draft proposed
American National Standard

T10
Project 1144D

Revision 8
September 23, 2002

Information Technology -
Fibre Channel Protocol for SCSI, Second Version (FCP-2)

This is a draft proposed American National Standard of Accredited Standards Committee NCITS. As such this is not a completed standard. The NCITS T10 Technical Committee may modify this document as a result of comments received during public review and its approval as a standard. Use of the information contained herein is at your own risk.

Technical editor:

Robert Snively
Brocade Communications Systems, Inc.
1745 Technology Drive
San Jose, CA 95110
(408) 487-8135
E-mail: rsnively@brocade.com

ISO LOGO

IEC LOGO

Reference number
ISO/IEC xxxxx: 2001
ANSI NCITS.xxx: 2001
Printed October 23, 2002

POINTS OF CONTACT:

NCITS T10 Chair

John Lohmeyer
LSI Logic
4420 Arrows West Drive
Colorado Springs, CO 80907
Tel: (719) 533-7560
Fax: (719) 533-7183
E-mail: lohmeyer@t10.org

NCITS T10 Vice Chair

George O. Penokie
Tivoli Systems
3605 Highway 52 N. MS 2C6
Rochester, MN 55901
Tel: (507) 253-5208
Fax: (507) 253-2880
E-mail: gpenokie@tivoli.com

NCITS Secretariat

NCITS Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005
Tel: 202-737-8888
Fax: 202-638-4922
E-mail: ncits@itic.org

T10 Reflector

Internet address for subscription to T10 reflector: majordomo@t10.org
The message body should be: subscribe t10

Internet address for distribution via SCSI reflector: 10@t10.org

Web site for T10.org information: http://www.t10.org

Document Distribution

Global Engineering
15 Inverness Way East
Englewood, CO 80112-5704
Telephone: 303-792-2181 or 800-854-7179
Facsimile: 303-792-2192

ABSTRACT

This standard describes the frame format and protocol definitions required to transfer commands and data between a SCSI (Small Computer System Interface) initiator and target using the Fibre Channel family of standards. The second version adds optional retransmission, task ordering, and confirmation capabilities.

PATENT STATEMENT

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard. As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, notice of one or more claims has been received.

By publication of this standard, no position is taken with respect to the validity of this claim or of any rights in connection therewith. The known patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Contents

Tables

Figures

Foreword

(This foreword is not part of BSR NCITS 350.)

The Fibre Channel Protocol for SCSI, Second version (FCP-2) defines a Fibre Channel mapping layer (FC-4) that uses the services defined by NCITS Project 1311D, *Fibre Channel Framing and Signaling Interface (FC-FS)* to transmit SCSI command, data, and status information between a SCSI initiator and a SCSI target. The use of the standard enables the transmission of standard SCSI command formats, the transmission of standard SCSI data and parameter strings, and the receipt of SCSI status and sense information across the Fibre Channel using only the standard Fibre Channel frame and sequence formats. The Fibre Channel protocol operates with Fibre Channel Classes of Service 1, 2, and 3 and operates across Fibre Channel fabrics and arbitrated loops.

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

This standard was processed and approved for submittal to ANSI by National Committee for Information Technology Standards (NCITS). Committee approval of the standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, NCITS had the following members:

NCITS Technical Committee T10 on Lower-Level Interfaces, which developed and reviewed this standard, had the following members:

John B. Lohmeyer, Chair

George O. Penokie, Vice-Chair

Ralph O. Weber, Secretary

Paul D. Aloisi
Tim Bradshaw
Charles Brill
Zane Daggett
Robert C. Elliott
Mark Evans
Leroy Fong
Bill Galloway
Edward A. Gardner
Robert Griswold
Randy Haagens
Edward Haske
Nathan Hastad
Gerald Houlder
Peter Johansson
Skip Jones
Tasuku Kasebayashi
Thomas J. Kulesza
Eugene Lew
Bill Mable
William P. McFerrin
Pete McLean
Kenneth Moe
Charles Monia
Dennis Moore
Jay Neer
Terence J. Nelson
Robert H. Nixon
Erich Oetting
David Peterson
Doug Piper
Bart Raudebaugh
Ron Roberts
Gary S. Robinson
Cris Simpson
Robert Snively
Ronald Stockford
Charles Tashbook
Mike Taylor
William C. Terrell
Douglas Wagner
Neil Wanamaker
Donald Woelz
I. Dal Allan (Alt.)
Tim Anderson (Alt.)
David Black (Alt.)
Robert Canniff (Alt.)
Doug Charnley (Alt.)
Terry Enright (Alt.)
Jie Fan (Alt.)
Donald R. Getty (Alt.)
Chuck Grant (Alt.)
Glen Griessler (Alt.)
William Ham (Alt.)
Rob Haydt (Alt.)
Emily Hill (Alt.)
Randall C. Hines (Alt.)
Titkwan Hui (Alt.)
Tom Jackson (Alt.)
David L. Jolley (Alt.)
Michael Karg (Alt.)
Jim Koser (Alt.)
Ben-Koon Lin (Alt.)
Tim Mackley (Alt.)
Kevin Marks (Alt.)
John Masiewicz (Alt.)
Patrick McGarrah (Alt.)
James McGrath (Alt.)
Brian McKean (Alt.)
E.J. Mondor (Alt.)
Richard Moore (Alt.)
Robert Morris (Alt.)
Franklin Ng (Alt.)
Vit Novak (Alt.)
William Petty (Alt.)
Darrell Redford (Alt.)
Chuck Rice (Alt.)
Charley Riegger (Alt.)
John P. Scheible (Alt.)
Paresh Sheth (Alt.)
Pete Tobias (Alt.)
John Tyndall (Alt.)
Rudolf Vitti (Alt.)
Dean Wallace (Alt.)
Michael Wingard (Alt.)

The NCITS Technical Committee T11 on Device Level Interfaces contributed a great deal of analysis, review, and content to this standard. It had the following members:

Kumar Malavalli, Chair

Ed Grivna, Vice-Chair

Neil Wanamaker, Secretary

Rick Allison
David Baldwin
Doug Berry
Charles Brill
Craig Carlson
Edward S. Chang
James Coomes
Zane Daggett
Jan V. Dedek
David Deming
Mike Fitzpatrick
David Ford
Michael S. Foster
Edward M. Frymoyer
Matt Gaffney
Wayne Gentry
Edward Grivna
Joe Gruba
Bill Ham
Norman Harris
David E. Instone
Mike Jensen
Michael J. Karg
Albert F. Kelley
Robert W. Kembel
Laura Kendrick
Bret Ketchum
Karl Kwiat
Neil MacLean
Kumar Malavalli
William R. Martin
Thomas Merphy
Ken Moe
Charles Monia
Dennis P. Moore
Hari Naidu
Jay H. Neer
James Nelson
Michael O'Donnell
Harry V. Paul
Robert K. Pedersen
David Peterson
Craig Prunty
Venkat Rangan
Curtis A. Ridgeway
Gary S. Robinson
Elizabeth G. Rodriguez
Roger Ronald
Roger Rose
Kenneth Samarra
John Scheible
John Schroeder
Pak Seto
Predrag Spasic
Gary R. Stephens
Craig Stuber
Etsuji Sugita
Tad Szostak
Rich Taborek
Luis Torres
Horst Truestedt
Adrienne Turenne
Fred Van Roessel
Matt Wakeley
Neil Wanamaker
Gary Warden
Naoki Watanabe
Kevin White
Michael Wingard
Leonard Young
Tatsua Arai (Alt.)
Bob Atkinson (Alt.)
Charles Binford (Alt.)
Ronald Bossard (Alt.)
Matt Brown (Alt.)
Terry Cobb (Alt.)
Doug Coleman (Alt.)
Robert Crawford (Alt.)
Kenneth Davis (Alt.)
Mike Dorsett (Alt.)
Bob Dugan (Alt.)
Neil Edmunds (Alt.)
Jie Fan (Alt.)
Marc Farley (Alt.)
Steve Farmer (Alt.)
Roy Funderburk (Alt.)
Ali Ghiasi (Alt.)
Gary Gold (Alt.)
Mark Hamel (Alt.)
Shin'ichi Iwano (Alt.)
Skip Jones (Alt.)
Julie Ann Kembel (Alt.)
Scott Kipp (Alt.)
Gregory Koellner (Alt.)
Michael Lamatsch (Alt.)
Alan Land (Alt.)
Tom Lindsay (Alt.)
Bill Lynn (Alt.)
Bill Mable (Alt.)
Roland Marbot (Alt.)
Chuck Mcknett (Alt.)
Greg McSorley (Alt.)
Kevin Monroe (Alt.)
Mike Morandi (Alt.)
David Neil (Alt.)
Bob Nixon (Alt.)
Stephen O'Neil (Alt.)
Robert Pearson (Alt.)
Ken Peters (Alt.)
Bart Raudebaugh (Alt.)
Bob Reynolds (Alt.)
Wayne Rickard (Alt.)
Suresh Roy (Alt.)
Frank Samela (Alt.)
Edward R. Schurig (Alt.)
Paul Suhler (Alt.)
Phillip Swart (Alt.)
Lloyd E. Thorsbakken (Alt.)
Jeanne Truestedt (Alt.)
Schelto van Doorn (Alt.)
Ralph Weber (Alt.)
Lise Wilson (Alt.)
Steven L. Wilson (Alt.)
Ray Yang (Alt.)
Bill Collette (Alt.)
Joe Graham (Alt.)
Jack E. Keller (Alt.)
John Lohmeyer (Alt.)
Taisei Miyake (Alt.)
George Penokie (Alt.)
Robert Snively (Alt.)
Brad Solomon (Alt.)
Jeffrey Stai (Alt.)

Introduction

The Small Computer System Interface (SCSI) command set is widely used and applicable to a wide variety of device types. The transmission of SCSI command set information across Fibre Channel links allows the large body of SCSI application and driver software to be successfully used in the high performance Fibre Channel environment.

This standard, *Fibre Channel Protocol for SCSI, Version 2* (FCP-2), describes the protocol for transmitting SCSI commands, data, and status using Fibre Channel FC-FS Exchanges and Information Units. Fibre Channel is a high speed serial architecture that allows either optical or electrical connections at data rates from 265 Mbits up to 4 Gbits per second. The topologies supported by Fibre Channel include point-to-point, fabric switched, and arbitrated loop. All Fibre Channel connections use the same standard frame format and standard hierarchy of transmission units to transmit the Information Units that carry SCSI information.

Fibre Channel Protocol for SCSI, Second Version (FCP-2) standard is divided into 12 clauses:

Clause 1 is the scope of this standard.

Clause 2 enumerates the normative references that apply to this standard.

Clause 3 describes the definitions, abbreviations, and conventions used in this standard.

Clause 4 provides an overview of the protocol for transmitting SCSI information over Fibre Channel.

Clause 5 describes the Information Units used to transfer SCSI commands, data, and status across a Fibre Channel connection.

Clause 6 describes the Basic Link Services and Extended Link Services used by the protocol for transmitting SCSI information over Fibre Channel.

Clause 7 describes the FC-GS-3 Name Server objects defined for FCP-2.

Clause 8 describes the FCP FC-4 Link Service definitions for the protocol for transmitting SCSI information over Fibre Channel.

Clause 9 describes the details of the Information Unit formats.

Clause 10 defines the SCSI management features for Fibre Channel, including the SCSI mode pages used by the protocol for transmitting SCSI information over Fibre Channel.

Clause 11 defines the timers used for FCP-2 error recovery algorithms.

Clause 12 defines the error recovery algorithms for FCP-2.

The Fibre Channel Protocol for SCSI, Second revision (FCP-2) standard has six annexes.

Annex A is a normative description of the relationship between the services defined by SAM-2 and the corresponding functions defined by this standard.

Annex B is an informative annex that provides examples of the protocol for transmitting SCSI information over Fibre Channel.

Annex C is an informative annex providing examples of the FCP-2 error recovery mechanisms.

Annex D is an informative annex describing techniques for discovering SCSI device capabilities over Fibre Channel.

Annex E is an informative annex providing examples of the content of ELSs used during FCP-2 recovery operations.

Annex F is an informative annex describing a mechanism to support bidirectional SCSI data transfer using SCSI commands.

Fibre Channel Protocol for SCSI, Second Version (FCP-2) is part of the SCSI family of standards developed by T10 to facilitate the use of the SCSI command sets for many different types of devices across many different types of physical interconnects. The architectural model for the family of standards is NCITS Project 1157D, *Information Technology - SCSI Architecture Model - 2 (SAM-2).*

American National Standard
for Information Technology –

Fibre Channel Protocol for SCSI, Second version (FCP-2)

1 Scope

This standard defines a second version of the SCSI Fibre Channel Protocol (FCP). This standard is a mapping protocol for applying the SCSI command set to Fibre Channel. This standard defines how the Fibre Channel services and the defined Information Units (IUs) are used to perform the services defined by the SCSI-3 Architecture Model - 2 (SAM-2). This second version includes additions and clarifications to the first version, removes information that is now contained in other standards, and describes additional error recovery capabilities for the Fibre Channel protocol.

2 Normative references

2.1 Qualification and availability of references

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

Copies of the following standards may 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) or via the World Wide Web at http://www.ansi.org.

Reference is made to these standards by their standard designations.

Additional availability contact information is provided below as needed.

2.2 Published standard references

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

ANSI NCITS 332:1999, *Fibre Channel Arbitrated Loop - 2 (FC-AL-2)*

NCITS TR-19:1998*, Fibre Channel Private Loop, SCSI Direct Attach (FC-PLDA) (Technical Report - Not an American National Standard)*

NCITS TR-20:1998*, Fibre Channel Fabric Loop Attachment (FC-FLA) (Technical Report - Not an American National Standard)*

NCITS TR-24:1999, *Fibre Channel - Tape Technical Report (FC-TAPE) (Technical Report - Not an American National Standard)*

2.3 References under development

At the time of publication, the following referenced standards were still under development by NCITS T10 and NCITS T11. For information on the current status of the standard, or regarding availability, contact the NCITS Secretariat, Information Technology Industry Council (ITI) at 202-737-8888 (phone) or by mail at 1250 Eye Street NW, Suite 200, Washington, DC 20005-3922.

NCITS Project 1331D, *Fibre Channel Framing and Signaling Interface (FC-FS)*

NCITS Project 1356D, *Fibre Channel Generic Services - 3 (FC-GS-3)*

NCITS Project 1157D, *SCSI Architecture Model - 2 (SAM-2)*

INCITS Project 1561D, *SCSI Architecure Model - 3 (SAM-3)*

NCITS Project 1236D, *SCSI Primary Commands - 2 (SPC-2)*

NCITS Project 1416D, *SCSI Primary Commands - 3 (SPC-3)*

Copies of these NCITS T10 and T11 draft standards are available for purchase from Global Engineering Documents. For further information, contact Global Engineering Documents at 800-854-7179 (phone) or 303-792-2181 (phone) or by mail at 15 Inverness Way East, Englewood, CO 80122-5704. The NCITS T10 draft standards are also available on the web site www.t10.org. The NCITS T11 draft standards are also available on the web site www.t11.org.

2.4 Other references

The following references are the product of the SFF committee. For information on the current status and availability of the documents, contact the SFF committee at 408-867-6630 (phone) or by mail at 14426 Black Walnut Court, Saratoga, CA 95070.

SFF document SFF-8067*, 40-pin SCA-2 Connector w/Bidirectional ESI*

3 Definitions, abbreviations and conventions

3.1 Definitions

3.1.1 access controls: Mechanisms allowing a managing application client to control the set of initiators that have access to a target. The access control is enforced by the target. See SPC-3.

3.1.2 access controls data: Information sent to the target by the managing application client that is used by the target to control the set of initiators that have access to the target. See SPC-3.

3.1.3 access controls enrollment state: A state established in the target by the managing application client. The state governs the behavior of the target in controlling the set of initiators that have access to the target. See SPC-3.

3.1.4 acknowledged class: Any class of service that acknowledges transfers. Acknowledged classes of service include Class 1, Class 2, and Class 4 service. See FC-FS.

3.1.5 address identifier: An address value used to identify source (S_ID) or destination (D_ID) of a frame. See FC-FS.

3.1.6 application client: An object that is the source of SCSI commands. See SAM-2.

3.1.7 application client buffer offset: Offset in bytes from the beginning of the application client’s buffer (data-in or data-out) to the location for the transfer of the first byte of a data delivery service request. See SAM-2.

3.1.8 autosense data: Sense data (see 3.1.51) that is returned in the FCP_RSP IU payload. See SAM-2.

3.1.9 command: A request describing a unit of work to be performed by a device server. See SAM-2.

3.1.10 command descriptor block: A structure used to communicate a command from an application client to a device server. See SAM-2.

3.1.11 data buffer size: Upper limit on the extent of the data (data-in or data-out) to be transferred by the SCSI command. See SAM-2.

3.1.12 data in delivery service: A confirmed service used by the device server to request the transfer of data to the application client. See SAM-2.

3.1.13 data out delivery service: A confirmed service used by the device server to request the transfer of data

from the application client. See SAM-2.

3.1.14 data overlay: The use of random buffer access capability where data is transmitted using the same application client buffer offset more than one time during the set of delivery actions performed by a single command. See SAM-2.

3.1.15 Destination_Identifier: The address identifier used to indicate the destination of the transmitted frame. See FC-FS.

3.1.16 device server: An object within the logical unit that executes SCSI tasks and enforces the rules for task management. See SAM-2.

3.1.17 discard: The term used in FC-FS to describe removing a frame or sequence from the destination buffer without making use of the frame or sequence and without notifying upper layers of the receipt of the frame or sequence. See FC-FS.

3.1.18 Exchange: The basic mechanism which transfers information consisting of one or more related non-concurrent Sequences which may flow in the same or opposite directions. The Exchange is identified by an Originator Exchange_ID (OX_ID) and a Responder Exchange_Identifier (RX_ID). See FC-FS.

3.1.19 Execute Command service: A peer-to-peer, confirmed service requested by the application client to perform a SCSI Command. See SAM-2.

3.1.20 FCP Exchange: A SCSI I/O Operation for the Fibre Channel FC-2 layer. The SCSI I/O Operation for Fibre Channel is contained in a Fibre Channel Exchange. See FC-FS and 4.1.

3.1.21 FCP I/O operation: A SCSI I/O Operation for the Fibre Channel FC-4 layer, as defined in this standard.

3.1.22 FCP_Port: An N_Port or NL_Port that supports the SCSI Fibre Channel Protocol.

3.1.23 fully qualified exchange identifier: A set of addresses and values used to uniquely identify an FCP I/O operation. See 5.1.

