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

FIBRE CHANNEL

PRIVATE LOOP

SCSI DIRECT ATTACH (FC-PLDA)

REV 1.10

X3 working draft proposed
Technical Report

February 17, 1997

Secretariat:
Information Technology Industry Council

ABSTRACT:

NOTE:
This is a draft proposed Technical Report of Accredited Standards Committee X3. As such, this is not a completed report. The X3T11 Technical Committee may modify this document as a result of comments received during public review and its approval as a technical report.

POINTS OF CONTACT:

Roger Cummings (X3T11 Chairman)
DISTRIBUTED PROCESSING TECHNOLOGY
140 Candace Drive
Maitland, FL 32751
Phone: (407) 830-5522 x348
Fax: (407) 260-5366
EMail: cummings_roger@dpt.com

Ed Grivna (X3T11 Vice Chairman)
Cypress Semiconductor
2401 East 86th Street
Bloomington, MN 55245
Phone: (612) xxx-xxxx
Fax: (612) 851-5087
E-Mail: elg@cypress.com

I. Dal Allan
(Fibre Channel Working Group Chairman)
ENDL
14426 Black Walnut Court
Saratoga, CA 95070
(408) 867-6630 Fax: (408) 867-2115
E-Mail: dal.allan@mcimail.com

Robert W. Kembel (Technical Editor)
Connectivity Solutions
3061 N. Willow Creek Drive
Tucson, AZ 85712
(520-881-0877) Fax: (520-881-0632)
E-mail: 73040.1376@compuserve.com

Summary of Changes, revision 1.10

- Definitions: removed definition of loop tenancy, clarified definition of loop circuit
- Table 2: changed the Relative Offset by Information Category to reflect the fact that this is a 16-bit field
- Table 5: Changed ABTS for SCSI initiator from ‘R’ to ‘I’ since the initiator is nor required to use this function
- Table 6: Changed ADISC support by a target to ‘R’ since the target doesn’t know whether an initiator will use ADISC or PDISC. This is consistent with note 1
- 5.8.1 and 5.8.2: Changed last bullet of each to LS_RJT from P_RJT to correct typographical error
- 5.8.3.2: Revised wording in several places to try and capture the proper wording for Sequence management
- Table 8: Rearranged last entry to clarify the wording for readability
- Table 8, Note 5: Added the qualifiers “Monitoring or Arbitrating States”
- 6.2.2: Clarified wording on item d.1
- 6.2.2: Removed erroneous initial phrase of item d.3 “As an alternative to items 3 and 4,”
- 6.2.2: Changed item e.2 from ‘transmit from one to Login_BB_Credit’ to “transmit up to Login_BB_Credit”
- 6.3: Removed phrase “or does not have any open Exchanges that would be affected by an AL_PA change”. This was added in rev. 1.9 as the result of a letter ballot comment but removed due to objections.
- 6.3.3: Removed last paragraph. This was added in rev. 1.9 as the result of a letter ballot comment but removed due to objections.
- 7: Added Table 9 back in. Added AL_TIME to timers, added R_T_TOV back in (it was removed in 1.8)
- 7.6: Removed third paragraph as it was redundant with the first paragraph of this clause
- Table 10: Changed Write XFER_RDY Disabled = 1 entries back to ‘P’ for both the Initiator and Target. This was changed in rev. 1.8 as the result of a letter ballot comment but removed due to objections.
- 8.2.1: Changed third paragraph from “For device types which” to “For SCSI commands which”
- Table 14: Changed SCSI initiator column such that initiators for stream devices are now ‘R’ to support untagged command queuing
- Table 16: Added a ‘-’ to those entries which are not meaningful for the listed action.

ANSI ®
TR X3.xxx-199x

draft proposed X3 Technical Report

Fibre Channel —
Private Loop SCSI Direct Attach (FC-PLDA)

Secretariat

Information Technology Industry Council

Approved ,199

American National Standards Institute, Inc.

Abstract

This profile selects and restricts logical options from the Fibre Channel Physical and Signalling, Fibre Channel Protocol for SCSI, Fibre Channel Arbitrated Loop, and Small Computer Systems Interface standards, such that any device complying with the profile should interoperate.

X3's Technical Report Series

This Technical Report is one in a series produced by the American National Standards Committee, X3, Information Technology. The secretariat for X3 is held by the Computer and Business Equipment Manufacturers Association (CBEMA), 1250 Eye Street, NW Suite 200, Washington DC 20005.

As a by-product of the standards development process and the resources of knowledge devoted to it, X3 from time to time produces Technical Reports. Such Technical Reports are not standards, nor are they intended to be used as such.

X3 Technical Reports are produced in some cases to disseminate the technical and logical concepts reflected in standards already published or under development. In other cases, they derive from studies in areas where it is found premature to develop a standard due to a still changing technology, or inappropriate to develop a rigorous standard due to the existence of a number of viable options, the choice of which depends on the user's particular requirements. These Technical Reports, thus, provide guidelines, the use of which can result in greater consistency and coherence of information processing systems.

When the draft Technical Report is completed, the Technical Committee approval process is the same as for a draft standard. Processing by X3 is also similar to that for a draft standard.

PATENT STATEMENT

CAUTION: The developers of this Technical Report have requested that holder's of patents that may be required for the implementation of the Technical Report, 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 Technical Report.

As of the date of publication of this Technical Report and following calls for the identification of patents that may be required for the implementation of the Technical Report, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any Technical Report it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this Technical Report.

Published by

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

Printed in the United States of America

Contents

Page

Annexes

Figures

Page

Tables

draft proposed X3 Technical Report for Information Technology—

Fibre Channel — Private Loop SCSI Direct Attach (FC-PLDA)

1 Introduction and scope

This Technical Report specifies Fibre Channel, SCSI-3 Fibre Channel Protocol for SCSI (FCP), and SCSI-3 command set options required for communication with private NL_Ports on SCSI initiators and targets. It is intended to serve as a guide whose primary objective is to maximize the likelihood of interoperability between conforming implementations. This guide prohibits and requires features which are optional as well as prohibiting the use of some non-optional features in the referenced ANSI standards.

A second objective of this guide is to simplify implementations and their associated documentation, testing, and support requirements. This means that there will be some optional features which are not mutually exclusive, but are still prohibited or required solely for the purpose of this simplification.

This guide is based on SCSI-3 command sets mapped to FCP using FC-AL. Internal characteristics of conformant implementations are not defined by this document. This document incorporates features from the standards identified in clause 2. Where needed, changes have been proposed to the appropriate ANSI X3 standards to ensure this document remains a strict subset of ANSI standards.

2 Normative references

The following standards contain provisions which, through reference in the text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this document 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), 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).

Additional availability contact information is provided below as needed.

2.1 Approved references

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

[2] ANSI X3.269-1996, *Fibre Channel Protocol for SCSI (FCP)*

[3] ANSI X3.272-1996, *Fibre Channel Arbitrated Loop (FC-AL)*

[4] ANSI X3.270-1996, *SCSI-3 Architecture Model (SAM)*

2.2 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of the document, or regarding availability, contact the relevant standards body or other organization as indicated.

[5] ANSI X3.xxx-199x, *SCSI-3 Primary Commands (SPC),* X3T10/995D/Rev 9b

[6] ANSI X3.xxx-199x, *SCSI-3 Block Commands (SBC)*, X3T10/996D/Rev 4

[7] ANSI X3.xxx-199x, *Fibre Channel - Physical and Signalling Interface-2 (FC-PH-2)*, X3T11/Project 901D/Rev 7.3

[8] ANSI X3.xxx-199x, *SCSI-3 Controller Commands (SCC)*, X3T10/1047D Rev 6a

[9] ANSI X3.xxx-199x, *Fibre Channel - Physical and Signalling Interface-3 (FC-PH-3)*, X3T11/Project 1119D/Rev 8.8

[10] ANSI X3.xxx-199x, *SCSI-3 Stream Device Command Set (SSC)*, X3T10/Project 997D/Rev 7

[11]ANSI X3.xxx-199x, *SCSI-3 Media Changer Command Set (SMC)*, X3T10/Project 999D/Rev 5

2.3 Other references

[12] *FCSI Common FC-PH Feature Sets Used in Multiple Profiles*, Rev 3.1

[13] *FCSI SCSI Profile*, Rev 2.2

[14] *SFF-8045 Specification for 40-pin SCA-2 Connector w/Parallel Selection,* Rev 3.3

3 Definitions and conventions

The following definitions, conventions, abbreviations, acronyms, and symbols apply to this document.

3.1 Definitions

3.1.1 Available BB_Credit: A variable used by a sequence initiator or responder to determine permission to transmit Class 3 frames. The transmitter may transmit one of the listed frames when Available BB_Credit is greater than 0. The rules for modifying Available BB_Credit are defined by FC-AL and summarized in 6.2.

3.1.2 Available_receive_buffers: The number of buffers in a receiving port which are available for receiving frames at link rate. Equal to the largest number of R_RDYs an NL_port can issue immediately upon transmission or receipt of an OPN. BB Flow control rules keep this number greater than or equal to the Available BB_Credit variable in a transmitting port. The number of available receive buffers must be greater than or equal to Login_BB_Credit upon receipt of any OPN.

3.1.3 Byte: A group of eight bits.

3.1.4 Hard Address: The AL_PA which an NL_Port attempts to acquire in the LIHA Loop Initialization Sequence.

3.1.5 Login_BB_Credit: On FC-AL, equal to the number of receive buffers that a receiving NL_port must have available when a loop circuit is established. Login_BB_Credit is established using PLOGI and may be discovered by using the PDISC Extended Link Service.

3.1.6 Loop Circuit: A bidirectional path that allows communication between two L_Ports.

3.1.7 Loop_ID: Loop_IDs are 7-bit values numbered contiguously from 0 to 126 decimal and represent the 127 valid AL_PAs on a loop. Loop_IDs correspond to the 7-bit SEL word in SFF-8045

[3] used for designating a Hard Addresses. See FC-AL, Annex K (Assigned Loop Identifier) for a complete mapping.

3.1.8 Node: An entity containing one or more N*_Ports controlled by a level above FC-2.

3.1.9 OPN Originator: The L_Port on an Arbitrated Loop that enters the OPEN state.

3.1.10 OPN Recipient: The L_Port on an Arbitrated Loop that enters the OPENED state.

3.1.11 Previously Acquired Address: During loop initialization, this is the AL_PA value which an L_Port attempts to acquire during the LIPA sequence.

3.1.12 Private loop device: A device with only Private NL_Ports.

3.1.13 Public loop device: A device with at least one Public NL_Port.

3.1.14 Private NL_Port: An NL_Port which is observing the rules of private loop behavior (see 4).

3.1.15 Public NL_Port: An NL_Port which attempts a fabric login and is permitted to open AL_PA='00'h. A Public NL_Port can observe the rules of either public or private loop behavior (see 4).

3.1.16 ULP process: A function executing within an FC node which conforms to Upper Level Protocol (ULP) defined protocols when interacting with ULP processes residing in other FC nodes.

3.2 Editorial conventions

A number of conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the following conventions:

– the first letter of each word in uppercase and the rest lowercase (e.g., Exchange, Class, etc.).

– a term consisting of multiple words, with the first letter of each word in uppercase and the rest lowercase, and each word separated from the other by an underscore (_) character. A word may consist of an acronym or abbreviation which would be printed in uppercase. (e.g., NL_Port, Transfer_Length, etc.).

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

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

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

In all of the figures, tables, and text, 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 set as required by the appropriate standard.

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

In several tables within this document, there is a column on the right side of the table labelled “Notes”. These notes are Normative and shall be considered requirements of this document.

In the event of conflicts between the text, tables, and figures in this document, the following precedence shall be used: text, tables, figures.

3.2.1 Binary notation

Binary notation may be used to represent some fields. Single bit fields are represented using the binary values 0 and 1. For multiple bit fields, the binary value is enclosed in single quotation marks followed by the letter b. For example, a four-byte Process_Associator field containing a binary value may be represented as ‘00000000 11111111 10011000 11111010’b.

3.2.2 Hexadecimal notation

Hexadecimal notation may be used to represent some fields. When this is done, the value is enclosed in single quotation marks preceded by the word hex. For example, a four-byte Process_Associator field containing a binary value of ‘00000000 11111111 10011000 11111010’b is shown in hexadecimal format as hex’00 FF 98 FA’.

3.3 Abbreviations, acronyms, and symbols

Abbreviations and acronyms applicable to this International Standard are listed. Definitions of several of these items are also found in 3.1.

3.3.1 Acronyms and abbreviations

ELS	Extended Link Service
FC-PH	ANSI X3.230-1994*, Fibre Channel Physical and Signaling Interface (FC-PH)*
FCP	Fibre Channel Protocol
IP	Internet Protocol
IU	Information Unit
FRU	Field Replaceable Unit
LAN	Local Area Network
LLC	Logical Link Control
MAC	Media Access Control
NFS	Network File System or Network File Server
SCSI	Small Computer System Interface
SFF	Small Form Factor
ULP	Upper Level Protocol
WAN	Wide Area Network
WWN	World Wide Name

3.3.2 Symbols

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

||	concatenation

3.4 Applicability and use of this document

This document specifies which features shall be used (“required”) and which features shall not be used (“prohibited”) by interoperating compliant Fibre Channel implementations. Use of some features is optional (“allowed”); these features may be used but compliant implementations are not required to do so. The only features or functions required or prohibited by this document are those which have been determined to affect interoperability.

The relationship between use (specified in this document) and support (implemented by an adapter designer) is subtle. If this document specifies that a feature must be used, then a compliant adapter must support it. In some cases, the specification is asymmetric: to ensure interoperability when an optional feature is used, this document mandates support for the infrastructure required to use the feature without specifying that the feature actually be used.

The requirements of this document are a proper subset of the various relevant standards. They prohibit use of many features and options in these standards; and interoperability is not guaranteed if these features are used. This document does not prohibit implementation of features, only their use.

Functions which are mandatory in the appropriate base standard are assumed to be implemented. Implementations may support features whose use is prohibited by this document and such prohibited features may be required for compliance with the relevant standards or other Profiles or applications not covered by this document.

3.5 Definitions and abbreviations used in Feature Set tables

Since the nature of this document is a profile, the usual definitions of the following words may not apply! Please read these definitions carefully!

Prohibited: If a feature is Prohibited, it means that it shall not be used between compliant implementations. An implementation may use the feature to communicate with non-compliant implementations. This document does not prohibit the implementation of features, only their use between compliant implementations. However, interoperability is not guaranteed if Prohibited features are used when communicating with devices conforming to this profile.

Required: If a feature or parameter value is Required, it means that it shall be used between compliant implementations. Compliant implementations are required to implement the feature. An implementation may use the feature or other features to communicate with non-compliant implementations. Interoperability with devices conforming to this profile is not guaranteed if Required features are not implemented.

Allowed: If a feature or parameter value is Allowed, it means that it may be used between compliant implementations. Compliant implementations are not required to implement the feature, but if they do, the feature shall be implemented as described in this document. Typically, the potential user of a feature may determine if the potential recipient supports that feature via a Required discovery process.

Invokable: If a feature or parameter value is Invokable, it means that it may be used between compliant implementations. Compliant implementations are required to implement the feature, and make available the use of the feature. Invokable is different than Allowable or Required in that an originator may invoke the feature if needed, but the originator is not required to invoke it, and may never need to. Typically, an Invokable feature is Required for implementation by the recipient of the feature.

Features in this document are summarized in the form of Feature Set tables. These tables indicate whether the feature is Required, Prohibited, Invokable, or Allowed for compliance with this report; or whether a parameter is Required to be a particular value for compliance with this document. Features or parameters which are not listed do not affect interoperability of Private Loop devices.

The following legend is used for table entries in these clauses:

`P' the implementation is Prohibited from using the specified feature when communicating with a PLDA-compliant implementation.

`R' the implementation is Required to support the specified feature when communicating with a PLDA-compliant implementation.

`A' use of the specified feature is Allowed when communicating with a PLDA-compliant implementation.

`I' the implementation may Invoke the specified feature when communicating with a PLDA-compliant implementation.

`n' the parameter shall be set to this value when communicating with a PLDA-compliant implementation.

`X' this parameter has no required value; any value is allowed when communicating with a PLDA-compliant implementation.

`-' or ‘n/a’this parameter or feature is not meaningful

A blank entry in a Feature Set table indicates that the feature is not part of that Feature Set.

For many features, explanatory text is provided in the form of notes following each Feature Set table. The “Notes” column in each Table contains the reference number of the note applying to an individual feature or group of features.

4 Private versus Public Port behavior

An NL_Port which attempts a Fabric Login (FLOGI) is called a Public NL_Port. An NL_Port which does not attempt a Fabric Login is called a Private NL_Port. Public NL_Ports whose Fabric Login fails revert to Private NL_Port behavior. Public NL_Ports may communicate with other Public NL_Ports, however, if a Public NL_Port wishes to communicate with a PLDA compliant Private NL_Port it must use the behavior described in this document. For example, a Public NL_Port in Figure 1 may be a Network File System (NFS) server which communicates with NFS Clients residing directly on the fabric using Internet Protocol (IP), and with local SCSI Targets on the same loop using FCP, if it follows the rules described in this document.

Devices with only Private NL_Ports are called private loop devices. A Private NL_Port may have concurrent open Exchanges with other Private and Public NL_Ports on the same loop.

Devices with at least one Public NL_Port are called public loop devices. A Public NL_Port may have concurrent open Exchanges with Private NL_Ports on the same loop, Public NL_Ports on the same loop, and N_Ports or NL_Ports external to the loop.

FC-AL allows one Private or Public NL_Port to optionally provide fabric services in the absence of an FL_Port. An NL_Port that does this is called an F/NL_Port. An F/NL_Port is not permitted on private loops described by this document.

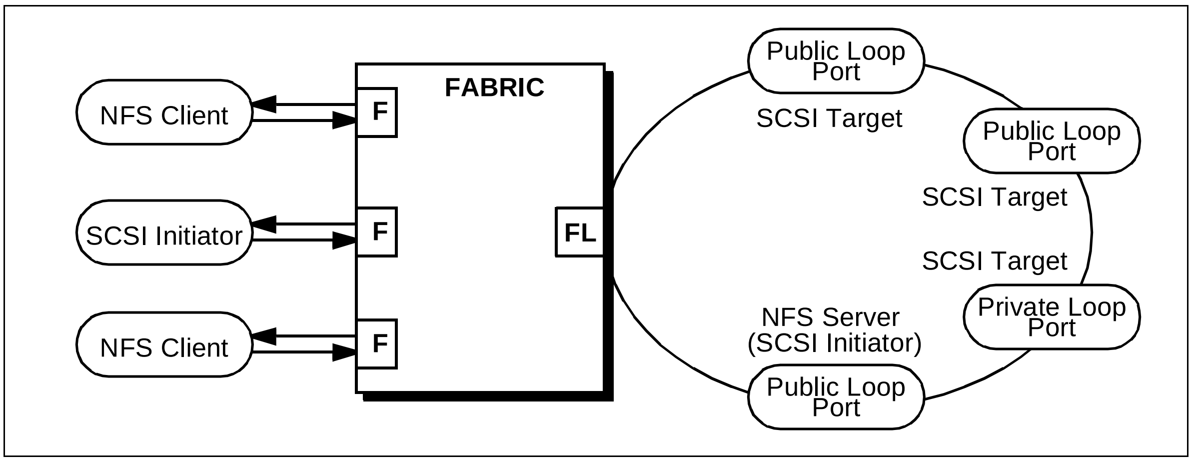

Figure 1 – Private and Public Loop Device Coexistence

A summary of Private NL_Port behavior is as follows:

Table 1 – Private vs. Public NL_Port Behavior

Behavior	Private NL_Port
NL_Port may open AL_PA = hex‘00’	Prohibited
Domain + Area of N_Port ID = hex‘0000’	Required
FL_Port may Open NL_Port	Prohibited
NL_Port may respond to AL_PA = hex‘00’ (i.e., F/NL_Port)	Prohibited
F/NL_Port may Open NL_Port	Prohibited
Public NL_Port may Open NL_Port	Allowed
Private NL_Port may Open Private or Public NL_Port	Allowed

If a Private NL_Port transmits a Class 3 frame with D_ID = hex’0000AB’ to a Public NL_Port with N_Port ID = hex’1234AB’, the Private NL_Port cannot rely on a response from the Public NL_Port.

Similarly, a Public NL_Port must be able to tolerate timeouts on Sequences containing Class 3 frames with a nonzero Domain+Area fields in the D_ID field if transmitted to a Private NL_Port (see 10.3, SCSI Target Discovery).

5 FC-2 feature sets

The tables in this clause list features described in FC-PH. These tables indicates whether the feature is required, prohibited, or optional for compliance with this specification. Features which are not listed do not affect interoperability of Private NL_Ports.

Reserved FC-PH fields are not required to be checked for zeroes. Validity bits set to 0 remove any requirement to check the corresponding field for zeroes (e.g., if F_CTL bit 3=0, receiving N_Ports are not required to verify that the parameter field in word 5 of the frame header contains zeroes).

5.1 Login and Node/Port naming

Both SCSI Targets and SCSI Initiators shall validate logins by comparing Port Name, Node Name (see annex D for a tutorial on World Wide Names), and N_Port ID. All three identifiers must match or a configuration change has occurred and LOGO is required, which terminates all open Exchanges with that SCSI Initiator or SCSI Target. Consider two examples of consecutive PDISC or ADISC requests arriving at a SCSI Target, or consecutive ACC responses arriving at a SCSI Initiator.

Example	Identifier	Login Database	New ADISC, PDISC, or ACC	Response
1	N_Port ID (24 bits)	00 00AA	00 **00AB**	LOGO
	Node Name (64 bits)	1000 ABCD EF12 3456	1000 ABCD EF12 3456	
	Port Name (64 bits)	1000 ABCD EF12 3457	1000 ABCD EF12 3457	
2	Node Name (64 bits)	1000 ABCD EF12 3456	1000 ABCD EF12 3456	LOGO
	Port Name (64 bits)	1000 ABCD EF12 3456	1000 ABCD EF**98 7654**	
	N_Port ID (24 bits)	00 00AA	00 00AA	

Example 1 may occur if a node with no Hard Address has been removed and reinserted into a reconfigured loop, and takes a different soft address.

Example 2 could occur if both ports of a node are connected to the same loop with the second port non-participating. The first port failed and was bypassed, and the second port assumed the AL_PA of the (now bypassed) first port.

These examples, and other situations, such as loop reconfigurations or the connection of two formerly independent loops into a single loop, could result in one or more of the identifiers changing. These situations are outside the scope of this document.

5.2 N_Port Common Service Parameters

Table 2 lists N_Port Common Service Parameters with usage defined by this document:

Table 2 – N_Port Common Service Parameters

Common Service Parameter	SCSI Initiator	SCSI Target	Notes
FC-PH Version			
Highest Version = hex‘20’	R	R	1
Lowest Version = hex‘20’	R	R	
Buffer-to-Buffer Credit (min)	0	0	
Common Features			
Continuously Increasing Relative Offset = 1	R	R	
Random Relative Offset = 1	P	P	
Valid Vendor Version Level = 0	R	R	2
N_Port/F_Port = 0	R	R	
Alternate BB_Credit Management = 1	R	R	
Buffer-to-Buffer Receive Data Field Size (min)	256	256	3
Total Concurrent Sequences (min)	1	1	
Relative Offset by Information Category = x‘0002’ (Information Category 1 only)	R	R	
E_D_TOV	see 7.4	see 7.4	

Notes:

1 The FC-PH version must include a version of FC-PH-2 or FC-PH-3 that includes PDISC, ADISC, TPR-LO, and RNC (previously known as RVU). PDISC, ADISC, and TPRLO are described in FC-PH-2, RNC is described in FC-PH-3. Hex‘20’ is the version assigned to FC-PH3.

