the manner in which the contents of the 32-byte field are to be interpreted. The meaning of bits 0-7 is as follows:

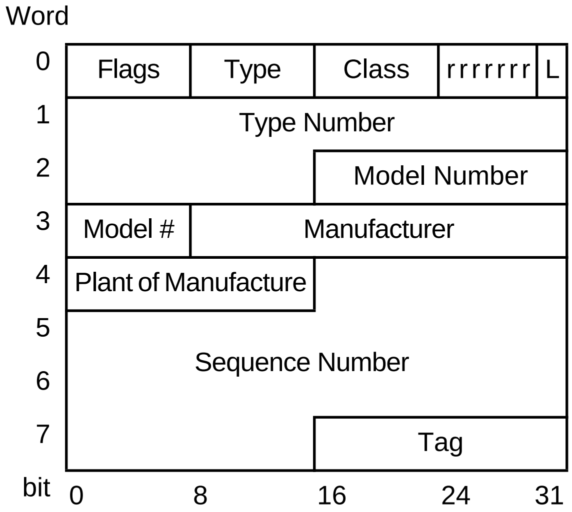


Figure 54 – Node-element-descriptor

Bits *Description*

0-1 *Field-identifier:* Bits 0-1 contain a two-bit code that identifies the contents of this 32-byte field. The values that may be specified and their meanings are as follows:

0	The field is unused.
1	The field contains a specific NEQ.
2	The field contains a general NEQ.
3	The field contains a NED.

The remainder of this subclause describes the contents of the 32-byte field for a field identifier value of 3, which indicates this is a NED.

2 *Token-indicator:* When one, bit 2 indicates that this NED is a token NED. A token NED shall be used to establish a relationship between two or more configuration records. The total collection of configuration records that contain the same token NED is called a *related-configuration-record set*.

The token NED shall always be the last NED of the configuration record and shall have nothing to do with the order in which node elements are accessed to perform I/O operations. The node-element ID contained in a token NED may be identical to the node-element ID contained in another NED in the same configuration record. This shall be the only case where two NEDs in the same configuration record can contain the same node-element ID. Only one NED of a configuration record shall be designated as a token NED. A token NED shall not be the only NED of a configuration record, shall not be followed by specific NEQs, and shall not be specified as an emulation NED. Flag bits 3 and 4 shall not both be zeros in a token NED. The level (L) bit (bit 31 of word 0 of the NED) shall have no meaning in a token NED.

NOTE – Certain control programs assume that related-configuration-record sets are disjoint with respect to their node elements. Therefore, if an NED that is neither an emulation NED nor a token NED appears in more than one configuration record, all the configuration records in which it appears should have the same token NED, in order to accommodate this control-program dependency.

3-4 Serial-number indicator. A serial number consists of the concatenation of the plant-of-manufacture designation and the sequence-number designation. Bits 3-4 contain a two-bit code that specifies the contents of the plant-of-manufacture and sequence-number fields as follows:

0 The serial number contained in this NED shall be identical to the serial number contained in the next NED that has a nonzero serial-number indicator.

1 The serial number contained in this NED is an installation-unique value that shall not necessarily be the same as the serial number of the node element described by this NED.

2 The serial number contained in this NED is the serial number of the node element described by this NED.

3 Same meaning as described for the value 2.

5 Reserved.

6 *Emulation NED:* When one, bit 6 indicates that this NED describes an emulated node element. An emulation NED shall provide a means for the I/O device to indicate to the program that the I/O device has the characteristics of another I/O device, while still providing accurate identification of itself. The actual node element that performs the emulation is the next node element that has bit 6 set to zero. When zero, bit 6 indicates that this NED does not describe an emulated node element.

7 Reserved.

Type: Byte 1 of word 0 contains a code that identifies the type of node element that is described by this NED. The values that may be specified and their meanings are as follows:

0	Unspecified node-element type.
1	The node element is an I/O device.
2	The node element is a control unit.
3-255	Reserved.

Class (CL): When the type field (byte 1 of word 0) contains a value of one (specifying that the node element is an I/O device), byte 2 of word 0 contains a code that specifies the class to which the I/O device belongs. When the type field contains a value other than one, byte 2 of word 0 shall have no meaning. The values that may be specified and their meanings are as follows:

0	Unspecified class
1	Direct access storage (DASD)
2	Magnetic tape
3	Unit record (input)
4	Unit record (output)
5	Printer
6	Communications controller
7	Terminal (full screen)
8	Terminal (line mode)
9	Channel-to-channel adapter (CTCA)
10	Switch
11	Protocol convertor
12-255	Reserved

Reserved (r): Bits 24-30 of word 0 are reserved.

Level (L): Bit 31 of word 0 when one, shall specify that this node element has no hierarchical relationship to the next NED in the configuration record. When zero, bit 31 shall specify that the node element described by this NED is accessed subsequent to the node element described by the next NED in the configuration record when an I/O operation is performed.

Type-number: Word 1 and bytes 0-1 of word 2 contain the six-character (0-9) EBCDIC type number of the node element described by this NED. The type number shall be right justified with leading EBCDIC zeros if necessary.

Model-number: Bytes 2-3 of word 2 and byte 0 of word 3 contain, if applicable, the three-character (0-9 or uppercase A-Z) EBCDIC model number of the node element described by this NED. The model number shall be right justified with leading EBCDIC zeros if necessary.

Manufacturer: Bytes 1-3 of word 3 contain a three-character (0-9 or uppercase A-Z) EBCDIC code that identifies the manufacturer of the node element described by this NED.

Plant-of-manufacture: Bytes 0-1 of word 4 contain a two-character (0-9 or uppercase A-Z) EBCDIC plant code that identifies the plant of manufacture for the node element described by this NED.

Sequence-number: Bytes 2-3 of word 4, words 5-6, and bytes 0-1 of word 7 contain the twelve-character (0-9 or uppercase A-Z) EBCDIC sequence number of the node element described by this NED. The sequence number shall be right justified with leading EBCDIC zeros if necessary.

Tag: Bytes 2-3 of word 7 contain a 16-bit unsigned binary integer that, together with the remaining contents of the node-element ID, uniquely identifies the node element described by this NED. The tag makes it possible to distinguish two or more node elements when the remaining portions of their node-element IDs are identical. For example, in the absence of the tag, the node-element IDs of two separate I/O devices contained in the same head-disk assembly (HDA) could be identical if the I/O devices have no serial numbers except that of the common HDA.