3.1.24 image pair: The originating and responding processes related by a Process Login operation. For the Fibre Channel protocol, the image pair is composed of one initiator and one target. See FC-FS.

3.1.25 I_T nexus loss: A condition resulting from the events defined by SAM-3 in which the SCSI device performs the operations described in SAM-3 and this standard.

3.1.26 I_T nexus loss event: A SCSI transport protocol specific event that triggers I_T nexus loss as described in SAM-3.

3.1.27 Information Unit: An organized collection of data specified by the Fibre Channel protocol to be transferred as a single Sequence by the Fibre Channel service interface. See FC-FS.

3.1.28 initiator: A SCSI device containing application clients that originate device service requests and task management functions to be processed by a target SCSI device. In this standard, the word “initiator” also refers to an FCP_Port using the Fibre Channel protocol to perform the SCSI initiator functions defined by SAM-2.

3.1.29 initiator identifier: Address a target uses to identify the initiator device. See SAM-2.

3.1.30 interconnect tenancy: The period of time that an FCP device owns or may access a shared Fibre Channel link such as an FC-AL loop. See 10.2.1.

3.1.31 logical unit: A target resident entity that implements a device model and processes SCSI commands sent by an application client. See SAM-2.

3.1.32 logical unit number: An encoded 64-bit identifier for a logical unit. See SAM-2.

3.1.33 loop initialization primitive: A primitive used in Fibre Channel arbitrated loops to start loop

initialization. See FC-AL-2.

3.1.34 Name_Identifier: A 64-bit identifier, with a 60-bit value preceded with a 4 bit Network_ Address_Authority Identifier, used to identify entities in Fibre Channel such as N_Port, Node, F_Port, or Fabric. See FC-FS.

3.1.35 Name Server: A Fibre Channel service accessed through a well-known address identifier that uses the Common Transfer (CT) protocol as defined in FC-GS-3 to allow a client to determine the address identifier and properties of devices attached to a Fibre Channel switching fabric. See FC-GS-3.

3.1.36 Node_Name: A Name_Identifier associated with a Node. See FC-FS.

3.1.37 NL_Port: An N_Port that contains arbitrated loop functions associated with the Fibre Channel arbitrated loop topology. See FC-AL-2.

3.1.38 N_Port: A hardware entity that supports the FC-FS FC-2 layer. It may act as an Originator, a Responder, or both. See FC-FS.

3.1.39 Originator: The logical function associated with an N_Port responsible for originating an Exchange. See FC-FS.

3.1.40 Originator Exchange Identifier: An identifier assigned by an Originator to identify an Exchange (see 3.1.50). See FC-FS.

3.1.41 Port Identifier: An address identifier, assigned to an N_Port or NL_Port during implicit or explicit Fabric Login. See FC-GS-3.

3.1.42 Port (N_Port) Login (PLOGI): The Fibre Channel Extended Link Service (ELS) that exchanges identification and operation parameters between an originating N_Port and a responding N_Port. See FC-FS.

3.1.43 Port_Name: A Name_Identifier associated with an N_Port or an NL_Port.

3.1.44 private loop: A loop operating with no attached fabric loop ports. See FC-PLDA.

3.1.45 public loop: A loop operating with an attached fabric loop port. See FC-FLA.

3.1.46 random buffer access: The occurrence of device server data transfer requests that request data transfers to or from segments of the application client's buffer with an arbitrary offset and extent. See SAM-2.

3.1.47 recovery abort: An FC-FS protocol that recovers FCP_Port resources by terminating the Exchange and FCP I/O operation. See 12.3.

3.1.48 request byte count: The number of bytes to be moved by a data delivery service request. See SAM-2.

3.1.49 Responder: The logical function in an N_Port responsible for supporting the Exchange initiated by the Originator in another N_Port. See FC-FS.

3.1.50 Responder Exchange Identifier: An identifier assigned by a Responder to identify an Exchange and meaningful only to the Responder. See FC-FS.

3.1.51 sense data: Data returned to an application client as a result of an autosense operation, asynchronous event report, or REQUEST SENSE command. See SPC-2.

3.1.52 Sequence: A set of one or more Data frames with a common Sequence_ID (SEQ_ID), transmitted unidirectionally from one N_Port to another N_Port with a corresponding response, if applicable, transmitted in response to each Data frame. See FC-FS.

3.1.53 Source_Identifier: The address identifier used to indicate the source port of the transmitted frame. See FC-FS.

3.1.54 SCSI device: A device that originates or services SCSI commands. See SAM-2.

3.1.55 SCSI I/O operation: An operation defined by a SCSI command, a series of linked SCSI commands or a task management function. See SAM-2.

3.1.56 status: A single byte returned by the device server to the application client to indicate the completion and completion state of a command. See SAM-2.

3.1.57 tag: The initiator-specified component of a task identifier that uniquely identifies one task among the several tasks coming from an initiator to a logical unit. The fully qualified exchange identifier performs the function of the SCSI tag in this standard. See 5.1 and SAM-2.

3.1.58 target: A SCSI device that receives SCSI commands and directs such commands to one or more logical units for execution. In this standard, the word “target” also refers to an FCP_Port using the Fibre Channel protocol to perform the SCSI target functions defined by SAM-2.

3.1.59 target identifier: Address of up to 64-bits that identifies a target. See SAM-2.

3.1.60 task: An object within the logical unit representing the work associated with a command or group of linked commands. See SAM-2.

3.1.61 task attribute: The queuing specification for a task (SIMPLE, ORDERED, HEAD OF QUEUE, ACA). See SAM-2.

3.1.62 task identifier: The information uniquely identifying a task. See SAM-2.

3.1.63 task management function: A peer-to-peer confirmed service provided by a task manager that may be invoked by an application client to affect the execution of one or more tasks. See SAM-2.

3.1.64 Worldwide_Name: A Name_Identifier which is worldwide unique, and represented by a 64-bit unsigned binary value. (FC-FS)

3.2 Abbreviations

CRN	Command Reference Number (See 9.1.2.2.)
D_ID	Destination_Identifier (See FC-FS.)
ELS	Extended Link Service (See FC-FS.)
FC	Fibre Channel (See FC-FS.)
FC-AL-2	NCITS 332:1999, *Fibre Channel Arbitrated Loop - 2* (See 2.2.)
FC-FLA	NCITS TR-20*, Fibre Channel Fabric Loop Attachment* (See 2.2.)
FC-FS	NCITS Project 1311D, *Fibre Channel Framing and Signaling Interface* (See 2.3.)
FC-GS-3	NCITS Project 1356D, *Fibre Channel Generic Services - 3* (See 2.3.)
FC-PLDA	NCITS TR-19*, Fibre Channel Private Loop, SCSI Direct Attach* (See 2.2.)
FC-TAPE	NCITS TR-24:1999, *Fibre Channel - Tape Technical Report* (See 2.2.)
FCP	X3.269-1996*, Fibre Channel Protocol for SCSI* (See 2.2.)
	Also: referring both to FCP and to this standard
FCP-2	This standard (Fibre Channel Protocol for SCSI, second version.)
FC-4	Fibre Channel Layer 4 mapping layer. (See FC-FS.)

FLOGI	Fabric Login Extended Link Service. (See FC-FS.)
FQXID	fully qualified exchange identifier (See 3.1.23.)
ID	identifier
IU	Information Unit (See FC-FS.)
LIFA	loop initialization fabric assigned, a step of the loop initialization sequence (See FC-AL-2.)
LIHA	loop initialization hard assigned, a step of the loop initialization sequence (See FC-AL-2.)
LIP	loop initialization primitive (See FC-AL-2.)
LIPA	loop initialization previously assigned, a step of the loop initialization sequence (See FC-AL-2.)
LISA	loop initialization soft assigned, a step of the loop initialization sequence (See FC-AL-2.)
LISM	loop initialization select master, a step of the loop initialization sequence (see FC-AL-2.)
LOGO	Log Out Extended Link Service (See FC-FS.)
NA	Not Applicable
OX_ID	Originator Exchange Identifier (See FC-FS.)
PLOGI	Port (N_Port) Login Extended Link Service (See FC-FS.)
RX_ID	Responder Exchange Identifier (See FC-FS.)
SAM-2	NCITS Project 1157D, *SCSI Architecture Model - 2* (See 2.3.)
SAM-3	INCITS Project 1561D, *SCSI Architecture Model - 3* (See 2.3.)
SCSI	Small Computer System Interface, any revision.
SCSI-3	Small Computer System Interface-3, the SCSI architecture specified by SAM-2 and extended by the companion standards referenced in SAM-2.
S_ID	Source_Identifier (See FC-FS.)
SPC-2	NCITS Project 1236D, revision 14*, SCSI Primary Commands - 2* (See 2.3.)
ULP	upper layer protocol (See FC-FS.)

3.3 Keywords

3.3.1 expected: A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

3.3.2 invalid: A keyword used to describe an illegal or unsupported bit, byte, word, field or code value. Receipt of an invalid bit, byte, word, field or code value shall be reported as error.

3.3.3 ignored: A keyword used to describe a bit, byte, word, field or code value that shall not be examined by the receiving SCSI device. The bit, byte, word, field or code value has no meaning in the specified context.

3.3.4 mandatory: A keyword indicating an item that is required to be implemented as defined in this standard.

3.3.5 may: A keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may

not").

3.3.6 may not: A keyword that indicates flexibility of choice with no implied preference (equivalent to "may or may not").

3.3.7 obsolete: A keyword indicating that an item was defined in a prior SCSI standard but has been removed from this standard.

3.3.8 optional: A keyword that describes features that are not required to be implemented by this standard. However, if any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard.

3.3.9 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients are not required to check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as an error.

3.3.10 restricted: A keyword referring to bits, bytes, words, and fields that are set aside for use in other SCSI standards. A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field for the purposes of the requirements defined in this standard.

3.3.11 shall: A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard. This standard prescribes no specific response by a component if it receives information that violates a mandatory behavior.

3.3.12 should: A keyword indicating flexibility of choice with a strongly preferred alternative; equivalent to the phrase "it is strongly recommended".

3.4 Editorial conventions

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in 3.1 or in the body of the standard where they first appear. Names of commands, statuses, task management functions, task attributes, Information Units, sense keys, additional sense codes, and additional sense code qualifiers are in all uppercase (e.g., REQUEST SENSE). Lowercase is used for words having the normal English meaning.

The names of fields are in small uppercase (e.g., ALLOCATION LENGTH). When a field name is a concatenation of acronyms, uppercase letters may be used for readability (e.g., NORMACA). Normal case is used when the contents of a field are being discussed. Fields containing only one bit are usually referred to as the NAME bit instead of the NAME field. Where fields defined in another standard are referenced in this standard, the capitalization conventions of the originating standard are used.

Numbers that are not immediately followed by lower-case b or h are decimal values.

Numbers immediately followed by lower-case b (0101b) are binary values.

Numbers or upper case letters immediately followed by lower-case h (FA23h) are hexadecimal values.

In all of the figures, tables, and text of this standard, the most significant bit of a binary quantity is shown on the left side. Bit order and byte order are as specified in FC-FS.

The ISO convention of numbering is used (i.e., the thousands and higher multiples are separated by a space). A comma is used as the decimal point. A comparison of the American and ISO conventions is shown below:

ISO	American
0,6	0.6

1 000	1,000
1 323 462,9	1,323,462.9

Lists sequenced by letters (e.g., a-red, b-blue, c-green) show no ordering relationship between the listed items. Numbered lists (e.g., 1-red, 2-blue, 3-green) show an ordering relationship between the listed items.

If a conflict arises between text, tables, or figures, the order of precedence to resolve the conflicts is text; then tables; and finally figures. Exceptions to this convention are indicated in the appropriate subclauses. Not all tables or figures are fully described in the text. Tables show data format and values. Notes do not constitute any requirements for implementors.

4 General

4.1 Structure and concepts

Fibre Channel (FC) is logically a point-to-point serial data channel. The architecture has been designed so that it may be implemented with high performance hardware that requires little real-time software management. The Fibre Channel Physical layer (FC-2 layer) described by FC-FS performs those functions required to transfer data from one N_Port or NL_Port to another. In this standard, N_Ports and NL_Ports capable of supporting Fibre Channel protocol transactions are collectively referred to as FCP_Ports. The FC-2 layer may be treated as a very powerful delivery service with information grouping and several defined classes of service.

A switching fabric allows communication among more than two FCP_Ports.

Fibre Channel Arbitrated Loop (FC-AL) is an alternative multiple port topology that allows communication between two ports on the loop or between a port on the loop and a port on a switching fabric attached to the loop.

An FC-4 mapping layer uses the services provided by FC-FS to execute the steps required to perform the functions defined by the FC-4. The protocol is described in terms of the stream of FC IUs and Exchanges generated by a pair of FCP_Ports that support the FC-4.

The detailed implementation that supports that stream is not defined, although Originator and Responder FCP_Ports are assumed to have a common service interface, for use by all FC-4s, that is similar in characteristics to the service interface defined in FC-FS. The requirements for the service interface for SCSI are contained in SAM-2.

Four kinds of functional management are defined by this standard.

Device Management

Task Management

Process Login/Logout Management

Link Management

The Fibre Channel protocol device and task management protocols define the mapping of the SCSI functions defined in SAM-2 to the Fibre Channel interface defined by FC-FS. Link control is performed by standard FC-FS protocols. The task management functions defined by SAM-2 are mapped as described in 4.7 of this standard. The I/O Operation defined by SAM-2 is mapped into a Fibre Channel Exchange. A Fibre Channel Exchange

carrying information for a SCSI I/O Operation is an FCP Exchange. The request and response primitives of an I/O Operation are mapped into Information Units (IUs) as shown in table 1.

Table 1 - SCSI and Fibre Channel protocol functions

SCSI function	FCP equivalent
I/O operation	Exchange
Protocol Service Request and Response	Sequence
Send SCSI Command Request	Unsolicited command IU (FCP_CMND)
Data delivery request	Data descriptor IU (FCP_XFER_RDY)
Data delivery action	Solicited data IU (FCP_DATA)
Send Command Complete Response	Command status IU (FCP_ RSP)
REQ/ACK for Command Complete	Confirmation IU (FCP_CONF)

The number of Exchanges that may simultaneously be open between an initiator FCP_Port and a target FCP_Port is defined by the FC-FS implementation. The architectural limit for this value is 65 535. The maximum number of active Sequences that may simultaneously be open between an initiator FCP_Port and a target FCP_Port is restricted by the allowable range of values of the Sequence ID to 256, as defined in FC-FS. To allow task management Exchanges to be originated, a certain number of extra Exchange IDs and at least one extra Sequence_ID should always be available.

4.2 Device management

An application client begins an FCP I/O operation when it invokes an Execute Command remote procedure call. (See SAM-2). The Execute Command call conveys a single request or a list of linked requests from the application client to the FCP service delivery subsystem. Each request contains all the information necessary for the execution of one SCSI command, including the local storage address and characteristics of data to be transferred by the command. The Fibre Channel protocol then performs the following actions using FC-FS services to perform the SCSI command. The execution of the individual steps of the protocol is consistent with the SCSI architectural model as defined by SAM-2.

The FCP_Port that is the initiator for the command starts an Exchange by sending an unsolicited command IU containing the FCP_CMND IU payload, including some command controls, addressing information, and the SCSI command descriptor block (CDB). The FCP_CMND IU payload is the Send SCSI Command protocol service request (see SAM-2) and starts the FCP I/O operation. The Exchange that is started is identified by its fully qualified exchange identifier (FQXID) during the remainder of the FCP I/O operation and is used only for the IUs associated with that FCP I/O operation. See 5.1.

When the device server for the command has completed the interpretation of the command, has determined that write data transfer is required, and is prepared to request the data delivery service, it sends a data descriptor IU containing the FCP_XFER_RDY IU payload to the initiator to indicate which portion of the data is to be transferred. The FCP_Port that is the initiator then transmits a solicited data IU to the target containing the FCP_DATA IU payload requested by the FCP_XFER_RDY IU. The FCP_XFER_RDY IU and FCP_DATA IU payloads constitute the Receive Data-Out protocol service request and Data-Out Received service confirmation described in SAM-2. Data delivery requests containing FCP_XFER_RDY IU and returning FCP_DATA IU payloads continue until the data transfer requested by the SCSI command is complete. One FCP_DATA IU shall

follow each FCP_XFER_RDY IU. If the system has mechanisms outside the scope of this standard for controlling the data transfer length, the transmission of the initial FCP_XFER_RDY IU may be disabled. See 10.2.10.

When the device server for the command has completed the interpretation of the command and has determined that read data transfer is required, the FCP_Port that is the target transmits a solicited data IU to the initiator containing the FCP_DATA IU payload. The FCP_DATA IU constitutes the Send Data-In protocol service request described in SAM-2. Data deliveries containing FCP_DATA IU payloads continue until all data described by the SCSI command is transferred.

After all the data has been transferred, the device server transmits the Send Command Complete protocol service response (described in SAM-2) by requesting the transmission of an IU containing the FCP_RSP IU payload. That payload contains the SCSI status and, if the SCSI status is CHECK CONDITION, the autosense data describing the condition. The FCP_RSP IU indicates completion of the SCSI command. If no command linking, error recovery, or confirmed completion is requested, the FCP_RSP IU is the final sequence of the Exchange. The device server determines whether additional linked commands are to be performed in the FCP I/O Operation. If this is the last or only command executed in the FCP I/O operation, the FCP I/O operation and the Exchange are terminated. If an FCP protocol error occurred during execution of the command, the FCP_RSP IU payload carries the FCP Response information instead of the SCSI status and autosense data.

When the command is completed, returned information is used to prepare and return the Command Complete Received protocol service confirmation to the application client that requested the operation. The returned status indicates whether or not the command was successful. The successful completion of the command indicates that the SCSI device performed the requested operations with the transferred data and that the information was successfully transferred to or from the initiator. Status other than successful completion indicates that either SCSI sense data or warnings about unexpected FCP behaviors are being provided. In this case, the sense data or warning is interpreted to determine whether the desired operation was successfully completed. The device server can optionally request a protocol service indication that confirms delivery of the FCP_RSP IU payload, as described in 4.4.