2 Profile versions are communicated via the Report Node Capabilities (RNC) ELS. The Valid Vendor Version Level shall not be set to 1 for communicating support of this document.

3 There is no restriction on the *maximum* frame size (up to the limits specified by FC-PH).

5.3 N_Port Class 3 Service Parameters

Table 3 lists Class 3 Service Parameters with usage defined by this document.

Table 3 – Class 3 Service Parameters

Class 3 Service Parameter	SCSI Initiator	SCSI Target	Notes
Class validity = 1	R	R	
Service Options			
Intermix Mode	-	-	
Stacked Connect Requests	-	-	
Sequential Delivery	-	-	
Dedicated Simplex	-	-	
Camp-On	-	-	
Buffered Class 1	-	-	
Initiator Control			
Sequence Initiator X_ID reassignment = ‘00’	R	R	
Initial Responder Process_Associator = ‘00’b	R	R	
Sequence Initiator ACK_0/ACK_N capable = ‘00’b	R	R	
ACK generation assistance = 0	R	R	
Initiator Data compression capable = 0	R	R	
Initiator Data compression History buffer size = ‘00’b	R	R	
Recipient Control			
ACK_0/ACK_N capable = ‘00’b	R	R	
X_ID interlock = 0	R	R	
Error Policy Supported=’00’b	R	R	
Categories per Sequence = ‘00’b (one)	R	R	
Data compression capable = 0	R	R	
Data compression History buffer size = ‘00’b	R	R	
Receive data field size (min)	256	256	
Concurrent Sequences > 0	R	R	
N_Port End-to-end Credit	n/a	n/a	
Open Sequences per Exchange > 0	R	R	1

Notes:

1 Situations may occur where successive Sequences are sent without confirmation of Sequence delivery. This can occur, for example, between consecutive FCP_DATA Sequences on a read operation, or between an FCP_DATA Sequence and FCP_RSP Sequence (see 8.6 for additional information on FCP Sequence delivery confirmation). Open Sequences per Exchange is used to determine the minimum number of different consecutive SEQ_IDs that shall be used when streamed Sequences occur (see 5.8.3.2).

5.4 Other FC-2 features

Table 4 lists other FC-2 features with usage defined by this document.

Table 4 – Other FC-2 Features

Feature	SCSI Initiator	SCSI Target	Notes
R_CTL Routing Bits			
FC-4 Device_Data frame (‘0000’b)	R	R	
Extended Link_Data frame (‘0010’b)	R	R	
Basic Link_Data frame (‘1000’b)	R	R	
Link_Control frame (‘1100’b)	P	P	1
Continue Sequence Condition = ‘00’b	R	R	
Continue Sequence Condition = ‘01’b, ‘10’b, or ‘11’b	P	P	
Ignore nonzero Continue Sequence values	R	R	
X_ID Interlock	P	P	2
Sequence Chaining (C_S bit in F_CTL = 0)	R	R	
Optional Headers (all)	P	P	
Node Name Format			
Uses a Registered Format	R	R	3
Uses a non-Registered Format	P	P	
Port Name Format			
Uses a Registered Format	R	R	3
Uses a non-Registered Format	P	P	
Node Name = N_Port Name	P	P	4
Continuously increasing Sequence count during consecutive Sequences within an Exchange	A	A	5
Consecutive Sequences to the same destination within an Exchange shall have different SEQ_IDs (required by FC-PH)	R	R	

Notes:

1 Link Control frames are specific to classes 1 and 2. A SCSI Initiator or Target is Allowed to return a P_RJT in the appropriate class of service to any Class 1 or 2 frame received if that class of service is not supported.

2 X_ID interlock only applies to classes 1 and 2. If an RX_ID other than hex’FFFF’ is used by an Exchange Responder, it shall return the RX_ID value in the first sequence transmitted as a Sequence Initiator (e.g., on the first XFER_RDY Sequence for read or write operations (if not suppressed) or the first Data Sequence for read operations).

3 Node Names and N_Port Names must use a format that contains a worldwide unique identifier issued by a recognized registration authority. The IEEE and IEEE Extended name formats are two examples of acceptable formats.

4 Some early implementations of single-port nodes with a non-replaceable port may have used the same identifier for both the Node Name and Port Name. The use of unique identifiers for both the Node Name and Port Name is recommended for future products.

5 Implementations may treat consecutive Sequences within the same Exchange as either streamed or non-streamed since $R\_A\_TOV_{seq\_qual}$=0. The use of continuously increasing SEQ_CNT is recommended as it may provide enhanced error detection.

5.5 Basic Link Service commands

Table 5 lists Basic Link Service commands with usage defined by this document

Table 5 – Basic Link Services

Feature	Orig. By SCSI Initiator	Resp. By SCSI Target	Orig. By SCSI Target	Resp. By SCSI Initiator	Note
No Operation (NOP)	P	-	P	-	
Abort Sequence (ABTS)	I	R	P	-	
Remove Connection (RMC) (Class 1 only)	P	-	P	-	

5.6 Extended Link Services

Table 6 lists Extended Link Services with usage defined by this document.

Devices which receive requests for Extended Link Services which are unsupported shall return LS_RJT with a reason code "Command not supported."

Table 6 – Extended Link Services

Extended Link Service	Orig. By SCSI Initiator	Resp. By SCSI Target	Orig. By SCSI Target	Resp. By SCSI Initiator	Note
Abort Exchange (ABTX)	P		P		
Advise Credit (ADVC)	P		P		
Discover Address(ADISC)	I	R	P		1
Discover F_Port Parameters (FDISC)	P		P		
Discover N_Port Parameters (PDISC)	I	R	P		1
ECHO	P		P		
Estimate Credit (ESTC)	P		P		
Establish Streaming (ESTS)	P		P		
Fabric Activate Alias_ID (FACT)	P		P		
Fabric Deactivate Alias_ID (FDACT)	P		P		
Fabric Login (FLOGI)	P		P		2
Get Alias_ID (GAID)	P		P		
Logout (LOGO)	I	R	I	R	
N_Port Activate Alias_ID (NACT)	P		P		
N_Port Deactivate Alias_ID (FDACT)	P		P		
N_Port Login (PLOGI)	R	R	P		
Process Login (PRLI)	R	R	P		
PRLI Common Service Parameters	P	P	P	P	
Single Service Parameter page per PRLI request	R	R	P	P	
Multiple Service Parameter pages per PRLI request	P	P	P	P	
ACC contains only those pages specified in PRLI	-	R	P		
Accept Response code="Command executed"	-	R	P		

Table 6 – Extended Link Services

Extended Link Service	Orig. By SCSI Initiator	Resp. By SCSI Target	Orig. By SCSI Target	Resp. By SCSI Initiator	Note
Process Logout (PRLO)	I	R	I	R	
Quality of Service Request (QoSR)	P		P		
Read Connection Status Block (RCS)	P		P		
Read Exchange Status Block (RES)					
Block (disk) type devices	P		P		
Stream (tape) type devices	I	A	I	A	4
Read Link Error Status Block (RLS)	I	A	P		3
Read VC Status (RVCS)	P		P		
Report Node Capabilities Information (RNC)	I	R	P		
Reinstate Recovery Qualifier (RRQ)	I	R	P		
Request Sequence Initiative (RSI)					
Block (disk) type devices	P		P		
Stream (tape) type devices	I	A	I	A	4
Read Sequence Status Block (RSS)					
Block (disk) type devices	P		P		
Stream (tape) type devices	I	A	I	A	4
Read Timeout Value (RTV)	P		P		
State Change Notification (SCN)	P		P		
TEST	P		P		
Third Party Process Logout (TPRLO)	I	R	P		

Notes:

1 See 10.3 for a description of SCSI Target Discovery. Only ADISC or PDISC is required by a SCSI Initiator, but not both. SCSI Targets must support receipt of both, for interoperability with SCSI Initiators that support one or the other.

2 Private NL_Ports are prohibited from performing FLOGI.

3 Not all fields in the LESB may be supported (see vendor-specific documentation). RLS can be a diagnostic tool for isolating link degradation. Using RLS, diagnostic software can use a combination of self-directed test frames, LPB, and LESB polling to determine which device or link segment is causing link errors.

4 These link services permit the progress in a Sequence or Exchange at the Sequence Responder to be made visible for recovery purposes by a Sequence Initiator (e.g., determine whether a Sequence was delivered before using ABTS).

5.7 Responses to Link Services

Table 7 summarizes the response that shall be generated as a result of receiving different Link Service requests when the recipient NL_Port is not currently logged in with the sending NL_Port.

Table 7 – Responses to Link Services from NL_Ports not Logged-In

Frame Received	NL_Port Not Logged In (PLOGI)	NL_Port Logged In (PLOGI)
ABTS	Discard and send LOGO	BA_ACC, BA_RJT (note 2)
ADISC	Discard and send LOGO	ACC (note 1), LS_RJT
LOGO	ACC	ACC
PDISC	Discard and send LOGO	ACC (note 1), LS_RJT
PLOGI	ACC	ACC
PRLI	Discard and send LOGO	ACC
PRLO	Discard and send LOGO	ACC, LS_RJT (note 3)
RLS	Discard and send LOGO	ACC
RNC	Discard and send LOGO	ACC
RRQ	Discard and send LOGO	ACC

Notes:

1 All three identifiers: N_Port ID, Port Name, and Node Name must match a logged-in NL_Port for ACC to be returned. If all three identifiers do not match LOGO is returned. If other conditions prevent execution of the ADISC or PDICS ELS, LS_RJT may be returned with the appropriate reason code.

2 BA_ACC or BA_RJT as specified in clause 9.3

3 If PRLI has not been successfully completed, set the reason code to “image pair does not exist”.

While not shown in the table, an NL_Port may return LS_RJT to any Extended Link Service request if that request is not properly formed.

An NL_Port which has discarded a frame due to the lack of appropriate N_Port login state shall respond by sending a LOGO to the NL_Port which originated the discarded request. No Exchange is created in the recipient NL_Port for the discarded request, and the originator of the discarded request terminates the Exchange associated with the discarded request and any other open Exchanges for the NL_Port sending the LOGO. The LOGO is not part of the Exchange associated with the discarded request.

5.8 Exchange and Sequence management

This clause describes required and optional behavior required of devices conforming to this profile.

5.8.1 Exchange Originator

The Originator of each Exchange shall assign an OX_ID to that Exchange. The Exchange Originator shall maintain OX_ID uniqueness with a given Responder.

Exchange Originators are not assured of RX_ID uniqueness across multiple Responders and must qualify the RX_ID with the Native Address Identifier of the Responder, if used.

Since the environment defined by this document insures in-order delivery of frames and assumes that frames are processed in the order received, a set of simplifying assumptions can be made regarding when an Exchange is open.

The Exchange Originator shall consider the Exchange open from the time the first frame of the first information unit (frame with the First_Sequence bit in the F_CTL field set to 1) is sent until one of the following occurs:

- the last frame of the last information unit (frame with the Last_Sequence bit in the F_CTL field set to 1) is received
- the Exchange is aborted using ABTS (see clause 9.3) and a response to the ABTS is received
- A LOGO is sent to, or received from, the Exchange Responder
- An LS_RJT is sent in response to an ADISC or PDISC during target discovery

5.8.2 Exchange Responder

Exchange responders may optionally assign an RX_ID or use the value of hex'FFFF'.

Exchange Responders in a multiple Originator environment are not assured of OX_ID uniqueness across multiple Originators and must qualify the OX_ID with the Native Address Identifier of the Originator. Exchange Responders are not required to check for OX_ID uniqueness with a given Originator.

The Exchange Responder shall consider the Exchange open from the time the first frame of the first information unit (frame with the First_Sequence bit in the F_CTL field set to 1) is received until one of the following occurs:

- the last frame of the last information unit (frame with the Last_Sequence bit in the F_CTL field set to 1) is sent
- the Exchange is aborted using ABTS (see clause 9.3)
- A LOGO is sent to, or received from, the Exchange Originator
- An LS_RJT is sent in response to an ADISC or PDISC during target discovery

5.8.3 Sequence management

The following clauses define Sequence management requirements for devices operating in accordance with this document.

5.8.3.1 Sequence open

The Sequence Initiator shall consider a Sequence open from the time that the first frame of the Sequence (the frame with the SOFi3 delimiter) is sent until one of the following occurs:

- the last frame of the Sequence (the frame with the EOFt delimiter) is sent and $R\_A\_TOV_{SEQ\_QUAL}$ has elapsed
- the Sequence is aborted using ABTS (see clause 9.3)
- A LOGO is sent to, or received from, the Sequence recipient

The Sequence Recipient shall consider a Sequence open from the time that the first frame of the Sequence (the frame with the SOFi3 delimiter) is received until one of the following occurs:

- the last frame of the Sequence (the frame with the EOFt delimiter) is received
- the Sequence is aborted using ABTS (see clause 9.3)
- A LOGO is sent to, or received from, the Sequence initiator

5.8.3.2 SEQ_ID usage

The following paragraphs summarize the rules governing reuse of SEQ_IDs (see also annex C).

For Sequences which transfer Sequence Initiative (other than the FCP_RSP which is described separately below),

a) An NL_Port may reuse a SEQ_ID for the same Exchange following confirmation of Sequence delivery (see 8.6 for a list of FCP Sequences whose delivery can be determined).

b) An NL_Port may reuse the SEQ_ID within a different Exchange (to the same, or a different destination NL_Port) immediately following transmission of the last frame of the Sequence without waiting for confirmation of Sequence delivery. The next Sequence sent by the Sequence initiator is not a consecutive Sequence since it is part of a different Exchange.

For Sequences which do not transfer Sequence Initiative,

a) Consecutive FCP_DATA Sequences for the same Exchange shall follow the FC-PH rules for streamed Sequences (an example of which occurs when multiple FCP_DATA Sequences are used during a single read operation). The rules for streamed Sequences are summarized in the following list:

1) The first FCP_DATA Sequence after transfer of Sequence Initiative is not a streamed Sequence. It may use any eligible SEQ_ID and the SEQ_CNT may be either zero or continuously increasing (incremented by one from the last frame of the previous Sequence of that Exchange).

2) The second, and subsequent, Sequences of a group of consecutive Sequences within the same Exchange should be treated as streamed. The following rules apply (X is the number of Open Sequences per Exchange from PLOGI (see table 3)):

i) X+1 different SEQ_IDs shall be used.

ii) The X+1 different SEQ_IDs may be assigned by the Sequence Initiator in any order (they are not required to be monotonically increasing).

iii) A SEQ_ID may be reused after X (where X is the number of Open Sequences per Exchange from PLOGI) different intervening SEQ_IDs have been used, provided that $R\_A\_TOV_{SEQ\_QUAL}$ has expired for all frames in the prior use of that SEQ_ID. The value for $R\_A\_TOV_{SEQ\_QUAL}$ is defined in 7.5.

iv) The Sequence Count (SEQ_CNT) across streamed Sequences shall be continuously increasing. The SEQ_CNT of the first frame of the streamed Sequence shall be incremented by one from the SEQ_CNT of the last frame of the previous Sequence.

NOTE – FC-PH3 adds a bit to PLOGI which indicates that a port uses continuously increasing SEQ_CNT on all Sequences within an Exchange with the first Sequence of the Exchange beginning at SEQ_CNT zero. Ports that set this bit during login are obligated to follow this behavior since the Sequence Recipient may use the bit to perform Sequence checking.

NOTE – Some existing devices are not capable of generating continuously increasing SEQ_CNT across consecutive Sequences. In this case, the following rules apply: A minimum of two different SEQ_IDs shall be used by the Sequence initiator and the SEQ_CNT across consecutive Sequences may restart at x‘0000’.

b) Since frame delivery is not confirmed, the Sequence Initiator shall not reuse a SEQ_CNT within a given Sequence. For Sequences with a beginning SEQ_CNT of zero, the SEQ_CNT is not allowed to wrap upon reaching hex‘FFFF’. For Sequences with a beginning SEQ_CNT of ‘N’ (where N is not zero) the SEQ_CNT may wrap upon reaching hex‘FFFF’ and continue from zero up to a value of N-1.

c) When there is additional data still to transfer and all of the data frames associated with a given Sequence have been sent, or all permissible SEQ_CNT values used, the Sequence Initiator shall begin a new Sequence in order to continue sending the data.

d) The Sequence initiator shall not be required to send the maximum allowed number of data frames per Sequence, even if there is additional data to send (for example, if the maximum Sequence size is 1000 frames, the Sequence initiator may end the current Sequence after 600 frames and start a new sequence to send any remaining frames).

e) During a read type command with at least one FCP_DATA Sequence, the FCP_RSP Sequence may optionally be considered a streamed Sequence and observe the rules for streamed Sequences listed above. In any case, the FCP_RSP shall not use the same SEQ_ID as the preceding FCP_DATA Sequence.

f) A SCSI Target may reuse a SEQ_ID for a Read FCP_DATA Sequence immediately following transmission of the final FCP_RSP of an Exchange.

Examples of SEQ_ID and SEQ_CNT usage during streamed Sequences may be found in annex C.

5.8.4 Sequence errors

The following conditions shall constitute a Sequence error when detected by the Sequence Recipient:

a) if a frame with an SOFi3 delimiter is received and the SEQ_CNT is not equal to zero or +1 from the SEQ_CNT of the last frame of the previous Sequence of that Exchange

b) if the SEQ_CNT of a received frame with an SOFn3 delimiter is not +1 greater than the previous frame received for that Sequence (i.e., a frame was lost)

NOTE – this should also detect the case where a frame with an SOFn3 delimiter is received for a SEQ_ID that is not currently open since the SEQ_CNT of the previous frame for that Sequence is undefined.

c) if a frame with an SOFi3 delimiter is received and the previous Sequence of that Exchange is still open (see 5.8.3.1 for a definition of when a Sequence is open)

d) if the relative offset in the parameter field of a received frame with an SOFn3 delimiter is not equal to the (relative offset + the payload size) of the previous frame received for that Sequence

e) The next frame of a Sequence is not received within E_D_TOV

f) If, during the same Sequence Initiative, a Sequence is received which has the same SEQ_ID as the previous Sequence of that Exchange

When a Sequence error is detected by the Sequence Recipient, it shall take the appropriate action as described in clause 9.

6 FC-AL feature set

This clause describes the use of Fibre Channel Arbitrated Loop features and capabilities by devices which conform to this document.

6.1 FC-AL features

Table 8 lists Fibre Channel Arbitrated Loop features with usage defined by this document.

Table 8 – FC-AL Features

Feature	SCSI Initiator	SCSI Target	Notes
Open Full Duplex - OPN(yx)			
Open Originator can send	I	I	
Open Recipient accepts	R	R	
Open Half Duplex - OPN(yy)			
Open Originator can send	I	I	
Open Recipient accepts	R	R	
Open Multicast/Selective Replicate - OPN(yr), OPN(fr)			
Open Originator	P	P	5
Unfairness	I	I	
Transfer State use	I	I	
LILP/LIRP			
Loop Master can originate	A	A	2
Non-loop Master L_Ports accept	A	A	2
Login_BB_Credit			
Advertise Login_BB_Credit = 0	A	A	
Advertise Login_BB_Credit > 0	A	A	
Accept Login_BB_Credit = 0	R	R	
Accept Login_BB_Credit > 0	R	R	3
LPEyx/LPByx/LPEfx transmission	I	P	1
MRKtx	P	R	4
Attempt to acquire Hard Address (if any) during LIHA sequence of loop initialization following a power cycle, power-on reset, or recognition of LIP(AL_PD,AL_PS)	R	R	

Notes:

1 LPEfx is useful for resetting the bypass circuits of NL_Ports which have been bypassed and have lost their AL_PAs due to an intervening LIP (and therefore cannot be enabled using an addressed LPEyx). SCSI Targets complying with SFF-8045 shall be able to receive LPE/LPB and toggle the corresponding enable signal.

2 The initialization Loop Master shall originate the LIRP/LILP Sequences only if the Loop Initialization identifier (in the LISA frame) is hex‘11050100’. Any L_Port not capable of supporting LIRP/LILP shall set the Loop Initialization identifier (in the LISA frame) to hex‘11050000’.

3 The actual value used shall be between zero and the Login_BB_Credit value received during PLOGI.

4 An NL_Port that receives a MRK shall attempt to forward the MRK. FC-AL states that an L_Port may remove a MRK if necessary for clock skew management.

5 FC-AL requires that an L_Port in the Monitoring or Arbitrating state shall forward the OPNr, whether it implements the associated function or not.

6.2 Alternate Login_BB_Credit management

OPN Originators may open full or half duplex, regardless of the value of Login_BB_Credit. See annex A for examples.

6.2.1 Login_BB_Credit=0

Interoperability with devices which advertise Login_BB_Credit=0 is required. When Login_BB_Credit=0 at the other L_Port, the following rules apply:

a) if the OPN Originator or OPN recipient has one or more available receive buffers, it shall transmit a number of R_RDYs equal to, or less than, the number of available receive buffers upon receipt or transmission of OPNy.

b) if the OPNy Recipient has no available receive buffers, it may respond with an immediate CLS rather than keep the circuit open until a buffer becomes available.

c) the Sequence initiator shall have an Available BB_Credit > 0 before transmitting a frame.

6.2.2 Login_BB_Credit>0

a) When an OPN Originator with Login_BB_Credit>0 transmits an OPNyx (full duplex OPN) to the other L_Port, it shall have Available BB_Credit equal to or greater than the Login_BB_Credit granted to the Open Recipient during PLOGI. This is to prevent buffer overrun if the OPN Recipient begins frame transmission immediately upon receipt of the OPNy.

b) The OPN Originator shall transmit no fewer than Login_BB_Credit R_RDYs to the OPN Recipient prior to sending a CLS.

c) When the OPN Recipient has granted the OPN Originator a Login_BB_Credit>0, it shall transmit no fewer than Login_BB_Credit R_RDYs to the OPN Originator following receipt of the OPNy and prior to sending a CLS.

d) When an L_Port has granted a Login_BB_Credit>0 to the other L_Port in the current loop circuit, that L_Port shall not transmit a CLS until it can guarantee that the number of available receive buffers upon receipt of an immediate, subsequent OPN will be equal to or greater than the maximum Login_BB_Credit granted to any L_Port. The required number of receive buffers is (X + max. Login_BB_Credit), where X is the number of frames that may still be received during the current loop circuit.

1) If one or more R_RDYs has been sent and a CLS has not been received, the L_Port shall not transmit a CLS unless it has a minimum of (MAX(Login_BB_Credit or number of R_RDYs sent) - number of frames received + maximum Login_BB_Credit granted to any L_Port) receive buffers available.

2) Upon receipt of a CLS, the L_Port may transmit a CLS once it has at least as many receive buffers available as the maximum Login_BB_Credit granted to any L_Port.

3) The open originator may send a CLS at any time after any required R_RDYs have been sent, provided that it delays entry to the Monitoring state until it has at least as many receive buffers available as the maximum Login_BB_Credit granted to any L_Port.

e) If a Sequence Initiator (either as OPN Originator or OPN Recipient) chooses to take advantage of nonzero Login_BB_Credit advertised by a Sequence Recipient, the following rules apply:

1) the Sequence Initiator must be aware of the value of Login_BB_Credit (or a lesser value) granted by the Sequence Recipient in order to know how many frames it may transmit without receiving any R_RDYs.

2) the Sequence Initiator may transmit up to Login_BB_Credit frames following the transmission or receipt of the OPN without waiting for R_RDYs,

3) the Sequence Initiator shall discard a number of initial R_RDYs equal to the value retained as the Login_BB_Credit value, regardless of how many frames may have already been transmitted, in order to avoid “double crediting” the Sequence Recipient which must observe rule c above.

After completing the Loop Initialization Procedure, Login_BB_Credit shall be zero with all devices until after completion of the SCSI Target Discovery protocol (see 10.3).