NOTE – For classes of node elements that are subassemblies of host-type SDCs, such as channel-to-channel adapters, model numbers may change as the result of concurrent upgrade of the SDC. Programs should be written to allow the model number to change without affecting operations.

16.4 Specific-node-element-qualifier

A specific node-element qualifier (NEQ) is a 32-byte field that contains device-dependent configuration information. A specific NEQ or a set of contiguous specific NEQs contains information regarding the node element described by the immediately preceding NED in a configuration record provided in response to a read-configuration-data command (see 15.3). The format of a specific NEQ is shown in figure 55.

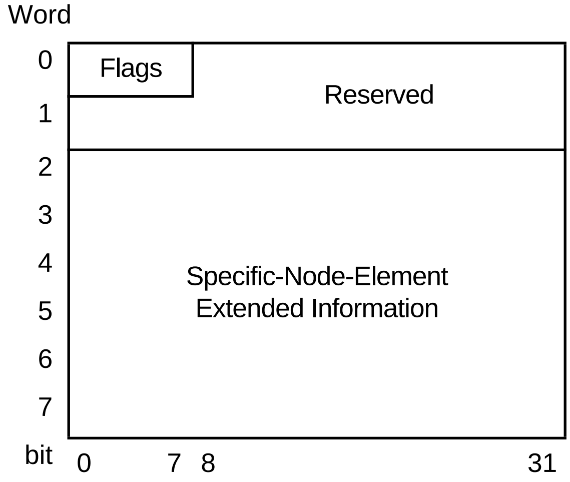


Figure 55 – Specific-node-element-qualifier

Flags: Byte 0 of word 0 describes the manner in which the contents of the 32-byte field are to be interpreted. The meaning of bits 0-7 is as follows:

Bits *Description*

0-1 Field identifier. Bits 0-1 contain a code that identifies the contents of this 32-byte field. The values that may be specified and their meanings are as follows:

- 0 The field is unused.
- 1 The field contains a specific NEQ.
- 2 The field contains a general NEQ.
- 3 The field contains a NED.

The remainder of this subclause describes the contents of the 32-byte field for a field-identifier value of one, which indicates that this is a specific NEQ.

2-7 Reserved.

Reserved (r): Bytes 1-3 of word 0 and word 1 are reserved.

Specific-node-element-extended-information: Words 2-7 contain node-element-dependent configuration information that applies to the node element described by the immediately preceding NED.

16.5 General-node-element-qualifier

A general node-element qualifier (NEQ) is a 32-byte field containing configuration information that applies to all of the node elements that are described in the configuration record in a configuration record provided in response to a read-configuration-data command (see 15.3). The format of a general NEQ is shown in figure 56.

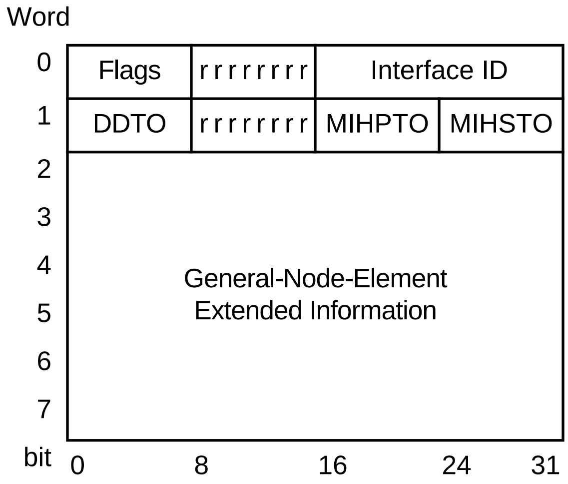


Figure 56 – General-node-element-qualifier

Flags: Byte 0 of word 0 describes the manner in which the contents of the 32-byte field are to be interpreted. The meaning of bits 0-7 is as follows:

Bits	*Description*
0-1	*Field identifier:* Bits 0-1 contain a code that identifies the contents of this 32-byte field. The values that may be specified and their meanings are as follows: 0 The field is unused. 1 The field contains a specific NEQ. 2 The field contains a general NEQ. 3 The field contains a NED.

The remainder of this subclause describes the contents of the 32-byte field for a field identifier value of 2, which indicates that this is a general NEQ.

2-7	Reserved.

Reserved (r): Byte 1 of word 0 and bytes 1-3 of word 1 are reserved.

Interface-ID: Bytes 2-3 of word 0 contain a value that shall uniquely identify the SDC interface that was used to request the configuration record. All configuration records that are requested on the same SDC interface shall contain the same value in the interface-ID field.

DDTO (device-dependent-time-out): Byte 0 of word 1 contains an unsigned binary integer that specifies a device-dependent time-out value. This value shall indicate the maximum time that the I/O device will take to perform its internal error recovery. The DDTO is expressed in seconds.

MIHPTO (missing-interrupt-handler-primary-time-out): Byte 2 of word 1 contains a value which specifies the time-out value recommended by the control unit which is used to set the missing-interrupt-handler time for the I/O device. This time shall be the longest time a channel program is expected to last taking into account any I/O device recovery mechanisms and any queuing time as a result of the presentation of busy or channel command-retry status if there is no value specified for the missing-interrupt-handler secondary timeout (MIHSTO). If a MIHSTO is specified, then this time shall be the longest time a channel program which contains no model-dependent long-running commands is expected to last, taking into account any device recovery mechanisms and any queuing time as the result of busy or channel command-retry status.

The MIHPTO value is a base-ten value obtained from the mantissa and exponent specified in the MIHPTO field.

The format of the MIHPTO field is shown in figure 57.

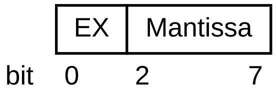


Figure 57 – MIHPTO field

When the mantissa is greater than zero, bits 0-1 of byte 2 of word 1 (EX) contain an unsigned binary integer that shall be the exponent for MIHPTO values greater than zero. When the mantissa is zero, no value shall be specified for the MIHPTO and bits 0-1 of byte 2 of word 1 (EX) shall have no meaning.

Bits 2-7 of byte 2 of word 1 (Mantissa) contain an unsigned binary integer that shall be the mantissa for the MIHPTO value. When this field contains a zero, no value shall be specified for the MIHPTO.