If the command is linked to another command, the FCP_RSP IU payload shall contain the proper status (i.e., INTERMEDIATE or INTERMEDIATE CONDITION MET) indicating that another command shall be executed. The target shall present the FCP_RSP using the IU that allows command linking, I5 (see 5.4). The initiator shall continue the same Exchange with an FCP_CMND IU, beginning the next SCSI command. All SCSI commands linked in the FCP I/O operation except the last are executed in the manner described above. SAM-2 defines the cases that interrupt and terminate a series of linked commands. In those cases, the FCP_RSP IU of the last command in the set of linked commands shall be transmitted using the IU that does not allow command linking, I4.

The number of FCP I/O operations that may be active at one time depends on the queuing capabilities of the FCP device. If command queueing resources are unavailable in the target when a command is received, the target returns TASK SET FULL status or BUSY status in the FCP_RSP IU as specified by SAM-2.

The Fibre Channel protocol takes full advantage of the multiplexing and shared bandwidth capabilities provided by various Fibre Channel classes of service. The protocol is designed to operate with any class of service and to provide options for reliable error detection and error recovery independent of the class of service.

SCSI allows the initiator function in any FCP_Port and the target function in any FCP_Port. For FCP I/O operations between a host and a peripheral subsystem, the host typically takes on the initiator role and the peripheral subsystem typically takes on the target role. For host to host communications, either one of the communicating pair may take on the initiator role. For device to device communications, typically used to

implement extended copy and other third-party operations, the initiator role is adopted by the managing FCP device.

The Fibre Channel protocol implements Asynchronous Event Reporting (see SAM-2) using the Asynchronous Event Notification (AEN) model in SPC-2. The AEN model reports asynchronous events by requiring that the peripheral FCP device take on the initiator role to deliver the asynchronous event sense data to the host, which acts as a target using the processor device model for the duration of the AEN reporting process.

4.3 Precise delivery of SCSI commands

In applications were SCSI communications between an application client and a device server are stateless, verification of the delivery and execution of SCSI commands is often not critical. Any changes in execution sequence caused by link failures or switch latencies are not important and the recovery and retry mechanisms may be executed while other activities are continued by the application client and the device server.

SAM-2 defines a mechanism to assure ordering of commands. If the initiator transmits a single command and waits for GOOD status before transmitting the next command, the commands are guaranteed to be executed in order.

This standard defines a second optional mechanism called precise delivery to assure ordering of commands. This may be used by any FCP device, but may be useful for devices performing ordered command queuing where device state is preserved from one command to the next. An application client may determine if a device server supports the precise delivery function by using the MODE SENSE and MODE SELECT commands to examine and set the enable precise delivery checking (EPDC) bit in the Fibre Channel Logical Unit Control page. See 10.3.

Precise delivery of SCSI commands uses the COMMAND REFERENCE NUMBER field in the FCP_CMND IU.

For each device server having the EPDC bit set to one, the application client places a one byte unsigned integer in the COMMAND REFERENCE NUMBER field of each command requiring precise delivery. The integer shall start at the reset value of one and shall be incremented by one for each command requiring precise delivery for that device server. Separate increment counters are maintained for each I_T_L nexus (i.e., each initiator maintains a separate counter for each device server using precise delivery). After the number of precisely delivered commands causes the integer to increment to 255, the integer wraps back to a value of one. The value of zero is reserved and shall be used for those commands that do not require precise delivery and for task management functions.

The following rules specify how the application client and device server use the CRN to determine that each command requiring precise delivery has been properly received and executed.

a) See tables 4 and 5 for the actions that cause the CRN to be transmitted by the initiator to be set to one and the CRN expected by the device server to be set to one.

b) The CRN shall be equal to one for the first FCP_CMND IU requiring precise delivery between the application client and device server and shall be incremented by one for each subsequent command requiring precise delivery.

c) The CRN shall wrap from 255 to one (i.e. a value of zero in the CRN field is not valid for an Exchange using precise delivery).

d) The initiator shall not transmit the same CRN again until delivery of the first FCP_CMND IU transmitted with that CRN has been confirmed by receipt of an FCP_XFER_RDY IU, an FCP_DATA IU, an FCP_RSP IU, an ACK, or a response to an REC.

e) The device server shall not accept a command with a nonzero CRN into the dormant or enabled state until after all commands with a previous CRN have been received by the device server. The commands shall be assumed to be received in the order of increasing CRN, highest CRN last. The order of execution of the commands shall be managed by the normal task set management algorithms.

f) The device server shall accept any valid command with a CRN of zero into the dormant or enabled state regardless of whether or not all commands with a nonzero CRN have been received. The order of execution of the commands shall be managed by the normal task set management algorithms. See SAM-2.

g) Task management functions shall have the CRN set to zero and shall not be tested for precise delivery by the device server.

Any command may use a CRN of zero if precise delivery is not required for that command. For example, commands such as INQUIRY, TEST UNIT READY, REPORT LUNS and MODE SENSE/SELECT used for booting and initialization may use a CRN of zero.

4.4 Confirmed completion of FCP I/O Operations

Some FCP devices require an acknowledgment of successful delivery of FCP_RSP information. Such an acknowledgment is provided by the confirmed completion function, optionally implemented by FCP-2 devices. PRLI parameters are used to determine that confirmed completion is accepted by an initiator and may be requested by a target communicating with that initiator.

A target may invoke the confirmed completion function by setting the FCP_CONF_REQ bit to one in the FCP_RSP IU. Upon receiving the request in the FCP_RSP IU, the initiator shall transmit an FCP_CONF IU to the target, indicating to the target that the FCP_RSP IU has been received by the initiator.

The confirmed completion function allows the retry of unsuccessful notifications of errors and confirms that the initiator and the target both agree upon the state of a state dependent device. Retry mechanisms for unsuccessful transmission of FCP_RSP IUs and FCP_CONF IUs are defined in this standard.

Confirmed completion shall not be requested for FCP_RSP IUs responding to task management requests.

If confirmed completion is not enabled, the FCP_CONF IU shall not be requested by the FCP_RSP IU.

If command linking is being performed, the target shall not request confirmed completion for an FCP_RSP IU containing INTERMEDIATE status. The target may request confirmed completion

a) when providing the FCP_RSP IU for the last command of the set of linked commands, or

b) when providing the FCP_RSP IU for a command that terminates linking because of an error or CHECK CONDITION status.

Confirmed completion may assist initiators and targets in many environments. Particular examples include:

a) The confirmed completion function may be used to confirm that an initiator has received an FCP_RSP IU reporting a SCSI CHECK CONDITION status, together with accompanying autosense data. Upon receiving the FCP_CONF IU, the target may discard its copy of the autosense data.

b) The confirmed completion function may be used to confirm that a queued SCSI command has been completed and that the completion information has been successfully transferred to the initiator. That allows subsequent queued state dependent operations to be performed, since the FCP_CONF IU confirms that the FCP_RSP IU has been received by the initiator.

c) The confirmed completion function may be used to confirm that an initiator has received the FCP_RSP IU for targets that require state dependent synchronization with initiators.

4.5 Retransmission of unsuccessfully transmitted data

Error detection and data retransmission algorithms are defined by clause 12.

The Read Exchange Concise (REC) Extended Link Service may be used by any initiator to determine the state of an ongoing Exchange. See 6.5. Those targets that do not support REC indicate that by performing a Link Service Reject (LS_RJT). See 8.3.

If an error is identified by any of the mechanisms defined by clause 12 and if the data retransmission capability is supported by both the initiator and target as indicated by the PRLI bits, the initiator may request retransmission using the Sequence Retransmission Request (SRR) FCP FC-4 Link Service Request. See 8.2. If the data retransmission capability is supported by both the initiator and the target, as indicated by the PRLI bits, both devices shall support REC.

4.6 Task retry identification

Task retry identification optionally provides an additional optional mechanism for relating commands that are being retried to the requests that are sensing the requirement for recovery (REC) and performing the recovery (SRR). The particular case that has been identified as a problem is related to the recovery procedure diagrammed in figure C.7. It is possible that initiators may re-use OX_ID values rapidly enough to create an ambiguous situation where the status being preserved in the target for possible retransmission and the new command being presented to the target may have the same OX_ID values. When recovery of a transmission failure for the new command is attempted, the target instead indicates that the recovery is related to the previous command's status and the initiator is provided status for the completed command. That information is mistakenly interpreted as status for the failed command. Many small variations on this scenario may exist.

Devices that agree to perform recovery and that may create such ambiguous cases should support task retry identification. If both devices agree to support task retry identification, a task retry identifier is provided in the PARAMETER field of each FCP_CMND IU frame. The Link Services associated with retransmission of IUs (REC and SRR) each contain the same task retry identifier, unambiguously relating them to the particular command. If the devices do not agree to support task retry identification, the PARAMETER field is zero for both the FCP_CMND IU frame and the REC and SRR frames.

4.7 Discovery of FCP capabilities

A number of Fibre Channel protocol capabilities require the knowledge and agreement of both the target and the initiator that such capabilities may or shall be used. Table 2 provides references to the discovery process for each of the Fibre Channel protocol capabilities.

Table 2 - Discovery of FCP-2 capabilities

Capability	Discovery mechanism	Reference
Initiator	Process Login	6.3.4
Target	Process Login	6.3.4
Initiator accepts data overlay	Process Login	6.3.4
Target performs data overlay	MODE SENSE command	10.2.8
Initiator generates FCP_CONF IU	Process Login	6.3.4
Target requests FCP_CONF IU	Process Login	6.3.4
Initiator performs REC	None required, Process Login allowed	4.5
Target accepts REC	LS_RJT if REC not accepted	4.5
Initiator performs SRR	Process Login	6.3.4
Target accepts SRR	Process Login	6.3.4
Initiator provides CRN	MODE SENSE command	4.3
Target accepts CRN	MODE SENSE command	10.3
Task Retry Identification	Process Login	6.3.4

4.8 Task management

An application client requests a task management function to control explicitly the execution of one or more FCP I/O Operations. See 9.1.2.4.

Some SCSI task management functions are mapped into FC-FS link services and others are mapped into control bits in the FCP_CMND IU. Task management functions that use the FCP_CMND IU are transmitted as the first IU in a new Exchange. A task management function that uses the FCP_CMND IU ends with an

FCP_RSP IU that indicates the completion status of the function. The FCP_CDB field in FCP_CMND IUs that perform task management functions is ignored. The mappings are explained in table 3.

Table 3 - Task management functions, SAM-2 to FCP

SAM-2 function	FCP equivalent	Optional/ Mandatory
TARGET RESET	FCP_CMND TARGET RESET	Mandatory
ABORT TASK	FCP recovery abort[1]	Mandatory
ABORT TASK SET	FCP_CMND ABORT TASK SET	Mandatory
CLEAR TASK SET	FCP_CMND CLEAR TASK SET	Mandatory
CLEAR ACA	FCP_CMND CLEAR ACA	Optional[2]
LOGICAL UNIT RESET	FCP_CMND LOGICAL UNIT RESET	Mandatory
Notes: 1) FC-FS basic link services are used to perform the ABORT TASK function. 2) CLEAR ACA is mandatory if Automatic Contingent Allegiance is used.		

FC-FS basic link services and extended link services are used to perform the ABORT TASK function, to recover Exchange resources, and to re-establish other initial conditions. FC-FS basic link services and extended link services may be used for additional control and management activities not listed in table 2. See 9.1.3.

4.9 Clearing effects of task management, FCP, FC-FS, and FC-AL-2 actions

Tables 4 and 5 summarize the FCP target objects that are cleared as a result of Fibre Channel Link actions and SCSI operations, respectively. A ‘Y’ in the corresponding column of either table indicates the object is cleared to its default, saved, or initial value within the device upon successful completion of the specified action. The clearing actions are applicable only to Sequences and Exchanges associated with Fibre Channel protocol actions. Sequences and Exchanges associated with other actions follow rules specified in FC-FS or other relevant protocol standards. 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. Rows indicating an effect for all initiator ports have the specified effect on all ports, regardless of the link that attaches the initiator port to the target.

Table 4 - Clearing effects of link related functions

Target object	Target Power Cycle	Reset LIP(y,x)[2]	LOGO[5], PLOGI	PRLI[4] PRLO[5],	TPRLO[3]	ABTS (Exchange)
	FC link action affecting target object					
PLOGI parameters						
For all logged-in initiator ports	Y	Y	N	N	N	N
Only for initiator port associated with the action	-	-	Y	N	N	N
Open FCP Sequences Terminated						
For all initiator ports with open FCP Sequences	Y	Y	N	N	Y	N
Only for initiator port associated with the action	-	-	Y	Y	-	N
Only for FCP Sequences associated with Aborted FCP Exchanges	-	-	-	-	-	Y
Login BB_Credit_CNT						
For all Logged-In L_Ports	Y	Y	-	N	N	N
For transmitting L_Port only	-	-	Y			
Hard Address Acquisition Attempted	Y[1]	Y[1]	N	N	N	N
PRLI parameters cleared						
For all logged-in initiator ports	Y	Y	N	N	N	N
Only for N_Port or L_Port associated with the action	-	-	Y	Y	N	N
CRN (Command Reference Number) (set to one)						
For all initiator ports	Y	Y	N	N	Y	N
Only for initiator port associated with the action	-	-	Y	Y	-	N

NOTES:

1 If the NL_Port has an AL_PA different than 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 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.

3 For TPRLO (Third Party Process Logout), actions listed shall be performed when the GLOBAL bit is set to one. If the GLOBAL bit is set to zero, then the actions listed under PRLI/PRLO shall be performed for the designated initiator. See FC-FS.

4 The Target shall clear the object only if ESTABLISH IMAGE PAIR is set to one and if the referenced image pair is FCP type. See 6.2.

5 LOGO and PRLO may be either implicit or explicit. Implicit LOGO and PRLO are specified in FC-FS and FC-FLA.

Table 5 - Clearing effects of initiator actions

Target object	Initiator action affecting target object			
	TARGET RESET	LOGICAL UNIT RESET[2, 3]	CLEAR TASK SET[2]	ABORT TASK SET[2]
PLOGI parameters				
For all logged-in initiator ports	N	N	N	N
Only for initiator port associated with the action	N	N	N	N
Open FCP Sequences Terminated				
For all initiator ports with open FCP Sequences	Y[1]	Y[1]	Y[1]	N
Only for initiator port associated with the action	-	-	-	Y[1]
Only for FCP Sequences associated with Aborted FCP Exchanges	-	-	-	-
Login BB_Credit_CNT				
For all Logged-In L_Ports	N	N	N	N
For transmitting L_Port only				
Hard Address Acquisition Attempted	N	N	N	N
PRLI parameters cleared				
For all logged-in initiator ports	N	N	N	N
Only for N_Port or L_Port associated with the action	N	N	N	N
CRN (Command Reference Number) (set to one)				
For all initiator ports	Y	Y	Y	N
Only for initiator port associated with the action	-	-	-	Y

NOTES:

1 Exchanges are cleared internally within the target, but open FCP Sequences shall be individually aborted by the initiator via the recovery abort protocol that also has the effect of aborting the associated FCP Exchange. See 12.3.

2 For multiple-LUN targets, CLEAR TASK SET, ABORT TASK SET, and LOGICAL UNIT RESET affect only the addressed LUN.

3 A definition of the LOGICAL UNIT RESET Task Management function is contained in the SAM-2 standard.

4.10 I_T nexus loss notification events

An FCP_Port shall deliver an I_T nexus loss notification (see SAM-3) for the following:

a) Sending or receiving LOGO (explicit or implicit);

b) Sending or receiving PRLO (explicit or implicit);

c) Receiving TPRLO;

d) Sending TPRLO with a Third Party Originator N_Port_ID (see FC-FS) that matches the N_Port_ID of the sending FCP_Port; or

e) Sending TPRLO with the GLOBAL bit set to one to a target port that has an I_T nexus with the sending initiator port.

4.11 Transport Reset notification events

An NL_Port shall deliver a Transport Reset notification (see SAM-3) for a Reset LIP(y,x) FC link event if the AL_PD matches the AL_PA of the receiving NL_Port.

4.12 Port login/logout

The N_Port Login (PLOGI) extended link service is optionally used to establish the Fibre Channel operating parameters between any two Fibre Channel ports, including FCP_Ports.

If a target receives a PLOGI request and it finds there are not enough login resources to complete the login, the target responds to the PLOGI with LS_RJT and reason code “unable to perform command request” and reason explanation “insufficient resources to support Login” as defined by FC-FS. By means outside the scope of this standard, the target may select another initiator and release some login resources by performing an explicit logout of the other initiator, freeing resources for a future PLOGI.

4.13 Process login/logout

The Process Login (PRLI) extended link service is optionally used to establish the FCP operating relationships between two FCP_Ports. Implicit PRLI/PRLO parameters may be defined for FCP_Ports. Such definitions are outside the scope of this standard.

4.14 Link management

FC-FS allows management protocols above the FC-FS interface to perform link data functions. The standard FC-FS primitive sequences, link management protocols, and basic and extended link services are used as required by FCP devices. Implicit login functions are allowed.

5 Fibre Channel protocol overview

5.1 FCP addressing and Exchange identification

The address of each FCP_Port is defined by its address identifier as described in FC-FS. Each FCP I/O operation is identified by the FCP I/O operation's fully qualified exchange identifier (FQXID). The FQXID is composed of the initiator address identifier, the target address identifier, the OX_ID and the RX_ID. Other definitions of FQXID are outside the scope of this standard. The method used to identify FCP I/O operations internal to the application client and the device server is not defined by this standard.

Addressability of logical units uses the logical unit number provided in the FCP_CMND IU. Subsequent identification of the FCP I/O operation and the Exchange which carries the protocol interactions for the FCP I/O operation uses the FQXID. FCP devices do not use the Process_Associator.

The target uses the OX_ID, and, if it has been assigned, the RX_ID to perform error recovery and task management functions. The task retry identifier is used as a supplemental task identifier if task retry identification is supported and enabled.

5.2 SCSI third-party device identifier for the Fibre Channel protocol

The SCSI RESERVE commands that use the 64-bit THIRD-PARTY DEVICE ID defined by SPC-2 shall use the FCP third-party device id format defined in table 6 to identify the specified third-party target.

Table 6 - FCP third-party device id format

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1–3	(MSB)		FCP_PORT IDENTIFIER					(LSB)
4–7	RESERVED							

FCP_PORT IDENTIFIER:

The FCP_PORT IDENTIFIER field defines the address identifier of the target that shall be used by the target for third-party addressing.