NOTE – This is essential for two reasons: The device at a particular address which had Login_BB_Credit > 0 may have been replaced by a device with Login_BB_Credit = 0. Until SCSI Target Discovery is complete, there is no way to be sure that the port is communicating with the Login_BB_Credit > 0 device. Secondly, a device may have been trying to balance credit by not sending CLS when the LIP occurred. The LIP could have prematurely forced the device into the MONITORING state at a time when it has insufficient buffers available. Requiring SCSI Target Discovery before assuming Login BB_Credit > 0 enables the device to free its buffers before attempting to resume normal operations.

6.2.3 OPN and CLS latencies

When Login_BB_Credit=0, CLS latency is minimized but a latency exists on every OPNy while waiting for one or more R_RDYs before frame transmission can begin.

When Login_BB_Credit>0, OPNy latency is minimized but CLS latency may increase while waiting for Login_BB_Credit buffers to become available. It is possible to reduce CLS latency when Login_BB_Credit>0 by increasing the available receive buffers within a Recipient. For example, a Sequence Recipient may choose to have:

Available_Rx_Buffers = Available BB_Credit (of current Sequence Initiator) + maximum Login_BB_Credit granted to any L_Port

at all times during a loop circuit with a Sequence Initiator. This would allow the Sequence Recipient as a CLS Recipient to forward a CLS immediately without having to wait for Login_BB_Credit to become available. A Sequence Recipient controls the Available BB_Credit variable in a Sequence Initiator through the return of R_RDYs.

Some NL_Ports may be tempted to reduce CLS latency in another fashion. To prevent buffer overruns, a CLS Recipient is only required to have maximum Login_BB_Credit granted to any L_Port buffers available before the next OPN is received, not necessarily before the CLS is forwarded. NL_Ports which initiate or forward a CLS before these buffers are available and rely on a (topology or application specific) delay between the time a CLS is transmitted or forwarded and the time an OPN is received do so at their own risk and with methods not defined in this document. This may result in lost frames on a subsequent open and the attendant error recovery.

6.3 Loop Initialization features

NL_Ports shall attempt to acquire their previously acquired AL_PA before they attempt to acquire their Hard Address, unless the NL_Port has experienced a power cycle, power-on reset, recognized a LIP(AL_PD,AL_PS) for that port, or any other event that causes the previously acquired AL_PA to be lost.

6.3.1 Initializing LIP(F7,F7) and LIP(F7,AL_PS)

If there are no address conflicts, these LIPs shall have no effect on existing Exchanges except that the Exchange in progress may be disrupted, resulting in a ULP timeout. LIP(F7,F7) is used if the initiating NL_Port has no valid AL_PA, and LIP(F7,AL_PS) is used if it does.

6.3.2 Selective Hard Reset LIP(AL_PD,AL_PS)

Upon recognition of LIP(AL_PD,AL_PS) the NL_Port designated by AL_PD shall perform the appropriate resetting actions specified in table 16. All other L_Ports shall treat this as LIP(F7,AL_PS). SCSI Targets are prohibited from issuing this LIP.

6.3.3 Loop Failure LIP(F8,AL_PS) and LIP(F8,F7)

A Loop Failure is defined as detection of any of the following:

a) Loss of Signal

b) Loss of Synchronization for longer than R_T_TOV

If a Loop Failure occurs, the L_Port which detects the failure shall issue LIP(F8,AL_PS) if it has a valid AL_PA, or LIP(F8,F7) if it doesn't.

6.3.4 Failure to obtain an AL_PA

An L_Port which loses its previously acquired address shall implicitly logout all NL_Ports regardless of whether or not that AL_PA was previously-assigned, hard-assigned, or soft-assigned. If no soft address is available, it shall go to non-participating mode. An NL_Port which has experienced a power cycle, received a power-on reset, or recognized a LIP(AL_PD,AL_PS) is not required to retain a previously acquired address to use during the next loop initialization.

7 Timers on Private Loop

FC-PH requirements for timers are summarized in annex B. This clause defines the timers and time-out values used for private loop implementations.

Table 9 – Timer Summary

Timer	Implemented by		Description	Value
	Initiator	Target		
AL_TIME	Y	Y	Arbitrated Loop Timeout Value	15 ms
LIS_HOLD_TIME	Y	Y	Loop Initialization Sequence Hold Time	1 ms
R_T_TOV	Y	Y	Receiver_Transmitter Timeout Value	100 ms
E_D_TOV	Y	N	Error_Detect_Timeout Value	2 sec
$R\_A\_TOV_{SEQ\_QUAL}$ $R\_A\_TOV_{ELS}$	Y Y	N N	Resource_Allocation Timeout Value	0 sec (note 1) 2 sec
RR_TOV	N	Y	Resource Recovery Timeout Value	2 sec
ULP_TOV	Y	N	Upper Level Protocol Timeout Value	>E_D_TOV

1 The division of R_A_TOV usage as specified in this table differs from FC-PH. This is due to the unique characteristics associated with the PLDA environment.

7.1 Arbitrated Loop Time (AL_TIME)

AL_TIME is specified as 15 ms, which represents two times the worst case round-trip latency for a very large loop.

7.2 Loop Initialization Sequence Hold Time (LIS_HOLD_TIME)

LIS_HOLD_TIME is the maximum amount of time between when a node receives a Loop Initialization Sequence until it forwards it to the next node. The value for LIS_HOLD_TIME is equal to 1ms.

7.3 Receiver_Transmitter Timeout (R_T_TOV)

The Receiver_Transmitter timeout value (R_T_TOV) is used by the receiver logic to detect Loss of Synchronization. The value for R_T_TOV is 100 ms.

7.4 Error_Detect Timeout (E_D_TOV)

E_D_TOV is the maximum time permitted for a Sequence Initiator between the transmission of consecutive Data frames within a single Sequence. This is also the minimum time that a Sequence Recipient shall wait for the reception of the next frame within a single Sequence before recognizing a Sequence timeout. The value is 2 seconds.

E_D_TOV must include the time required to gain access to the loop in addition to the actual frame transmission time.

SCSI Target devices may optionally implement this timer, it is not required by this profile.

7.5 Resource Allocation Timeout (R_A_TOV)

R_A_TOV has two separate components which are labeled $R\_A\_TOV_{SEQ\_QUAL}$ and $R\_A\_TOV_{ELS}$.

$R\_A\_TOV_{SEQ\_QUAL}$ is used to define the minimum amount of time that a Sequence initiator shall wait before reusing the Sequence qualifiers (SEQ_ID and SEQ_CNT). This value is also the minimum amount of time that a SCSI Initiator shall wait following receipt of the ACC to ABTS before transmitting a Reinstate Recovery Qualifier (RRQ) ELS. $R\_A\_TOV_{SEQ\_QUAL}$ is equal to 0 seconds.

NOTE – using a value of 0 for this time out value assumes that a Sequence initiator does not transmit any frames for a Sequence after an ABTS is sent for that Sequence. If a design uses a queueing mechanism for the transmission of Sequences, the queue for a given Sequence must be empty before an ABTS for that Sequence can be sent, or the act of sending the ABTS purges the queue.

A value of twice $R\_A\_TOV_{ELS}$ is used to determine the minimum time that the Originator of an Extended Link Service or FC-4 Extended Link Service request shall wait for the response to that request. $R\_A\_TOV_{ELS}$ is equal to 2 seconds, therefore the minimum wait shall be twice this time, or 4 seconds.

7.6 Resource Recovery Timer (RR_TOV)

RR_TOV is the minimum time a SCSI Target shall wait for a specific SCSI Initiator to perform Exchange Authentication (see 10.4) following the completion of the Loop Initialization Protocol (i.e., the receipt of CLS while in the OPEN-INIT state). RR_TOV is also the minimum time a SCSI Target shall wait for a SCSI Initiator response following transfer of Sequence Initiative from the SCSI Target to the SCSI Initiator (e.g., following transmission of the FCP_XFER_RDY during a write command). A suggested default value is 2 seconds.

Upon expiration of RR_TOV, a SCSI Target may implicitly or explicity perform a LOGO with that SCSI Initiator, terminate all open Exchanges for that SCSI Initiator, and reclaim the resources associated with those Exchanges. The suggested default value of 2 seconds was felt to be a sufficient amount of time for a SCSI Initiator to complete the post-LIP processing and initiate the Exchange Authentication process with each SCSI Target on the loop.

NOTE – A proposal has been prepared (see annex E) to specify this timeout value through a SCSI mode page in order to allow a system to tailor this value to the needs of that system.

7.7 Upper Level Protocol Timeout (ULP_TOV)

This is an operation-specific timer maintained by the Upper Level Protocol. The minimum value used for ULP_TOV shall be E_D_TOV.

It is used to time the completion of Exchanges associated with ULP operations. Since the amount of time required varies depending upon the operation, the value assigned for this timer is determined by operation being timed. Some operations may require extended periods of time to complete (for example, Format Unit with Immediate = 0).

8 SCSI-FCP Feature Set

The following Feature Sets are described in FCP. The tables in this clause indicates whether the feature is required, prohibited, or optional for compliance with this specification.

8.1 Process Login parameters

Table 10 lists Process Login parameters with usage defined by this document.

Table 10 – PRLI Parameters

Feature	SCSI Initiator	SCSI Target	Notes
Originator Process Associator Valid = 1	P	P	
Responder Process Associator Valid = 1	P	P	
Originator Process Associator	P	P	
Responder Process Associator	P	P	
Establish Image Pair (bit 13) = 1	R	R	2
Establish Image Pair (bit 13) = 0	A	R	2
Initiator Function =1	R	A	1
SCSI Target Function = 1	A	R	1
Data Overlay Allowed = 1	P	P	
Data + Response in same Sequence (Read) = 1	P	P	
Command + Data in same Sequence (Write)= 1	P	P	
Read XFER_RDY Disabled = 1	R	R	
Write XFER_RDY Disabled = 1	P	P	

Notes:

1 Allowed for XOR functionality

2 This bit must be set to one in the PRLI payload (requesting the image pair be established) and must be set to a one in the PRLI ACC payload when the image pair has been successfully established before sending FCP frames. The bit may be set to zero for the purpose of checking another NL_Port's capabilities.

8.2 FCP Information Units

From a SCSI Initiator to a SCSI Target, the T1 (CMD/Task Mgmt with SI transferred), and T6 (Write Data with SI transferred) are required. All other SCSI Initiator to SCSI Target IUs are prohibited.

From a SCSI Target to a SCSI Initiator, the I1 (FCP_XFER_RDY on Write), I3 (Read Data), I4 (FCP_RSP) are required. All other SCSI Target to SCSI Initiator IUs are prohibited. SCSI Targets shall respond to Task Management functions using I4. Examples of write and read type operations are shown in Figure 2.

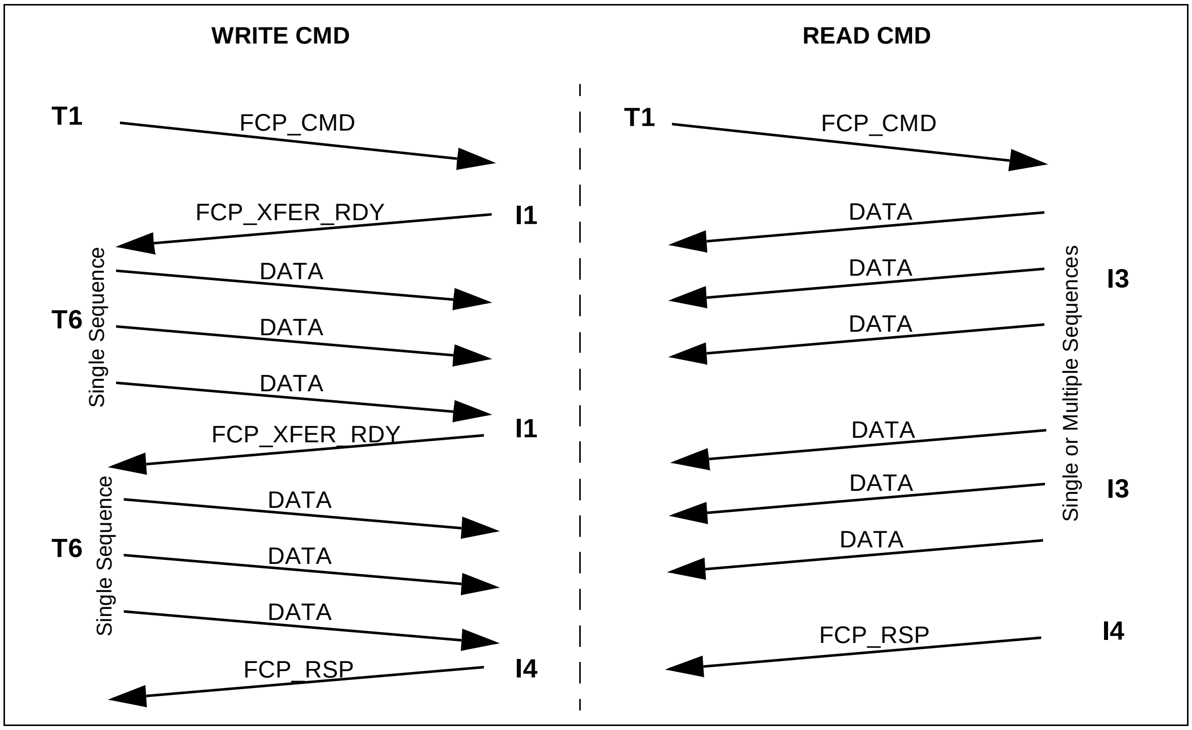

Figure 2 – FCP Read/Write IU Examples

8.2.1 FCP_CMND IU (T1)

The T1 Information Unit shall be a single-frame Sequence.

SCSI Targets that report a peripheral device type of Array Controller (hex'C') shall conform to the SCC defined LUN addressing mechanism. Other SCSI Targets shall address the LUNs using the “first level addressing” field of SCC (reference [9]) with “peripheral device addressing method” (00b) as shown in figure 3, unless they report a different LUN addressing mechanism in response to a Report LUN command. For example:

Figure 3 – FCP 8-byte LUN

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00	LUN	00	00	00	00	00	00

FCP_DL in the FCP_CMND payload should always be equal to the number of bytes expected to be transferred for the command. For SCSI commands which specify the Transfer Length in the CDB in logical blocks, the FCP_DL should be equal to (Transfer_Length x Block_Size). SCSI Targets are not required to check that the FCP_DL is in fact the (Transfer_Length x Block_Size).

On read commands, the SCSI Initiator is responsible for ensuring the amount of data returned is equal to the amount specified by FCP_DL, even if the Status returned is GOOD. If the amount does not match FCP_DL, a command-specific ULP recovery action may need to be invoked. Since there are no transfers of Sequence Initiative during read operations once the SCSI Target receives the T1 Information Unit, status may be returned as GOOD even though some of the data was not received by the SCSI Initiator. This may occur as the result of lost or corrupted frames in the read data.

NOTE – The manner in which a SCSI Initiator determines that the correct amount of data is returned is implementation dependent and may include counting the number of bytes returned, computing the number of bytes received by use of the relative offsets, or other means not specified.

8.2.2 FCP_XFER_RDY IU

The FCP_XFER_RDY IU (I2) shall not be used during read type (data in) operations. This shall be indicated by setting the ‘READ XFER_RDY DISABLED’ bit during process login.

During write type (data out) operations, FCP_XFER_RDY (I1) shall be sent prior to each Write Data Sequence. The I1 Information Unit shall be a single-frame Sequence.

If the Enable Modify Data Pointer (EDMP) in the SCSI Disconnect-reconnect mode page = ‘1’b, then for transfers consisting of multiple Write Data Sequences, the DATA_RO parameter contained in consecutive FCP_XFER_RDY Sequences does not have to be continuously increasing. This provides the logical equivalent of SCSI Modify Data Pointers. NL_Ports conforming to this profile shall insure that data overlay does not occur as a result of the values chosen for the DATA_RO parameter. This is not the same as the Random Relative Offset N_Port login parameter, which refers to Relative Offset on consecutive frames within a Sequence.

The DATA_RO parameter in the FCP_XFER_RDY IU on write type operations for disk devices shall be a multiple of 8 bytes.

The BURST_LEN parameter in the FCP_XFER_RDY IU on write type operations shall be less than or equal to the amount of data remaining to be transferred for the write task and not exceed the maximum burst size in the Disconnect-reconnect mode page.

8.2.3 FCP_DATA IU

The FCP_DATA IU is used to transfer the data associated with an operation, if any. This data may be logical data to or from the medium, as well as command parameter data (such as Mode Select data) or command response data (such as Mode Sense data). FCP does not distinguish between logical data and command parameter or response data.

The T6 and I3 Information units may be either single-frame or multiple-frame Sequences.

Within a Sequence, only the FCP_DATA frame containing the largest Relative Offset may contain fill bytes.

NOTE – Some first generation implementations require that all but the last frame of a multi-frame Sequence transmitted to the disk drive be equal in size to the receive data field size in the N_Port common service parameters. See vendor documentation for more information.

8.2.3.1 Write FCP_DATA IU (T6)

The Parameter field of the first frame of a Write data Sequence shall be set to the Relative Offset specified by the corresponding FCP_XFER_RDY. On subsequent frames of the Sequence, continuously increasing Relative Offset shall be used.

8.2.3.2 Read FCP_DATA IU (I3)

Within Read Data Sequences, the Relative Offset on consecutive frames within that Sequence shall be continuously increasing. If the Enable Modify Data Pointer (EDMP) in the SCSI Disconnect-reconnect mode page = ‘1’b, then for read-type commands with multiple data (I3) Sequences, the starting Relative Offsets (the Relative Offsets in the first frame of each Sequence) need not be continuously increasing. NL_Ports conforming to this profile shall insure that data overlay does not occur as a result of the values chosen for the starting Relative Offsets.

During read-type commands with multiple data (I3) Sequences, it is recommended that the NL_Ports treat all data Sequences other than the first as streamed Sequences and follow the associated rules for streamed Sequences (see 5.8.3 and annex C).

The Relative Offset in the first frame of each Read Data Sequence for disk type devices shall be a multiple of 8 bytes.

8.2.4 FCP_RSP (I4)

The I4 Information Unit from a disk type device shall be a single-frame Sequence. An I4 from a non-disk type Target may be a multiple-frame sequence.

According to FC-PH, the first two bits [31:30] of the first word of a command status frame payload fall into one of the following categories: 00 (successful and complete), 01 (successful but incomplete), 10 (unsuccessful but complete), 11 (unsuccessful and incomplete).

Since the first word of FCP_RSP frames are reserved in FCP, these bits are set to zero, regardless of the content of the SCSI Status portion of the payload (which may indicate the command was unsuccessful and/or incomplete). SCSI Initiators shall not rely on bits 31:30 of word 0 in FCP_RSP to determine success or completion status of a command.

An FCP_RSP following a data-in Sequence (I3) may or may not be treated as a streamed Sequence.

8.2.4.1 FCP_RSP Residual checking

SCSI Targets that transfer exactly FCP_DL data bytes during the FCP_DATA IUs shall set the FCP_RESID_UNDER to ‘0’b. SCSI Targets that transfer less than FCP_DL data bytes during the FCP_DATA IUs shall set the FCP_RESID_UNDER to ‘1’b.

When FCP_RESID_UNDER is set to ‘0’b, the SCSI Initiator may be able to determine if all of the expected data was transferred by comparing the FCP_DL to the actual number of bytes transferred. If these values are not the same the ULP may be notified so that the appropriate action can be taken.

If the FCP_RESID_UNDER bit is set to ‘1’b, a transfer that did not fill the buffer to the expected displacement FCP_DL was performed and the value of FCP_RESID shall be equal to:

FCP_DL - highest offset of any byte transmitted

Failure to transfer FCP_DL bytes does not necessarily indicate an error for some devices and commands.

If the FCP_RESID_OVER bit is set, the transfer was truncated because the data transfer required by the SCSI command extended beyond the displacement value of FCP_DL. Those bytes that could be transferred without violating the FCP_DL value may or may have been transferred. The value of FCP_RESID shall be equal to:

(Transfer length required by command) - FCP_DL

Use of FCP_RESID_OVER is required for both read and write type commands.

During commands that do not contain an FCP_DATA IU, FCP_RESID_UNDER and FCP_RESID_OVER shall both be set to ‘0’b, and the value of the FCP_RESID is undefined.

8.2.4.2 FCP_RSP payload

Table 11 lists those FCP_RSP fields with usage defined by this document.

Table 11 – FCP_RSP Payload

Feature	SCSI Initiator	SCSI Disk Target	Other SCSI Target	Notes
FCP_SNS_LEN	R	<=96	<=255	1
FCP_SNS_INFO	R	I	I	
Length of Additional Sense Bytes in FCP_SNS_INFO	R	<=78	<=247	1
FCP_RSP_LEN	0 or 8	0 or 8	0 or 8	
FCP_RSP_INFO	R	I	I	2
FCP_RESID_UNDER	R	I	I	
FCP_RESID_OVER	R	I	I	
FCP_RESID	R	I	I	

Notes:

1 SCSI single-LUN direct access SCSI Targets have this restriction to keep FCP_RSP payload to less than or equal to 128 bytes. Other device types (e.g., controller devices) may return sense information (FCP_SNS_INFO) up to the SCSI limit of 255 bytes, and SCSI Initiators which support multiple device types may have to support the largest value allowed by the SCSI-3 Primary Command set. The FCP_RSP Payload for a disk therefore consists of a maximum of 96 bytes of Sense Data, and 32 bytes of other information.

2 If present, this field shall contain the FCP_RSP code in an 8 byte field in the format specified by FCP (see table 12).

8.2.4.3 FCP_RSP Code

The FCP_RSP_INFO field contains information describing only the protocol failures detected during the execution of a FCP I/O Operation. The FCP_RSP_INFO does not contain link error information, since FC-PH provides the mechanisms for presenting such errors. The FCP_RSP_INFO does not contain SCSI logical unit error information, since that is contained in the FCP_STATUS and FCP_SNS_INFO fields. RSP_CODE values of hex’04’ and hex’05’ are not valid responses to SCSI commands.

The RSP_CODE is independent of the SCSI Status and should be examined prior to interpretation of the SCSI Status. RSP_CODEs of hex’04’ and hex’05’ are not valid responses to SCSI commands. For other, non-zero values of the RSP_CODE, the SCSI Status may not be valid.

The result of a Task Management function is indicated by codes hex’00’, hex’04’ and hex’05’ in the RSP_CODE of the FCP_RSP_INFO field and shall be as shown in table 12.

Table 12 – Task Management Function RSP_CODES

RSP_CODE	Meaning
00	No Failure or Task Management function complete
01	FCP_DATA length different than BURST_LEN
02	FCP_CMND Fields Invalid
03	FCP_DATA RO mismatch with FCP_XFER_RDY DATA_RO
04	Task management function was not performed because the function is unsupported
05	Task management function was not performed, but the function is supported.
06-FF	Reserved

8.3 Task Management Flags and Information Units

All SCSI Initiators shall send profile supported Task Management functions using T1. All SCSI Targets shall return FCP_RSP to Task Management functions using I4. The RSP_CODE in the FCP_RSP_INFO field shall indicate the result of the Task Management function. The SCSI Status byte and FCP_SNS_INFO shall be ignored for I4 information units sent in response to a Task Management function.

Table 13 lists FCP Task Management functions with usage defined by this document. Only one Task Management function shall be selected at a time.

Table 13 – FCP Task Management Flags

Feature	SCSI Initiator	SCSI Target	Notes
Terminate Task = 0	R	R	
Clear ACA = 1	note 1	note 1	
Target Reset = 1	I	R	2
Clear Task Set = 1	I	R	2
Abort Task Set = 1	I	R	2

Notes:

1 Clear ACA is only required by the SCSI Target if it indicates support of ACA via the NormACA bit in the INQUIRY data. Clear ACA is required by SCSI Initiators which support ACA (NACA bit in CDB=1) when communicating with SCSI Targets which have indicated support of ACA.

2 See 9.3 for SCSI Initiator requirements following transmission of these Task Management functions.

8.4 FCP Task Attributes

Table 14 lists FCP Task Attributes with usage defined by this document.