The decode of the MIHPTO exponent field is shown in table 38:

Table 38 – MIHPTO exponent field definition

Exponent Value	MIHPTO Value
0 0	Increments of 1s of seconds
0 1	Increments of 10s of seconds
1 0	Increments of 100s of seconds
1 1	Increments of 1,000s of seconds

MIHSTO (missing-interrupt-handler-secondary-time-out): Byte 3 of word 1 contains a value which specifies the control-unit's recommended value for the program's extended-missing-interrupt-handler time-out value for the I/O device. The MIHSTO value is the larger of:

- the maximum expected duration of any long-implicit allegiance formed for device-dependent long-running commands.
- the maximum expected duration of a device-dependent long-busy condition.

The MIHSTO value shall be the maximum expected duration of long-busy conditions or the longest expected execution time for channel programs containing a model-dependent long-running command, taking into account any device recovery mechanisms and any queueing time as a result of busy or channel command-retry status.

The missing-interrupt-handler-secondary-time-out value is a base-ten value based on the mantissa and exponent specified in the MIHSTO field.

The format of the MIHSTO field is shown in figure 58.

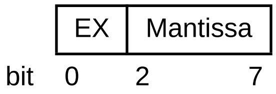


Figure 58 – MIHSTO field

When the mantissa is greater than zero, bits 0-1 of byte 3 of word 1 (EX) contain an unsigned binary integer that shall be the exponent for MIHSTO values greater than zero. When the mantissa is zero, no value shall be specified for the MIHSTO and bits 0-1 of byte 2 of word 1 (EX) shall have no meaning.

Bits 2-7 of byte 3 of word 1 (Mantissa) contain an unsigned binary integer that is the mantissa for the MIHSTO value. When this field contains a zero, no value is specified for the MIHSTO.

The decode of the MIHSTO exponent field is shown in table 39:

Table 39 – MIHSTO exponent field definition

Exponent Value	MIHSTO Value
0 0	Increments of 1s of seconds
0 1	Increments of 10s of seconds
1 0	Increments of 100s of seconds
1 1	Increments of 1,000s of seconds

General-node-element-extended-information: Words 2-7 contain node-element-dependent configuration information that shall apply to the collection of node elements that are described in the configuration record.

Annex A

(informative)

Status combinations

The following rules indicate status combinations which are appropriate or inappropriate, depending on the state of an I/O operation when status is presented by the I/O device. If a status-byte is accepted by the channel and the status-byte contains a combination of status bits that is inappropriate at the time the status is presented, an error condition may be recognized by the channel. If such an error condition is recognized, command-chaining is suppressed.

Appropriate and inappropriate status combinations are described below and, in most cases, shown in a corresponding table.

The device-status-byte is represented in each table by a two-digit hexadecimal value where the 16 possible values for each digit are shown as follows:

- *first hex digit:* Represents bit positions 0-3 (attention, status-modifier, control-unit-end, and busy, respectively);
- *second hex digit:* Represents bit positions 4-7 (channel-end, device-end, unit-check, and unit-exception, respectively).

In these tables, symbols are used to indicate whether the status is appropriate or otherwise, as follows:

• - Combination of bits is inappropriate for the device-status-byte;

A - Combination of bits is appropriate for the device-status-byte;

E - Device-level error is detected for all-zero combination.

A.1 Non-CC command response status

When status is presented in response to a command which is not the result of command-chaining (CC), the following status combinations are considered inappropriate:

a) The status-byte contains all zeros;

b) The status-byte contains the device-end bit set to one and both the channel-end and busy bits set to zeros;

c) The status-byte contains any combination of control-unit-end, status-modifier, and attention bits set to ones, and no other status bit is set to one;

d) The status-byte contains the control-unit-end bit set to one, the busy, channel-end, and device-end bits set to zeros, and any combination of unit-check and unit-exception bits set to ones.

Table A.1 shows appropriate and inappropriate status combinations for status in response to a command when not command-chaining.

Table A.1 – Non-CC command response status

First Hex Digit	Second Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0_	E	A	A	A	•	•	•	•	A	A	A	A	A	A	A	A
1_	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
2_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
3_	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
4_	•	A	A	A	•	•	•	•	A	A	A	A	A	A	A	A
5_	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
6_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
7_	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
8_	•	A	A	A	•	•	•	•	A	A	A	A	A	A	A	A
9_	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
A_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
B_	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
C_	•	A	A	A	•	•	•	•	A	A	A	A	A	A	A	A
D_	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
E_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
F_	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A

A.2 CC command response status

When status is presented in response to a command frame which is a result of command-chaining (CH bit set to one), the following status combinations are considered inappropriate:

a) The status-byte contains all zeros;

b) The status-byte contains the device-end bit set to one and both the channel-end and busy bits set to zeros;

c) The status-byte contains any combination of control-unit-end, status-modifier, and attention bits set to ones, and no other status bit is set to one;

d) The status-byte contains the control-unit-end bit set to one, the busy, channel-end, and device-end bits set to zeros, and any combination of unit-check and unit-exception bits set to ones;

e) The status-byte contains the busy bit set to one, except when only the busy and device-end bits are set to ones or when only the busy and attention bits are set to ones;

f) The status-byte contains the status-modifier bit set to one, any combination of unit-check and unit-exception bits set to ones, and all other status bits set to zeros.

Table A.1 shows appropriate and inappropriate status combinations for status in response to a command when command-chaining.

Table A.2 – CC command response status

First Hex Digit	Second Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0_	E	A	A	A	•	•	•	•	A	A	A	A	A	A	A	A
1_	•	•	•	•	A	•	•	•	•	•	•	•	•	•	•	•
2_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
3_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
4_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
5_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
6_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
7_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
8_	•	A	A	A	•	•	•	•	A	A	A	A	A	A	A	A
9_	A	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
A_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
B_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
C_	•	A	A	A	•	•	•	•	A	A	A	A	A	A	A	A
D_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
E_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
F_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

A.3 Non-immediate command accept status

When the first status is transferred after a command is accepted as a nonimmediate operation, (provided there is no intervening cancel, selective-reset, or system-reset on the same logical path), the following status combinations are considered inappropriate:

a) The status-byte contains the channel-end bit set to zero;

b) The status-byte contains the busy bit set to one.

Table A.1 shows appropriate and inappropriate status combinations for the first status transferred after a command is accepted as a non-immediate operation.