5.3 Use of World Wide Names

As specified in FC-FS, each Fibre Channel node and each Fibre Channel port shall have a Worldwide_Name. The Worldwide_Name shall be a unique name using one of the formats defined by FC-FS and its extensions. Each target and its associated logical units has knowledge of the Port_Name of each initiator through the Fibre Channel login process. As a result, the relationship between address identifier of the initiator and a persistent reservation for a logical unit may be adjusted as defined in SPC-2 during those reconfiguration events that may change the address identifier of the initiator. If a target receives a PRLI or a PLOGI from an initiator FCP_Port with a previously known Worldwide_Name but with a changed initiator identifier, the device server shall assign the new initiator identifier to the existing registration and reservation to the initiator port having the same Worldwide_Name.

Each logical unit shall be able to present a Worldwide_Name through the INQUIRY command vital product data device identification page as defined by SPC-2. For devices compliant with this standard and having a LUN 0,

the Worldwide_Name of the logical unit having a LUN of 0 may be the same as the Node_Name of the target. The Worldwide_Name for the port shall be different from the Worldwide_Name for the node.

5.4 FCP Information Units (IUs)

The IUs used by the Fibre Channel protocol and their characteristics are shown in table 7 for IUs sent to targets, and in table 8 for IUs sent to initiators. Each IU shall be contained in a single Sequence. Each Sequence carrying an FCP IU shall contain only one IU. Examples of typical Fibre Channel protocol operations using these IUs are included in Annex B.

Table 7 - FCP Information Units (IUs) sent to targets

IU	SCSI-3 primitive	Data block		F/M/L	SI	M/O
		CAT	Content			
T1	Command / Task Mgmt Rqst	6	FCP_CMND	F	T	M
T2	Command request	6	FCP_CMND	F	H	O
T3	Command request (Linked)	6	FCP_CMND	M	T	O
T4	Command request (Linked)	6	FCP_CMND	M	H	O
T6	Data Out action	1	FCP_DATA	M	T	M
T12	Confirm	3	none	L	T	O

Notes:
- T5, T7, T8, T9, T10, and T11 are obsolete.
- T2 and T4 are only permitted when transfer ready IUs are disabled. See
- T3 and T4 are only permitted for linked SCSI commands.
- T2 and T4 allow optional sequence streaming during Write operations.
- T12 is only permitted in response to an I5 frame requesting the confirmed completion protocol. See table 8

Key:

- IU — Information Unit identifier
- CAT — Information category of Device_Data frames carrying the data block (see FC-FS)
- CONTENT — Contents (payload) of data block
- F/M/L — First/Middle/Last Sequence of Exchange (FC-FS)
  - F First
  - M Middle
  - L Last
- SI — Sequence Initiative: Held or Transferred (FC-FS)
  - H Held
  - T Transferred
- M/O — Mandatory/Optional Sequence
  - M Mandatory
  - O Optional

Table 8 - FCP Information Units (IUs) sent to initiators

IU	SCSI-3 primitive	Data block		F/M/L	SI	M/O
		CAT	Content			
I1	Data delivery request	5	FCP_XFER_RDY (Write)	M	T	M
I3	Data In action	1	FCP_DATA	M	H	M
I4	Command/Task Mgmt Response	7	FCP_RSP	L	T	M
I5	Response (Linked or confirm request)	7	FCP_RSP	M	T	O

Notes:
I2, I6, and I7 are obsolete.
I5 is permitted for linked SCSI commands or to request the confirm completion protocol.
I3 allows optional sequence streaming to I3, I4, or I5.

Key:

IU	Information Unit identifier
CAT	Information category of Device_Data frames carrying the data block (FC-FS)
CONTENT	Contents (payload) of data block
F/M/L	First/Middle/Last Sequence of Exchange (FC-FS) F First M Middle L Last
SI	Sequence Initiative: Held or Transferred (FC-FS) H Held T Transferred
M/O	Mandatory/Optional Sequence M Mandatory O Optional

5.5 Fibre Channel protocol standard formats

The Fibre Channel protocol requires only the standard FC-2 services as described in FC-FS. FC-FS provides an informative outline of the FC_PH_SEQUENCE.request used by the Fibre Channel protocol to provide all the information to manage the protocol. No additional capabilities are required to transmit the required FCP IUs. Corresponding FC_PH_SEQUENCE.indication information is provided to the Sequence Recipient to properly type and categorize the received IUs. Classes of service that provide end-to-end acknowledgement, including Class 1 and Class 2 service, provide the FC_PH_SEQUENCE.confirmation primitive that may be used to assist in management of the Fibre Channel protocol. The use of the FC_PH_SEQUENCE_TAG.indication primitive may be required by the Sequence Initiator.

5.6 FC-FS mappings to SCSI-3 functionality

5.6.1 FC-FS frame header

The format of the standard FC-FS header as used by the Fibre Channel protocol is defined in table 9.

Table 9 - FCP frame header

Bits Word	31– 24	23–16	15–08	07–00
0	R_CTL	D_ID		
1	CS_CTL	S_ID		
2	TYPE	F_CTL		
3	SEQ_ID	DF_CTL	SEQ_CNT	
4	OX_ID		RX_ID	
5	PARAMETER			

All fields use the standard FC-FS definitions. The following explanations of some of the fields provide information about the use of those fields to implement FCP functionality.

5.6.2 Frame header fields

5.6.2.1 R_CTL

The values in the R_CTL field identify the frame as part of an FCP I/O Operation and identify the information category. All Sequences containing FCP command, data, response, and data descriptor information shall be composed of Device_Data frames.

The information category associated with each IU is defined in table 7 and table 8.

5.6.2.2 D_ID

The value in the D_ID field is the D_ID of the frame. For FCP FC-4 Device_Data frames, the D_ID transmitted by the Exchange Originator is the address identifier of the target. The D_ID transmitted by the Exchange Responder is the address identifier of the initiator.

5.6.2.3 CS_CTL

The values in the CS_CTL field are defined by FC-FS for class specific control information and do not interact with the Fibre Channel protocol.

5.6.2.4 S_ID

The value in the S_ID field is the S_ID of the frame. For FCP FC-4 Device_Data frames, the S_ID transmitted by the Exchange Originator is the address identifier of the initiator. The S_ID transmitted by the Exchange Responder is the address identifier of the target.

5.6.2.5 TYPE

The value in the TYPE field shall be 08h for all frames of SCSI FCP Sequences.

5.6.2.6 F_CTL

The bits in the F_CTL field manage the beginning and normal or abnormal termination of Sequences and Exchanges. The bits and definitions shall be as defined by FC-FS.

5.6.2.7 SEQ_ID

The value in the SEQ_ID field identifies each Sequence between a particular Exchange Originator and Exchange Responder with a unique value as defined by FC-FS.

5.6.2.8 DF_CTL

The bits in the DF_CTL field indicate any optional headers that may be present. The Fibre Channel protocol does not use any optional headers.

5.6.2.9 SEQ_CNT

The value in the SEQ_CNT field indicates the frame order within the Sequence as defined by FC-FS.

5.6.2.10 OX_ID

The value in the OX_ID field is the Originator Exchange Identifier and is one of the identifiers contained in the FQXID. The OX_ID shall be assigned and shall have a value other than FFFFh.

5.6.2.11 RX_ID

The value in the RX_ID field is the Responder Exchange Identifier and is one of the identifiers contained in the FQXID. The RX_ID shall have the unassigned value of FFFFh until the Exchange Responder assigns a different value in its response to the Exchange Originator. The Exchange Originator shall use the value assigned by the Exchange Responder for subsequent frames.

5.6.2.12 PARAMETER

The PARAMETER field has two definitions for Device_Data frames with the FCP type.

For frames of the solicited data category (FCP_DATA IUs), the PARAMETER field shall contain a relative offset. The RELATIVE OFFSET PRESENT bit of the F_CTL field shall be set to one, indicating that the PARAMETER field value is a relative offset. For the solicited data category (FCP_DATA IUs), the relative offset is the application client buffer offset as described by SAM-2. For solicited data category frames, the relative offset shall have a value that is 0 modulo 4 (i.e., each frame of each FCP_DATA IU shall begin on a word boundary).

For frames of the unsolicited control category (FCP_CMND IUs), the PARAMETER field value depends on whether task retry identification is active. If the target and initiator have agreed upon performing task retry identification, the PARAMETER field shall contain the task retry identifier. If the target and initiator have not agreed upon performing task retry identification, the PARAMETER field shall contain a value of zero. In both cases, the RELATIVE OFFSET PRESENT bit of the F_CTL field shall be set to zero.

For all other Device_Data frames with the FCP type, the RELATIVE OFFSET PRESENT bit of the F_CTL field shall be set to zero and the PARAMETER field shall contain a value of zero.

For FCP FC-4 Link Service frames, the PARAMETER field is specified in the description of the individual link services.

6 FCP basic and extended link service definitions

6.1 Overview of link service requirements

The FCP link-level protocol includes the basic link services and extended link services defined by FC-FS. The protocol also includes the Process Login and Process Logout extended link services defined by FC-FS, the Process Login FCP Service Parameter pages defined in this standard, and the Read Exchange Concise extended link service with usage as defined in this standard.

Link-level protocols are used to configure the FC environment, including the establishment of configuration information and address information. FCP devices introduced into a configuration or modifications in the addressing or routing of the configuration may require the login and discovery procedures to be executed again.

6.2 Overview of Process Login/Logout

The extended link service command of Process Login (PRLI) is defined in FC-FS. The PRLI allows for a process at one FCP_Port to be related to a corresponding process at another FCP_Port as an image pair. In addition, the PRLI allows one or more FC-4 capabilities to be reported by the initiating FCP_Port to the recipient FCP_Port. The recipient FCP_Port indicates its acceptance or rejection of the capabilities in its response to the PRLI request. The PRLI is optional, since implicit login may be established by configuration conventions outside the scope of this standard.

PRLI requests shall only be initiated by devices having the initiator capability. Devices having only target capability shall not perform a PRLI request. An initiator shall have successfully completed a PRLI with a target that establishes an image pair before any FCP IUs are exchanged. An image pair may also be established by an implicit PRLI established by methods outside the scope of this standard. An image pair is removed by a PRLO (see 6.4). If an image pair is not established by an initiator to a target, the initiator and target shall not exchange any FCP IUs. Any FCP IUs received by a target from an N_Port or NL_Port that has not established an image pair with that target shall be discarded. In addition, a target that receives an FCP_CMND IU from an N_Port or NL_Port that has not established an image pair with that target shall discard the FCP_CMND IU and respond with an explicit PRLO.

Process_Associators shall not be used in initiators and targets. If multiple images are required in an initiator, they shall be provided by transparent aliasing of the N_Port Identifier of the initiator. If multiple images are required in a target, they shall be provided by SCSI logical units.

The creation of image pairs behind an FCP_Port has no effect on the Fibre Channel protocol.

The FC-4 Service Parameter pages for the Fibre Channel protocol are defined in the following subclauses.

Execution of a PRLI or PRLO extended link service command performs the clearing actions defined in 4.9.

Process Login has two actions that can be performed, selected by the ESTABLISH IMAGE PAIR bit. (See 6.3.4.)

a) Informative action
Service parameter information is exchanged during the Process Login enabling subsequent negotiation for image pair establishment.

b) Binding action
Service parameter information is exchanged that establishes an image pair relationship between processes in the communicating N_Port or NL_Ports. The relationship does not allow any communication types or paths other than those established by the PRLI.

6.3 Process Login (PRLI)

6.3.1 Use of Process Login by the Fibre Channel protocol

The PRLI request is transmitted from an Originator FCP_Port to a Responder FCP_Port to identify to the destination the capabilities that the Originator FCP_Port expects to use with the Responder FCP_Port and to determine the capabilities of the Responder. (See FC-FS.)

If the PRLI is requesting an informative action by setting the ESTABLISH IMAGE PAIR bit to zero, the PRLI accept reports the capabilities of the Responder to the Originator.

If the PRLI is requesting a binding action by setting the ESTABLISH IMAGE PAIR bit to one, the PRLI accept reports the capabilities of the responder to the Originator and establishes an image pair. An image pair shall be established only if the FCP devices have complementary initiator and target capabilities. If both FCP devices have both initiator and target capabilities, a single image pair allows both initiators to access the complementary target capabilities of the other device in the pair. Some capabilities require support by both the Originator and Responder before they can be used (see 6.3.4). The IMAGE PAIR ESTABLISHED bit in the PRLI accept indicates that an image pair was successfully established.

A reason code shall be provided if the PRLI request is incorrect or if a requested image pair cannot be established.

A link service reject (LS_RJT) indicates that the PRLI request is not supported or is incorrectly formatted.

The PRLI common service parameters and accept response codes are defined in FC-FS. FC-4 service parameters for mappings other than the Fibre Channel protocol are defined in other FC-4 standards.

6.3.2 Process_Associator requirements

Operation of the PRLI depends on the Originator's and Responder's requirements for Process_Associators as specified in FC-FS. Process_Associators are not used in the Fibre Channel protocol and shall not be used by FCP devices.

The Fibre Channel protocol assumes that the Originator has knowledge of the capabilities of the Responder. That information may be obtained by performing a PRLI requesting an informative action or by other mechanisms outside the scope of the this standard.

6.3.3 New or repeated PRLI

After the completion of any new or repeated binding PRLI, all clearing actions specified in 4.9 shall be performed.

After the completion of any new or repeated informative PRLI, the state of the Originator and Responder remains unchanged.

If the change in parameters affects any outstanding FCP Exchanges, those Exchanges shall be terminated by the initiator using a recovery abort operation. A recovery qualifier may be established after the recovery abort, temporarily restricting the choice of OX_ID values by the initiator and RX_ID values by the target. Only actions for image pairs that are being referenced by the PRLI are affected.

FCP devices may have default PRLI information provided in a manner outside the scope of this standard. Such devices do not require the execution of a PRLI to perform normal FCP I/O operations. If default PRLI information is complete enough so that login (PLOGI) is sufficient to perform an implicit PRLI, then PLOGI shall perform the same clearing actions and establish the same Unit Attention condition that would normally be performed and established by PRLI.

6.3.4 Process Login request FCP Service Parameter page format

The FCP Service Parameter page for the Process Login request is shown in table 10.

Table 10 - FCP Service Parameter page, PRLI request

FCP service parameter	Word	Bit
SCSI FCP (08h)	0	31–24
reserved for TYPE code extension	0	23–16
ORIGINATOR PROCESS_ASSOCIATOR VALID	0	15
RESPONDER PROCESS_ASSOCIATOR VALID	0	14
ESTABLISH IMAGE PAIR	0	13
RESERVED	0	12–0
ORIGINATOR PROCESS_ASSOCIATOR	1	31–0
RESPONDER PROCESS_ASSOCIATOR	2	31–0
RESERVED	3	31–10
TASK RETRY IDENTIFICATION REQUESTED	3	9
RETRY	3	8
CONFIRMED COMPLETION ALLOWED	3	7
DATA OVERLAY ALLOWED	3	6
INITIATOR FUNCTION	3	5
TARGET FUNCTION	3	4
OBSOLETE	3	3
OBSOLETE	3	2
READ FCP_XFER_RDY DISABLED (shall be one)	3	1
WRITE FCP_XFER_RDY DISABLED	3	0

Word 0, Bits 31–24: FCP specific code: The value of 08h in this byte indicates that this Service Parameter page is defined for the Fibre Channel protocol. (See FC-FS.)

Word 0, Bit 15: ORIGINATOR PROCESS_ASSOCIATOR VALID: The ORIGINATOR PROCESS_ASSOCIATOR VALID bit is defined in FC-FS. For the Fibre Channel protocol, the ORIGINATOR PROCESS_ASSOCIATOR VALID bit shall be zero, indicating that the ORIGINATOR PROCESS_ASSOCIATOR is not valid.

Word 0, Bit 14: RESPONDER PROCESS_ASSOCIATOR VALID: The RESPONDER PROCESS_ASSOCIATOR VALID bit is defined in FC-FS. For the Fibre Channel protocol, the RESPONDER PROCESS_ASSOCIATOR VALID bit shall be zero, indicating that the RESPONDER PROCESS_ASSOCIATOR is not valid.

Word 0, Bit 13: ESTABLISH IMAGE PAIR**:** If the ESTABLISH IMAGE PAIR bit is set to zero, the PRLI only exchanges service parameters as defined in FC-FS.

If the ESTABLISH IMAGE PAIR bit is set to one, the PRLI exchanges service parameters and attempts to establish an image pair as defined in FC-FS.

Word 1: ORIGINATOR PROCESS_ASSOCIATOR: The ORIGINATOR PROCESS_ASSOCIATOR field is the Originator Process_Associator as defined by FC-FS.

Word 2: RESPONDER PROCESS_ASSOCIATOR: The RESPONDER PROCESS_ASSOCIATOR field is the Responder Process_Associator as defined by FC-FS.

Word 3, Bit 9: TASK RETRY IDENTIFICATION REQUESTED: When the TASK RETRY IDENTIFICATION REQUESTED bit is set to one, the Originator of the PRLI requests that the task retry identification function be used. If both the Originator of the PRLI and the Responder to the PRLI request that the task retry identification function be used, then it shall be used between the initiator and all logical units for that port. The PARAMETER field for each FCP_CMND IU shall be set to a unique non-zero value. The PARAMETER field for any REC Extended Link Service request or SRR FCP FC-4 Link Service request for that command shall be set to the same value.

When the TASK RETRY IDENTIFICATION REQUESTED bit is set to zero by either the Originator of or the Responder to the PRLI, the task retry identification function shall not be used. The PARAMETER fields for FCP_CMND IUs, for REC Extended Link Service requests, and for SRR FCP FC-4 Link Service requests shall be zero.

Word 3, Bit 8: RETRY: When the RETRY bit is set to one, the Originator or Responder is indicating that it supports as an initiator the capability of requesting a retransmission of unsuccessfully transmitted data or as a target the capability of performing a requested retransmission. When the RETRY bit is set to zero, the Originator or Responder is indicating that it does not support the capability of requesting or performing retransmissions of unsuccessfully transmitted data.

If the process has both initiator and target capabilities, the RETRY bit shall apply to both. SRR may be both transmitted by and accepted by the process.

An initiator and target shall use the retransmission capability only if the RETRY bit is set in both the request payload and in the accept payload. If the RETRY bit is set to zero in either the request payload or the accept payload, the SRR shall not be performed by the initiator. If an SRR FCP FC-4 Link Service is received by a target that has set the RETRY bit to zero, the SRR shall be rejected with FCP_RJT.

If the image pair is allowed to use the retransmission capability, overlay of data as defined for SRR shall be allowed regardless of the state of the DATA OVERLAY ALLOWED bit.