Table 14 – FCP Task Attributes

Feature	SCSI Initiator	SCSI Disk Target	SCSI Tape Target	Notes
Untagged				
Block (disk) devices	P	P		
Stream (tape) devices	R		R	
Simple Queue Type	R	R	A	
Ordered Queue Type	I	R	A	
Head of Queue Type	I	R	A	
Auto Contingent Allegiance Type	I	R	R	1

Notes:

1 ACA Type is only required by the SCSI Target if it advertises support of ACA via the NormACA bit in the INQUIRY data. ACA Type is required by SCSI Initiators which use ACA (NACA bit in CDB=1).

8.5 Other FCP features

Table 15 lists Other FCP features with usage defined by this document.

Table 15 – Other FCP Features

Feature	SCSI Initiator	SCSI Target	Notes
FCP_LUN	R	R	
FCP_LUN (0)	I	R	
INQUIRY of FCP_LUN (0)	I	R	
INQUIRY of FCP_LUN (>0)	I	A	1
Process Login	R	R	
Process Logout	R	R	2

Notes:

1 Some SCSI Targets will require multiple LUNs (e.g., disks over 8 Gbyte with 512 byte block size on some OS’s necessitates more than 1 LUN).

2 PRLO may be issued following receipt of a frame from a device which has not performed PRLI as a notification. A PRLI has the effect of implicit PRLO.

8.6 FCP Sequence delivery confirmation

Since this document describes operations in a Class 3 environment, there are no Acknowledgments to provide direct confirmation of Sequence delivery. In many cases, Sequence delivery can be inferred following transmission of a SCSI Initiator to SCSI Target Sequence which transfers Sequence Initiative. The following cases provide implicit confirmation of Sequence delivery.

a) Receipt of the following confirms that the FCP_CMND IU (T1) was delivered:

1) Receipt of a Write XFER_RDY IU (I1) following transmission of an FCP_CMND IU containing a write type command

2) Receipt of a Read FCP_DATA IU (I3) following transmission of an FCP_CMND IU containing a read type command

3) Receipt of FCP_RSP IU (I4) following transmission of an FCP_CMND IU containing any type of command

b) Receipt of the following confirms that a write FCP_DATA IU (T6) was delivered:

1) Receipt of a Write XFER_RDY IU (I1)

2) Receipt of FCP_RSP IU (I4)

c) Delivery of the following Information Units can not be determined directly (delivery failure is detected by a timeout condition or other mechanism):

1) Read FCP_DATA IU (I3)

2) FCP_RSP IU (I4)

9 Error detection and recovery

This clause describes the error actions to be taken by the SCSI Initiator and SCSI Target upon detection of an error condition.

NOTE – For SCSI Stream Devices, especially buffered tape devices, error recovery operations and commands may be incomplete at this time and further development of these clauses may be required.

9.1 Error detection by SCSI Initiator

The ABTS protocol (described in 9.3) shall be invoked by a SCSI Initiator when any of the following conditions occur:

a) A Sequence error is detected (see 5.8.4 for a list of Sequence error conditions)

b) ULP_TOV has expired and an Exchange has not completed (see 5.8 for conditions indicating Exchange completion).

The description of Loop Initialization (see 6.3) contains other requirements for Loop specific error recovery.

9.2 Error detection by SCSI Target

When a SCSI Target detects a Sequence error, it shall discard that Sequence, and all remaining Sequences for the Exchange containing the Sequence in error. A Target may attempt to send an FCP_RSP for the Exchange with the Sequence error in order to send SCSI Status and Sense information, if the Target receives Sequence Initiative for that Exchange. If no explicit notification is provided to the SCSI Initiator the Exchange eventually times out when ULP_TOV expires (see 9.4).

| SCSI Targets may implement RR_TOV as described in clause 7.6 to facilitate recovery of resources allocated to a non-respondent SCSI Initiator. If a SCSI Target implements an RR_TOV timer for this purpose, it may send a LOGO to the SCSI Initiator and terminate all open Exchanges for that SCSI Initiator upon detection of the following:

a) The SCSI Initiator has failed to perform SCSI Target Authentication within RR_TOV

b) The SCSI Initiator has failed to transmit the next expected Sequence of an Exchange for all open Exchanges with a specific SCSI Initiator within RR_TOV (e.g., FCP write data in response to an FCP_XFER_RDY)

SCSI Targets compliant with this document are not permitted to invoke the ABTS protocol.

9.3 Disk error recovery using ABTS protocol

The ABTS-Last Sequence of Exchange (ABTS-LS) protocol is required, which uses ABTS to abort entire Exchanges. The unit of error recovery for this document is an Exchange, not a Sequence. The protocol defining retransmission of Exchanges following ABTS is beyond the scope of this document.

9.3.1 SCSI Initiator ABTS behavior

Only a SCSI Initiator may transmit ABTS. This document does not define the protocol by which multiple SCSI Initiators communicate or synchronize shared peripherals.

ABTS may be transmitted even if Sequence Initiative is not held. Following the transmission of ABTS, any Device_Data frames received for this Exchange shall be discarded until the BA_ACC with “Last Sequence of Exchange” bit is received from the SCSI Target.

ABTS may not take effect immediately. For example, if ABTS is sent following transmission of a Read command, the SCSI Target may send back all or some of the requested read data before replying with the BA_ACC to the ABTS (or the data may already be in flight at the time the ABTS was sent). The SCSI Initiator must receive this data and provide BB_Credit in order for the SCSI Target to send the BA_ACC.

There are circumstances where a configuration change event could occur very quickly, which results in link errors and a ULP_TOV timeout, but no LIP is generated to reconfigure the loop. A SCSI Initiator shall attempt the ABTS protocol following ULP_TOV timeout.

The ABTS protocol shall be invoked:

a) for all open Exchanges for all LUNs on a designated SCSI Target:

1) following receipt of “Function Complete” to SCSI Target Reset

2) following receipt of CHECK CONDITION status with Sense Key=Unit Attention and ASC/ASCQ of

i) “Power on, reset, or bus device reset occurred” (29 XX)

ii) Microcode changed (3F 01)

b) for all open Exchanges for the specified LUN on a designated SCSI Target:

1) following receipt of “Function Complete” to Abort Task Set

2) following receipt of “Function Complete” to Clear Task Set

3) upon receipt of CHECK CONDITION status with Sense Key=Unit Attention and ASC/ASCQ of “Command cleared by another Initiator” (2F 00)

c) for a specific Exchange on a specified LUN on a designated SCSI Target:

1) upon ULP_TOV timer expiration (no STATUS returned for a Command within an application-specified amount of time)

2) upon recognition of any of the Sequence errors described in clause 5.8.4

Following receipt of the BA_ACC in response to an ABTS, and after $R\_A\_TOV_{SEQ\_QUAL}$ has elapsed, the SCSI Initiator shall transmit RRQ.

If a proper BA_ACC, BA_RJT, LOGO, or PRLO is not received from the SCSI Target within E_D_TOV, second level error recovery as described in clause 9.3.3. shall be performed.

9.3.2 SCSI Target ABTS behavior

When ABTS is received at the SCSI Target, it shall abort the designated Exchange and return one of the following responses:

a) the SCSI Target shall discard the ABTS and return LOGO if the NL_Port issuing the ABTS is not currently logged in (no PLOGI)

b) the SCSI Target shall return BA_RJT with Last Sequence of Exchange bit set to one if the received ABTS contains an RX_ID, other than ‘FFFF’, which is unknown to the SCSI Target. The reason code shall be “Logical Error” with a reason code explanation of “Invalid OX_ID/RX_ID combination”

c) otherwise, the SCSI Target shall return BA_ACC with Last Sequence of Exchange bit set to one

Upon transmission of any of the above responses, the SCSI Target may reclaim any resources associated with the designated Exchange after $R\_A\_TOV_{SEQ\_QUAL}$ has elapsed or a Reinstate Recover Qualifier (RRQ) extended link service request has been received.

When RRQ is received at the SCSI Target, it shall respond according to the same rules stated for the ABTS. If the Exchange resources were not reclaimed upon responding to the ABTS, they shall be reclaimed at the time the response to the RRQ is sent.

SCSI Targets shall qualify ABTS based only upon D_DI || S_ID || OX_ID, not RX_ID, since the RX_ID is not guaranteed to be known by a SCSI Initiator.

9.3.3 Second-level error recovery

If a response to the ABTS is not received within E_D_TOV, the SCSI Initiator shall:

a) send the ABTS again

b) if a response to the second ABTS is not received within E_D_TOV, the SCSI Initiator shall explicitly logout (FC-PH N_Port Logout, LOGO) the SCSI Target and may issue the Selective Reset LIP (LIP,AL_PD,AL_PS) to reset the SCSI Target. All outstanding Exchanges with that SCSI Target are terminated at the SCSI Initiator

9.3.4 Abort Sequence (ABTS) frame

Only the SCSI Initiator shall transmit an ABTS frame. When it does so, the specified fields shall be set as shown in figure 4.

Figure 4 – ABTS Frame

	Field	Sub-field	Content
Frame Header	F_CTL	Sequence Context	Initiator (even though the ABTS initiator may not have Sequence Initiative for the Sequence being aborted).
		Sequence Initiative	Transferred, even if the ABTS initiator did not hold Sequence Initiative prior to the ABTS.
	SEQ_ID		If ABTS is sent by the Sequence Initiator and the Sequence is still open, the SEQ_ID of the open Sequence in the Exchange being aborted is used. Otherwise, the SEQ_ID is any SEQ_ID not currently open (for any Exchange) between that pair of ports.
	SEQ_CNT		SEQ_CNT of last frame transmitted in an Open Sequence + 1. If no Sequence is open, then SEQ_CNT=zero.
	OX_ID		OX_ID = same as that assigned by the Exchange Originator (SCSI Initiator) for the Exchange being aborted.
	RX_ID		Set to hex’FFFF’ or the same as that assigned by the Exchange Responder (SCSI Target) for the task being aborted.

9.3.5 Basic Accept (BA_ACC) frame to ABTS

A SCSI Target may accept ABTS with BA_ACC. When it does so, the BA_ACC shall be as shown in figure 5.

9.3.6 Basic Reject (BA_RJT) frame to ABTS

A SCSI Target may reject ABTS with BA_RJT. When it does so, the BA_RJT shall be as shown in figure 6.

Figure 5 – BA_ACC Frame to ABTS

	Field	Sub-field	Content
Frame Header	OX_ID		OX_ID from ABTS frame
	RX_ID		RX_ID from ABTS frame
	F_CTL	Last_Sequence	1
		Sequence Context	Recipient
Payload	SEQ_ID validity		hex'00'
	SEQ_ID byte		invalid (don't care)
	OX_ID		OX_ID from ABTS frame
	RX_ID		RX_ID from ABTS frame
	Lowest SEQ_CNT		hex'0000'
	Highest SEQ_CNT		hex'FFFF'

Figure 6 – BA_RJT Frame to ABTS

	Field	Sub-field	Content
Frame Header	OX_ID		OX_ID from ABTS frame
	RX_ID		RX_ID from ABTS frame
	F_CTL	Last_Sequence	1
		Sequence Context	Recipient
Payload	Byte 0	Reserved	hex'00'
	Byte 1	Reason Code	hex'03' (Logical error)
	Byte 2	Reason Explanation	hex'03' (Invalid OX_ID-RX_ID combination)
	Byte 3	Vendor Unique	hex'00'

9.3.7 Reinstate Recovery Qualifier (RRQ)

The format of the RRQ is shown in figure 7.

Figure 7 – Reinstate Recovery Qualifier

	Field	Content
Frame Header	OX_ID	Identifier of a new exchange
	RX_ID	hex'FFFF'
Payload	Originator S_ID	Source_ID of the SCSI Initiator
	OX_ID	OX_ID of XCHG that was previously aborted with ABTS
	RX_ID	RX_ID of XCHG that was previously aborted with ABTS

Following successful completion of the RRQ, the SCSI Target shall respond with ACC.

9.4 SCSI Target error behavior

If an error is detected by a SCSI Target while it has Sequence Initiative, the only permissible recovery action is the transmission of FCP_RSP with CHECK CONDITION status and an appropriate Sense Key/ASC/ASCQ.

If an error is detected by a SCSI Target while it does not have Sequence Initiative, it must wait until it has been given Sequence Initiative before it can return CHECK CONDITION.

If ULP resources in the SCSI Target are unavailable when a command is issued, the SCSI Target shall return TASK SET FULL status in FCP_RSP (i.e., shall not return BUSY).

If a SCSI Target receives a PLOGI request and it detects a limitation of login resources,

a) if there are no open tasks with another logged-in SCSI Initiator, that other SCSI Initiator may be implicitly logged out so that the PLOGI request can be processed.

b) if all logged-in SCSI Initiators have open tasks, the SCSI Target may respond to the PLOGI with LS_RJT and reason code “unable to perform command request” and reason code explanation “insufficient resources to support Login”.

9.5 Task Management and multiple-initiator SCSI Targets

If a SCSI Target Reset or Clear Task Set management function is received by a SCSI Target which has multiple SCSI Initiators logged in with it, then the SCSI Target shall:

a) create a Unit Attention Condition for all other SCSI Initiators (an FCP_RSP may have been transmitted but not received by the SCSI Initiator, or the SCSI Initiator may have transmitted a command which has not yet been received by the SCSI Target).

b) clear all resources associated with the cleared Exchanges, per SCSI Architectural Model (SAM).

c) return FCP_RSP upon completion of (a) and (b). The payload shall be zeroes with the exception of the FCP_RSP_LEN_VALID bit, FCP_RSP_LEN (which shall be set equal to 8), and the FCP_RSP_INFO.

Upon discovery of the Unit Attention Condition set in a), SCSI Initiators shall issue ABTS for all commands which are outstanding for the appropriate LUN or LUNs at that SCSI Target as described in 9.3.1. From a SCSI Initiator perspective, this is all commands for which FCP_RSP has not been received.

9.6 SCSI Target Exchange origination capability

The only time that a SCSI Target is required to have Exchange origination capability is to originate a LOGO or PRLO request. All other Exchanges are originated by the SCSI Initiator.

9.7 Responses to FCP-level frames before PLOGI or PRLI

If a SCSI Target receives an FCP_CMND from an NL_Port with which it has not successfully completed N_Port Login (PLOGI), it shall discard the FCP_CMND and send LOGO to that NL_Port. No Exchange is created in the SCSI Target for the discarded request, and the originator of the discarded request terminates the Exchange associated with the discarded request and any other open Exchanges for the SCSI Target sending the LOGO. The LOGO is not part of the Exchange associated with the discarded request.

If a SCSI Target receives an FCP_DATA Sequence from and NL_Port with which it has not successfully completed N_Port Login (PLOGI), it shall discard all frames of that Sequence and may send LOGO. If LOGO is sent, all open Exchanges for the NL_Port that sent the FCP_DATA Sequence shall be terminated.

If a SCSI Target receives an FCP_CMND from an NL_Port with which it has not successfully completed Process Login (PRLI), it shall discard the FCP_CMND and send PRLO to the SCSI Initiator. No Exchange is created in the recipient NL_Port for the discarded request, and the originator of the discarded request terminates the Exchange associated with the discarded request.

If a SCSI Target receives an FCP_DATA Sequence from a SCSI Initiator with which it has not successfully completed Process Login (PRLI), it shall discard all frames of that Sequence and may send PRLO.

For the action taken on any other received frame, see table 7.

10 SCSI features

This clause describes those SCSI features with usage defined by this document.

10.1 Auto Contingent Allegiance (ACA)

During the discovery process, FCP SCSI Initiators shall use INQUIRY to discover whether or not Logical Units support ACA.

ACA shall be used (NACA=1 in CDB) only when both the sending SCSI Initiator and receiving Logical Unit support ACA.

ACA shall not be used (NACA=0 in CDB) when either the sending SCSI Initiator or the receiving SCSI Target does not support ACA. If a SCSI Target receives a command with NACA=0 and that command fails, autosense data is cleared upon transmission of the FCP_RSP to the faulted SCSI Initiator and no ACA condition results.

10.2 SCSI Status

SCSI Targets are required to be able to return the following status values. Status values not listed are optional (not required to be returned by SCSI Targets, not relied upon by SCSI Initiators).

- ACA ACTIVE (if ACA is supported by both the SCSI Initiator and SCSI Target)
- RESERVATION CONFLICT
- BUSY
- GOOD
- CHECK CONDITION
- TASK SET FULL

10.3 SCSI Target Discovery

When the possibility of a configuration change exists, a SCSI Initiator may want to rediscover the new configuration. The SCSI Target Discovery procedure for a SCSI Initiator is:

For all valid AL_PAs:

```
OPN(AL_PA)
  IF OPN successful, THEN
    Send ADISC or PDISC to D_ID = hex'0000' || AL_PA
    IF (LOGO returned or Node Name or Port Name changed) THEN
      Send PLOGI to D_ID = hex'0000' || AL_PA
      IF PLOGI successful THEN
        IF (no hard address conflicts or application tolerant of hard address conflicts)
          Send PRLI to D_ID = hex'0000' || AL_PA
            IF PRLI successful, THEN
              Send FCP_CMND with INQUIRY CDB to D_ID = hex'0000' || AL_PA(LUN_0)
            ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
NEXT AL_PA
```

In order to determine whether or not an OPN was “successful” in the above procedure, an NL_Port must be able to either:

a) detect when an OPN has not been intercepted by the designated AL_PA

b) detect that an R_RDY or CLS has not been received from the AL_PA specified in an OPN within E_D_TOV of sending that OPN

c) detect that a CLS was received in response to the OPN. In this case, the Target Discovery procedure should be retried at a later time.

If the SCSI Target Discovery procedure revealed that there was a Hard Address conflict (i.e., some NL_Port was unable to acquire its Hard Address), then the application may choose to operate in spite of that conflict. If this is the case, then the discovery procedure can continue with the PRLI and subsequent SCSI INQUIRY command. If the application is not tolerant of Hard Address conflicts, the SCSI Initiator may choose not to use that NL_Port.

Through the use of the SCSI Target Discovery procedure, the SCSI Initiator has the ability to assemble a database of [AL_PA:Node Name:Port Name:N_Port ID] tuples for all responding AL_PAs. There are several confirmations a SCSI Initiator can perform on that database to determine which SCSI Targets it can continue to communicate with which are not defined by this document. See 10.4 for rules regarding completion of open Exchanges.

NOTE – Not all initiators may perform the exact steps described in the above algorithm, although a SCSI Initiator is required to issue ADISC or PDISC to all SCSI Targets it is logged in with within RR_TOV of receiving LIP if it wants to remain logged in with those SCSI Targets.

The ADISC/PDISC procedure is designed to avoid the abnormal termination of all open Exchanges when a new device is inserted into the loop, or when a device powers on.

NOTE – Since devices are not required to respond to Class 3 frames which have a D_ID which does not match the full 24-bit N_Port identifier of the receiving NL_Port, this may result in timeouts during the SCSI Target discovery process if a SCSI Initiator sends a frame to a Public NL_Port using a D_ID of hex'0000' || AL_PA or to a Private NL_Port using a D_ID with the upper 16 bits non-zero. Therefore, for performance reasons SCSI Initiators should originate PDISC or ADISC Exchanges by transmitting the ELS Sequence without waiting for the response. SCSI Initiators may need to originate multiple concurrent Exchanges in order to hide multiple timeouts from the user.

10.4 Exchange Authentication following LIP

If the node is preparing to change the login parameters of any NL_Port, the node shall perform explicit logout with all NL_Ports currently logged-in with that NL_Port before allowing the new parameters to take effect. If there are open and active Sequences with the NL_Ports to be logged out, the device shall discard any frames received in those Sequences and send a LOGO, if one has not already been sent.

10.4.1 SCSI Initiator Exchange Authentication

Following transmission of the CLS signalling completion of loop initialization, SCSI Initiators are required to authenticate each SCSI Target with which they have completed PLOGI.

a) A SCSI Initiator shall suspend execution of all open tasks with a SCSI Target following loop initialization until authentication is complete. Following transmission or forwarding of a LIP, only frames related to SCSI Target authentication (i.e., ADISC, PDISC, or PLOGI) shall be transmitted to that SCSI Target before frames for other exchanges to that SCSI Target are transmitted.

b) For each SCSI Target with which it has successfully complete PLOGI, a SCSI Initiator shall originate ADISC or PDISC to that SCSI Target such that the ADISC or PDISC request Sequence arrives at each SCSI Target within RR_TOV of completing loop initialization.

c) For every SCSI Target that returns an ACC, if the [N_Port ID:Port Name:Node Name] triplet for that SCSI Target:

1) does not match the triplet of a logged-in SCSI Target, the SCSI Initiator shall transmit LOGO to that SCSI Target.

2) does match the triplet of a logged-in SCSI Target, the SCSI Initiator may resume all tasks with that SCSI Target.

d) SCSI Initiator Exchange authentication is complete when the ACC to an ADISC or PDISC has been received. If the SCSI Initiator fails to receive an ACC within R_A_TOV, it shall implicitly logout the SCSI Target and terminate all tasks for that SCSI Target.

e) Following loop initialization, any frames received (other than ACC to an ADISC, PDISC, or PLOGI, or a LOGO request) from a SCSI Target which has not been authenticated shall be discarded. This means they shall not be considered part of any valid Exchange, and only R_RDYs to maintain BB_Credit shall be sent in response to these frames.

f) If a SCSI Initiator receives a LOGO from an NL_Port during this procedure, it shall terminate all open Exchanges with that NL_Port and send an ACC.

10.4.2 SCSI Target Exchange Authentication

Following completion of loop initialization, SCSI Targets are required to wait for exchange authentication by the SCSI Initiators with which they have completed PLOGI.

a) For each SCSI Initiator that has open tasks, a SCSI Target shall suspend tasks associated with that SCSI Initiator until an ADISC or PDISC is received from that SCSI Initiator. Following transmission or forwarding of a LIP, only frames related to SCSI Initiator authentication (i.e., ACC to an ADISC or PDISC) shall be transmitted to that SCSI Initiator before frames for other exchanges to that SCSI Initiator are transmitted.

b) For every SCSI Initiator which sends an ADISC or PDISC, if the [N_Port_ID:Port_Name:Node_Name] triplet for that SCSI Initiator:

1) does not match the triplet of a logged-in SCSI Initiator, the SCSI Target shall transmit LOGO to that SCSI Initiator and terminate all Exchanges associated with that SCSI Initiator.

2) does match the triplet of a logged-in SCSI Initiator, the SCSI Target shall send the ACC to the extended link service. The SCSI Target may then resume all tasks with that SCSI Initiator.

c) If a SCSI Target does not receive a PDISC or ADISC from each logged-in SCSI Initiator within RR_TOV of completing loop initialization, then it may implicitly logout that SCSI Initiator and terminate all Exchanges associated with that SCSI Initiator.

d) SCSI Target authentication is complete when the ACC to an ADISC or PDISC has been transmitted.

e) Following loop initialization, any frames received (other than ADISC, PDISC, or PLOGI) from a SCSI Initiator that has not been authenticated shall be discarded. Only R_RDYs to maintain BB_Credit shall be sent in response to those frames.

10.5 Clearing effects of ULP, FCP, FC-PH, and FC-AL actions

Table 16 summarizes which FCP SCSI Target objects are cleared as a result of a listed FCP SCSI Initiator action. A 'Y' in the corresponding column indicates the object is cleared to its default or power-on value within the device upon completion of specified action. An 'N' in the corresponding column indicates the object is not affected by the specified action. A '-' in the column indicates that the action is not applicable.