Table A.3 – Non-immediate command accept status

First Hex Digit	Second Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0_	E	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
1_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
2_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
3_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
4_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
5_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
6_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
7_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
8_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
9_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
A_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
B_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
C_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
D_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
E_	•	•	•	•	•	•	•	•	A	A	A	A	A	A	A	A
F_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

A.4 First status after CE without DE

Table A.1 shows appropriate and inappropriate status combinations for the first status after channel-end (CE) without device-end (DE).

For the first status that is presented using a status frame having the AS bit set to one after the channel accepts a status-byte with any of the following conditions:

- only the channel-end (CE) bit set to one and command-chaining was indicated;
- only the channel-end, unit-check, and status-modifier bits set to ones when command retry was requested and command-chaining was indicated;
- the channel-end bit set to one and the device-end (DE) bit set to zero and command-chaining was not indicated (provided the channel did not send an intervening selective-reset, system-reset, or command).

The following status combinations are considered inappropriate:

a) The status-byte contains the busy bit set to one;

b) The status-byte contains the channel-end bit set to one

c) The status-byte does not contain either the device-end bit set to one or the combination of control-unit-end and unit-check bits set to ones.

Table A.4 – First status after CE without DE

First Hex Digit	Second Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0_	E	•	•	•	A	A	A	A	•	•	•	•	•	•	•	•
1_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
2_	•	•	A	A	A	A	A	A	•	•	•	•	•	•	•	•
3_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
4_	•	•	•	•	A	A	A	A	•	•	•	•	•	•	•	•
5_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
6_	•	•	A	A	A	A	A	A	•	•	•	•	•	•	•	•
7_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
8_	•	•	•	•	A	A	A	A	•	•	•	•	•	•	•	•
9_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
A_	•	•	A	A	A	A	A	A	•	•	•	•	•	•	•	•
B_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
C_	•	•	•	•	A	A	A	A	•	•	•	•	•	•	•	•
D_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
E_	•	•	A	A	A	A	A	A	•	•	•	•	•	•	•	•
F_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

A.5 Selective-reset UC request status

When status is presented in a status frame sent in response to a selective-reset frame containing a request for unit check (RU=1 or RO=1), the following status combinations are considered inappropriate, except under the condition in which the status response indicates control unit busy; that is, the status frame contains the AS bit set to zero and only the status-modifier and the busy bits set to ones in the status-byte of the status frame:

a) The status-byte contains the unit-check bit set to zero;

b) The status-byte contains the busy bit set to one.

Table A.1 shows appropriate and inappropriate status combinations for the status in response to a selective-reset frame containing a request for unit check.

Table A.5 – Selective-reset UC request status

First Hex Digit	Second Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0_	E	•	A	A	•	•	A	A	•	•	A	A	•	•	A	A
1_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
2_	•	•	A	A	•	•	A	A	•	•	A	A	•	•	A	A
3_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
4_	•	•	A	A	•	•	A	A	•	•	A	A	•	•	A	A
5_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
6_	•	•	A	A	•	•	A	A	•	•	A	A	•	•	A	A
7_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
8_	•	•	A	A	•	•	A	A	•	•	A	A	•	•	A	A
9_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
A_	•	•	A	A	•	•	A	A	•	•	A	A	•	•	A	A
B_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
C_	•	•	A	A	•	•	A	A	•	•	A	A	•	•	A	A
D_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
E_	•	•	A	A	•	•	A	A	•	•	A	A	•	•	A	A
F_	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Annex B
(informative)

Self-description model supplementary information

The material in this annex is provided to assist the reader in understanding the concepts of device self-description and does not constitute a precise definition of the self-description architecture. See clause 16 for self-description architecture.

B.1 Self-description model

Self-description is the ability of components of a system to provide unique identification information about themselves dynamically on request. Information about characteristics is also provided. The points at which products communicate externally are identified, and subassemblies of a product may also be identified. Identifiers associated with a product are decided upon during product design. The abstract model described in this section is intended to illustrate the capabilities of the architecture that permit programming to obtain identification information.

Figure B.1 illustrates the structural objects of the self-description architecture that allow the program to obtain identification information. The relationships possible among *self-describing components* (SDCs) are typified by an SDC (A), its *associated* SDC (B), and a *neighbor* SDC. Connections from one SDC to another SDC exist between *communication points* (CPs) on the SDC. *Addressable elements* (AEs), and *identifiable subassemblies* (ISAs) are indicated, as is a command source with predetermined information about the SDC.

A channel subsystem, or a channel program, typifies a *command source*. Device addresses or, more generally, the addresses of addressable elements in the model, are typically predefined for use by a program in an I/O instruction which designates the target of an I/O operation. A ULP instruction designating a device and also designating a channel program containing a sense-ID command is a usual means by which a program commences extracting identification information from an SDC. For some SDCs, the code points for the channel commands by which identifying information is requested are themselves provided dynamically in response to a sense-ID command.

The request for information can take one of several forms:

a) In the first form, the communication point at which the request is directed is not explicitly named, implying that the response is to contain information about the connected communication point. Some of the information is provided in a structure called a node descriptor (ND). In addition, at least one node qualifier (NQ) must be provided. Further, sufficient NQs must also be provided to identify all of the identifiable communication points of the SDC and optionally any associated SDCs. The ND and NQs are blocked in a single record;

b) In the second form, a communication point is explicitly designated. In this case, NQs are not provided, but the ND of either the designated communications point or of its neighbor may be requested;

c) On SBCON interfaces, link-level means are defined for retrieving identification information. Information is provided only in the form of NDs and without requiring an explicit designation of the target point, constituting a third form for acquiring identification information. As a case of particular interest, a channel subsystem is an instance of an SDC with respect to link-level protocols, and identification information can be solicited from a channel subsystem by a connected device and vice versa.