Word 3, Bit 7: CONFIRMED COMPLETION ALLOWED: When the CONFIRMED COMPLETION ALLOWED bit is set to one, the Originator's or Responder's initiator function has the capability of supporting confirmed completion. When the CONFIRMED COMPLETION ALLOWED bit is set to zero, the initiator function does not have the capability of supporting confirmed completion. The CONFIRMED COMPLETION ALLOWED bit shall be zero for FCP devices having only target function. If the initiator function supports confirmed completion, then a target may optionally request an FCP_CONF IU by setting the FCP_CONF_REQ bit to one as specified by 4.4. If the initiator function does not have the capability of supporting confirmed completion, the target shall not set the FCP_CONF_REQ bit to one.

Word 3, Bit 6: DATA OVERLAY ALLOWED: When the DATA OVERLAY ALLOWED bit is set to one, the Originator or Responder is indicating that its initiator function has the capability of supporting data overlay. When the data overlay allowed bit is set to zero, the initiator function does not have the capability of performing data overlay. The data overlay allowed bit shall be zero for FCP devices having only target function. If the initiator function supports data overlay, then a target may optionally perform random buffer access that performs a transfer to or from the same offset in the application client buffer more than once during execution of a command.

Data transmission requested by the initiator during the optional retry procedures defined by this standard is managed by the initiator. Such data retransmissions are not considered data overlays, even if retransmission occurs to the same offset in the application client buffer.

Word 3, Bit 5: INITIATOR FUNCTION: When the INITIATOR FUNCTION bit is set to one, the Originator or Responder is indicating it has the capability of operating as an initiator. When the INITIATOR FUNCTION bit is set to zero, the process does not have the capability of operating as an initiator.

Word 3, Bit 4: TARGET FUNCTION: When the TARGET FUNCTION bit is set to one, the Originator or Responder is indicating that it has the capability of operating as a target. When the TARGET FUNCTION bit is set to zero, the process does not have the capability of operating as a target. Both the INITIATOR FUNCTION and the TARGET FUNCTION bits may be set to one. If neither the INITIATOR FUNCTION nor the TARGET FUNCTION bit is set to one, the service parameters for the FCP Service Parameter page are assumed to be invalid. A Responder receiving such an invalid FCP Service Parameter page shall notify the Originator with a PRLI accept reason code of INVALID SERVICE PARAMETERS OF PAGE and the IMAGE PAIR ESTABLISHED bit set to zero. An Originator receiving such an invalid FCP Service Parameter page shall not perform Fibre Channel protocol operations with the Responder.

Word 3, Bit 1: READ FCP_XFER_RDY DISABLED: The READ FCP_XFER_RDY DISABLED bit shall be set to one. Targets shall not send FCP_XFER_RDY on read operations.

Word 3, Bit 0: WRITE FCP_XFER_RDY DISABLED: When the WRITE FCP_XFER_RDY DISABLED bit is set to zero, FCP_XFER_RDY IUs shall be transmitted by the target to request each of the SCSI write FCP_DATA IUs from the initiator. When the WRITE FCP_XFER_RDY DISABLED bit is set to one, FCP_XFER_RDY IUs shall not be used before the first FCP_DATA IU to be transferred in the SCSI write operation. If both the Originator and Responder choose to disable write FCP_XFER_RDY IUs, then all FCP I/O operations performing SCSI writes between the FCP_Ports shall operate without using the FCP_XFER_RDY IU before the first FCP_DATA IU. The FCP_XFER_RDY IU shall be transmitted to request each additional FCP_DATA IU, if any. If either the Originator or the Responder requires the use of FCP_XFER_RDY IUs during SCSI writes, then the Exchange Responder shall transmit an FCP_XFER_RDY IU requesting each FCP_DATA IU, including the first, from the Exchange Originator.

6.3.5 Process Login accept FCP Service Parameter page format

The FCP Service Parameter page for the Process Login accept is shown in table 11.

Table 11 - FCP Service Parameter page, PRLI accept

FCP service parameter	Word	Bit
SCSI FCP (08h)	0	31–24
reserved for TYPE Code Extension	0	23–16
ORIGINATOR PROCESS_ASSOCIATOR VALID	0	15
RESPONDER PROCESS_ASSOCIATOR VALID	0	14
IMAGE PAIR ESTABLISHED	0	13
RESERVED	0	12
ACCEPT RESPONSE CODE	0	11–8
RESERVED	0	7–0
Originator Process_Associator	1	31–0
Responder Process_Associator	2	31–0
RESERVED	3	31–10
TASK RETRY IDENTIFICATION REQUESTED	3	9
RETRY	3	8
CONFIRMED COMPLETION ALLOWED	3	7
DATA OVERLAY ALLOWED	3	6
INITIATOR FUNCTION	3	5
TARGET FUNCTION	3	4
OBSOLETE	3	3
OBSOLETE	3	2
READ FCP_XFER_RDY DISABLED (shall be one)	3	1
WRITE FCP_XFER_RDY DISABLED	3	0

With the following exceptions, the service parameter definitions are identical for the PRLI request and accept FCP Service Parameter pages.

Word 0, Bit 13: IMAGE PAIR ESTABLISHED: The IMAGE PAIR ESTABLISHED bit is defined in FC-FS. If the IMAGE PAIR ESTABLISHED bit is set to zero, the image pair was not established. The ACCEPT RESPONSE CODE has additional information.

If the IMAGE PAIR ESTABLISHED bit is set to one, the image pair was established.

PRLI ACCEPT RESPONSE CODE: The PRLI ACCEPT RESPONSE CODE field is defined in FC-FS. The values of the PRLI ACCEPT RESPONSE CODE field indicate whether the image pair was successfully created. If the image pair could not be created, the value of the PRLI ACCEPT RESPONSE CODE indicates why the request failed or was rejected.

6.4 Process Logout (PRLO)

The Process Logout (PRLO) request is transmitted from an Originator FCP_Port to a Responder FCP_Port to indicate to the Responder that the image pair specified in the FCP Service Parameter pages of the PRLO is being discontinued by the Originator. If the PRLO logs out the image pair between an initiator and a target, then all clearing actions specified in 4.9 shall be performed and an I_T nexus loss notification shall be delivered (see 4.10).

For the Fibre Channel protocol, the PRLO FCP Service Parameter page identifies an image pair where neither the Originator or Responder supports Process_Associators by marking the Originator Process_Associator and Responder Process_Associator as invalid. After PRLO, no further Fibre Channel protocol communication is possible between those N_Ports or NL_Ports.

The PRLO accept (ACC) is returned to the Originator FCP_Port to indicate that the Responder FCP_Port recognizes that the image pair is being discontinued. The ACC shall present a response FCP Service Parameter page for the request FCP Service Parameter page. It is not an error to perform a PRLO for an image pair that does not exist.

A link service reject (LS_RJT) indicates that the PRLO request is invalid and not accepted.

The PRLO common service parameters and accept response codes are defined in FC-FS.

6.5 Read Exchange Concise (REC)

See FC-FS for a description of the REC Extended Link Service. FCP-2 specific usage of REC is as follows:

a) If the task retry identification function is active for the Originator and the Responder, the PARAMETER field of the request Sequence shall contain the task retry identifier for the task specified by the OX_ID and RX_ID;

b) If the destination FCP_Port of the REC request determines that the Originator S_ID, OX_ID, RX_ID, or task retry identifier are inconsistent, then it shall respond with an LS_RJT Sequence with a reason code of “Logical error” and a reason explanation of “Invalid OX_ID-RX_ID combination”;

c) The REC shall be sent in a new Exchange. The Exchange shall be ended by the response to the REC;

d) If the RX_ID in the REC request payload was FFFFh, the RX_ID in the REC accept payload may be set to the value selected by the Responder when the first frame of the Exchange was received; and

e) The FC4VALUE field is the number of bytes successfully received by the Device Server for a write or the number of bytes transmitted by the target for a read.

7 FC-4 specific name server objects

7.1 Overview of FC-4 specific objects for the Fibre Channel protocol

The Name Server for a Fibre Channel fabric is defined by FC-GS-3. Certain FCP specific objects are defined in this clause for use by the Name Server. FC-GS-3 provides complete descriptions of the operations which can be performed to register objects with a Name Server and to query the Name Server for the value of the objects.

7.2 FC-4 Features object

The FC-4 Features object defines a 4-bit field for each possible FC TYPE code. The object is a 32-word array of 4-bit values. The 4-bit FC-4 Feature field for FCP is inserted in bits 3 - 0 of word 1. The format of the 4-bit FC-4 Feature field for FCP is shown in table 12.

Table 12 - FCP definition of FC-4 Feature bits

Word 1 bit	Description of bit
3	reserved
2	reserved
1	FCP initiator function supported
0	FCP target function supported

An FCP_Port may register its FC-4 Features object with a Name Server using the RFF_ID Request CT_IU, which provides the FC-4 Features object as one of the parameters in the RFF_ID Request CT_IU.

The FC-4 Features object may be obtained by any N_Port or NL_Port from a Name Server using a GFF_ID Request CT_IU, which requests the FC-4 Features object for a specified Port Identifier. The object is provided in the GFF_ID Accept CT_IU.

A list of all the Port Identifiers matching the domain and area addressing and a specified FC-4 Features object may be obtained by any N_Port or NL_Port from a Name Server using the GID_FF Request CT_IU. The FC-4 Features object is a parameter in the GID_FF Request CT_IU.

7.3 FC-4 Descriptor object

The FC-4 Descriptor object is an object up to 255 bytes in length defined for each FC-4 TYPE. For target FCP_Ports, the object shall be the INQUIRY data returned by the INQUIRY command for logical unit 0 of the FCP_Port, as specified by SPC-2. For FCP_Ports implementing only the initiator function, the object shall have a length of zero, indicating that no object is available.

An FCP_Port may register its FC-4 Descriptor object with a Name Server using the RFD_ID Request CT_IU, which provides a list of one or more FC-4 Descriptor objects as parameters of the Request CT_IU.

The FC-4 Descriptor object may be obtained by any N_Port or NL_Port from a name server using a GFD_ID Request CT_IU, which requests the FC-4 Descriptor object for a specified Port Identifier and one or more FC-4 TYPEs. The FC-4 Descriptor objects, one for each specified FC-4 TYPE, are provided in a list in the GFD_ID Accept CT_IU.

8 FC-4 Link Service definitions

8.1 FC-4 Link Services for the Fibre Channel protocol

The format for FC-4 Link Service requests is defined by FC-FS. R_CTL bits 31-28 (Word 0) are set to 0011b to indicate that the frame is an FC-4 Link_Data frame. The TYPE field for FCP FC-4 Link Service frames is set to 0000 1000b. The Information Category bits R_CTL 27-24 shall be set to unsolicited control (0010b) for request Sequences and solicited control (0011b) for response Sequences. The FCP FC-4 Link Service Requests and Responses defined in this standard are shown in table 13.

Table 13 - FCP FC-4 Link Service Requests and Responses for FCP-2

Encoded Value Word 0 of Payload (bits 31-24)	Description	Abbr.	Request/ Response	Reference
14h	Sequence Retransmission Request	SRR	Request	8.2
01h	FCP FC-4 Link Service Reject	FCP_RJT	Response	8.3

8.2 Sequence Retransmission Request (SRR)

The SRR FCP FC-4 Link Service request Sequence is transmitted by an initiator to request that a target retransmit information or request retransmission of information for the specified Exchange.

If the task retry identification function is active for the Originator and the Responder, the PARAMETER field of the request Sequence shall contain the task retry identifier for the task specified by the OX_ID and RX_ID.

If the target FCP_Port determines that the Originator S_ID, OX_ID, RX_ID, or task retry identifier are inconsistent, then it shall respond with an FCP FC-4 Link Service Reject Sequence with a reason code of “Unable to perform the command request” and a reason explanation of “Invalid OX_ID-RX_ID combination”.

If the target is unable to retransmit the Sequence or data at the requested Relative Offset, the target shall respond with an FCP FC-4 Link Service Reject Sequence with a reason code of “Unable to perform the command request” and a reason explanation of “Unable to supply requested data”.

If the initiator receives an FCP FC-4 Link Service Reject response, the initiator shall terminate the Exchange referenced by the SRR using recovery abort. See 12.3.2.

The SRR shall be sent in a new Exchange. The Exchange shall be ended by the response to the SRR.

Sequence Initiative for the Exchange referenced by the SRR shall be transferred to the target to resend the requested Sequence.

For unacknowledged classes, the Sequence Count for retransmitted FCP_DATA IU shall start at zero, even if continuously increasing sequence count is being used. For acknowledged classes, the Sequence Count for retransmitted FCP_DATA IU shall start at one higher than the last Sequence Count used in the exchange to prevent its being within the range of the Recovery_Qualifier.

Addressing:

The S_ID field designates the initiator requesting the information retransmission. The D_ID field designates the target that is to receive the request. In the event that the target responds to the SRR with an FCP FC-4 Link Service Reject, the target shall return CHECK CONDITION status with the sense key set to HARDWARE ERROR and an additional sense code of INITIATOR DETECTED ERROR MESSAGE RECEIVED. A target that has agreed during PRLI to support retransmission should not reject requests for retransmission of the requested

frames unless unusual conditions make the retransmission impossible. SRR requests for exchanges involving logical units that do not support retransmission on a target that supports retransmission for other logical units shall be rejected with an FCP FC-4 Link Service Reject containing a reason code of “unable to support command request” and a reason code explanation of “unable to supply requested data”.

Payload for SRR FCP FC-4 Link Service request:

The format of the Payload is shown in table 14.

Table 14 - SRR Payload

Bits / Words	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0	14000000h			
1	OX_ID		RX_ID	
2	RELATIVE OFFSET			
3	R_CTL FOR IU	reserved		

The R_CTL FOR IU field encoding is as described in FC-FS (i.e., Data Descriptor (FCP_XFER_RDY IU), Command Status (FCP_RSP IU), Solicited Data (FCP_DATA IU)).

The RELATIVE OFFSET parameter is only valid if the R_CTL FOR IU field is set to 01h for Solicited Data or to 05h for Data Descriptor. The RELATIVE OFFSET field contains the Relative Offset of the lowest byte the initiator has identified as requiring retransmission. The two low-order bits of the RELATIVE OFFSET field shall be zero, such that the data to be retransmitted begins on a four-byte boundary.

The amount of data to transfer is implicitly the remainder of that for the Exchange.

Possible responses to SRR FCP FC-4 Link Service request:

FCP FC-4 Link Service Reject

Signifies rejection of the SRR request.

SRR Accept

Signifies that the Payload is accepted.

SRR Accept Payload:

The payload for the SRR Accept is shown in table 15.

Table 15 - SRR Accept Payload

Bits / Words	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0	02000000h			

8.3 FCP FC-4 Link Service Reject

The FCP FC-4 Link Service Reject (FCP_RJT) shall notify the originator of an FCP FC-4 Link Service request that the FCP FC-4 Link Service request Sequence has been rejected. A four-byte reason code shall be contained in the Data_Field. See table 16.

An FCP FC-4 Link Service Reject may be a response Sequence to any FCP FC-4 Link Service request.

Addressing:

The D_ID field designates the source of the FCP FC-4 Link Service request being rejected. The S_ID field designates the destination of the request FCP FC-4 Link Service frame Sequence being rejected.

Payload for FCP FC-4 Link Service Reject:

The first word of the Payload shall contain the FCP FC-4 Link Service Reject code (01000000h). The next four bytes of this field shall contain a reason code and reason explanation for rejecting the request. The format of the FCP FC-4 Link Service Reject Payload is shown in table 16.

Table 16 - FCP FC-4 Link Service Reject (FCP_RJT) Payload

Bits / Words	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0	01000000h			
1	RESERVED	REASON CODE	REASON EXPLANATION	VENDOR SPECIFIC

The reason codes for FCP FC-4 Link Service Reject are specified in table 17.

Table 17 - FCP FC-4 Link Service Reject reason codes

Encoded Value (Bits 23-16)	Description
01h	Invalid FCP FC-4 Link Service Command code
03h	Logical error
05h	Logical busy
07h	Protocol error
09h	Unable to perform command request
0Bh	Command not supported
FFh	Vendor Specific Error (See Bits 7-0)
other	Reserved

The first error condition encountered shall be the error reported.

FCP FC-4 Link Service Reject Reason Code Descriptions

Invalid FCP FC-4 Link Service Command code: The FCP_LS Command code in the Sequence being rejected is invalid.

Logical error: The request identified by the FCP_LS Command code and Payload content is invalid or logically

inconsistent for the conditions present.

Logical busy: The Link Service is logically busy and unable to process the request at this time.

Protocol error: This indicates that an error has been detected that violates the rules of the FC-FS signaling protocol, but that is not specified by other error codes.

Unable to perform command request: The Recipient of a Link Service command is unable to perform the request at this time.

Command not supported: The Recipient of a Link Service command does not support the command requested.

Vendor Specific Error: The Vendor Specific Error bits may be used by Vendors to specify additional reason codes.

FCP FC-4 Link Service Reject Reason explanation

Table 18 lists the reason code explanations for FCP FC-4 Link Service commands.

Table 18 - FCP FC-4 Link Service Reject reason code explanation

Encoded Value (Bits 15-8)	Description	Applicable commands
00h	No additional explanation	SRR
03h	Invalid OX_ID-RX_ID combination	SRR
2Ah	Unable to supply requested data	SRR

9 FCP Information Unit (IU) formats

9.1 FCP_CMND IU

9.1.1 FCP_CMND IU format

The FCP_CMND IU carries either a SCSI Command or a task management request. If an invalid combination of bits is set in the FCP_CMND IU, the target shall respond with an FCP_RSP IU containing a RSP_CODE field set to “FCP_CMND Fields Invalid”. The FCP_CMND IU shall contain the values and control fields defined in table 19 in its payload.

Table 19 - FCP_CMND IU Payload

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	FCP_LUN						
7								(LSB)
8	COMMAND REFERENCE NUMBER							
9	RESERVED					TASK ATTRIBUTE		
10	TASK MANAGEMENT FLAGS							
11	ADDITIONAL FCP_CDB LENGTH = (N-27)/4						RDDATA	WRDATA
12	(MSB)	FCP_CDB						
27								(LSB)
28	(MSB)	ADDITIONAL FCP_CDB						
n								(LSB)
n+1	(MSB)	FCP_DL						
n+2								
n+3								
n+4								(LSB)

9.1.2 FCP_CMND IU Field descriptions

9.1.2.1 FCP_LUN

The FCP logical unit number (FCP_LUN) field contains the address of the destination logical unit in the attached subsystem. See SAM-2.

Each target shall accept an INQUIRY command addressed to logical unit zero. If LUNs other than logical unit zero are supported by the target, logical unit zero shall implement the REPORT LUNS command. See SPC-2.