Table 16 – Clearing Effects of SCSI Initiator Actions

	FCP SCSI Initiator Action Clears Object								
FCP SCSI Target Object	**Power Cycle**	**Reset LIP(y,x)[3]**	**LOGO, PLOGI**	**ABTS w/Last Seq.**	**PRLI, PRLO**	**TPRLO[5]**	**SCSI Target Reset**	**Clear Task Set[4]**	**Abort Task Set[4]**
PLOGI parameters									
For all logged-in SCSI Initiator ports	Y	Y	N	N	N	N	N	N	N
Only for SCSI Initiator port initiating action	-	-	Y	N	N	N	N	N	N
Open Sequences Terminated									
For all SCSI Initiator ports with open Sequences	Y	Y	N	N	N	Y	Y[2]	N	N
Only for SCSI Initiator port initiating action	-	-	Y	Y	Y	-	-	Y[2]	Y[2]
BB_Credit_CNT	Y	Y	Y	N	N	N	N	N	N
Hard Address Acquisition Attempted	Y[1]	Y[1]	N	N	N	N	N	N	N
PRLI parameters cleared									
For all logged-in SCSI Initiator ports	Y	Y	N	N	N	N	N	N	N
Only for ports of specified TYPE	-	-	N	N	Y	Y	N	N	N
Only for port initiating action	-	-	Y	N	Y	N	N	N	N
Open Tasks (Exchanges) Aborted									
All tasks for all SCSI Initiator ports with open tasks	Y	Y	N	N	N	Y	Y	Y	N
All tasks, only for port initiating action	-	-	Y	N	Y	-	-	-	Y
Only specified task for port initiating action	-	-	N	Y	N	-	-	-	N
SCSI Target mode page parameters restored from saved pages									
For all SCSI Initiators	Y	Y	N	N	N	Y	Y	N	N
Only for port initiating action	-	-	Y	N	Y	-	-	N	N
Pre-existing UA condition cleared									
For all SCSI Initiators	Y	Y	N	N	N	Y	Y	N	N
Only for port initiating action	-	-	Y	N	Y	-	-	N	N
Pre-existing ACA condition cleared									
For all SCSI Initiators	Y	Y	N	N	N	Y	Y	N	N
Only for port initiating action	-	-	Y	N	Y	-	-	N	N

Notes:

1 If the NL_Port has an AL_PA different that its hard address and the NL_Port experiences a power cycle or recognizes LIP(AL_PD,AL_PS), the NL_Port shall relinquish its current AL_PA and attempt to acquire its hard address.

2 Tasks are cleared internally within the SCSI Target, but open Sequences must be individually aborted by the SCSI Initiator via the ABTS_LS protocol which also has the effect of aborting the associated Exchange.

3 This is also known as LIP(AL_PD,AL_PS). If the destination recognizes a selective hard reset LIP where the AL_PD matches the AL_PA of the receiving NL_Port, the receiving NL_Port shall perform the behavior described in this column.

4 For multiple-LUN SCSI Targets, Clear Task Set, and Abort Task set affect only the addressed LUN, not the entire SCSI Target.

5 Actions listed shall be performed when the Global bit = '1'b. If the Global bit = '0'b, then the actions listed under PRLI/PRLO should be performed for the designated SCSI Initiator.

11 SCSI-3 block devices

This clause describes behaviors and characteristics applicable to SCSI-3 block devices.

11.1 Applicable Classes of Service

SCSI-3 block devices conforming to this document shall use Class 3 service with parameters as described in Table 3.

11.2 Asynchronous Event Notification (AEN)

The use of Asynchronous Event Notification by SCSI block type devices is prohibited.

11.3 Command Linking

Command Linking is Prohibited by all SCSI Initiators communicating with block devices conforming to this document. The Link and Flag bits of the CDB shall be set to zeroes.

11.4 Disk device commands

Commands and features within commands which are not listed are optional. Interoperability between SCSI Initiators and SCSI Targets is not guaranteed if optional commands or features are used. If support of a field is listed as Required without specifying a value for that field, it is assumed that all valid values of the field must be supported as specified by the appropriate ANSI standard. Any unlisted commands/features/settings are implicitly Invokeable by the SCSI Initiator, and Allowed by the SCSI Target.

Table 17 – SCSI Disk Device Commands

Feature	Initiator	Target	Doc	Notes
FORMAT UNIT FmtData CmpLst Defect list format Interleave 0 0 ‘000’b 0	I	R	SBC	
INQUIRY	I	R	SPC	
Standard INQUIRY data (bytes 0-35)	I	R		
EVPD=1	I	R		
Vital Product Data page codes:				
hex‘00’ (Supported vital product pages)	I	R		
hex‘80’ (Unit serial number page)	I	R		
hex‘81’ (Implemented operations definition page)	I	A		
hex‘82’ (ASCII implemented operations definition page)	I	A		
hex‘83’ (Device identification page)	I	R		
MODE SELECT(10)	I	R	SPC / SBC	
PF=1	R	R		
SP=1	I	A		
MODE SENSE(10)	I	R	SPC / SBC	
DBD=0	I	R		
PC = ‘00’b	I	R		
PC = ‘01’b	I	R		
PC = ‘10’b	I	R		
PC = ‘11’b	I	A		
PERSISTENT RESERVE IN/PERSISTENT RESERVE OUT	I	A	SPC	
READ(6)	I	A	SBC	

Table 17 – SCSI Disk Device Commands

Feature	Initiator	Target	Doc	Notes
READ(10)	I	R	SBC	
DPO = 0	I	R		
DPO = 1	I	A		
FUA = 0	I	R		
FUA = 1	I	A		
RelAdr = 0	I	R		
RelAdr = 1	I	A		
READ CAPACITY	I	R	SBC	
RelAdr = 0	I	R		
RelAdr = 1	I	A		
PMI = 0	I	R		
PMI = 1	I	A		
READ DEFECT DATA	I	R	SBC	
PLIST | GLIST = ‘00’b	I	R		
PLIST | GLIST = ‘01’b	I	R		
PLIST | GLIST = ‘10’b	I	R		
PLIST | GLIST = ‘11’b	I	R		
Defect List Format = ‘000’b	R	R		
REPORT LUN	I	A	SPC	
RESERVE(6)/RELEASE(6)	I	A	SBC	
RESERVE(10)/RELEASE(10)	I	R	SBC	
Extent=1	I	A		
Extent=0	R	R		
3rdPty=1	I	A		
REQUEST SENSE	I	R	SPC	
SEND DIAGNOSTIC	I	R	SPC	
SelfTest=1	I	R		
SelfTest=0	I	A		
START/STOP UNIT	I	R	SBC	
Immed = 0	I	R		
Immed = 1	I	R		
LEOJ = 0	I	A		
LEOJ = 1	I	A		
Start = 0	I	R		
Start = 1	I	R		
TEST UNIT READY	I	R	SPC	
WRITE(6)	I	A	SBC	

Table 17 – SCSI Disk Device Commands

Feature	Initiator	Target	Doc	Notes
WRITE(10)	I	R	SBC	
DPO = 0	I	R		
DPO = 1	I	A		
FUA = 0	I	R		
FUA = 1	I	A		
RelAdr = 0	R	R		
RelAdr = 1	P	A		
WRITE BUFFER	I	R	SPC	
Mode = ‘100’b	I	A		
Mode = ‘101’b	I	R		
All other modes	P	A		

11.4.1 Mode Select/Sense (10) parameters (direct access)

Table 18 lists Mode Select(10) and Mode Sense(10) command parameters with usage defined by this document. The following legend is used for entries in this table.

‘R’ A SCSI Target must be able to return a value for the parameter in a MODE SENSE(10) command. If changeable, a SCSI Target must be able to accept a different value for this parameter in a MODE SELECT(10) command.

‘A’ A SCSI Target may be able to accept changes to this parameter in a MODE SELECT(10) command if changeable, but must be able to return a value for it in a MODE SENSE(10) command. SCSI Initiators are prohibited from relying on SCSI Target support of these features. All unlisted features implicitly fall into this category.

Table 18 – Disk Mode Select/Sense Parameters

Parameter	SCSI Target	Notes
Mode Parameter Header (10)	R	
Medium Type per SBC	R	
WP=1	A	
DPOFUA=1	A	
Block Descriptors		
Number returned by SCSI Target=1	R	
Block length an integer multiple of 2 bytes	R	5
Control Mode Page		
RLEC=0	R	
Queue Algorithm Modifier=0	R	
Queue Algorithm Modifier=1	A	
Qerr=0	R	
Dque=0	R	
RAC=0	R	
RAC=1	A	
EECA (from SCSI-2)=0	R	
RAERP/UAAERP/EAERP=’000’b	R	
Ready AER holdoff period=0	R	
Busy timeout period = hex’FFFF’	R	
Busy timeout period >hex ’0000’	A	

Table 18 – Disk Mode Select/Sense Parameters

Parameter	SCSI Target	Notes
Disconnect-reconnect Page	R	
Buffer Full/Empty Ratio	R	1
Maximum Burst Size > 0	R	2
Maximum Burst Size = 0	A	
Connect Time Limit = 0	R	3
Bus Inactivity Limit = 0	R	
Disconnect time limit = 0	R	
Enable Modify Data Pointer (EMDP) bit=0	R	
Enable Modify Data Pointer (EMDP) bit=1	A	4
DImm = 0	R	
DTDC = ‘000’b	R	
DTDC = ‘001’b	P	
DTDC = ‘011’b	A	

Notes:

1 Example: consider a SCSI Target with ten 512-byte buffers and a buffer full ratio of 3Fh (63 decimal). The formula is:

$$\mathrm{INT}\left[\frac{\textit{buffer full ratio}}{256} \times \textit{number of buffers}\right] = \mathrm{INT}\left[\frac{63}{256} \times 10\right] = 2$$

Which means that on read operations, the SCSI Target should attempt to arbitrate for the loop and transmit data when 2 or more buffers are full.

2 Regardless of whether or not FCP_XFER_RDY is used, the Maximum Burst Size corresponds to the maximum FCP_Data Sequence length (in bytes) sent by a SCSI Initiator or SCSI Target. The Maximum burst size is a multiple of 512 bytes with a value of one representing 512 bytes, two means 1024 bytes, etc. More than one Sequence may be sent in a single loop circuit. More than one loop circuit may be used to transmit a single Sequence.

3 Connect time limit corresponds to the maximum time that the SCSI Target’s NL_Port may keep the loop. The SCSI Target’s NL_Port shall send CLS, and subsequently relinquish control of the loop and return to the MONITORING state, no later than this time limit after transmitting each OPNy. A value of zero means there is no time limit.

4 EMDP=1 indicates that Sequences may be sent with offset values in any order. EMDP=0 indicates Sequences shall be sent with offset values in ascending offset order. This bit does not affect the order of frames within a Sequence. SCSI Targets which can return FCP_XFER_RDY DATA_RO parameters which are not continuously increasing within an Exchange implicitly support EMDP=1.

5 Devices conforming to this document are required to start Sequences on boundaries that are multiples of 8 bytes. If the device block length is not an integer multiple of 8 bytes, then multiple Sequence data transfer operations must insure that each data sequence, except the last, transfers a multiple of 8 bytes.

12 SCSI Stream Devices

This clause describes commands and features applicable to SCSI stream devices with usage defined by this profile.

NOTE – For SCSI Stream Devices, especially buffered tape devices, error recovery operations and commands may be incomplete at this time and further development of these clauses may be required.

12.1 Applicable Classes of Service

SCSI-3 stream devices conforming to this document shall use Class 3 service with parameters as described in Table 3.

12.2 Asynchronous Event Notification (AEN)

The use of AEN by stream devices conforming to this profile is prohibited.

12.3 Command Linking

Command Linking is allowed by all stream devices conforming to this profile. Support for command linking is identified in the Inquiry data. The Flag bit of the CDB shall be set to zero.

12.4 Sequential device commands

Commands and features within commands which are not listed are optional. Interoperability between SCSI Initiators and SCSI Targets is not guaranteed if optional commands or features are used. If support of a field is listed as Required without specifying a value for that field, it is assumed that all possible values of the bit or field must be supported per the appropriate ANSI standard. Any unlisted commands/features/settings are implicitly Invokeable by the SCSI Initiator, and Allowed by the SCSI Target.

Table 19 – SCSI Tape Device Commands

Feature	Initiator	Target	Doc	Notes
ERASE	I	R	SSC	
IMMED	I	R		
LONG	I	R		
SHORT	I	A		
FORMAT MEDIUM	I	A	SSC	
INITIALIZE ELEMENT STATUS	I	A	SMC	2
INQUIRY	I	R	SPC	
Standard INQUIRY data (bytes 0-35)	I	R		
EVPD=1	I	R		
Vital Product Data page codes:				
hex‘00’ (Supported vital product pages)	I	R		
hex‘80’ (Unit serial number page)				
hex‘81’ (Implemented operations definition page)				
hex‘82’ (ASCII implemented operations definition page)				
hex‘83’ (Device identification page)	I	A		
LOAD UNLOAD	I	R	SSC	
EOT = 1	I	A		
Immed = 1	I	R		
Load = 1	I	A		
Load = 0	I	R		
Reten = 1	I	A		

Table 19 – SCSI Tape Device Commands

Feature	Initiator	Target	Doc	Notes
LOCATE	I	R		
BT = 0	I	R		
BT = 1	I	A		
CP = 0	I	R	SSC	
CP = 1	I	A		
Immed=0	I	R		
Immed=1	I	R		
LOG SELECT	I	A		
SP = 0	I	R		
SP = 1	I	A		
PCR = 1 & PLL = 0	I	R	SPC	
PC = ‘00’b	I	A		
PC = ‘01’b	I	R		
PC = ‘10’b	I	A		
PC = ‘11’b	I	A		
LOG SENSE	I	R		
SP = 0	I	R		
SP = 1	I	A		
PPC = 0	I	R		
PPC = 1	I	A		
PC = ‘00’b	I	A		
PC = ‘01’b	I	R	SPC	
PC = ‘10’b	I	A		
PC = ‘11’b	I	A		
Page Select 00	I	R		
Page Select 02	I	R		
Page Select 03	I	R		
Page Select 06	I	R		
Page Select 0C	I	R		
MODE SENSE/MODE SELECT (6 and 10)	I	R	SPC	
DBD = 0	I	R		
DBD = 1	I	R		
PC = ‘00’b	I	A		
PC = ‘01’b	I	R		
PC = ‘10’b	I	A		
PC = ‘11’b	I	A		
Page 01	I	R	SPC	
Page 02	I	R	SPC	
Page 0A	I	R	SSC	
Page 10	I	R	SSC	
Pages 11-14	I	A	SSC	
Media changer specific pages:	I	A		2
Page hex‘1D’	I	R	SMC	
Page hex‘1F’	I	R		
MOVE MEDIUM	I	A	SMC	2
PERSISTENT RESERVE IN/PERSISTENT RESERVE OUT (when SCSI Target supports multiple ports)	I	R	SPC	

Table 19 – SCSI Tape Device Commands

Feature	Initiator	Target	Doc	Notes
PREVENT/ALLOW MEDIUM REMOVAL	I	A	SSC	
READ	I	R	SSC	
Fixed = 1	I	R		
Fixed = 0	I	A		
SILI = 0	I	R		
SILI = 1 (Required if RBL values differ)	I	A		
READ BLOCK LIMITS	I	R	SSC	
READ BUFFER	I	A	SPC	
READ ELEMENT STATUS	I	A	SMC	
Vol Tag = 0	I	R		
Vol Tag = 1	I	A		
READ POSITION	I	R	SSC	
BT = 0	I	R		
BT = 1	I	A		
READ REVERSE	I	A	SSC	
Fixed = 1	I	R		
Fixed = 0 (Required if RBL values differ)	I	A		
SILI = 0	I	R		
SILI = 1 (Required if RBL values differ)	I	A		
Byte Ord = 0	I	A		
Byte Ord = 1	I	A		
RECEIVE DIAGNOSTIC RESULTS	I	A	SPC	
RECOVER BUFFERED DATA	I	R	SSC	
Fixed = 1	I	R		
Fixed = 0 (Required if RBL values differ)	I	A		
SILI = 0	I	R		
SILI = 1 (Required if RBL values differ)	I	A		
FIFO	I	A		
LIFO	I	R		
RELEASE(6)	I	A	SPC	
3rd Party = 0	I	R		
3rd Party = 1	P	P		
RELEASE(10)	I	R	SPC	
3rd Party = 0	I	R		
3rd Party = 1	I	A		
REPORT DENSITY SUPPORT	I	R	SSC	
REPORT LUN	I	A	SPC	
RESERVE(6)	I	A	SPC	
3rd Party = 0	I	R		
3rd Party = 1	P	P		
RESERVE(10)	I	R	SPC	
3rd Party = 0	I	R		
3rd Party = 1	I	A		
REWIND	I	R	SSC	
Immed = 0	I	R		
Immed = 1	I	R		

Table 19 – SCSI Tape Device Commands

Feature	Initiator	Target	Doc	Notes
SEND DIAGNOSTIC	I	R	SPC	
ST = 1	I	R		
ST = 0	I	A		
SPACE	I	R	SSC	
Forward	I	R		
Reverse	I	R		
Code = ‘000’b	I	R		
Code = ‘001’b	I	R		
Code = ‘010’b	I	A		
Code = ‘011’b	I	R		
Code = ‘100’b	I	A		
Code = ‘101’b	I	A		
TEST UNIT READY	I	R	SPC	
VERIFY	I	A	SSC	
Fixed = 1	I	R		
Fixed = 0 (Required if RBL values differ)	I	A		
BytCmp = 0	I	R		
BytCmp = 1	I	A		
Immed = 0	I	R		
Immed = 1 & BytCmp = 0	I	R		
Immed = 1 & BytCmp = 1	I	A		
WRITE	I	R	SSC	
Fixed = 1	I	R		
Fixed = 0	I	A		
WRITE BUFFER	I	A	SPC	
Mode = ‘110’b (Download ucode w/ offsets)	I	R		
Mode = ‘111’b (Download ucode w/ offsets & save)	I	R		
WRITE FILEMARK	I	R	SSC	
Count = 0	I	R		
Count > 0	I	R		
Immed	I	R		

1 FCP requires the use of Autosense

2 Medium changer commands and features are only required when the MCHGR field in Inquiry data is set to ‘1’b

Annex A
(informative)

Login_BB_Credit Examples

Available BB_Credit: Port as Sequence Initiator has permission to send up to this number of full length frames.

Min_available_RX_buffers: Port as Sequence Recipient has room to receive at least this number of full length frames.

In the tables below, these credit variables are modified following the actions described in the columns labelled OPN Initiator X and OPN Recipient Y.

A.1 Open half-duplex, Login_BB_Credit = 0

Table A.1 – Open Half Duplex (OPNyy), Login_BB_Credit=0

Step	OPN Originator X	Avail BB_ Credit	Min_available RX_buffers	OPN Recipient Y	Avail BB_ Credit	Min_available RX_buffers
A	Login_BB_Credit = 00	0	N/A (1/2 dup)	Login_BB_Credit = 00	0	0
B	Tx OPN Half Duplex Tx 0 to N R_RDYs (note 1)	0	N/A (1/2 dup)			
C				Rx OPN Half Duplex Tx 2 R_RDYs	0 0	0 2
D	Rx 2 R_RDYs Tx 2 Frames Tx CLS	2 0 0	N/A (1/2 dup) N/A (1/2 dup) N/A (1/2 dup)			
E				Rx 2 frames (note 2) Rx CLS Tx CLS	0 0 0	0 0 0
F	Rx CLS	0	N/A		0	0

Notes

1 The OPN Originator may transmit one R_RDY for each available receive buffer. The OPN Originator is not required to have any receive buffers available when the Login_BB_Credit is = 0. The OPN Recipient is not permitted to transmit data frames even if one or more R_RDYs are received.

2 OPN Recipient may transmit 0, 1, or 2 R_RDYs here with no effect

A.2 Open full-duplex, Login_BB_Credit = 0

In this example, the OPN Originator transmits a full-duplex OPNyx, but forces half-duplex operation by not transmitting any R_RDYs following the OPNyx. L_Ports that grant a Login_BB_Credit = 0 are not required to have any receive buffers available at the time an OPN is sent or received.

Table A.2 – Open Full duplex, Login_BB_Credit = 0

Step	OPN Initiator X	Avail BB_ Credit	Min_available RX_buffers	OPN Recipient Y	Avail BB_ Credit	Min_available RX_buffers
A	Login_BB_Credit = 0	0	0	Login_BB_Credit = 0	0	0
B	Tx OPN Full Duplex Tx 0 to N R_RDYs (note 1)	0 0	0 0 to N			
C				Rx OPN Full Duplex Rx 0 to N R_RDYs Tx 2 R_RDYs	0 0 to N 0 to N	0 0 2
D	Rx 2 R_RDYs Tx 2 frames Tx CLS	2 0 0	0 to N 0 to N 0 to N 0 to N			
E				Rx 2 frames (note 2) Rx CLS Tx CLS	1 1 0	0 0 0
F	Rx CLS	0	0		0	0

Notes

1 The OPN Originator may transmit one R_RDY for each available receive buffer. The OPN Originator is not required to have any receive buffers available when the Login_BB_Credit is = 0.

2 OPN Recipient may transmit 0, 1, or 2 R_RDYs here with no effect

A.3 Open full-duplex, Login_BB_Credit = 0

In this example, the OPN Originator transmits a full-duplex OPNyx and sends 2 R_RDYs that enable the OPN Recipient to transmit 2 frames following receipt of the OPNyx.

Table A.3 – Open Full duplex, Login_BB_Credit = 0

Step	OPN Originator X	Avail BB_ Credit	Min_available RX_buffers	OPN Recipient Y	Avail BB_ Credit	Min_available RX_buffers
A	Login_BB_Credit = 0	0	0	Login_BB_Credit = 0	0	0
B	ARB Win Arbitration Tx OPN Full Duplex Tx 2 R_RDYs	0 0 0 0	0 0 0 2			
C				Rx OPN Full Duplex Rx 2 R_RDYs Tx 2 R_RDYs Tx 1 frame	0 2 2 1	0 0 2 2
D	Rx 2 R_RDYs Rx 1 frame Tx 2 frames Tx CLS	2 2 0 0	2 1 1 1			
E				Rx 2 frames (note 1) Rx CLS (note 2) Tx CLS	1 1 0	0 0 0
F	Rx CLS	0	1		0	0

Notes

1 OPN Recipient may transmit 0, 1, or 2 R_RDYs here with no effect

2 Since the Available BB_Credit is = 1, the OPN Recipient may transmit one frame here before transmitting the CLS if desired

A.4 Open full-duplex and half-duplex, Login_BB_Credit>0

This example demonstrates FC-AL BB_Credit rules using both the full-duplex OPNyx and the half-duplex OPNyy. It is not intended to demonstrate FCP over FC-AL, or any particular implementation an L_Port.