An addressable element of an SDC is a subassembly of the SDC which must provide separate identification information in the form of a node-element descriptor (NED) and possibly one or more node-element qualifiers (NEQ). There may also be other related identifiable subassemblies. Identifiable subassemblies differ from addressable elements in that they are not addressable by the command source. (A vendor may choose to so identify a significant

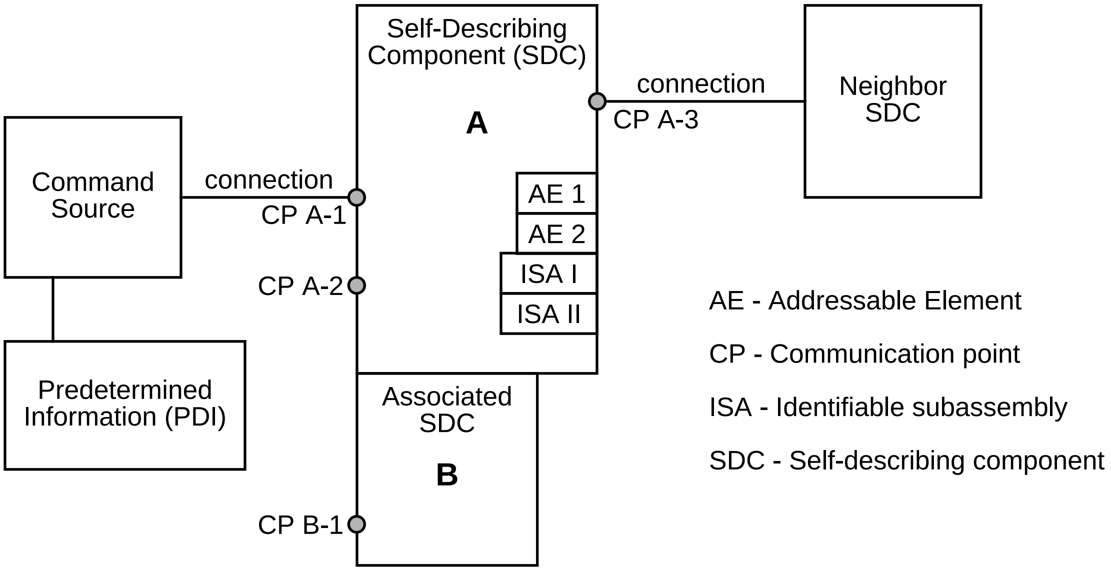


Figure B.1 – Self-description model

field-replaceable unit (FRU) that is not itself an addressable element.) Whether there are identifiable subassemblies is a model-dependent design choice; if there are, information about them is included when information is provided about an addressable element. Information about identifiable subassemblies is also provided in the form of an NED, possibly accompanied by one or more NEQs. A record may be formed of one or more NEDs, each with its related NEQs, along with a token NED and a final general NEQ. The same token NED may appear in more than one record to identify a model-dependent configuration association. A final general NEQ contains information not specific to a single subassembly.

Not all SDCs contain addressable elements. In these cases, communication is restricted to the use of link-level protocols or certain special-purpose facilities.

All of the structures, NDs, NQs, NEDs and NEQs, provide information that may be used by querying algorithms to exhaustively obtain the information available from an SDC, including sufficient information to parse the records.

The names of the information-supplying structures begin with the term *node*. In the context of an SDC, a node comprises the subassemblies identified by the NED and NEQ information obtained by commands entering an SDC through a specific communication point.

The name of the communication point to which the command source is attached consists of two parts. The first part is the name of the SDC; the second part identifies the particular communication point within the SDC. ND and NQ information may identify communication points with different first-part names. These identify associated SDCs. NED and NEQ information cannot be obtained through a communication point of an SDC about an associated SDC.

A command entering through one communication point may query another communication point about what the second communication point is connected to. If a connection exists, the second communication point is connected to a neighbor SDC. The second communication point returns information about its neighbor in the form of an ND.

B.2 Sample I/O-configuration elements

Figure B.2 illustrates a disk subsystem, storage control SC1, that is connected to two channel subsystems, CS1 and CS2. The storage control has two control units, CU1 and CU2, each providing two SBCON interfaces, so that there are four SBCON interfaces (A, B, C, and D) to the storage control. The disk subsystem also contains two strings of disk drives, String1 and String2, with four disks in each string. The disks in String1 have addresses 0, 1, 2, and 3; those in String2 have addresses 8, 9, A, and B. Each

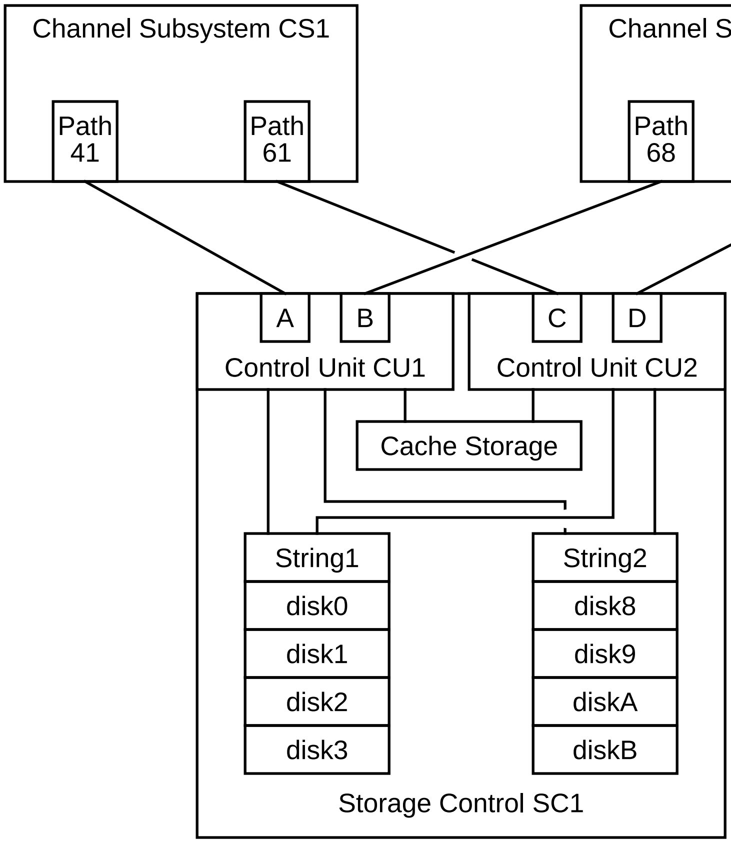


Figure B.2 – Sample portion of an I/O configuration

control unit has one path to each disk string. Thus, each channel path can access each disk drive. The storage control also has storage that can be used as a cache for data on the disk drives. The storage control can provide self-description information.

From channel subsystem CS1, channel paths 41 and 61 are attached to interfaces A and C; from channel subsystem CS2, channel paths 68 and 88 are attached to interfaces B and D. Thus each channel subsystem has two paths to each disk drive in the disk subsystem.

B.2.1 Mapping to the model

In terms of figure B.1, the self-description model, SC1 is an SDC with four communications points, the four channel interfaces, which define four nodes. SC1 also has eight addressable elements (the disk drives) and five identifiable subassemblies (the control units, the strings, and the cache storage).