If the FCP_LUN field contains a valid logical unit address the command shall be routed to the addressed logical unit. If the addressed logical unit does not exist, the target shall report that the logical unit number is not valid or that the logical unit is not installed as defined by SPC-2.

9.1.2.2 COMMAND REFERENCE NUMBER

The COMMAND REFERENCE NUMBER (CRN) field contains the number sent by the initiator to assist in performing precise delivery checking for FCP commands. If precise delivery is enabled, a nonzero value in the CRN field shall be treated as a command reference number in determining the receipt and ordering of commands from a particular initiator to the particular logical unit as described in 4.3. If precise delivery is enabled, a zero value in the CRN field indicates that command shall not be verified for precise delivery. If precise delivery checking is not enabled, the COMMAND REFERENCE NUMBER field shall be ignored by the device server. If the FCP_CMND IU specifies a task management function, the CRN field shall be reserved and set to zero and the FCP_CMND IU shall not be verified for precise delivery.

9.1.2.3 TASK ATTRIBUTE

The TASK ATTRIBUTE field contains values that specify the task attribute (see SAM-2) associated with the CDB, as shown in table 20.

Table 20 - TASK ATTRIBUTE field values

Value, bits 2–0	task attribute
000b	SIMPLE
001b	HEAD OF QUEUE
010b	ORDERED
100b	ACA
101b	UNTAGGED
others	Reserved

SIMPLE requests that the task be managed according to the rules for a SIMPLE task attribute.

HEAD OF QUEUE requests that the task be managed according to the rules for a HEAD OF QUEUE task attribute.

ORDERED requests that the task be managed according to the rules for an ORDERED task attribute. Mechanisms to assure delivery of commands to a device server in the correct order are described in 4.3.

ACA requests that the task be managed according to the rules for an automatic contingent allegiance (ACA) task attribute.

UNTAGGED requests that the task be managed according to the rules for an untagged task. Only one untagged task shall exist for each logical unit / initiator pair. Requesting a second untagged command for the same logical unit / initiator pair shall be treated as an overlapped command. See SAM-2.

9.1.2.4 TASK MANAGEMENT FLAGS

The TASK MANAGEMENT FLAGS field contains flags that request that a task management function be performed. Task management functions shall be requested by the initiator (Exchange Originator) using a new Exchange. If any task management flag bit is set to one, the FCP_CDB field, the FCP_DL field, the TASK ATTRIBUTES field, the RDDATA bit, and the WRDATA bit shall be ignored. If more than one task management flag bit is set to one in any FCP_CMND IU, a task management function shall not be executed and the FCP_RSP IU that indicates completion of the task management function shall contain a RSP_CODE field set to “FCP_CMND fields invalid”.

The clearing actions performed by task management functions are shown in table 5. The placement of the task management flags is shown in table 21.

Table 21 - task management Flags

bit	task management function
7	obsolete
6	CLEAR ACA
5	TARGET RESET
4	LOGICAL UNIT RESET
3	reserved
2	CLEAR TASK SET
1	ABORT TASK SET
0	reserved

The **CLEAR ACA** bit, when set to one, causes the ACA condition to be cleared. When the task manager clears the auto contingent allegiance condition, any task within that task set may be completed subject to the rules for task management specified by SAM-2. If there is no ACA condition present, the CLEAR ACA Task Management function shall be accepted and the normal Task Management function complete RSP_CODE shall be contained in the returned FCP_RSP IU.

When set to zero, the ACA condition remains unchanged.

The use of the ACA bit in the CDB control field and the implementation of ACA is described in SAM-2.

If the ACA bit in the CDB control field is set to zero, the autosense operation performed by the presentation of the FCP_RSP IU shall clear the contingent allegiance condition. Depending on the mode page parameters that have been established (see SPC-2), additional FCP I/O operations may have to be aborted by the recovery abort as part of the process of clearing the automatic contingent allegiance.

The CLEAR ACA is transmitted by the initiator (Exchange Originator) using a new Exchange.

Support of the CLEAR ACA bit is mandatory in the Fibre Channel protocol if the FCP device sets the NORMACA bit to one in the INQUIRY data. It shall not be sent to a target with a NORMACA bit equal to zero in the INQUIRY data.

The **TARGET RESET** bit, when set to one, performs a TARGET RESET task management function to the FCP device as defined in SAM-2. TARGET RESET aborts all tasks for all initiators. A unit attention condition is created for all initiators. Support of the TARGET RESET bit is mandatory for the Fibre Channel protocol.

The TARGET RESET is transmitted by the initiator (Exchange Originator) using a new Exchange. TARGET RESET resets internal states of the target as shown in 4.9. Exchange resources to be cleared may be cleared by one or more of the following mechanisms.

a) A recovery abort sequence (see 12.3.1) may be generated by the initiator that sent the TARGET RESET for each task known to that initiator.

b) A task for an initiator other than the initiator that sent the TARGET RESET may be ended in the target. The initiator for that task shall determine by a time-out that the task did not finish. Subsequent retries fail because the task resources have been cleared in the target, so the initiator shall clear the Exchange resources with a recovery abort sequence. See 12.4.1.

c) A task for an initiator other than the initiator that sent the TARGET RESET may be completed by returning SCSI status of CHECK CONDITION with a sense key of UNIT ATTENTION and an ASC/ASCQ of POWER ON, RESET, OR BUS DEVICE RESET OCCURRED. The initiator shall then clear all other tasks for that target using the ABORT TASK task management function. See 9.1.3.

NOTE 1 -SAM-2 has defined TASK ABORTED completion status for tasks terminated by a TARGET RESET task management function if the Control mode page indicates that the TASK ABORTED status is supported.

The **LOGICAL UNIT RESET** bit, when set to one, performs a LOGICAL UNIT RESET task management function as defined in SAM-2. LOGICAL UNIT RESET aborts all tasks in the task set for the logical unit and performs a LOGICAL UNIT RESET for all dependent logical units. Support of the LOGICAL UNIT RESET bit is mandatory for the Fibre Channel protocol.

The LOGICAL UNIT RESET is transmitted by the initiator (Exchange Originator) using a new Exchange. LOGICAL UNIT RESET resets the internal states of the target and logical unit as shown in 4.9. Exchange resources to be cleared may be cleared by the following mechanisms.

a) A recovery abort sequence (see 12.3.1) may be generated by the initiator that sent the LOGICAL UNIT RESET for each task in the logical unit known to that initiator.

b) A task for an initiator other than the initiator that sent the LOGICAL UNIT RESET may be ended in the target. The initiator for that task shall determine by a time-out that the task did not finish. Subsequent retries fail because the task resources have been cleared in the target, so the initiator shall clear the Exchange resources with a recovery abort sequence. See 12.4.1.

c) A task for an initiator other than the initiator that sent the LOGICAL UNIT RESET may be completed by returning SCSI status of CHECK CONDITION with a sense key of UNIT ATTENTION and an ASC/ASCQ of POWER ON, RESET, OR BUS DEVICE RESET OCCURRED. The initiator shall then clear all other tasks for that target and logical unit using the ABORT TASK task management function. See 9.1.3.

NOTE 2 - SAM-2 has defined TASK ABORTED completion status for tasks terminated by a LOGICAL UNIT RESET task management function if the Control mode page indicates that the TASK ABORTED status is supported.

The **CLEAR TASK SET** bit causes all tasks from all initiators in the specified task set to be aborted as defined in SAM-2. A unit attention condition is created for all initiators other than the initiator that sent the CLEAR TASK SET that had tasks in the task set. Support of the CLEAR TASK SET bit is mandatory for the Fibre Channel protocol.

The CLEAR TASK SET is transmitted by the initiator (Exchange Originator) using a new Exchange. CLEAR TASK SET resets internal states of the target as shown in 4.9. Exchange resources to be cleared may be cleared by one or more of the following mechanisms.

a) A recovery abort sequence (see 12.3.1) may be generated by the initiator that sent the CLEAR TASK SET for each task known to that initiator.

b) A task for an initiator other than the initiator that sent the CLEAR TASK SET may be ended in the target. The initiator for that task shall determine by a time-out that the task did not finish. Subsequent retries fail because the task resources have been cleared in the target, so the initiator shall clear the Exchange resources with a recovery abort sequence. See 12.4.1.

c) A task for an initiator other than the initiator that sent the CLEAR TASK SET may be completed by returning SCSI status of CHECK CONDITION with a sense key of UNIT ATTENTION and an ASC/ASCQ of POWER ON, RESET, OR BUS DEVICE RESET OCCURRED. The initiator shall then clear all other tasks for that target using the ABORT TASK task management function. See 9.1.3.

NOTE 3 - SAM-2 has defined TASK ABORTED completion status for tasks terminated by a CLEAR TASK SET task management function if the Control mode page indicates that the TASK ABORTED status is supported.

The **ABORT TASK SET** bit causes all tasks in the task set from the initiator requesting the ABORT TASK SET to be aborted as defined in SAM-2. Support of the ABORT TASK SET bit is mandatory in the Fibre Channel protocol.

The ABORT TASK SET is transmitted by the initiator (Exchange Originator) using a new Exchange. ABORT TASK SET resets internal states of the target as shown in 4.9. Exchange resources to be cleared may be cleared by a recovery abort sequence (see 12.3.1) generated by the initiator that sent the ABORT TASK SET for each task known to the initiator.

9.1.2.5 ADDITIONAL FCP_CDB LENGTH

The ADDITIONAL FCP_CDB LENGTH field contains the length in 4-byte words of the ADDITIONAL FCP_CDB field. The value of the ADDITIONAL FCP_CDB LENGTH field shall be zero for task management requests.

9.1.2.6 RDDATA

The RDDATA bit, when set to one, specifies that the initiator expects FCP_DATA IUs for the task to be in the direction opposite to the direction of the FCP_CMND IU. This is a SCSI read operation.

9.1.2.7 WRDATA

The WRDATA bit, when set to one, specifies that the initiator expects FCP_DATA IUs for the task to be in the same direction as the FCP_CMND IU. This is a SCSI write operation. If both RDDATA and WRDATA are set to zero, there shall be no FCP_DATA IUs and FCP_DL shall be zero. The initiator shall not set both the RDDATA and the WRDATA bits to one except as defined by Annex .

9.1.2.8 FCP_CDB

The FCP_CDB field contains the CDB to be sent to the addressed logical unit. The maximum CDB length is 16 bytes unless the ADDITIONAL FCP_CDB_LENGTH field has specified that there is an ADDITIONAL_FCP_CDB field. The FCP_CDB shall be ignored if any task management flag is set to one.

The CDB format is defined by SAM-2 and the contents of the CDB are defined in the SCSI command standards. Bytes between the end of a CDB and the end of the FCP_CDB field or, if applicable, the ADDITIONAL_FCP_CDB field shall be reserved.

9.1.2.9 ADDITIONAL_FCP_CDB

The ADDITIONAL_FCP_CDB field contains any CDB bytes beyond those contained within the 16 byte FCP_CDB field. The ADDITIONAL_FCP_CDB field shall not be present if any task management flag is set to one. The contents of the field shall be those bytes of an extended CDB beyond the first 16 bytes of the CDB as defined in the SCSI command standards.

9.1.2.10 FCP_DL

The FCP_DL field contains a count of the greatest number of data bytes to be transferred to or from the application client data buffer by the SCSI CDB. The FCP_DL field is the data buffer size defined by SAM-2. An FCP_DL value of zero indicates that no data transfer is expected regardless of the state of the RDDATA and WRDATA bits and that no FCP_XFER_RDY or FCP_DATA IUs shall be transferred.

9.1.3 Additional mechanisms for performing task management functions - ABORT TASK

The **ABORT TASK** task management function causes the target to abort the specified task using the recovery abort protocol, if the task exists. The action is defined in SAM-2. The ABORT TASK is performed by the initiator (Exchange Originator) using the recovery abort (see 12.3.1). The specified Exchange shall be terminated by the initiator using the recovery abort. To be compliant with FC-FS, the ABORT TASK may not immediately release all Exchange resources, since a Recovery_Qualifier may be established to allow for the management of information that may already have been delivered to the fabric.

In addition to recovering Exchange resources that may have been left unavailable while executing task management functions, recovery abort may be used to recover Exchange resources left in an undefined state by any of the task abort events defined in SAM-2 or by any similar events.

9.2 FCP_XFER_RDY IU

9.2.1 Overview and format of FCP_XFER_RDY IU

The FCP_XFER_RDY IU indicates that the target is prepared to receive part or all of the data for a write command. The FCP_XFER_RDY IU contains those parameters of the SAM-2 data delivery service required by the initiator, including the length and beginning relative offset of the FCP_DATA IU that is requested. Since the target has established buffering and caching resources based on the requested data, the initiator shall provide the described data in the requested FCP_DATA IU. The initiator shall be ready to transmit any part or all of the number of bytes indicated in the FCP_DL field if requested.

FCP_XFER_RDY IUs shall be transmitted preceding each write FCP_DATA IU when the WRITE FCP_XFER_RDY DISABLED bit is set to zero by PRLI. If the target and initiator have negotiated write FCP_XFER_RDY disabled, FCP_XFER_RDY IUs shall be transmitted to request each write FCP_DATA IU after the first FCP_DATA IU of the command. The first FCP_DATA IU is transmitted without a preceding FCP_XFER_RDY IU. See 6.3.4.

The first 8 bytes of the FCP_XFER_RDY IU payload are defined by FC-FS for all IUs of category 5, the data descriptor category. The fields defined in FC-FS are given FCP names for use in this document. The format of the FCP_XFER_RDY IU payload is shown in table 22.

Table 22 - FCP_XFER_RDY IU payload

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
				FCP_DATA_RO				
3								(LSB)
4	(MSB)							
				FCP_BURST_LEN				
7								(LSB)
8								
				RESERVED				
11								

9.2.2 FCP_DATA_RO

The FCP_DATA_RO field contains a value specifying the relative offset in the PARAMETER field for the first data byte of the requested FCP_DATA IU. See 5.6.2.12. This may be used by the target to request data out of order on writes if allowed by the EMDP bit in the disconnect-reconnect page of the MODE SELECT and MODE SENSE commands. See 10.2. This is the same as the SAM-2 application client buffer offset.

The FCP_DATA_RO field shall have a value that is 0 modulo 4 (i.e., each FCP_DATA IU shall begin on a word boundary).

9.2.3 FCP_BURST_LEN

The FCP_BURST_LEN field contains a value indicating the amount of buffer space prepared for the next FCP_DATA IU and requests the transfer from the initiator of an IU of that length. The value in the FCP_BURST_LEN field is the same as the SCSI data delivery request byte count. See SAM-2.

The value in the FCP_BURST_LEN field shall not exceed the maximum burst length defined by the disconnect-reconnect page of MODE SELECT and MODE SENSE commands. See 10.2.7. The sum of the value of FCP_BURST_LEN field and the value of FCP_DATA_RO shall not exceed the value of FCP_DL. The value in the FCP_BURST_LEN field shall not be zero.

9.3 FCP_DATA IU

The data associated with a particular FCP I/O operation is transmitted in the same exchange that sent the FCP_CMND IU requesting the transfer.

SCSI data transfers may be performed by one or more data delivery requests, each one performing a transfer no longer than the maximum burst length defined by the parameters of the disconnect-reconnect page of the MODE SENSE and MODE SELECT commands.

If more than one FCP_DATA IU is used to transfer the data, the relative offset value in the PARAMETER field is used to ensure that the SCSI data is reassembled in the proper order. See 5.6.2.12. If an FCP_XFER_RDY IU is used to describe a data transfer and the first frame of the requested FCP_DATA IU has a relative offset that differs from the value in the FCP_DATA_RO field of the FCP_XFER_RDY IU, the target shall post the error code “FCP_DATA Parameter mismatch with FCP_DATA_RO” in the FCP_RSP_INFO field of the FCP_RSP IU.

If required by the PRLI FCP service parameters, each Data Out action FCP_DATA IU shall be preceded by an FCP_XFER_RDY IU containing a standard data descriptor payload that indicates the location and length of the data delivery. If the PRLI FCP Service Parameters specify WRITE FCP_XFER_RDY DISABLED, the first FCP_DATA IU shall be transmitted without a preceding FCP_XFER_RDY IU.

During any data transfer, the initiator shall have available a buffer of length FCP_DL. The buffer contains data to be transferred to the target if the operation is a write operation (i.e., an operation that uses Data Out actions, IU T6). The buffer receives the data if the operation is a read operation (i.e., an operation that uses the Data In action, IU I3). The target shall never request or deliver data outside the buffer length defined by FCP_DL. If the command requested that data beyond FCP_DL be transferred, the FC_RSP IU shall contain the FCP_RESID_OVER bit set to one. The command is completed normally except that data beyond the FCP_DL count shall not be transferred and that the appropriate overrun condition is presented. See 9.4.4.

During a write operation that is not using FCP_XFER_RDY IUs, the initiator indicates that it has transferred all the required data by transferring initiative to the target. The initiator shall not transfer data outside the buffer length defined by FCP_DL. If the write operation requires a total amount of data less than the amount of data provided by the initiator, the target shall discard the excess bytes. Because there were fewer bytes provided

than required by FCP_DL, the FCP_RESID_UNDER bit shall be set to 1 in the FCP_RSP IU. The command is completed according to the rules specified by the SCSI command set for that command.

If the PRLI service parameter DATA OVERLAY ALLOWED for the initiator is set to one, the target may request that data be overlaid. If the PRLI service parameter DATA OVERLAY ALLOWED for the initiator is zero, the target shall not request that data be overlaid. If data overlay is not allowed and the target attempts to overlay data, the initiator may not be able to guarantee data integrity and may indicate service delivery failure. Data retransmission as part of an error recovery process is not considered data overlay, even if retransmission occurs to the same offset in the application client buffer.

The target may request data bursts in any order if allowed by the EMDP bit in the disconnect-reconnect page of the MODE SELECT and MODE SENSE commands. See 10.2. By the time data transfer has been terminated, all data between the offset of zero and the highest offset shall have been transferred. If error conditions occur that prevent the transfer of data in the middle of a data transfer, the FCP_SNS_INFO shall indicate that only data from the offset of zero up to the first byte of missing data has been transferred. Even if data of a higher offset was successfully transferred, it shall not be considered valid.

If the amount of data returned does not match the number of bytes of data calculated from the value of FCP_DL and the value of FCP_RESID (see 9.4.8), the error detection and recovery procedure described in clause 12 may be invoked or the FCP I/O operation may be terminated with a recovery abort or other failure indication. The mechanism an initiator uses to determine that the correct amount of data has been returned is vendor specific. Data that has been retransmitted and overlaid shall be counted only once for the purposes of calculating residual values.