Table A.4 – Open Full and half duplex, Login_BB_Credit>0

Step	OPN Originator X	Avail BB_ Credit	Min available RX_buffers	OPN Recipient Y	Avail BB_ Credit	Min available RX_buffers
A	Login completed with Y, Login_BB_Credit = 3	0	3	Login completed with X, Login_BB_Credit = 2	0	2
B	ARB Win Arbitration Tx OPN Full Duplex Tx 3 R_RDYs Tx 1 frame Tx CLS	0 2 2 1 1 0	3 3 3 3 3 3			
C				Rx OPN Full Duplex Rx, discard 3 R_RDYs Tx 2 R_RDYs Rx 1 Frame Rx CLS	3 3 3 3 3	2 2 2 1 1
D	Rx, discard 2 R_RDYs	1	3	Tx 1 R_RDY (optional) Tx 3 frames Tx CLS (note 1)	3 3 0	2 2 2
E	Rx 1 R_RDY (if sent) Rx 3 frames Rx CLS	2 1 1	3 0 0	ARB	0 0 0	2 2 2
F	Return to Monitoring State (note 2)	0	3	Win Arbitration Tx OPN Full Duplex Tx 2 R_RDYs Tx 3 frames Tx CLS	0	2
G	Rx OPN Full Duplex Rx, discard 2 R_RDYs Tx 3 R_RDYs Rx 3 frames Rx CLS	2 2 2 2 2	3 3 3 0 0			
H	Tx 2 Frames Tx CLS	0 0	0 3 (note 1)	Rx, discard 3 R_RDYs	0	2
I	ARB			Rx 2 frames Rx CLS	0 0	0 0
J				Return to Monitoring State (note 2)	0	2
K	Win Arbitration Tx OPN Half Duplex Tx 3 R_RDYs Tx 2 frames Tx CLS	2 2 2 2 0	 3 3 3 3			
L				Rx OPN Half Duplex Rx, discard 3 R_RDYs Rx 2 frames Tx 2 R_RDYs Tx CLS	0 0 0 0 0	2 0 0 0 2
M	Rx 2 R_RDYs Rx CLS	0	3		0	0

Table A.4 – Open Full and half duplex, Login_BB_Credit>0

Step	OPN Originator X	Avail BB_ Credit	Min available RX_buffers	OPN Recipient Y	Avail BB_ Credit	Min available RX_buffers
Notes 1 Since the OPN Recipient has received a CLS from the OPN Originator, it may transmit a CLS once it has Login_BB_Credit receive buffers available. 2 The OPN Originator is only required to have Login_BB_Credit buffers available at the time that L_Port enters the Monitoring State. This may necesitate delaying entry to the Montioring State until the required number of buffers are available. 3 FC-AL requires that a port transmit a number of R_RDYs equal to or greater than the Login_BB_Credit. No exception is made for the half-duplex OPNyy.						

Annex B
(informative)

FC-PH Timer Requirements

For the purposes of background information, the following timer requirements are transcribed from FC-PH revision 4.3 with Errata. FC-PH defines R_T_TOV, E_D_TOV, and R_A_TOV. All timers must be accurate to +20%/-0% of the stated value.

B.1 R_T_TOV

R_T_TOV refers to the Receiver_Transmitter_Timeout value, and is specified by FC-PH to be 100 ms. It is used to detect loss of synchronization at the receiver.

a) Loss of Synchronization for more than R_T_TOV while not in the Offline State is considered a link failure condition.

b) R_T_TOV timeout during the Link Reset Protocol is considered a link failure condition

c) If a port remains in the LR Transmit , LR Receive, LRR Receive State for more than R_T_TOV, a Link Reset Protocol timeout shall be detected which results in a Link Failure condition (and the NOS Transmit State is entered)

d) While in the NOS Receive State, an R_T_TOV period is started when NOS is no longer recognized and no other events occur which cause a transition out of the NOS Receive State. If the timeout period expires, the port shall enter NOS Transmit State.

e) While a port is attempting to go Online, if no Primitive Sequence is received or event detected which causes the port to exit the OLS Transmit State after R_T_TOV, the port shall enter the Wait for OLS State.

f) If Loss of Synchronization is detected for more than R_T_TOV while in the OLS Receive State or while in the OLS Transmit State at an appropriate time in Link Initialization, the port shall enter the Wait for OLS state.

B.2 E_D_TOV

E_D_TOV refers to Error_Detect_Timeout value. It is used to detect error conditions, and represents a reasonable timeout value for the detection of a response to a timed event. A default value is 10 seconds. However, a valid E_D_TOV shall also adhere to the proper relationship to R_A_TOV. When an N_Port performs fabric login, the common service parameters provided by the F_Port specify the proper value for E_D_TOV. When an N_Port performs N_Port login in a point-to-point topology, the common service parameters provided by each N_Port provide a value for E_D_TOV. If the two values differ, the larger value shall be used by each N_Port.

a) “Each Data frame within a Sequence shall be transmitted within an E_D_TOV timeout period to avoid timeout errors at the destination N_Port.” (FC-PH 4.3, clause 20.2)

b) “Link Control frames shall be transmitted within an E_D_TOV timeout period of the event which causes transmission of the Link Control frame.” (FC-PH 4.3, clause 20.3)

c) “if the Sequence Initiator attempts to abort a Sequence using ABTS (Abort Sequence Protocol) and it detects a Sequence timeout (E_D_TOV) waiting for the ACK frame in response to the ABTS, it shall abort the Exchange (ABTX)...” (FC-PH 4.3, clause 21.1.2)

d) “The RTV Link Service request Sequence requests an N_Port or F_Port (hex 'FFFFFE') to return the Resource_Allocation_Timeout Value (R_A_TOV) and the Error_Detect_Timeout Value (E_D_TOV) in the Accept reply Sequence.” (FC-PH clause, 21.4.13)

e) “Word 3 (E_D_TOV value) shall only be meaningful by an N_Port in a point-to-point topology. In a topology other than point-to-point, word 3 shall not be meaningful. The E_D_TOV value shall be specified as a count of 1 ms increments. ... The E_D_TOV value in the Accept shall be greater than or equal to the value in the PLOGI. The E_D_TOV value in the Accept shall be the value used by each N_Port. R_A_TOV shall be a value twice the E_D_TOV value in a point-to-point topology.” (FC-PH 4.3, clause 23.6.3.7)

f) “Each Data frame in a Sequence shall be transmitted within an E_D_TOV timeout period of the previous Data frame transmitted within the same Sequence. Otherwise, a Sequence timeout shall be detected.” (FC-PH 4.3, clause 24.3.5)

g) “Each ACK shall be transmitted within an E_D_TOV timeout period of the event which prompts the initiative to transmit an ACK frame. For example, when using ACK_1, it shall be trans mitted within E_D_TOV of the Data frame reception. Whereas, when using ACK_0, it shall be transmitted within E_D_TOV of receiving the last Data frame of the Sequence.” (FC-PH 4.3, clause 24.3.7)

h) “In Class 1 a missing Data frame error due to timeout is detected by the Sequence Recipient if a partial Sequence has been received and the next expected Data frame (current SEQ_CNT+1, except when a wrap to zero occurs) is not received within an E_D_TOV timeout period.” (FC-PH 4.3, clause 24.3.9)

i) “In Class 2 and 3, with out of order delivery, a potentially missing Data frame is detected if a frame is received with a SEQ_CNT that is not one greater than the previously received frame, except when a SEQ_CNT wrap to zero occurs. If the potentially missing Data frame is not received within the E_D_TOV timeout period, a missing frame error is detected.” (FC-PH 4.3, clause 24.3.9)

j) “In Class 2, with in order delivery, a potentially missing Data frame is detected if a frame is received with a SEQ_CNT that is not one greater than the previously received frame, except when a SEQ_CNT wrap to zero occurs. If the potentially missing Data frame is not received within the E_D_TOV timeout period, a missing frame error is detected.” (FC-PH 4.3, clause 24.3.9)

k) “The types of Sequence errors that shall be detected by an N_Port include: (FC-PH 4.3, clause 24.3.10.1)

1) - detection of a missing frame based on SEQ_CNT,

2) - detection of a missing frame based on a timeout (E_D_TOV),...”

l) “An Exchange shall be remain Open until: (FC-PH 4.3, clause 24.3.14)

1) - the last Sequence of the Exchange completes normally,

2) - a timeout period of E_D_TOV has elapsed since the last Sequence of the Exchange completed abnormally, or

3) - the Exchange is successfully aborted with ABTX (which includes a Recovery_Qualifier timeout, if necessary).”

m) “If a Sequence error is detected or the E_D_TOV expires when the Sequence Recipient is withholding the last ACK for a Sequence and waiting to send other ACKs for that Sequence, the Sequence Recipient supporting discard policy shall set Abort Sequence bits and transmit the last ACK (see 24.6.5).” (FC-PH 4.3, clause 26.4.3.4)

n) “An N_Port uses the E_D_TOV timeout period after a connect-request has been transmitted (part of Sequence timeout) whether the connect-request is a normal request or a Stacked request. That is, an ACK response shall be received within an E_D_TOV timeout period or an F_BSY shall be returned from the Fabric to the N_Port. If either condition is not met within E_D_TOV, the N_Port detects a Sequence timeout and Connection Recovery (see 28.8) is performed.” (FC-PH 4.3, clause 28.5.2)

o) Retransmission of a class 2 data frame in response to F_BSY or P_BSY shall occur in less than E_D_TOV.

p) A Link timeout error shall be detected if one or more R_RDYs are not received within E_D_TOV after BB_Credit_CNT has reached zero. The Link Reset Protocol is then invoked.

q) A Link timeout error shall be detected if, in class 1, all Active Sequences have timed out (E_D_TOV)

r) When a Sequence Recipeint detects and E_D_TOV Sequence timeout, the Recipient shall terminate the Sequence and update the Exchange Status Block.

s) With misordered transmission, the credit for a lost ACK cannot be recovered until E_D_TOV has expired.

B.3 R_A_TOV

R_A_TOV represents the Resource_Allocation_Timeout value. One use is to determine when to reinstate a Recovery Qualifier when out of order delivery is possible (either due to fabric misordering of connectionless traffic or BSY retries by the Sequence Initiator).

When an N_Port performs fabric login, the common service parameters provided by the F_Port specify the proper value for R_A_TOV. In a point-to-point topology, R_A_TOV shall be set to twice the E_D_TOV value. When R_A_TOV is used to determine when to reuse an N_Port resource such as a Recovery Qualifier, the resource shall not be reused until R_A_TOV has expired for all frames previously transmitted which fall within the SEQ_CNT range of the Recovery Qualifier.

a) R_A_TOV is E_D_TOV plus twice the maximum time a frame may be delayed in a fabric and still be delivered.

b) The default value is 120 seconds.

c) If a Recovery Qualifier is used in class 2 or 3, a Reinstate Recovery Qualifier (RRQ) ELS shall be issued after R_A_TOV has expired.

d) If the R_A_TOV is larger than the Expiration Time value in the Expiration Security header, the rules regarding R_A_TOV shall be followed regardless of the value of the Expiration Time.

e) After transmitting LCR, the N_Port shall wait R_A_TOV before initiating new Sequences to the destination N_Port.

f) The originator of an ELS shall detect an Exchange error following Sequence Initiative transfer if the Reply Sequence to the ELS is not received within 2 x R_A_TOV.

g) If a Recovery Qualifier has been established, the ABTS initiator shall issue an RRQ after waiting R_A_TOV.

h) If an ABTX originator requires a Recovery Qualifier, it shall transmit an RRQ after sending the ABTX and waiting R_A_TOV.

i) The originator of an FC-4 Link Service Exchange shall detect an Exchange error following Sequence Initiative transfer if the reply Sequence is not received within R_A_TOV.

j) During reLogin with the Fabric, if the N_Port detects the F_Port name has changed since the last Fabric Login, the N_Port shall wait R_A_TOV before initiating or accepting communication with other N_Ports.

k) R_A_TOV is specified in 1ms increments.

l) In the Discard multiple Sequences Error Policy in class 3, the Sequence Recipient shall not be required to utilize a timeout value of R_A_TOV following detection of a missing frame. Therefore, frames may be discarded by the Sequence Recipient for an Exchange for the Sequence in error and all subsequent Sequences received by that port if other detection mechanisms are not utilized by the Sequence Initiator.

m) In class 3, the Sequence Initiator considers the Sequence to be Open until the deliverability is confirmed or R_A_TOV has expired.

n) In the ABTS protocol, a Recovery Qualifier may be specified by the Recipient in the Basic Accept. The X_ID values in that Recovery Qualifier shall be retired by both N_Ports for R_A_TOV in classes 2 and 3.

o) In class 2, if a Sequence has been aborted and the Sequence Recipient supplies the Recovery Qualifier, the Sequence Initiator shall not transmit any data frames within that range within R_A_TOV. Both the Sequence Initiator and Sequence Recipient discard frames within that range. After R_A_TOV has expired, the Sequence Initiator shall reinstate the Recovery Qualifier using RRQ.

p) If an ABTS is transmitted in classes 2 or 3, the Recovery Qualifier shall be timed out by the ABTS initiator for R_A_TOV. After R_A_TOV has expired, the ABTS initiator shall issue an RRQ.

q) In classes 2 or 3, after an ABTS intiiator receives the BA_ACC, it shall discard any frames received for the specified SEQ_CNT range until R_A_TOV has expired.

r) Annex U: R_A_TOV is ... twice the longest amount of time that a frame may stay in the Fabric before it may be delivered (i.e., before it must be discarded).

Annex C
(informative)

Sequence Management and Error Detection

C.1 FCP commands

This annex illustrates execution of FCP commands under normal and error conditions and discusses how the Sequences are managed for those commands.

C.1.1 Write command

Figure C.1 illustrates the Information Unit and frame flow for a normal FCP Write type command. In this example, two FCP_DATA (T6) Sequences are required to transfer the data.

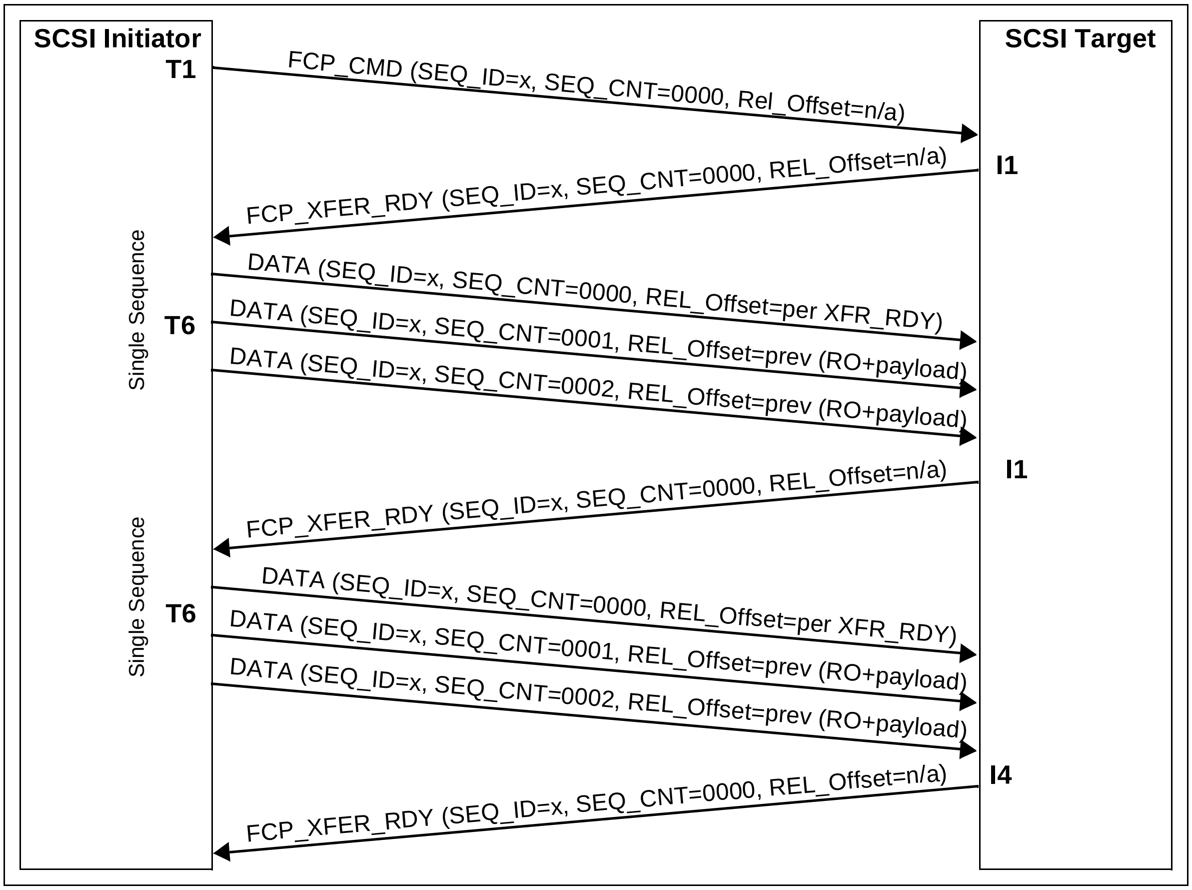

Figure C.1 – Normal FCP Write Command Example

The write command begins when the Initiator sends the FCP_CMND Sequence to the target. The FCP_CMND is a single-frame Sequence for operations compliant with this document. When the target is ready to begin the data transfer, it sends an FCP_XFER_RDY (I1) Sequence to the SCSI Initiator. The FCP_XFER_RDY specifies how much data is to be transferred in the subsequent FCP_DATA Sequence, and from what offset in the application client buffer. Receipt of the FCP_XFER_RDY indicates that the SCSI Target received and processed the FCP_CMND Sequence.

The SCSI Initiator begins the FCP_DATA Sequence by transmitting the first frame of that Sequence. The SEQ_CNT is normally set to x'0000' and the Relative Offset is set from the DATA_RO field of the FCP_XFER_RDY. As each subsequent frame of the FCP_DATA Sequence is sent, the SEQ_CNT is incremented by +1 from the previous frame and the Relative_Offset in the parameter field of the frame header is set to the Relative_Offset of the previous frame + payload length of the previous frame (i.e., the Relative_Offset = the offset of the first byte of the payload of the current frame which must be +1 from the offset of the last byte of the previous data frame).

When the last data frame associated with the FCP_DATA Sequence is sent, the SCSI Initiator sets the Sequence Initiative bit in the frame header enabling the Target to send another FCP_XFER_RDY or the FCP_RSP. Receipt of either by the Initiator confirms delivery of the preceeding FCP_DATA Sequence.

If additional data remains to be transferred, the Target sends another FCP_XFER_RDY instructing the Initiator to send another FCP_DATA Sequence. The FCP_XFER_RDY can request the next sequential data associated with the command, or a non-sequential segment of data associated with the command.

If all data has been transferred, or the Target desires to end the data transfer, the Target sends an FCP_RSP Sequence containing the Status and FCP_Residual. When the FCP_RSP is received by the Initiator, it knows that the operation has completed in the Target.

C.1.1.1 Error during FCP_CMND Sequence

If an error occurs during the FCP_CMND Sequence such that the FCP_CMND is unrecognizable or not received by the Target, the target never begins command execution for that command. The SCSI Initiator detects an error after ULP_TOV expires for that operation and sends ABTS to abort the Exchange.

The Target never received the command, so no Exchange is active for that operation when the ABTS is received. The Target sends an ACC to the ABTS with the Last_Sequence (LS) bit set.

C.1.1.2 Error during FCP_XFER_RDY Sequence

If an error occurs during the FCP_XFER_RDY Sequence such that the FCP_XFER_RDY is unrecognizable or not received by the SCSI Initiator the operation hangs with the Target waiting for the data and the SCSI Initiator waiting for the FCP_XFER_RDY.

If the Target uses the RR_TOV timer to time the return of the requested data, the Target will detect that the data failed to arrive before the timer expired. Since the Target is prohibited from sending ABTS, the Target is not able to abort the operation, but must wait for the SCSI Initiator to issue and ABTS.

If the RR_TOV has expired for all outstanding FCP_XFER_RDY Sequences with a specific SCSI Initiator, the Target may choose to send LOGO and terminate all active operations with that initiator.

The SCSI Initiator will detect that an error has occured when the ULP_TOV expires for the command.

C.1.1.3 Error during FCP_DATA Sequence

If an error occurs on the first or middle frame of a multi-frame FCP_DATA Sequence such that the frame is unrecognizable or not received by the SCSI Target, the Target will detect the error on the next frame of the Sequence. The SEQ_CNT of the next frame will not be +1 from the prior frame of that Sequence (since the first frame was lost) nor will the relative offset in the parameter field of the frame header equal the expected relative offset provided in the FCP_XFER_RDY that requested the data. When this condition is detected, the Target discards the sequence in error.

If an error occurs on the last frame of a multi-frame FCP_DATA Sequence such that the frame is unrecognizable or not received by the SCSI Target, the Target detects the error since the next expected frame was not received within E_D_TOV.

If an error occurs on a single-frame FCP_DATA Sequence such that the frame is unrecognizable or not received by the SCSI Target, or all frames of the FCP_DATA Sequence so that the entire Sequence is lost, the Target cannot determine if the FCP_XFER_RDY was received or the FCP_DATA

Sequence was in error. The action taken in either case is the same as described earlier for FCP_XFER_RDY errors.

C.1.1.4 Error during FCP_RSP Sequence

If an error occurs during the FCP_RSP Sequence such that the FCP_RSP is unrecognizable or not received by the SCSI Initiator the initiator detects an error when the ULP_TOV expires and sends ABTS to the SCSI Target.

The operation is considered complete at the SCSI Target once the FCP_RSP is sent, so no Exchange is active for that operation when the ABTS is received. The Target sends an ACC to the ABTS with the Last_Sequence (LS) bit set.

C.1.2 Read command

The read command begins when the Initiator sends the FCP_CMND Sequence to the target. The FCP_CMND is a single-frame Sequence for operations compliant with this document. This is the first Sequence shown in Figure C.2.

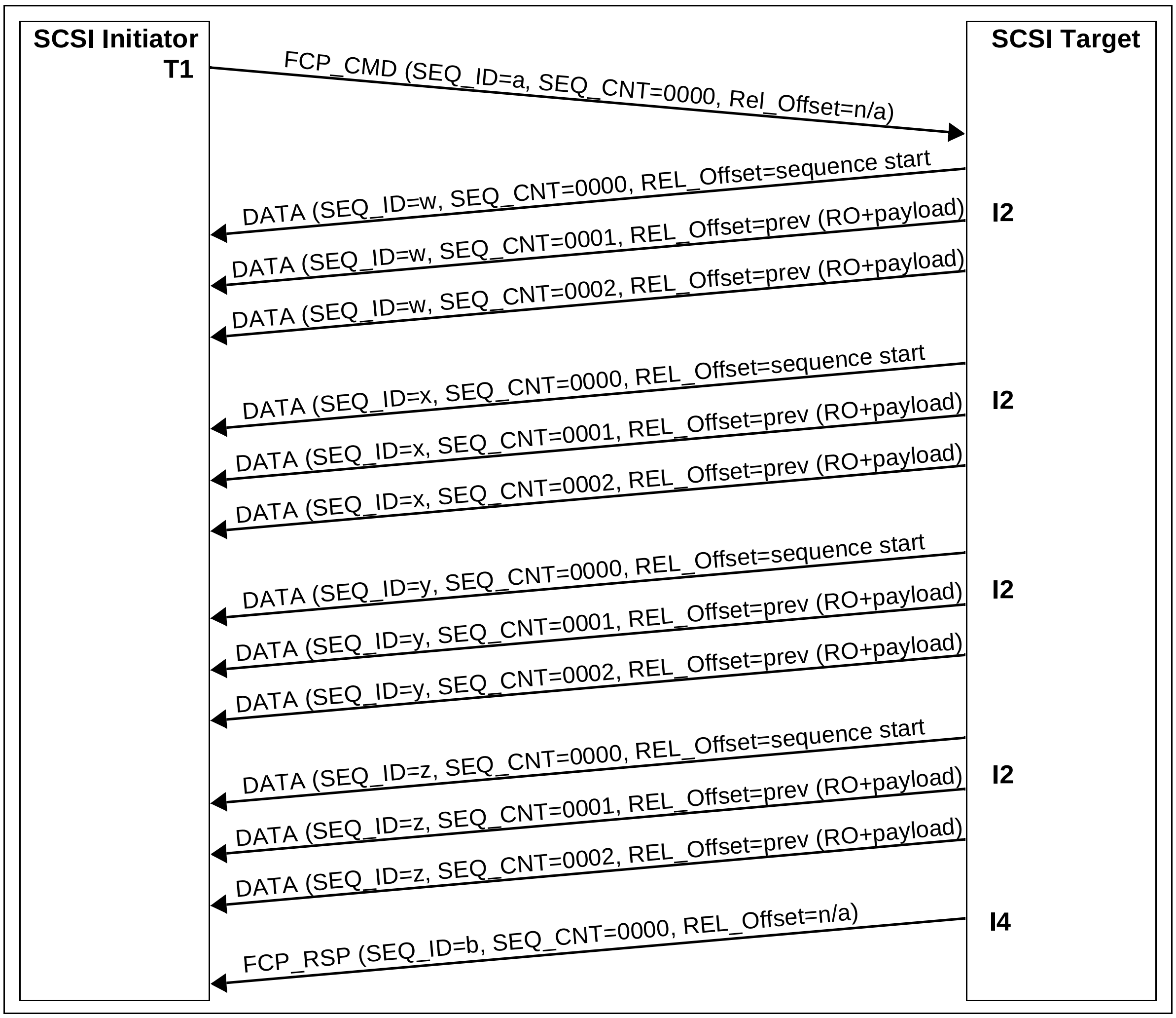

Figure C.2 – Normal FCP Read Command Example

When the Target is ready to begin the read data transfer, it starts sending the first data Sequence (labled SEQ_ID=w in Figure C.2). The first data frame of the Sequence establishes the initial relative offset of the Sequence and each subsequent frame of the Sequence must have continously increasing relative offsets.