Each channel subsystem is a command source for itself and can also provide commands from its associated processors. Considering one channel subsystem as the command source, the other channel subsystem is a neighbor node of the storage control. Because each control unit can access each disk, no associated SDC appears in the sample.

B.2.2 Sample nodes

When the channel subsystems initialize their channel paths, they acquire the node descriptors of the attached control units. In this case, the command source is the interface protocol, and the predetermined information is the channel path. This process is described in 10.1. Thus CS1 learns that path 41 is attached to node SC1.A and that path 61 is attached to node SC1.C; similarly, CS2 learns that paths 68 and 88 lead to nodes SC1.B and SC1.D, respectively. (In the above, the command source of the model is the interface protocol, and the predetermined information is the type and number of the channel path.) Also as part of the process of initializing the channel paths, SC1

Table B.1 – Results of RNI command

Interface Identifier	Node Selector		
	0	1	2
0	ND for SC1.A; NQ for A, B, C, and D	CE+DE+UC	CE+DE+UC
A	ND for SC1.A; NQ for A, B, C, and D	ND for SC1.A	ND for CS1.41
B	ND for SC1.A; NQ for A, B, C, and D	ND for SC1.B	ND for CS2.68
C	ND for SC1.A; NQ for A, B, C, and D	ND for SC1.C	ND for CS1.61
D	ND for SC1.A; NQ for A, B, C, and D	ND for SC1.D	ND for CS2.88

learns that A is connected to CS1.41, B to CS2.68, C to CS1.61, and D to CS2.88.

A channel program comprising a set-interface-ID command chained to a read-node-identifier (RNI) command initiated for disk 0 along path 41 with a node selector containing a value of 0 would result in the device returning a node descriptor for SC1.A followed by a node qualifier identifying interfaces A, B, C, and D. Table B.1 summarizes the results of read-node-identifier commands directed to device 0 with various node selectors established. Note that the channel program is terminated prematurely when the value of the node selector is 1 or 2 and the device does not recognize the interface identifier (interface identifier 0, in this example).

B.2.3 Sample node elements

Figure B.3 shows the structure of the configuration record provided when disk 1 is accessed from channel-interface B. The configuration records for disks 0, 2, and 3 are similar, except for the I/O-device NED, which identifies the specific disk.

Figure B.4 shows the structure of the configuration record provided when disk 9 is accessed from channel-interface A. The configuration records for disks 8, A, and B are similar, except for the I/O-device NED, which identifies the specific disk.

Specific node-element qualifiers (NEQs) are omitted from the illustrations, although they may exist following any non-token NED. The placement of the unused space in the sample configuration records is not significant; in general, the program must examine each 32-byte field within the record to determine whether it contains an NED, an NEQ, or is unused.

Word	
0	I/O Device NED SC1.Disk1
8	NED for String1 SC1.String1
16	NED for the Cache SC1.Cache
24	unused space
32	Control Unit NED SC1.CU1
40	Token NED SC1.CU1
48	unused space
56 64	General NEQ

Figure B.3 – Configuration record for disk1

B.2.4 Program use of self-description data

In an ESA/390 environment, the program has access to I/O devices through the channel subsystem, which provides a subchannel for each I/O device in the configuration. The channel subsystem itself needs to be informed of the

Word	
0	I/O Device NED SC1.Disk9
8	NED for String2 SC1.String2
16	NED for the Cache SC1.Cache
24	unused space
32	Control Unit NED SC1.CU1
40	Token NED SC1.CU1
48	unused space
56 64	General NEQ

Figure B.4 – Configuration record for disk9

control units, I/O devices, and addresses used on each channel path, as well as the device numbers to be assigned to each I/O device. (An I/O-configuration program (IOCP) is one means typically used to provide this information to the channel subsystem.)

In like manner, control programs such as OS/VS2 (MVS) and VM/ESA have used similar input to generate the control blocks they use to manage I/O devices and operations. The MVS control block is called a *unit control block (UCB)*; the rest of this discussion uses UCB for the control block used by the program to represent a device. Part of the initialization procedures for control programs has been verification that the device accessible through a subchannel is one that was defined to the control program, and that each channel path available in a given subchannel leads to the same device. Such a procedure might include:

a) For each subchannel with a valid device number, locate the UCB with the same device number;

b) Use a sense-ID command to verify that the class and type of the device match those specified to the control program for that device number;

c) If the device has a *volume label*, read the label and save it for later use;

d) If the device supports path groups, establish path-group information at the device;

e) Repeat for each path available in the subchannel.