FC-FS specifies the mechanisms used to transfer an IU. The mechanisms vary with the Class of Service being used and the service parameters that are in effect.

9.4 FCP_RSP IU

9.4.1 Overview and format of FCP_RSP IU

The FCP_RSP IU provides completion information for FCP I/O operations. The information includes SCSI status, protocol verification, and any applicable autosense data. The FCP_RSP IU shall return the completion status of all task management functions using the FCP_RSP_INFO field.

The bits and fields in bytes 10 and 11 summarize the completion status of an FCP I/O operation and indicate the meaning and validity of other fields in the FCP_RSP IU. Bytes 10 and 11 shall be zero upon successful completion of an FCP I/O operation, indicating that no other information is present in the FCP_RSP IU. A nonzero value in either byte 10 or byte 11 should cause the application client to examine the fields in FCP_RSP IU to determine whether a failure, a retryable temporary condition, or an expected response occurred.

If command linking is being performed, an FCP_RSP IU is provided for each command. For linked commands, INTERMEDIATE status or INTERMEDIATE - CONDITION MET status indicates successful completion of a command with no other information valid if all other fields are zero. If command linking is requested, the use of the INTERMEDIATE or INTERMEDIATE - CONDITION MET status indicates that linking shall be performed. The LINKED COMMAND COMPLETE or LINKED COMMAND COMPLETE (WITH FLAG) Service Response defined by SAM-2 is implicit in the presentation of INTERMEDIATE or INTERMEDIATE - CONDITION MET status in the FCP_RSP IU.

If data retransmission is enabled and a Sequence error is detected, a target shall not transmit an FCP_RSP IU with CHECK CONDITION status. See 12.3.5 for additional target error recovery.

If a SCSI device error is detected by a target while the target has Sequence Initiative for the Exchange associated with the error, the target should complete any Sequence that has already been started, keep

Sequence Initiative and transmit an FCP_RSP IU with CHECK CONDITION status and the sense data that describes the error. If a SCSI device error is detected by a target while the target does not have Sequence Initiative for the Exchange associated with the error, it shall wait until Sequence Initiative has been returned and then transmit an FCP_RSP IU with CHECK CONDITION status and the sense data that describes the error.

In the event that Sequence Initiative is not received within RR_TOV, the target may implicitly terminate the affected Exchange (see 11.4).

The content of the FCP_RSP IU is indicated in table 23.

Table 23 - FCP_RSP IU Payload

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
7								
8	RESERVED							
9	RESERVED							
10	RESERVED				FCP_ CONF_ REQ	FCP_ RESID_ UNDER	FCP_ RESID_ OVER	FCP_ SNS_ LEN_ VALID	FCP_ RSP_ LEN_ VALID
11	SCSI STATUS CODE							
12	(MSB) FCP_RESID							
15								(LSB)
16	(MSB) FCP_SNS_LEN (= n)							
19								(LSB)
20	(MSB) FCP_RSP_LEN (= m)							
23								(LSB)
24	(MSB) FCP_RSP_INFO (m bytes long)							
23+m								(LSB)
24+m	(MSB) FCP_SNS_INFO (n bytes long)							
23+m+n								(LSB)

9.4.2 FCP_CONF_REQ

An FCP_CONF_REQ bit of one indicates that the initiator shall transmit an FCP_CONF IU to confirm receipt of the FCP_RSP Sequence. An FCP_CONF_REQ bit of zero indicates that the initiator shall not transmit an FCP_CONF IU.

9.4.3 FCP_RESID_UNDER

An FCP_RESID_UNDER bit of one indicates that the FCP_RESID field is valid and contains the count of bytes that were expected to be transferred, but were not transferred. The application client should examine the FCP_RESID field in the context of the command to determine whether or not an error condition occurred.

9.4.4 FCP_RESID_OVER

An FCP_RESID_OVER bit of one indicates that the FCP_RESID field is valid and contains the count of bytes that could not be transferred because the FCP_DL was not sufficient. The application client should examine the FCP_RESID field in the context of the command to determine whether or not an error condition occurred.

9.4.5 FCP_SNS_LEN_VALID

An FCP_SNS_LEN_VALID bit of one indicates that the FCP_SNS_INFO field contains valid information and that the FCP_SNS_LEN field is valid and non-zero and that it contains the count of bytes in the FCP_SNS_INFO field. The application client should examine the FCP_SNS_INFO field to determine whether or not an error condition occurred.

An FCP_SNS_LEN_VALID bit of zero indicates that the FCP_SNS_LEN field is not valid and shall be treated as if its value were zero. See 9.4.9.

9.4.6 FCP_RSP_LEN_VALID

An FCP_RSP_LEN_VALID bit of one indicates that the FCP_RSP_INFO field contains valid information and that the FCP_RSP_LEN field is valid and non-zero and that it contains the count of bytes in the FCP_RSP_INFO field. The application client should examine the FCP_RSP_INFO field to determine whether or not an error condition occurred. When the FCP_RSP_LEN_VALID bit is one, the content of the SCSI STATUS CODE field is not reliable and shall be ignored by the initiator.

For task management functions transmitted to the target using an FCP_CMND IU, the FCP_RSP_LEN_VALID bit shall be set to one, the FCP_RSP_LEN field shall be set to the specified value, and the information in the RSP_CODE field shall indicate the completion status of the task management function.

An FCP_RSP_LEN_VALID bit of zero indicates that the FCP_RSP_LEN field is not valid and shall be treated as if its value were zero. When the FCP_RSP_LEN_VALID bit is zero, the FCP_RSP_INFO field shall have a length of zero and shall not be present.

9.4.7 SCSI STATUS CODE

The SCSI STATUS CODE field contains the status code for the SCSI command being completed, as defined by SAM-2.

9.4.8 FCP_RESID

If the FCP_RESID_UNDER bit or the FCP_RESID_OVER bit is set to one, the FCP_RESID field contains a count of the number of residual data bytes that were not transferred in the FCP_DATA IUs for this SCSI command. Upon successful completion of an FCP I/O operation, the residual value is normally zero and the FCP_RESID value is not valid. FCP devices having indeterminate data lengths may have a nonzero residual byte count after completing valid operations. Targets are not required to verify that the data length implied by the contents of the CDB cause an overrun or underrun before beginning execution of a SCSI command.

If the FCP_RESID_UNDER bit is set to one, a transfer that did not fill the buffer to the expected displacement FCP_DL was performed and the value of FCP_RESID is defined as follows:

FCP_RESID = FCP_DL - (highest offset of any byte transmitted + 1)

A condition of FCP_RESID_UNDER may not be an error for some FCP devices and some commands.

If the FCP_RESID_OVER bit is set to one, 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 be transferred. The FCP_RESID is defined as follows:

FCP_RESID = (Transfer length required by command) - FCP_DL

If a condition of FCP_RESID_OVER is detected, the termination state of the FCP I/O operation is not certain. Data may or may not have been transferred and the SCSI status byte may or may not provide correct command completion information.

If the FCP_RESID_UNDER and the FCP_RESID_OVER bits are set to zero, the FCP_RESID field is not meaningful and may have any value. The FCP_RESID field is always included in the FCP_RSP IU.

NOTE 4 - Some early target implementations presented the FCP_RSP IU without the FCP_RESID, FCP_SNS_LEN, and FCP_RSP_LEN fields if the FCP_RESID_UNDER, FCP_RESID_OVER, FCP_SNS_LEN_VALID, and FCP_RSP_LEN_VALID bits were all set to zero. Initiators should be tolerant of this non-standard behavior.

9.4.9 FCP_SNS_LEN

If the FCP_SNS_LEN_VALID bit is one, the FCP_SNS_LEN field specifies the number of valid bytes of FCP_SNS_INFO.

If the FCP_SNS_LEN_VALID bit is zero, the FCP_SNS_LEN field is not valid and shall be treated as if its value were zero. No FCP_SNS_INFO is provided.

The FCP_SNS_LEN field is always included in the FCP_RSP IU.

9.4.10 FCP_RSP_LEN

If the FCP_RSP_LEN_VALID bit is one, the FCP_RSP_LEN field specifies the number of valid bytes of FCP_RSP_INFO. The number shall be 4, or 8. Other values of length are reserved for future standardization.

If the FCP_RSP_LEN_VALID bit is zero, the FCP_RSP_LEN field is not valid and shall be treated as if its value were zero. No FCP_RSP_INFO is provided.

The FCP_RSP_LEN field is always included in the FCP_RSP IU.

9.4.11 FCP_RSP_INFO

The FCP_RSP_INFO field contains information describing only the protocol failures detected during the execution of an FCP I/O operation. If none of the specified protocol failures have occurred, the FCP_RSP_INFO field shall not be included in the FCP_RSP IU and the FCP_RSP_LEN_VALID bit shall be zero. The FCP_RSP_INFO does not contain link error information, since FC-FS provides the mechanisms for presenting such errors. The FCP_RSP_INFO field does not contain SCSI logical unit error information, since that is contained in the FCP_SNS_INFO field as described in 9.4.12. The FCP_RSP_INFO field shall contain valid information if the target

detects any of the conditions indicated by an FCP RSP_CODE. The format of the FCP_RSP_INFO field is specified in table 24.

Table 24 - FCP_RSP_INFO field format

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
2								
3	RSP_CODE							
4	RESERVED							
7								

The valid RSP_CODE values are specified in table 25

Table 25 - RSP_CODE definitions

RSP_CODE definition	Value
Task Management function complete	00h
FCP_DATA length different than FCP_BURST_LEN	01h
FCP_CMND fields invalid	02h
FCP_DATA Parameter mismatch with FCP_DATA_RO	03h
Task Management function rejected	04h
Task Management function failed	05h
Reserved	06h – FFh

The completion status of the task management function is indicated by the RSP_CODE. If the Exchange is aborted before the FCP_RSP IU is returned, the completion status is unknown. If the RSP_CODE indicates “Task Management function failed”, the state of the target is unknown. Values 04h and 05h are not valid responses to SCSI commands.

Activities started by a task management function may continue after the FCP_RSP IU for the task management has been delivered.

9.4.12 FCP_SNS_INFO

The FCP_SNS_INFO field contains the autosense data specified by SPC-2. The proper FCP_SNS_INFO shall be presented when the SCSI status byte of CHECK CONDITION is presented as specified by SAM-2. If no conditions requiring the presentation of SCSI sense data have occurred, the FCP_SNS_INFO field shall not be included in the FCP_RSP IU and the FCP_SNS_LEN_VALID bit shall be zero. FCP devices shall perform autosense.

9.5 FCP_CONF IU

The FCP_CONF IU has no payload. It is used as described in 4.4 for an initiator to confirm the receipt of the FCP_RSP IU from a target. The frame shall be transmitted by an initiator when the confirmed completion protocol is supported by both the target and the initiator and when the confirmation has been requested by the FCP_CONF_REQ bit in the FCP_RSP IU.

10 SCSI mode parameters for the Fibre Channel protocol

10.1 Overview of mode page codes for the Fibre Channel protocol

Clause 10 describes the block descriptors and the pages used with MODE SELECT and MODE SENSE commands that control and report the behavior of the Fibre Channel protocol. All mode parameters not defined in this standard shall control the behavior of the FCP devices as specified in the appropriate command set standard. The mode pages are addressed to the device server of a logical unit. The logical unit shall provide the appropriate control parameters, if any, to the state machine implementing the connection to the Fibre Channel loop or link in a vendor specific manner. The mode pages associated with Fibre Channel protocol operation are listed in table 26.

Table 26 - Mode page codes for FCP

Page code	Description	Clause
02h	Disconnect-reconnect page	10.2
18h	Fibre Channel Logical Unit Control page	10.3
19h	Fibre Channel Port Control page	10.4
3Fh	Return all pages (valid only for the MODE SENSE command)	SPC-2

10.2 Disconnect-Reconnect mode page

10.2.1 Overview and format of Disconnect-Reconnect mode page for FCP

The disconnect-reconnect page (see table 27) provides the application client the means to attempt to optimize the performance of the service delivery subsystem. This subclause specifies the parameters defined by SPC-2 that are used by FCP devices and defines how FCP devices interpret the parameters. The initiator communicates with the device server to determine what values are most appropriate for a device server. The device server communicates the parameter values in this mode page to the target port, normally the Fibre Channel interface circuitry. This communication is internal to the target and FCP device and is outside the scope of this standard. If a field or bit contains a value that is not supported by the FCP device, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN PARAMETER LIST.

Table 27 - Disconnect-reconnect page (02h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	RESERVED	PAGE CODE (02h)					
1	PAGE LENGTH (0Eh)							
2	BUFFER FULL RATIO							
3	BUFFER EMPTY RATIO							
4	(MSB)	BUS INACTIVITY LIMIT						
5								(LSB)
6	(MSB)	DISCONNECT TIME LIMIT						
7								(LSB)
8	(MSB)	CONNECT TIME LIMIT						
9								(LSB)
10	(MSB)	MAXIMUM BURST SIZE						
11								(LSB)
12	EMDP	FAA	FAB	FAC	RESTRICTED	RESTRICTED		
13	RESERVED							
14	(MSB)	FIRST BURST SIZE						
15								(LSB)

An interconnect tenancy is the period of time when an FCP device owns or may access a shared Fibre Channel interconnect. For FC-AL-2 loops and Fibre Channel Class 1 connections, a tenancy typically begins when an FCP device successfully opens the connection and ends when the FCP device releases the connection for use by other device pairs. Data and other information transfers take place during interconnect tenancies.

Point-to-point or fabric-attached Class 2 or Class 3 links and many other configurations do not have a concept of interconnect tenancy and may perform transfers at any time.

10.2.2 BUFFER FULL RATIO

The BUFFER FULL RATIO field indicates to the device server, during read operations, how full the buffer should be prior to requesting an interconnect tenancy. Device servers that do not implement the requested ratio should round down to the nearest implemented ratio as defined in SPC-2. FCP devices attached to links that do not have the concept of interconnect tenancy shall round the ratio to zero and transmit data in a vendor specific manner.

The value contained in the BUFFER FULL RATIO field is defined by SPC-2.

10.2.3 BUFFER EMPTY RATIO

The BUFFER EMPTY RATIO field indicates to the device server, during write operations, how empty the buffer should be prior to transmitting an FCP_XFER_RDY IU that requests the initiator to send data. Device servers that do not implement the requested ratio should round down to the nearest implemented ratio as defined in SPC-2.

The value contained in the BUFFER EMPTY RATIO field is defined by SPC-2.

10.2.4 BUS INACTIVITY LIMIT

The BUS INACTIVITY LIMIT field indicates the maximum time that the target is permitted to maintain an interconnect tenancy without data or information transfer, measured in transmission word increments. If the bus inactivity limit is exceeded or if the bus is inactive and the target holding the bus detects that the limit is going to be exceeded, the device server shall end the interconnect tenancy. This value may be rounded as defined in SPC-2. A value of zero indicates that there is no bus inactivity limit.

NOTE 5 - Because of the low overheads associated with initiating and closing bus tenancy on Fibre Channel links, device servers should end tenancies immediately upon completing the required transfers.

The BUS INACTIVITY LIMIT is not applicable for FCP devices attached to links that do not have the concept of interconnect tenancy.

10.2.5 DISCONNECT TIME LIMIT

The DISCONNECT TIME LIMIT field indicates the minimum delay between interconnect tenancies measured in increments of 128 transmission words. Targets in configurations having the concept of interconnect tenancy shall delay at least this time interval after each interconnect tenancy before beginning arbitration. The device server may round this value to any value it prefers. A value of zero indicates that the disconnect time limit does not apply.

The DISCONNECT TIME LIMIT is not applicable for FCP devices attached to links that do not have the concept of interconnect tenancy.

10.2.6 CONNECT TIME LIMIT

The CONNECT TIME LIMIT field indicates the maximum duration of a single interconnect tenancy, measured in increments of 128 transmission words. If the connect time limit is exceeded the device server shall conclude the interconnect tenancy, within the restrictions placed on it by the applicable Fibre Channel configuration. The device server may round this value to any value it prefers. A value of zero indicates that there is no connect time limit.

The CONNECT TIME LIMIT is not applicable for FCP devices attached to links that do not have the concept of interconnect tenancy.

10.2.7 MAXIMUM BURST SIZE FIELD

The MAXIMUM BURST SIZE field indicates the maximum size of FCP_DATA IU that the device server shall transfer to the initiator or request from the initiator. This parameter does not affect how much data is transferred in a single interconnect tenancy. This value is expressed in increments of 512 bytes (e.g., a value of 1 means 512 bytes, two means 1024 bytes, etc.). The device server may round this value down as defined in SPC-2. A value of zero indicates there is no limit on the amount of data transferred per data transfer operation. This value shall be implemented by all FCP devices. The application client and device server may use the value of this parameter to adjust internal maximum buffering requirements.

10.2.8 ENABLE MODIFY DATA POINTERS (EMDP)

The ENABLE MODIFY DATA POINTERS (EMDP) bit indicates whether or not the target may use the random buffer access capability to reorder FCP_DATA IUs for a single SCSI command. If the EMDP bit is set to zero, the target shall generate continuously increasing relative offset values for each FCP_DATA IU for a single SCSI command. If the EMDP bit is set to one, the target may transfer the FCP_DATA IUs for a single SCSI command in any order. An EMDP bit of zero prohibits data overlay, even if it is allowed by the state of the PRLI DATA OVERLAY ALLOWED bit. The EMDP bit does not affect the order of frames within a Sequence. The EMDP function is optional for all FCP devices.

10.2.9 FAA, FAB, FAC

The fairness access (FA) bits, FAA, FAB, and FAC, indicate whether a target in a loop configuration shall use the access fairness algorithm when beginning the interconnect tenancy. An FA bit set to one indicates that the target shall use the access fairness algorithm for the specified frames. An FA bit set to zero indicates that the target may choose to not use the access fairness algorithm. The FAA bit controls arbitration when the target wishes to send one or more FCP_DATA IU frames to an initiator. The FAB bit controls arbitration when the initiator wishes to send one or more FCP_XFER_RDY IU frames to a target. The FAC bit controls arbitration when the target wishes to send an FCP_RSP IU frame to an initiator or when the initiator wishes to send an FCP_CMND IU frames to target. If the target intends to send multiple frame types, it may choose to not use the access fairness algorithm if any applicable FA bit is set to zero. FCP devices attached to links that do not have the concept of interconnect tenancy shall ignore the FA bits. The FA bits are optional for all FCP devices.