When the first data Sequence is complete, the Target may begin sending the next data Sequence (labled SEQ_ID=x in Figure C.2). The first data frame of this Sequence establishes the initial relative offset for the Sequence and each subsequent frame of this Sequence must have continously increasing relative offsets.

The Target continues to send FCP_DATA Sequences until all of the requested data has been transferred. The Target may treat consecutive FCP_DATA Sequences as either streamed or non-streamed Sequences. While this is not strictly required by FC-PH under the conditions defined by this document (since R_A_TOV is zero a Sequence is considered closed immediately upon transmission of the final frame), following streamed Sequence rules may provide enhanced error detection capabilities when one or more entire Sequences are lost due to error conditions.

FC-PH defines a streamed Sequence as follows (from FC-PH 18.6):

“If the Sequence Initiator initiates a new Sequence for the same Exchange before receiving the final ACK (EOFt, EOFdt) for the previous Sequence in Class 1 and 2, or before R_A_TOV has expired for all frames of a Class 3 Sequence, it is termed a streamed Sequence. If streamed Sequences occur, it is the responsibility of the Sequence Initiator to use X+1 different consecutive SEQ_IDs where X is the number of Open Sequences per Exchange (see 23.6.8.8). For example, if X=2 from Login, then consecutive SEQ_IDs of 11-93-22-11-93 is acceptable.”

FC-PH subsequently defines rules for the SEQ_CNT for streamed and non-streamed cases. It says (from FC-PH 18.8):

“If a Sequence is streamed, the sequence count of the first Data frame of the Sequence shall be incremented by one from the sequence count of the last Data frame of the previous Sequence (this is termed continuously increasing SEQ_CNT). If a Sequence is non-streamed, the starting sequence count may be continuously increasing or binary zero.

FC-PH goes on to say:

“If consecutive non-streamed Sequences for the same Exchange occur during a single Sequence Initiative, it is the responsibility of the Sequence Initiator to use a different SEQ_ID for each consecutive Sequence. For example, consecutive SEQ_IDs of 21-74-21-74 is acceptable. The examples show when a SEQ_ID shall be allowed to be repeated. A series of SEQ_IDs for the same Exchange may also be random and never repeat (also see 24.3.4).”

Therefore, it is also permissable to use just two SEQ_IDs and alternate between them. For example, in figure C.2, Sequence x could use a SEQ_ID=21, Sequence y could use SEQ_ID=74, Sequence z could use SEQ_ID=21 and Sequence z (the FCP_RSP) could use SEQ_ID=74.

FC-PH subsequently defines rules for the SEQ_CNT for streamed and non-streamed cases. It says (from FC-PH 18.8):

“If a Sequence is streamed, the sequence count of the first Data frame of the Sequence shall be incremented by one from the sequence count of the last Data frame of the previous Sequence (this is termed continuously increasing SEQ_CNT). If a Sequence is non-streamed, the starting sequence count may be continuously increasing or binary zero.”

After all of the FCP_DATA Sequences have been sent, the Target sends the FCP_RSP Sequence to signal completion of the command.

Since there is no transfer of Sequence Initiative after a FCP_DATA Sequence has been sent, the Target has no confirmation of successful Sequence delivery to the SCSI Initiator.

C.1.2.1 Error during FCP_CMND Sequence

If an error occurs during the FCP_CMND Sequence such that the FCP_CMND is unrecognizable or not received by the Target, the target never begins command execution for that command. The SCSI Initiator detects an error after ULP_TOV expires for that operation and sends ABTS to abort the Exchange. This is identical to a write command.

The Target never received the command, so no Exchange is active for that operation when the ABTS is received. The Target sends an ACC to the ABTS with the Last_Sequence (LS) bit set.

C.1.2.2 Error during FCP_DATA Sequence

On read commands, the SCSI Initiator is responsible for ensuring the amount of data returned is equal to the amount specified by FCP_DL, even if the Status returned is GOOD. If the amount does not match FCP_DL, a command-specific ULP recovery action may need to be invoked (see 8.2.1).

Since there are no transfers of Sequence Initiative during read operations once the Target receives the T1 Information Unit, status may be returned as GOOD even though some of the data was not received by the SCSI initiator. This may occur as the result of lost or corrupted frames in the read data.

The manner in which a SCSI Initiator determines that the correct amount of data is returned is implemenation dependent and may include counting the number of bytes returned, computing the number of bytes received by use of the relative offsets of each Sequence, or other means not specified.

C.1.2.2.1 Lost Frame(s) during FCP_DATA Sequence

If an error occurs on the first frame of an FCP_DATA Sequence such that the frame is unrecognizable or not received by the SCSI Initiator the initiator detects an error when the frame is received. The detection occurs as a result of one or more of the following:

a) Receipt of a data frame for a Sequence without receipt of the first frame of that Sequence is an error in an in-order environment.

b) It is required that the SEQ_CNT of each received frame within a Sequence in an in-order environment be +1 greater than the SEQ_CNT of the previous frame of that Sequence.

c) It is required that the relative offset of each received frame within a Sequence in an in-order environment, other than the first frame, be equal to the relative offset of the previous frame of that Sequence + the length of the payload of that frame.

d) Detection of a lost frame if the next expected frame of a Sequence is not received within E_D_TOV of the previous frame of that Sequence.

If an error occurs on the second or subsequent frame of an FCP_DATA Sequence such that the frame is unrecognizable or not received by the SCSI Initiator the initiator detects an error when it recognizes condition b), c), or d) above.

If an error occurs on the last frame of an FCP_DATA Sequence such that the frame is unrecognizable or not received by the SCSI Initiator the initiator detects an error when it recognizes one or more of the following conditions:

a) Receipt of a data frame for a new Sequence without receipt of the last frame a prior Sequence in an in-order environment.

b) Detection of a lost frame if the next expected frame of a Sequence is not received within E_D_TOV of the previous frame of that Sequence.

Upon detection of any of the listed conditions, the SCSI Initiator detects a Sequence error and takes the appropriate action as described in 9.3.1.

C.1.2.2.2 Loss of entire FCP_DATA Sequence

If an error occurs such that one or more complete FCP_DATA Sequence are unrecognizable or not received by the SCSI Initiator the initiator detects an error when it recognizes one or more of the following conditions:

a) The SEQ_CNT of the first frame of a received Sequence is not equal to zero or +1 from the SEQ_CNT of the last frame of the previous Sequence of that Exchange (continously increasing SEQ_CNT is recommended to enhance error detection).

b) The SEQ_ID of the first frame of a received Sequence is the same as the SEQ_ID of the previous frame of that Exchange (consecutive Sequences for the same Exchange must have different SEQ_IDs). While it may not be intuitively obvious, using the minimum number of SEQ_IDs during an operation provides improved error detection since the probability of receiving consecutive Sequences with the same SEQ_ID is inversely related to the number of SEQ_IDs used.

c) The amount of data returned for the entire operation is not equal the the FCP_DL.

Upon detection of any of the listed conditions, the SCSI Initiator detects a Sequence error and takes the appropriate action as described in 9.3.1.

C.1.2.2.3 Loss of multiple Sequence fragments

If an error occurs such that the ending portion of one FCP_DATA Sequence, one or more complete intermediate FCP_DATA Sequence, and the beginning of another FCP_DATA Sequence are unrecognizable or not received by the SCSI Initiator such that the two Sequence fragments appear to be a single Sequence, the initiator detects an error when it recognizes one or more of the following conditions:

a) It is required that the SEQ_CNT of each received frame within a Sequence in an in-order environment be +1 greater than the SEQ_CNT of the previous frame of that Sequence.

b) It is required that the relative offset of each received frame within a Sequence in an in-order environment, other than the first frame, be equal to the relative offset of the previous frame of that Sequence + the length of the payload of that frame.

Upon detection of any of the listed conditions, the SCSI Initiator detects a Sequence error and takes the appropriate action as described in 9.3.1.

C.1.2.3 Error during FCP_RSP Sequence

If an error occurs during the FCP_RSP Sequence such that the FCP_RSP is unrecognizable or not received by the SCSI Initiator the initiator detects an error when the ULP_TOV expires and sends ABTS to the SCSI Target.

The operation is considered complete at the SCSI Target once the FCP_RSP is sent, so no Exchange is active for that operation when the ABTS is received. The Target sends an ACC to the ABTS with the Last_Sequence (LS) bit set.

C.1.2.4 Streamed Sequences during reads

Some implementations may choose to treat consecutive FCP_DATA Sequences and the FCP_RSP Sequence as streamed Sequences. While this is not strictly required by FC-PH under the conditions defined by this document, it may provide enhanced error detection capabilities when one or more entire Sequences are lost.

The following examples illustrate the usage of SEQ_ID and SEQ_CNT during streamed Sequences.

C.1.2.4.1 Streamed Sequences Example 1

Figure C.3 illustrates a read operation consisting of four FCP_DATA Sequences followed by an FCP_RSP Sequence. In this example, the number of Open Sequences per Exchange (referred to as X later) is assumed to be equal to 2 and the maximum Sequence size (as determined by the SCSI maximum burst size in the Disconnect/Reconnect mode page) is equal to three frames.

The Sequence initiator begins transmission of the first FCP_DATA IU and assigns SEQ_ID=11. The SEQ_CNT of the first frame of this Sequence may be either zero or continously increasing (incre-

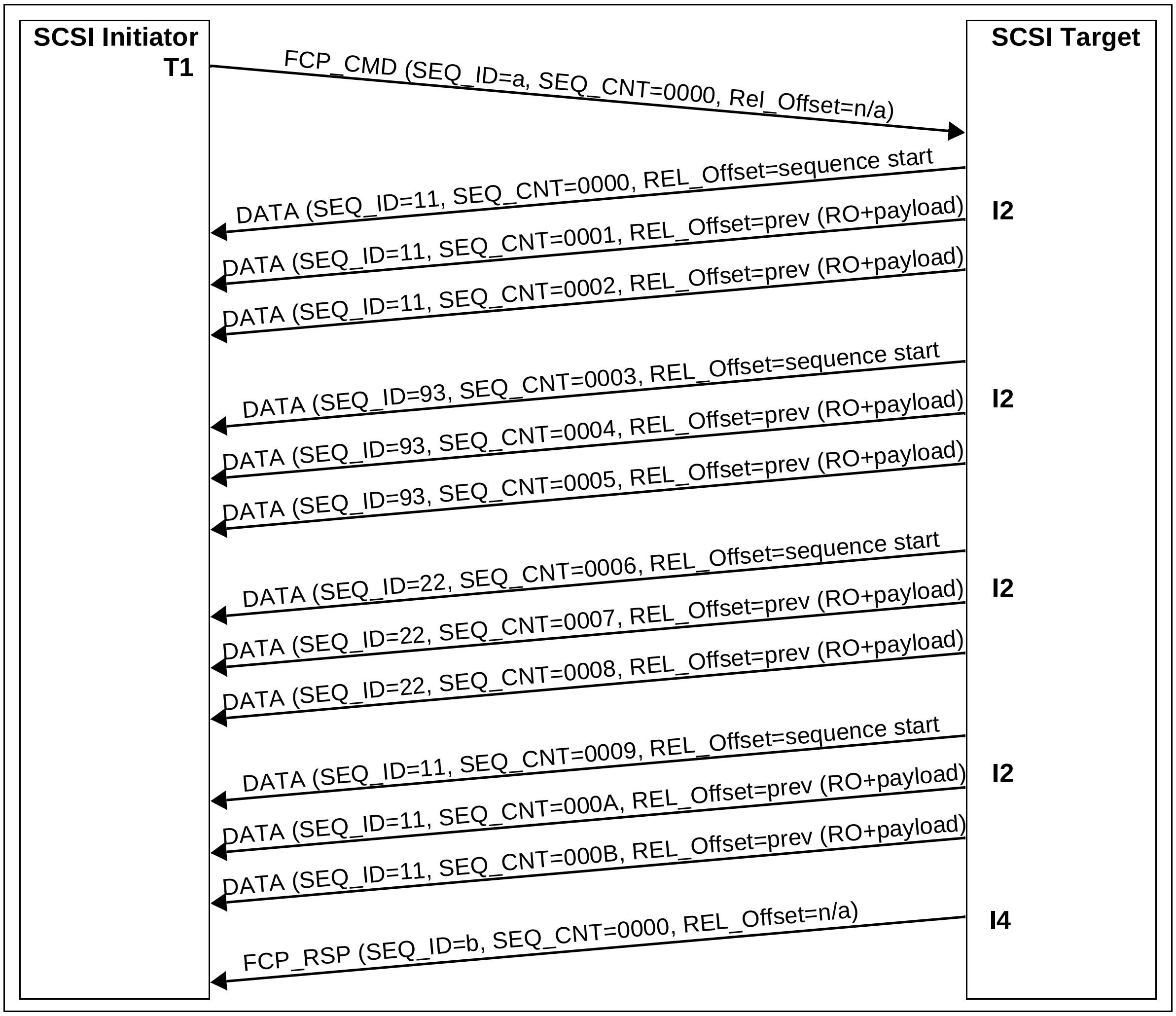

Figure C.3 – Streamed Sequence Example 1

mented by +1 from the last frame of the previous Sequence). In this example, the SEQ_CNT of the first frame is zero.

As each frame of the Sequence is sent, the SEQ_CNT is incremented by +1 until the last frame of the Sequence has been sent. In this example, the maximum Sequence size is limited to 3 frames by the SCSI maximum burst size, therefore the Sequence consists of three frames with SEQ_CNTs of hex‘0000’, hex‘0001’, and hex‘0002’.

Since there is additional data to send, a new Sequence must be started. In this case, SEQ_ID=93 is used. The Sequence initiator does not have to assign SEQ_IDs in increasing order. Since SEQ_ID=93 is a streamed Sequence, it must use continuously increasing sequence count for the frames in that Sequence. Therefore, the SEQ_CNTs continue by incrementing +1 from the SEQ_CNT of the last frame of SEQ_ID=11 and are hex‘0003’, hex‘0004’, and hex‘0005’. Again, since the maximum burst size limited the Sequence size to 3 frames, a new Sequence must be started to send additional data.

At this point, two Sequences have been sent which is equal to the maximum number of Open Sequences per Exchange PLOGI parameter. In a Class 1 or Class 2 environment, the next Sequence could not be initiated until delivery of one of the prior Sequences was confirmed. In Class 3, however, there is not mechanism to confirm delivery of those Sequences, so the R_A_TOV value is used to determine when the X+1th SEQ_ID may be sent. Since R_A_TOV is defined to be 0 seconds, the X+1th SEQ_ID may be sent immediately.

This allows SEQ_ID=22 to be sent using frames with SEQ_CNTs of hex‘0006’, hex‘0007’, and hex‘0008’. Upon completion of SEQ_ID=22 the requirement to use X+1 different SEQ_IDs has been met allowing the next Sequence to reuse SEQ_ID=11. Since this is still a streamed Sequence, the SEQ_CNTs continuously increase using values of hex‘0009’, hex‘000A’, and hex‘000B’. At this point, all of the FCP_DATA Sequences have been sent.

The FCP_RSP Sequence may or may not be considered a streamed Sequence by implementions conpliant with this document. In this example, the FCP_RSP uses SEQ_ID=93 and SEQ_CNT=’0000’ to complete the Exchange. The FCP_RSP shall not use the same SEQ_ID as the preceeding FCP_DATA Sequence since that would violate FC-PH clause 18.6.

C.1.2.4.2 Streamed Sequence example 2

Figure C.4 illustrates a scenario in which all 65,536 SEQ_CNT values are used within a single Sequence. The number of Open Sequences per Exchange is assumed to be equal to 2 and the maximum Sequence size (as determined by the SCSI maximum burst size in the Disconnect/Reconnect mode page) is equal to, or greater than, 65,536 frames. Even though the maximum Sequence size is 65,536 frames, the Sequence initiator is not obligated to send the maximum number frames per Sequence and may send fewer frames before starting the next Sequence.

This example illustrates that even with the use of streamed Sequence rules, loss of a complete Sequence may be undetected since the SEQ_CNT of the next frame may appear correct even though an entire Sequence was lost. For example, if Sequence SEQ_ID=93 was lost, the SEQ_CNT of the first frame of SEQ_ID=22 is continously increasing from the last frame of SEQ_ID=11.

C.1.2.4.3 Streamed Sequence example 3

Figure C.5 illustrates a scenario in which the SEQ_CNT wraps within a single Sequence. The number of Open Sequences per Exchange is assumed to be equal to 2 and the maximum Sequence size (as determined by the SCSI maximum burst size in the Disconnect/Reconnect mode page) is equal to hex‘F000’ frames.

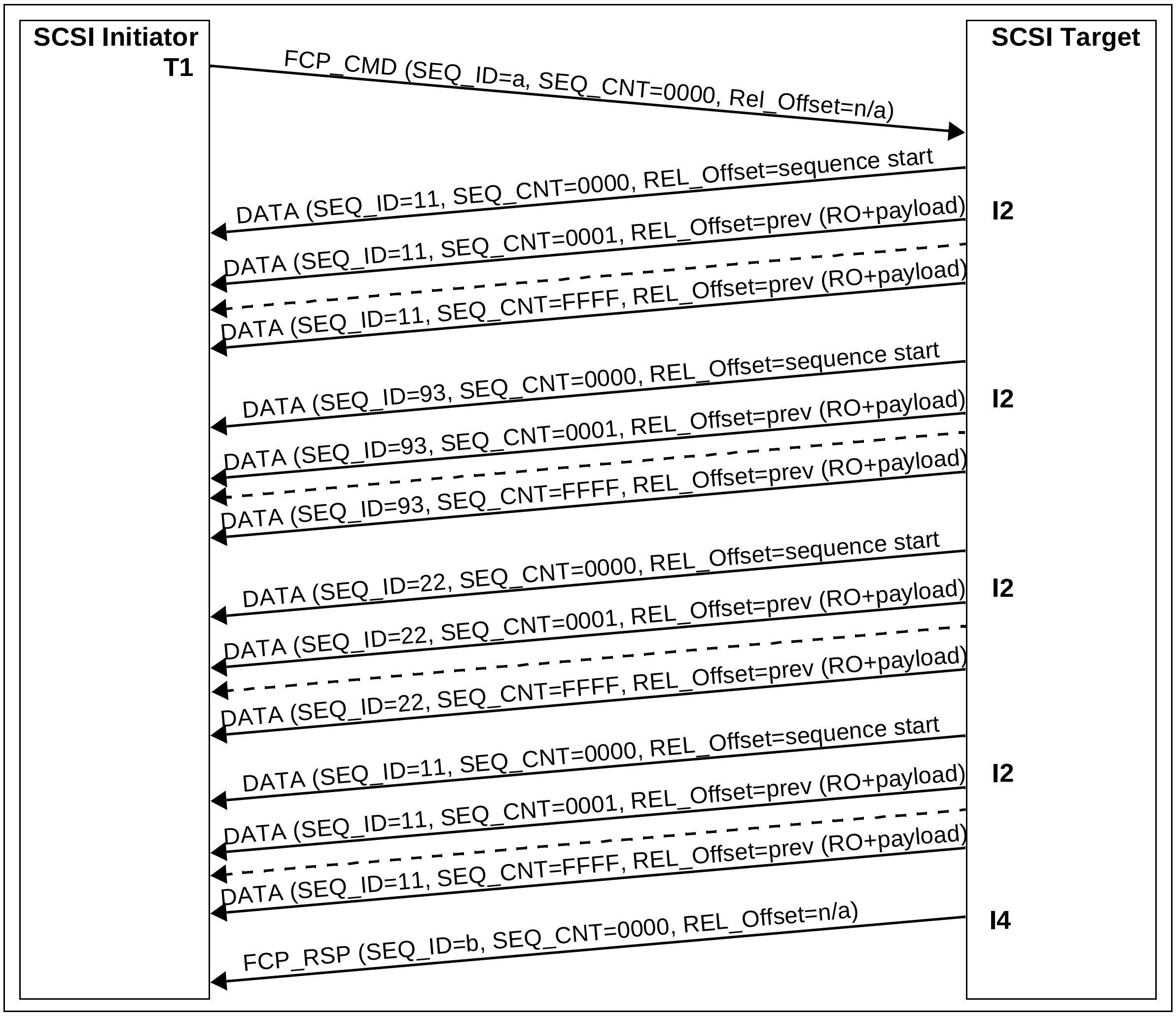

Figure C.4 – Streamed Sequence Example 2

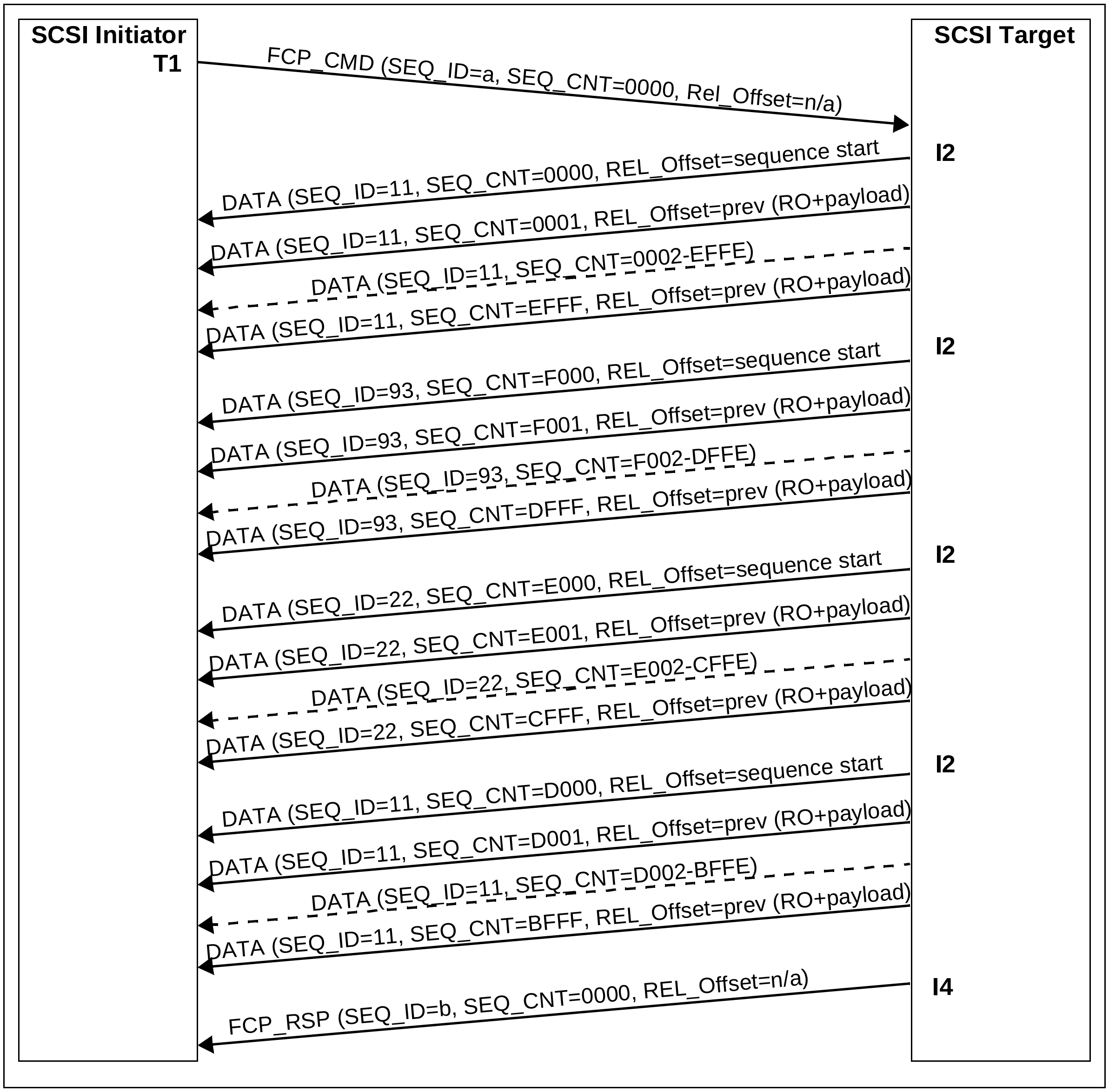

Figure C.5 – Streamed Sequence Example 3

Annex D
(informative)

IEEE Global Identifiers

This material is taken from X3T11/95-160 and X3T11/95-161 which provide a tutorial on the 48-bit Global Identifier format and the Fibre Channel use of IEEE Global Identifiers.