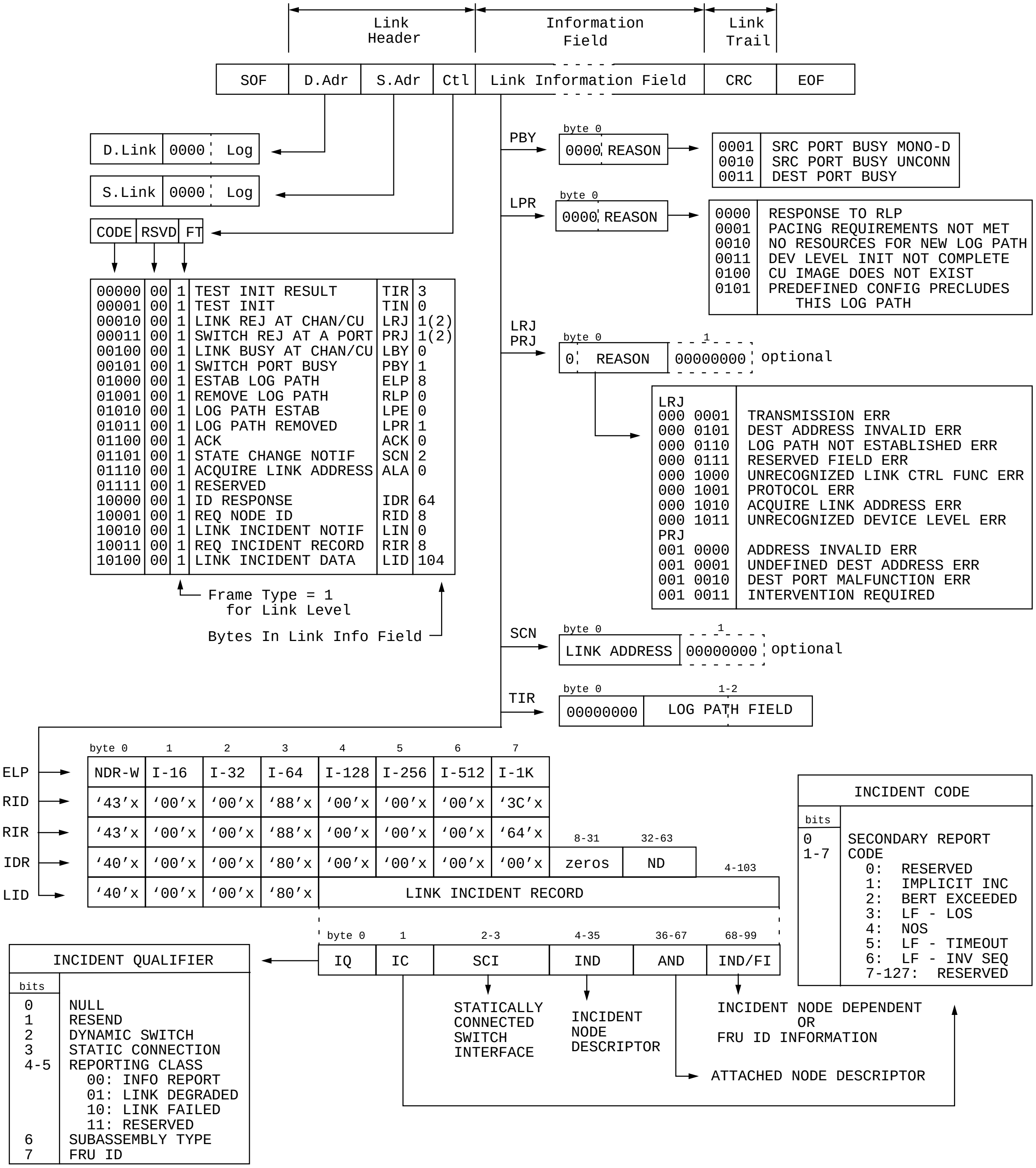
The presence of the world-wide unique identifier in the NED allows the control program to ensure that the same device is accessed on each available path in the subchannel. Furthermore, the control program can ensure that a device reporting itself as newly available is the same one that was previously accessed by the subchannel, by comparing the new node-element ID with the one saved at initialization.

Annex C

(informative)

Frame formats

Link Frame Format



00000	00	1	TEST INIT RESULT	TIR	3
00001	00	1	TEST INIT	TIN	0
00010	00	1	LINK REJ AT CHAN/CU	LRJ	1(2)
00011	00	1	SWITCH REJ AT A PORT	PRJ	1(2)
00100	00	1	LINK BUSY AT CHAN/CU	LBY	0
00101	00	1	SWITCH PORT BUSY	PBY	1
01000	00	1	ESTAB LOG PATH	ELP	8
01001	00	1	REMOVE LOG PATH	RLP	0
01010	00	1	LOG PATH ESTAB	LPE	0
01011	00	1	LOG PATH REMOVED	LPR	1
01100	00	1	ACK	ACK	0
01101	00	1	STATE CHANGE NOTIF	SCN	2
01110	00	1	ACQUIRE LINK ADDRESS	ALA	0
01111	00	1	RESERVED		
10000	00	1	ID RESPONSE	IDR	64
10001	00	1	REQ NODE ID	RID	8
10010	00	1	LINK INCIDENT NOTIF	LIN	0
10011	00	1	REQ INCIDENT RECORD	RIR	8
10100	00	1	LINK INCIDENT DATA	LID	104

0001	SRC PORT BUSY MONO-D
0010	SRC PORT BUSY UNCONN
0011	DEST PORT BUSY

0000	RESPONSE TO RLP
0001	PACING REQUIREMENTS NOT MET
0010	NO RESOURCES FOR NEW LOG PATH
0011	DEV LEVEL INIT NOT COMPLETE
0100	CU IMAGE DOES NOT EXIST
0101	PREDEFINED CONFIG PRECLUDES THIS LOG PATH

LRJ	
000 0001	TRANSMISSION ERR
000 0101	DEST ADDRESS INVALID ERR
000 0110	LOG PATH NOT ESTABLISHED ERR
000 0111	RESERVED FIELD ERR
000 1000	UNRECOGNIZED LINK CTRL FUNC ERR
000 1001	PROTOCOL ERR
000 1010	ACQUIRE LINK ADDRESS ERR
000 1011	UNRECOGNIZED DEVICE LEVEL ERR
PRJ	
001 0000	ADDRESS INVALID ERR
001 0001	UNDEFINED DEST ADDRESS ERR
001 0010	DEST PORT MALFUNCTION ERR
001 0011	INTERVENTION REQUIRED

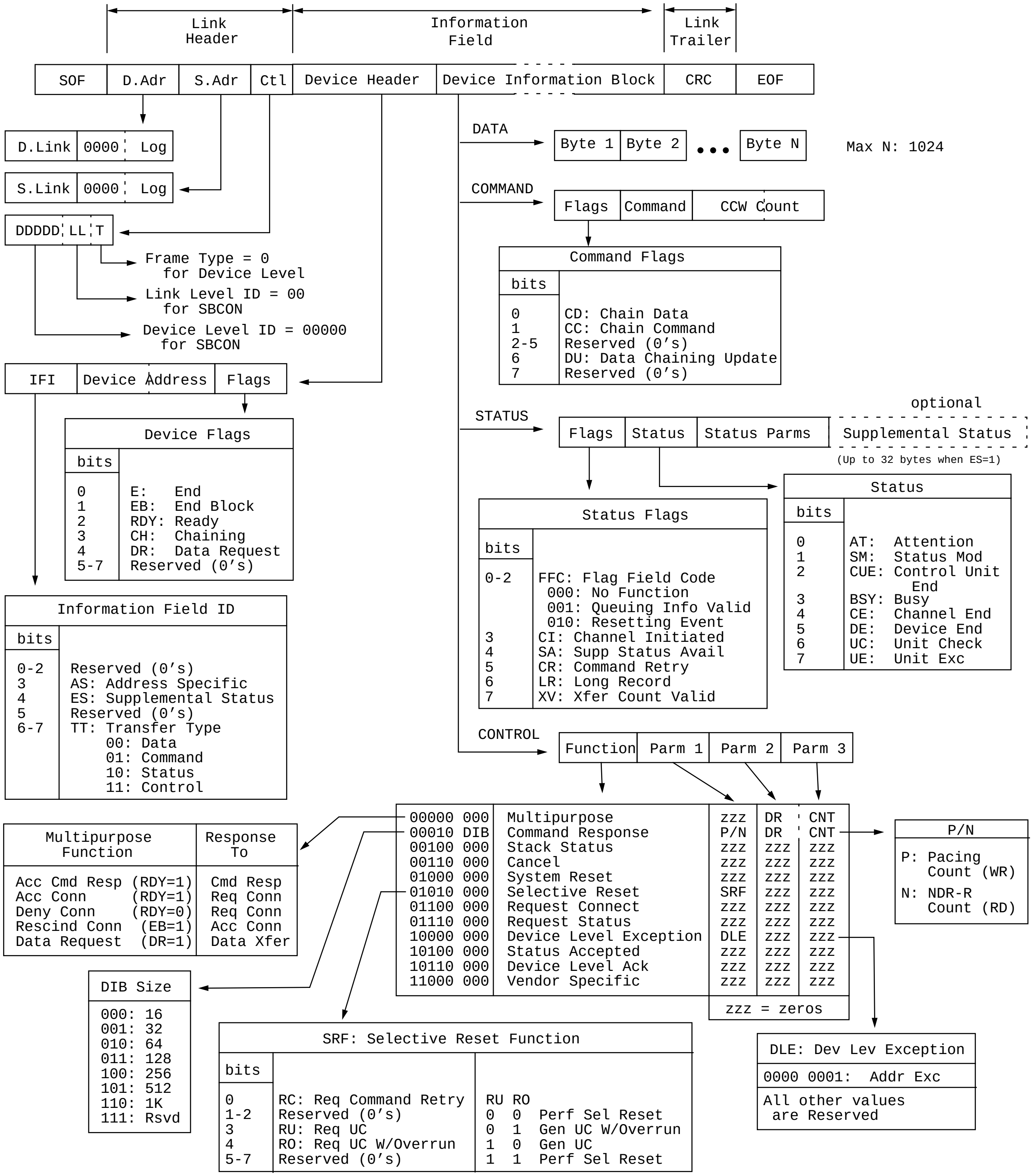
	byte 0	1	2	3	4	5	6	7	8-31	32-63
ELP	NDR-W	I-16	I-32	I-64	I-128	I-256	I-512	I-1K		
RID	‘43’x	‘00’x	‘00’x	‘88’x	‘00’x	‘00’x	‘00’x	‘3C’x		
RIR	‘43’x	‘00’x	‘00’x	‘88’x	‘00’x	‘00’x	‘00’x	‘64’x		
IDR	‘40’x	‘00’x	‘00’x	‘80’x	‘00’x	‘00’x	‘00’x	‘00’x	zeros	ND
LID	‘40’x	‘00’x	‘00’x	‘80’x	LINK INCIDENT RECORD (4-103)					

byte 0	1	2-3	4-35	36-67	68-99
IQ	IC	SCI	IND	AND	IND/FI

INCIDENT CODE	
bits	
0	SECONDARY REPORT
1-7	CODE 0: RESERVED 1: IMPLICIT INC 2: BERT EXCEEDED 3: LF - LOS 4: NOS 5: LF - TIMEOUT 6: LF - INV SEQ 7-127: RESERVED

INCIDENT QUALIFIER	
bits	
0	NULL
1	RESEND
2	DYNAMIC SWITCH
3	STATIC CONNECTION
4-5	REPORTING CLASS 00: INFO REPORT 01: LINK DEGRADED 10: LINK FAILED 11: RESERVED
6	SUBASSEMBLY TYPE
7	FRU ID

Device Frame Format



Command Flags

bits	
0	CD: Chain Data
1	CC: Chain Command
2-5	Reserved (0’s)
6	DU: Data Chaining Update
7	Reserved (0’s)

Device Flags

bits	
0	E: End
1	EB: End Block
2	RDY: Ready
3	CH: Chaining
4	DR: Data Request
5-7	Reserved (0’s)

Information Field ID

bits	
0-2	Reserved (0’s)
3	AS: Address Specific
4	ES: Supplemental Status
5	Reserved (0’s)
6-7	TT: Transfer Type 00: Data 01: Command 10: Status 11: Control

Status Flags

bits	
0-2	FFC: Flag Field Code 000: No Function 001: Queuing Info Valid 010: Resetting Event
3	CI: Channel Initiated
4	SA: Supp Status Avail
5	CR: Command Retry
6	LR: Long Record
7	XV: Xfer Count Valid

Status

bits	
0	AT: Attention
1	SM: Status Mod
2	CUE: Control Unit End
3	BSY: Busy
4	CE: Channel End
5	DE: Device End
6	UC: Unit Check
7	UE: Unit Exc

Multipurpose Function		Response To
Acc Cmd Resp	(RDY=1)	Cmd Resp
Acc Conn	(RDY=1)	Req Conn
Deny Conn	(RDY=0)	Req Conn
Rescind Conn	(EB=1)	Acc Conn
Data Request	(DR=1)	Data Xfer

00000 000	Multipurpose	zzz	DR	CNT
00010 DIB	Command Response	P/N	DR	CNT
00100 000	Stack Status	zzz	zzz	zzz
00110 000	Cancel	zzz	zzz	zzz
01000 000	System Reset	zzz	zzz	zzz
01010 000	Selective Reset	SRF	zzz	zzz
01100 000	Request Connect	zzz	zzz	zzz
01110 000	Request Status	zzz	zzz	zzz
10000 000	Device Level Exception	DLE	zzz	zzz
10100 000	Status Accepted	zzz	zzz	zzz
10110 000	Device Level Ack	zzz	zzz	zzz
11000 000	Vendor Specific	zzz	zzz	zzz

zzz = zeros

P/N

P: Pacing Count (WR)
N: NDR-R Count (RD)

DIB Size

000: 16
001: 32
010: 64
011: 128
100: 256
101: 512
110: 1K
111: Rsvd

SRF: Selective Reset Function

bits				
0	RC: Req Command Retry	RU	RO	
1-2	Reserved (0’s)	0	0	Perf Sel Reset
3	RU: Req UC	0	1	Gen UC W/Overrun
4	RO: Req UC W/Overrun	1	0	Gen UC
5-7	Reserved (0’s)	1	1	Perf Sel Reset

DLE: Dev Lev Exception

0000 0001: Addr Exc

All other values are Reserved

Index