D.1 IEEE 48-Bit Global Identifier format

The IEEE defined 48-bit global identifier (GID-48) is assigned by a manufacturer who has been assigned a 24-bit company_id value by the IEEE Registration Authority. The 48-bit identifier is a concatenation of this 24-bit company_id (in the most significant bits) and a 24 bit extension identifier (in the least significant bits) assigned by the organization with that company_id.

The IEEE administers the assignments of 24-bit company_id values. The assignments of these values are public, so that a user of a GID-48 value can identify the organization which provided that value. IEEE has no control over the assignments of 24-bit extension identifiers and assumes no liability for assignments of duplicate GID-48 identifiers.

For example, assume that the company_id value is hex ’ACDE48’ and the extension identifier is hex ’234567’. The GID-48 value generated from these two numbers is hex ’ACDE48234567’.

	Most Significant Byte					Least Significant Byte
	Company_ID			Company assigned extension		
hex	AC	DE	48	23	45	67
binary	1010 1100	1101 1110	0100 1000	0010 0011	0100 0101	0110 0111
original byte address	A+0	A+1	A+2	A+3	A+4	A+5

If provided in byte-addressable media, the original byte-address order is specified: the most through least significant bytes of the GID-48 value are contained within the lowest through highest byte addresses, as illustrated above.

When transferred to other standard-specific locations (within a disk file or network packet, for example) the relative ordering of the bytes may be changed, as specified within the applicable standard.

D.2 Fibre Channel use of GID-48

The Fibre Channel Fabric model defines a single, homogeneous 24-bit address space. These short addresses are the basis for routing frames within a Fabric. However, the protocol supported by the N_Port/F_Port interface is also capable of transporting “long” identifiers of up to 64 bits in length. These identifiers may be globally unique types defined by a number of standards authorities. FC-PH generically refers to these identifiers as World Wide Names (WWNs).

These WWNs have two different uses:

a) to provide unique identification of items within a FC configuration,

b) to facilitate routing in situations where the FC configuration is part of a larger WAN, or where the ULPs being transported by FC (e.g., SCSI-3 or IEEE 802.2 LLC) assume the use of those long identifiers.

For the first use, it is acceptable that a single, unique IEEE 48-bit GID-48 be provided for each Fabric and each Node, and this is then used with a local qualifier to form a 60-bit IEEE Extended identifier to identify the specific port.

For the second use, it is strongly suggested that each N_Port and F_Port be assigned its own IEEE GID-48 bitidentifier where:

a) an application designed for an IEEE 802 LAN is moved with minimal modification into a FC environment,

b) communication occurs between an FC-LE Entity and an IEEE 802 station on a LAN, or

c) FC provides an intermediate part of a path between two IEEE 802 stations on LANs (i.e., conforms to the IEEE 802.1G remote bridging requirements).

In addition, when bridging techniques are being used to communicate between entities implementing FC-LE and systems on an IEEE 802 LAN, it is required that each N_Port and F_Port be assigned its own IEEE GID-48 to allow:

a) an FC-LE Entity to address the system on an IEEE 802 LAN, and

b) the system on an IEEE 802 LAN to use ULAs to address the FC-LE Entity.

Only one of the formats defined as usable for a WWN is an IEEE 48-bit format. This is specified to contain a 48-bit IEEE Standard 802.1A Universal LAN MAC Address (ULA). The ULA is represented as an ordered string of six bytes numbered from 0 to 5. The least significant bit of byte 0 is identified as the Individual/Group Address (I/G) bit. The next least significant bit is identified as the Universally or Locally Administered Address (U/L) bit. It is further specified that layout of the bytes in two FC standard 32-bit words is as follows:

Most Significant Bits				Least Significant Bits			
63 56	55 48	47 40	39 32	31 24	23 16	15 08	07 00
		ULA Byte 0	ULA Byte 1	ULA Byte 2	ULA Byte 3	ULA Byte 4	ULA Byte 5

Note that all fields in the FC-PH standard are defined as having the most significant bit in the highest numbered bit position, and the most significant byte in the position closest to the start of the FC-PH “packet”.

The Fibre Channel Arbitrated Loop topology (FC-AL) uses ULAs in the same manner and format as other Fibre Channel topologies (point-to-point, fabric).

D.3 Use of WWNs by this document

Per FC-PH 23.6.4 and 23.6.5, the N_Port_Name and Node_Name communicated in PLOGI, ADISC, and PDISC are 8 byte identifiers which are independent of any network addressing that may occur within the same node or N_Port. In other words, it cannot be assumed that identifiers communicated in the Network Header have any relationship to the identifiers communicated in the PLOGI, ADISC, and PDISC extended link services per this document.

Since the WWNs used are 60 bits, there are 4 unused bits in an 8-byte field. These unused bits identify the format of the WWN and shall be the most significant bits of the 64-bit identifier as illustrated in figure D.1. The format shown is the “IEEE Extended” format.

Within the 12-bit extension, bits 59:56 are used to identify a specific port associated with the node assigned the GID-48 identifier present if bits 47:00. Bits 55:48 of the 12-bit extension are set to zeros. A future version of FC-PH may change the defintion of the IEEE Extended format to allow bits 55:48 to be used as an extension to the Company assigned value in bits 23:00.

Figure D.1 – PLDA Use of IEEE Extended Format WWN

<table>
<tr><td colspan="3">Most Significant Bits</td><td colspan="2" align="right">Least Significant Bits</td></tr>
<tr><td>63 60</td><td>59 56</td><td>55 48</td><td>47 24</td><td>23 00</td></tr>
<tr><td rowspan="2">‘0010’b</td><td>Port Number</td><td>zeros</td><td>Company_ID</td><td>Company assigned value</td></tr>
<tr><td colspan="2">12-bit extension to GID-48</td><td colspan="2">GID-48</td></tr>
</table>

Annex E
(proposed normative)

Resource Recovery Timer Proposal

This annex describes a proposed change to the SCSI Primary Command document to add a definition of a Resource Recovery timer (RR_TOV). The RR_TOV is used to determine when a SCSI Target may terminate all open Exchanges with a specific SCSI Initiator and reclaim the resources associated with those Exchanges. This timer is used by this document following LIP to define the minimum amount of time that a SCSI Target waits for an initiator to perform Exchange Authentication (see 10.4).

It is proposed to use byte 13 of the Disconnect-reconnect mode page or a Fibre Channel specific mode page to implement this timer with the following definition:

Figure E.1 – Resource Recovery Timer

7	6	5	4	3	2	1	0
Time Units			Time Value				
000 = 10^{-6} seconds			1 to 32 microseconds				
001 = 10^{-5} seconds			10 to 320 microseconds				
010 = 10^{-4} seconds			100 to 3200 microseconds (3.2 milliseconds)				
011 = 10^{-3} seconds			1 to 32 miilliseconds				
100 = 10^{-2} seconds			10 to 320 miilliseconds				
101 = 10^{-1} seconds			100 to 3200 miilliseconds (3.2 seconds)				
110 = 10^{0} seconds			1 to 32 seconds				
111 = 10^{1} seconds			10 to 320 seconds (5 min. 20 seconds)				

Bits 7:5 define the time scale used to determine the timer value. Bits 4:0 define the actual timer value. A value of ‘00000’b in bits 4:0 indicates indicates that the timer is disabled and Resource Recovery shall not occur.

Annex F
(informative)

Hot Swap Considerations

An FC-AL system required continuity of the complete interconnected loop in order to maintain communication. Some method must be used to minimize the effect of any action that disrupts the loop's continuity. A common action that disrupts an active loop is the addition or removal of devices from the loop or connectors from a hub. This is generically described as 'device swapping'.

The purpose of this annex is to describe the issues related to minimizing the disruption to the loop interconnect during device swapping operations. The examples are intended to demonstrate possible approaches to these issues, but other approaches may be used.

F.1 Overview

A goal of many storage subsystems is to minimize the disruption caused at the application software level when various disturbances occur at the physical storage device level.

A subsystem that permits low-level (hardware) disruptions to affect application level I/O operations may be less than ideal for many environments. For example, if as a result of a device swapping operation, a device suddenly disappears for several seconds and then reappears at a different ID on the loop, the operating system's device driver has a difficult task to try to hide the underlying complexity from applications. If this recovery is to be done without input from an operator, then the device driver must recognize that the device is gone, determine whether to pause application I/Os or fail them, recognize that a replacement device has been inserted, and finally recognize the new device (possibly at a different ID) is a replacement for the previous drive.

Another example is the case where a multi-host clustered system is on one arbitrated loop. At a given point in time, there may be hundreds of commands queued at various points in the system. These commands may have been issued in a specific sequence which must be maintained in order to guarantee data integrity. Recovery from errors in this case is quite complex; the frequency of errors must be minimized.

A final example occurs if a device is simply removed from a system without notice. The system must rely on command timeouts or failures to detect removal so failover procedures can be initiated. The use of timeouts to control such a system is not desirable because they introduce user-visible latency.

Since device drivers typically perform several retries if a SCSI command fails, it will take some time until the driver recognizes that the device is actually no longer available. Since pending commands in the device may or may not have completed before the device was removed, the state of the data on the device is not defined.

A step back from full automation of the device-replacement scenario is one where a human operator performs certain notification steps to let the operating system know of the impending device swap. In response to the notification of the pending device removal, the device driver may flush its current outstanding commands (queued in the driver, the host adapter, the interconnect itself, and the device), spin down the device, and report that the device is now prepared for removal. A "swap manager" function is needed to coordinate the interaction between the operator, driver, and interconnect. This swap manager may reside in an application on a given host, in a distributed network application, or in a processor in a subsystem enclosure.

F.2 Bypass Circuits

Throughout this discussion it is assumed that a loop bypass circuit, or a hub, is used at all points where a device can be inserted into, or removed from, a loop. The question of bit level discontinuities is not considered since it is not directly relevant to the device swapping process, although retiming elements may be an integral part of a bypass function.

Concepts:

a) Bypass functions are needed to maintain loop continuity when a device is removed.

b) The function is typically mounted on a backplane or in a hub or device enclosure (not in the device itself!).

c) The bypass function may be associated with signal retiming circuits, both because of the need to provide signal retiming and also as a method of detecting that a device has failed or been removed.

d) The bypass function is activated either by the device to which it is attached, by a downstream retiming circuit, or by a signal line from some control function.

e) When the bypass function is active, a device may listen on the loop but not transmit.

F.3 Command Sequentiality Issues

FC-AL loops, by their nature preserve delivery ordering of all information sent at the lowest level. Thus, frames are not reordered as they might be in a Fibre Channel switched fabric. However, this low level characteristic does not guarantee that SCSI command sequentiality is preserved.

For example, if a Sequence delivery error occurs, the associated Sequence and Exchange are aborted according to the PLDA error policy. This causes the SCSI command (using the FCP protocol) to terminate unexpectedly. The SCSI initiator detects the error and may return an error status to the operating system or retry the command at the device driver level. In the meantime, additional commands that have already been sent by the driver may have been acted upon by the storage device.

For disk drives, and other random-access devices, this may not be a problem at the device level because the data may be directly accessed by the device in any order. Non-sequentiality is a problem in two areas, first, for those devices that have intrinsically sequential operation (such as stream devices or media movers), and second, for those system environment where SCSI command sequentiality is required. The latter case can occur when a write to a specific location is subsequently followed by another write to the same location. If an error occurs on the first write and the second write completes, retrying the first write would result in incorrect data on the device.

Certain application programs and operating system activities require “atomic” manipulation of a device. This means that the SCSI commands for the activity are processed in sequence with no other commands intervening at the device.

Using ordered command queueing as described in the SCSI-3 direct access device model on parallel busses, a “blocking” command may be issued. A pair of these commands define the boundaries of an atomic operation, and the queue of SCSI commands is managed by the device to meet the atomic requirement. Unordered commands cannot be executed at an unexpected time.

In an FC-AL environment, any command may be lost as described earlier. This is disastrous if the command that is lost is one of the blocking commands, since unordered commands may be mixed with ordered commands in an unpredictable way.

The operating system may manage sequentiality as follows. When an atomic operation is required the operating system stops issuing commands and waits until all commands have completed. Then, the first ordered command is issued and the operating system waits until the command completes. Following this, a second ordered command may be issued with the operating system waiting until that command completes. This process continues until the atomic operation is complete. This approach cripples the performance advantages of the SCSI-3 command queueing and incurs additional FC-AL overhead.

F.4 Device Swap Nomenclature

There are many different definitions of the terms “hot swap” or “hot plug”. For the purposes of this annex, the following definitions are used based on the electrical situation at the connector when the device is connected or disconnected:

warm swap: devices on the loop are powered, but all are quiescent (only signals at the device connector

hot swap: device to be swapped is quiescent, others may be active

yank swap: device to be swapped may be actively using the loop

From the system viewpoint there are two broad classes of device swap designs. The first is the minimalist approach where the operating system is not made aware in advance of the impending device swap operation and must recover from the swap on its own. The second class is where the operator informs the system in advance of the device swap operation so that the system may prepare for it.

In the latter case, there is a logical entity called the “swap manager” in the system that manages the swapping process. The swap manager may be integral with the host device driver or it may be located at some other point in the distributed storage system. For example, the swap manager might be an FC-AL node in a subsystem enclosure that has local connections to the devices in the enclosure as well as a loop connection to the drivers in the hosts. The swap manager may be located in a hub, or there may be a distributed implementation of the swap manager that shares subsystem configuration information between several physical locations. The actual implementation of a swap manager and communications with the manager may be vendor specific.

The swap manager may have a loop connection to the host, or may have a non FC-AL signal path to the devices. For the purposes of this annex, any non FC-AL signal path, either to a device or host is called an “out of band” signal.

In this annex, the above three definitions and two classes are reduced to two methods. The first, “uncontrolled swap method”, is electrical yank swap with no prior notice to the system. The second, “controlled swap method”, is electrical warm swap achieved by coordination control within a swap manager.

F.4.1 Uncontrolled Swap Method

In an uncontrolled swap, there is no advance notice to the system of the impending swap operation. The operator simply walks up to the cabinet or hub and removes the device or disconnects the connector. The host operating system(s) must recover from this event. Electrically, this method allows for a “best effort” attempt to minimize disruption at the connector, including use of the port bypass circuit and any automatic means of detecting that the device has been removed or inserted. Uncontrolled device swapping on a hub that does not have a swap manager is disruptive even if the hub has operator controls and indicators associated with activating the port bypass circuits since these controls do not notify the host of the impending removal.

An uncontrolled swap device removal causes loop continuity to be lost when the port bypass circuit is activated. This may cause the failure of one or more frames that attempt to pass through the point of removal resulting in an ULP timeout and associated recovery.

The downstream device may lose synchronization and require some time to regain synchronization, but this action may not trigger a loop initialization (LIP) unless the R_T_TOV period is exceeded (100 ms.).

Uncontrolled device insertion may also cause a disruption if deactivating the port bypass circuit causes one or more frames to be affected.

F.4.2 Controlled Swap Method

In the case of a controlled swap, the swap manager takes control of the loop to insure that there is no user information on the loop when the port bypass circuit is activated or deactivated. This is therefore an electrical warm swap situation.

There are two subsets of this method depending upon whether the communication between the device and swap manager is cone via the FC-AL loop or by a separate connection.

F.4.2.1 Loop Connection to Swap Manager

When the communication between the swap manager, port bypass circuits, and related hardware is done via the FC-AL loop, the swap manager may reside at any location that is serviced by the loop facilitating a distributed swap management system. A local “about to be removed” switch near the device could be wired to the device allowing the device to communicate this status to the remote swap manager. A local indicator could be used to inform the operator of the status of the loop and/or port bypass circuit. The swap manager would control this indicator by communicating with the device.

F.4.2.2 Out-of-Band Connection to Swap Manager

In this case the communication between the swap manager and port bypass circuits and related hardware is done via some connection other than the FC-AL loop. This could be a connection between each bypass circuit and a central manager in each enclosure, or it could be via an Ethernet or serial line connection to a remote manager or terminal. A local “about to be removed” switch near the device could be wired directly to the swap manager along with any indicator lights.

F.5 Device Swap Procedures

Assume a running loop with in-progress activity passing by a currently bypassed position with no attached device.

F.5.1 Device Insertion

a) Operator plugs in the device using a SFF-8045 device, HSSDC connector or DB-9 connector to a hub. No prior notice to the system is needed when using either method since the port bypass circuit remains enabled.

b) The new device is powered on via the SFF-8045 (or SFF-8067) connector pins or other method. The port bypass circuit remains enabled until the device performs its internal self-tests.

c) Device performs its internal Power-On Self Test (POST). The action taken upon successful completion of the self test depends upon whether an uncontrolled or controlled swap method has been chosen.

d) Uncontrolled swap insertion method:

1) Device disables the port bypass circuit. This is triggered by the completion of POST. The device uses the appropriate SFF-8045 or SFF-8067 signal line to control the bypass circuit. In a hub environment, the port bypass circuit is deactivated by a switch or loss-of-signal detector.

2) Disabling the port bypass circuit disrupts the loop (this disruption may or may not cause a loop initialization).

3) Device enters the INITIALIZING state and begins transmitting LIP. Deactivating the port bypass circuit and transmitting LIP causing one or more frames to be lost or discarded.

4) As a result of the lost or damaged frames one or more sequence failures may be detected following completion of loop initialization. This causes the Exchange to be terminated due to detection of the error or an ULP timeout.

5) The operating system performs the appropriate recovery action(s) to reposition the device, if appropriate, and retry any failed commands.

e) Controlled Swap Insertion Method

1) Device waits for activation by the swap manager (this is not currently specified by FC-AL which assumes that the device will begin loop initialization upon completion of POST).

2) Operator notifies the swap manager that the device has been inserted by means of a control or console command. There are a number of means by which the swap manager can be informed that the device has been inserted. They include:

i) A push-button is located close to the drive with an electrical connection to the device through a device connector. When the push-button is depressed, the device sends an “I’m here” message to the swap manager. This tells the host to prepare to quiesce the loop.

ii) When the device finishes its POST, it proactively sends an “I’m here” announcement message to the swap manager.

iii) In the case of a hub, the push-button is close to the connector for the remote sub-loop, and when depressed sends an “I’m here” message to the swap manager.

iv) If the push-button is on the panel of a subsystem enclosure, the subsystem may send a global “I’m here” message to the swap manager.

3) The swap manager arbitrates for the loop in order to insure that there are no frames on the loop when the port bypass circuit is deactivated. Upon winning arbitration, the swap manager sends a Loop Port Enable (LPEyx) primitive sequence to the device (note that this requires that the device have an AL_PA such as an assumed hard assigned AL_PA or the Loop Port Enable All (LPE(fx) primitive sequence be used).

4) The device deactivates the port bypass circuit. The downstream device experiences a phase discontinuity due to deactivation of the port bypass circuit. Synchronization may or may not be lost as a result. If synchronization is lost, it will be regained from the LPE primitive sequence.

5) The added device may initiate loop initialization by transmitting one of the LIP primitive sequences (mode page settings may direct it not to initiate loop initialization). Loop initialization may be used to acquire an AL_PA and provide notification that the configuration has changed.

6) Loop initialization passes a 128-bit AL_PA bit map around the loop four times. Each time around, the bit map is used to accumulate AL_PA assignment information from the L_Ports.

i) The LIFA sequence is used to gather Fabric assigned AL_PA values.

ii) The LIPA sequence gathers Previously assigned AL_PA values.

iii) The LIHA sequence gathers Hard assigned AL_PA values (i.e., set by backpanel wiring or address switches).

iv) The LISA sequence gathers Soft assigned AL_PA values.

7) As a result of the AL_PA assignment process, those devices that had either a AL_PA value prior to initialization use slot-related retain those AL_PA values (Fabric sustained or Previously assigned values). Devices which have a Hard Assigned AL_PA value (such as might be assigned via the device connector such as described by SFF-8045 or other similar specification) use that value when they first acquire an AL_PA. The previously assigned AL_PA value is probably not retained when device power is removed, although FC-AL allows it to either be retained or lost.

8) Following loop initialization, every L_Port has a unique AL_PA and is in the MONITORING state (or is in the non-participating mode). This allows normal loop operations from an FC-AL level, but additional activity may be required before SCSI operations can resume. See Target Discovery, 10.3 and Exchange Authentication 10.4 for examples. After completion of any necessary post initialization processing, operations with devices other than the newly inserted one may resume.

9) The newly inserted device performs any necessary login procedures (i.e., Fabric Login, N-Port Login, FC-4 Login) and can then begin operations.

F.5.2 Device Removal

Assume a currently running loop with operations in progress and a frame in transit through a currently disabled port bypass circuit with a connected L_Port.

a) Controlled Swap Removal:

1) Operator notifies the system of the impending removal by means of an operator control or system command. The swap manager negotiates with the device driver to insure that all pending commands has been quiesced and any necessary cache management or buffer management has been completed.

2) The swap manager arbitrates for the loop, and upon winning arbitration send an Loop Port Bypass LPB(yx) primitive sequence to the L_Port (in a hub environment, some hubs may provide the capability to specify a bypass function within the hub).

3) The device or hub activates the port bypass circuit. This may cause a phase discontinuity at the next downstream device, but synchronization can be regained using the LPB primitive sequence.

4) The swap manager ends transmission of the LPB sequence and relinquishes control of the loop allowing operations to resume. No recovery or discovery process is necessary.

b) Uncontrolled Swap Removal

1) Operator does not provide notice to the system of the impending removal, but instead simply removes the device from the loop. Loop initialization may or may not occur as a result of removing the device since the interruption is short enough that a loop failure may not result.

2) One or more in-progress frames may be corrupted due to the removal.

3) In Class 1 or Class 2, the error probably results in a sequence timeout due to the lost or corrupted frames. In Class 3, the error may be detected through a ULP timeout. In either case, the ULP recognizes that the device no longer exists and ceases operations with that device.

4) Since device removal means that there is now an additional AL_PA available on the loop, it may be desirable to perform a loop initialization to allow a non-participating device to attempt to acquire an AL_PA and join loop operations. The loop initialization would be initiated by the swap manager in the host or controller.

F.6 Summary

The arbitrated loop topology imposes addition considerations on the addition and removal of devices to the configuration. This is due to the fact that all devices attached to the loop are ‘in-series’ with the flow of information and inserting or removing devices disturbs that flow. With the proper attention to the insertion or removal process, interruption of that flow can be minimized allowing devices to be added to a loop, or removed from a loop, with a minimum of disruption.

U

W

X