10.2.10 FIRST BURST SIZE

When write transfer ready is disabled, the FIRST BURST SIZE field indicates the maximum amount of data that shall be transmitted in the first FCP_DATA IU sent from the initiator to the target. If all data is transmitted in the first IU, no subsequent FCP_XFER_RDY IUs shall be transmitted by the target. If the maximum amount of data has been transmitted, but more data remains to be transferred, the target shall request that data with subsequent FCP_XFER_RDY IUs.

When write transfer ready is enabled, the FIRST BURST SIZE field is ignored and permission to transmit data from the initiator to the target is managed using FCP_XFER_RDY IUs. For data transmissions from the target to the initiator, the FIRST BURST SIZE field is ignored.

The FIRST BURST SIZE field value is expressed in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1024 bytes). A value of zero indicates that there is no first burst size limit. The FIRST BURST SIZE field shall be implemented by all FCP devices that support the disabling of write transfer ready. The application client and device server may use the value of this parameter to adjust internal maximum buffering requirements.

10.3 Fibre Channel Logical Unit Control mode page

The Fibre Channel Logical Unit Control mode page (see table 28) contains those parameters that select FCP logical unit operation options. The page shall be implemented by all FCP devices. The implementation of any parameter and its associated functions is optional. The page follows the MODE SENSE and MODE SELECT command rules specified by SPC-2.

Table 28 - Fibre Channel Logical Unit Control page (18h)

Byte \ Bit	7	6	5	4	3	2	1	0
0	PS	RESERVED	PAGE CODE (18h)					
1	PAGE LENGTH (06h)							
2	RESERVED				PROTOCOL IDENTIFIER (FCP = 0h)			
3	RESERVED							EPDC
4	RESERVED							
5	RESERVED							
6	RESERVED							
7	RESERVED							

An ENABLE PRECISE DELIVERY CHECKING (EPDC) bit of one indicates that the logical unit shall use the precise delivery function defined by this standard. See 4.3. When the EPDC bit is set to zero, the target shall not use the precise delivery function and shall ignore the contents of the CRN field. The EPDC bit is valid for all types of link connections. If the precise delivery function is not supported and the Fibre Channel Logical Unit Control is supported by the target, the EPDC bit shall be masked as not changeable and shall follow the MODE SENSE and MODE SELECT command rules specified by SPC-2.

If the Fibre Channel Logical Unit Control page is not supported by a target, the initiator shall assume that the precise delivery function is not supported.

10.4 Fibre Channel Port Control mode page

10.4.1 Overview and format of Fibre Channel Port Control mode page

The Fibre Channel Port Control mode page contains those parameters that select FCP_Port operation options. The page shall be implemented by LUN 0 of all FCP devices. The page shall not be implemented by logical units other than LUN 0. The implementation of any bit and its associated functions is optional. The page follows the MODE SENSE and MODE SELECT command rules specified by SPC-2.

Some of the bits defined by the Fibre Channel Port Control page require the FCP_Port to violate one or more of the Fibre Channel standards. The non-standard behaviors have been identified as useful for certain specialized operating environments.

The format of the Fibre Channel Port Control page is shown in shown in table 29.

Table 29 - Fibre Channel Port Control page (19h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	RESERVED	PAGE CODE (19h)					
1	PAGE LENGTH (06h)							
2	RESERVED				PROTOCOL IDENTIFIER (FCP = 0h)			
3	DTFD	PLPB	DDIS	DLM	RHA	ALWI	DTIPE	DTOLI
4	RESERVED							
5	RESERVED							
6	RESERVED					RR_TOV UNITS		
7	RESOURCE RECOVERY TIME-OUT VALUE (RR_TOV)							

10.4.2 DISABLE TARGET ORIGINATED LOOP INITIALIZATION (DTOLI)

A DISABLE TARGET ORIGINATED LOOP INITIALIZATION (DTOLI) bit of one indicates that a target attached by an arbitrated loop shall not generate a LIP following insertion into the loop. The target shall respond to a LIP when it is received. When the DTOLI bit is set to zero, the target attached by an arbitrated loop shall generate LIP(F7,xx) after it enables a port into a loop. If the target is attached to an arbitrated loop and detects loop failure at its input, it shall follow the error initialization process defined by FC-AL-2 regardless of the state of the DTOLI bit. Targets not attached to an arbitrated loop shall ignore the DTOLI bit.

10.4.3 DISABLE TARGET INITIATED PORT ENABLE (DTIPE)

A DISABLE TARGET INITIATED PORT ENABLE (DTIPE) bit of one indicates that a target attached to an arbitrated loop shall wait for an initiator to send the Loop Port Enable (LPE) primitive sequence before inserting itself into a loop (see FC-AL-2). The target shall wait in a participating state with the Port Bypass circuit, if any, set to bypass the target. The target uses the hard address available in the SCA-2 SFF-8067 connector or in device address jumpers to determine whether LPE primitive sequences are addressed to it. An LPE primitive sequence addressed to the broadcast address shall also cause the target to insert itself into the loop. When the DTIPE bit is set to zero, the target shall enable itself onto the loop in according to the rules specified in FC-AL-2. Targets not attached to an arbitrated loop shall ignore the DTIPE bit.

10.4.4 ALLOW LOGIN WITHOUT LOOP INITIALIZATION (ALWLI)

An ALLOW LOGIN WITHOUT LOOP INITIALIZATION (ALWLI) bit of one indicates that a target attached to an FC-AL-2 loop shall use the hard address available in the Single Connector Attach - 2 (SCA-2) SFF-8067 connector or in device address jumpers, enter the monitoring state in participating mode, and accept logins without using the loop initialization procedure (see FC-AL-2). When the ALWLI bit is set to zero, the target shall perform the normal loop initialization procedure before entering the monitoring mode and accepting a login ELS. Targets not attached to an arbitrated loop shall ignore the ALWLI bit.

10.4.5 REQUIRE HARD ADDRESS (RHA)

A REQUIRE HARD ADDRESS (RHA) bit of one indicates that a target attached to an arbitrated loop shall only attempt to obtain its hard address available in the SCA-2 SFF-8067 connector or device address jumpers during loop

initialization. The target shall not attempt to obtain an address during the LISA phase of initialization (see FC-AL-2). If there is a conflict for the hard address selection during loop initialization or the target does not have a valid hard address available, the target shall enter the nonparticipating state. If the target detects loop initialization while in the nonparticipating state, the target shall again attempt to get its hard address. If the hard address has not changed from the address obtained in a previous successful loop initialization, the target shall attempt to obtain the address in the LIFA phase if a valid Fabric Login exists or LIPA phase of loop initialization. If the hard address has changed, the target shall attempt to obtain the new address in the LIHA phase.

When the RHA bit is set to zero, the target follows the normal initialization procedure, including the possibility of obtaining a soft address during the loop initialization process.

Targets not attached to an arbitrated loop shall ignore the RHA bit.

10.4.6 DISABLE LOOP MASTER (DLM)

A DISABLE LOOP MASTER (DLM) bit of one indicates that a target attached to an FC-AL-2 loop shall not participate in loop master arbitration and shall not become loop master. The target shall only repeat LISM frames it receives. When the DLM bit is set to zero, the target may participate in loop master arbitration in the normal manner and, if successful, may become loop master during the loop initialization process. Targets not attached to an arbitrated loop shall ignore the DLM bit.

10.4.7 DISABLE DISCOVERY (DDIS)

A DISABLE DISCOVERY (DDIS) bit of one indicates that a target without a valid FLOGI attached to an arbitrated loop shall not require receipt of Address or Port Discovery (ADISC or PDISC ELSs) following loop initialization as described in FC-PLDA and FC-FLA. The logical units shall resume processing tasks on completion of loop initialization. When the DDIS bit is set to zero, the target shall wait to complete target discovery as defined by FC-PLDA, FC-FLA, and FC-TAPE before allowing processing of tasks to resume.

Targets not attached to an arbitrated loop shall ignore the DDIS bit. A target with a valid FLOGI shall ignore the DDIS bit.

10.4.8 PREVENT LOOP PORT BYPASS (PLPB)

A PREVENT LOOP PORT BYPASS (PLPB) bit of one indicates that a target attached to an FC-AL-2 loop shall ignore any Loop Port Bypass (LPB) and Loop Port Enable (LPE) primitive sequences. The loop port shall always remain participating. When PLPB bit is set to zero, the target allows the Loop Port Bypass (LPB) and Loop Port Enable (PBE) primitive sequences to control the port bypass circuit and participation on the loop as specified by FC-AL-2. Targets not attached to an arbitrated loop shall ignore the PLPB bit.

The DTIPE and PLPB bits shall not both be set to one at the same time. When an invalid bit combination is sent by the application client the device server shall return CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN THE PARAMETER LIST.

10.4.9 DISABLE TARGET FABRIC DISCOVERY (DTFD)

A DISABLE TARGET FABRIC DISCOVERY (DTFD) bit of one indicates that a target attached by an arbitrated loop shall not recognize the presence of a fabric loop port on the loop. The target shall perform only the private loop functions defined for targets defined by FC-PLDA and FC-TAPE. When DTFD bit is set to zero, the target attached by an arbitrated loop shall discover a fabric loop port if present on the loop and perform the public loop functions defined for targets by FC-FLA. Targets not attached to an arbitrated loop shall ignore the DTFD bit.

10.4.10 RESOURCE RECOVERY TIME-OUT VALUE (RR_TOV)

The RR_TOV is defined by bytes 6 and 7 in the following manner. See 11.4.

The RR_TOV UNITS field indicates the units for the RR_TOV field value, according to table 30.

Table 30 - Values for RR_TOV UNITS

Byte 6			Units of measure for RR_TOV
bit 2	bit 1	bit 0	
0	0	0	No timer is specified
0	0	1	0.001 seconds
0	1	1	0.1 seconds
1	0	1	10 seconds
All other values			Reserved

The RR_TOV field indicates the number of time units specified by the RR_TOV UNITS field that shall be used by the timer that performs the RR_TOV time-out functions. If no timer is specified, the RR_TOV value shall be ignored by the device server and a vendor specific default value shall be used for RR_TOV. See 11.4 and FC-PLDA for the RR_TOV time-out functions.

11 Timers for FCP operation and recovery

11.1 Summary of timers for the Fibre Channel protocol

This clause indicates the use of timers defined by other standards in performing the FCP-2 recovery procedures. In addition, the clause defines those timers used only by this standard.

Table 31 - Timer summary

Timer	Implementation Mandatory (M) or Optional (O)		Description	Default Value	Notes	Ref
	Initiator	Target				
E_D_TOV	M	O	Error_Detect_Time-out Value	2 sec.	2,3	11.2
$R\_A\_TOV_{SEQ\_QUAL}$	M	O	Resource_Allocation Time-out Value	Private loop = 0 sec. Public loop = 10 sec.	1,2	11.3
$R\_A\_TOV_{ELS}$	M	M		Private loop = 2 sec. Public loop = 10 sec.	1,2	11.3
RR_TOV		M	Resource Recovery Time-out Value	If RETRY bit is set to 0: 2 sec. If RETRY bit is set to 1: ≥ REC_TOV + $2xR\_A\_TOV_{ELS}$ + 1 sec.		11.4
REC_TOV	M	M	REC Time-out Value	≥ E_D_TOV + 1 sec. (minimum)	4	11.5
ULP_TOV	M		Upper Level Protocol Time-out Value	If RETRY bit is set to 0: ≥ Operation-specific timer + E_D_TOV + 1 sec. If RETRY bit is set to 1: ≥ Operation-specific timer + 2xRR_TOV		11.6

NOTES:

1 R_A_TOV is defined by FC-FS.
This standard defines the default R_A_TOV for Sequence Qualifiers as zero for private loops and 10 seconds for public loops. This standard defines the default R_A_TOV for ELS responses as 2 seconds for private loops and 10 seconds for public loops. If ELSs are used to set R_A_TOV, the same value is applied for both uses. Other Fibre Channel standards may specify different default values for R_A_TOV for different topologies.

2 Targets that support Class 2 delivery service shall implement this timer.

3 E_D_TOV default time-out values are defined by FC-FS, FC-PLDA, and FC-FLA. ELSs are provided to set values other than the default value. This standard defines the default value required by the recovery protocol, deriving the value as follows:

a Public loop devices compliant with FC-FLA use an E_D_TOV value of 2 seconds before fabric login and the value obtained in the FLOGI ACC after fabric login.

b Private loop devices compliant with FC-PLDA use the default E_D_TOV value of 2 seconds.

c Devices attached through a fabric or point-to-point connection use the default E_D_TOV value specified by FC-FS before fabric login and the value obtained in the FLOGI ACC after fabric login.

4 REC_TOV is required by the target for FCP_CONF IU error detection.

11.2 Error_Detect Time-out (E_D_TOV)

E_D_TOV is a general error detect time-out value. Its use is specified in FC-FS, FC-AL, FC-PLDA, FC-FLA, and FC-TAPE. For FCP-2 Sequence recovery, it is used to time the following:

a) the maximum time permitted for a Sequence Initiator between the transmission of consecutive data frames within a single Sequence;

b) the minimum time that a Sequence Recipient shall wait for the reception of the next frame within a single Sequence before recognizing a Sequence time-out; and

c) the minimum time a Sequence Initiator shall wait for an ACK response before it considers the ACK to be missing and begins recovery actions.

Target devices that support Class 2 shall implement this timer for the purpose of timing out missing ACKs.

Loop attached Class 2 devices may require a complete fairness cycle plus the fabric and link delay times before an ACK is received.

11.3 Resource Allocation Time-out (R_A_TOV)

R_A_TOV has two separate components, 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_Qualifier associated with an aborted Sequence. The Sequence_Qualifer is composed of the S_ID, D_ID, OX_ID, RX_ID, and SEQ_ID. This value is also the minimum amount of time that a Sequence Initiator shall wait following receipt of the BA_ACC to ABTS before transmitting a Reinstate Recovery Qualifier (RRQ) ELS.

Using a value of zero 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 queuing mechanism for the transmission of Sequences, the queue for a given Sequence shall be empty before an ABTS for that Sequence is sent, or the act of sending the ABTS purges the queue.

A value of two times $R\_A\_TOV_{ELS}$ is used to determine the minimum time that the Originator of an Extended Link Service or FC-4 Link Service request shall wait for the response to that request.

After completion of FLOGI, Public Loop devices shall use the value of R_A_TOV specified by the Fabric in the FLOGI ACC.

11.4 Resource Recovery Timer (RR_TOV)

RR_TOV is the minimum time a target shall wait for a specific initiator to perform Exchange Authentication following the completion of the Loop Initialization Protocol (i.e., the receipt of CLS while in the OPEN-INIT state). See FC-PLDA. RR_TOV is also the minimum time a target shall wait for an initiator response following transfer of Sequence Initiative from the target to the initiator (e.g., following transmission of the FCP_XFER_RDY IU during a write command). If either of these two conditions is not recovered successfully before expiration of RR_TOV, a target may implicitly or explicitly perform a LOGO with that initiator, terminate all open Exchanges for that initiator, and reclaim the resources associated with those Exchanges. See 12.4.1.5.

The value of RR_TOV may be set using the Fibre Channel Port Control mode page. See 10.4.10.

11.5 Read Exchange Concise Time-out Value (REC_TOV)

REC_TOV is used by the initiator to provide a minimum polling interval for REC and by the target for FCP_CONF IU error detection. The REC_TOV timer shall be implemented such that at least one REC_TOV period passes

between transmission of a command and the first polling for Exchange status with the REC Extended Link Service. Table 32 describes REC_TOV usage pertaining to the initiator.

Table 32 - Initiator REC_TOV Usage

Timer starts or restarts after:	Timer stops without sending REC after:
FCP_CMND IU has been sent.	FCP_RSP IU is received for the FCP_CMND IU or the Exchange is aborted.
FCP_DATA IU Sequence has been sent by the initiator (optional)	FCP_RSP IU is received for the FCP_CMND IU or the Exchange is aborted.
REC was sent for an FCP_CMND IU and an ACC was received indicating the command is in progress (i.e. REC polling interval).	FCP_RSP IU is received for the FCP_CMND IU or the Exchange is aborted.

Table 33 describes REC_TOV usage pertaining to the target.

Table 33 - Target REC_TOV usage

Timer starts after:	Timer stops without sending REC after:
FCP_RSP IU requesting an FCP_CONF IU has been sent.	FCP_CONF IU is received or the Exchange is aborted.

11.6 Upper Level Protocol Time-out (ULP_TOV)

ULP_TOV is an operation-specific timer maintained by the Upper Level Protocol. ULP_TOV 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 the operation being timed. Some operations may require extended periods of time to complete.

ULP timers take into account response time increments caused by command queuing and multi-initiator congestion.

12 Link error recovery procedure

12.1 Overview

12.1.1 Exchange level error recovery

This standard provides several mechanisms for FCP devices to identify protocol errors caused by frames and responses that have been corrupted and discarded in accordance with the requirements of FC-FS. See 12.2 for a list of these mechanisms.

To recover from these errors, all FCP compliant initiators shall be capable of invoking the recovery abort function to terminate a failing exchange and to recover the associated resources as described in 12.3. All FCP compliant targets shall be capable of executing the requested recovery abort to finish clearing the exchange and to recover the associated resources. The failed command may then be reissued by higher level programs according to protocols beyond the scope of this standard.

This standard has expanded the error detection capabilities defined by the FCP standard by allowing the optional use of the REC Extended Link Service to monitor the progress of active Exchanges. An FCP-2 device may optionally accept or reject error detection inquiries.

12.1.2 Sequence level error recovery

To recover from errors, FCP-2 compliant devices may optionally perform retransmission procedures that allow the commands to be completed without requiring higher level programs to perform command retries. Such recovery may be useful for SCSI logical units that depend critically on command ordering and maintaining records of internal device state. The initiator and the target shall agree to perform retransmission using the SRR FCP FC-4 Link Service by setting the RETRY bit to one in PRLI before performing the retransmission of individual IUs. See 6.3.4. An FCP-2 device that has agreed to perform retransmission shall use and accept the REC Extended Link Service and SRR FCP FC-4 Link Service as defined by this standard to perform the retransmission.

Even after agreeing to perform retransmission, the initiator may choose to request a recovery abort and the target shall be able to accept and perform the recovery abort.

While the basic error detection and error recovery procedures are class independent, acknowledged classes of services may use the acknowledgement mechanism as an additional error detection feature and may use optional mechanisms defined in FC-FS to assist in the recovery process.

This clause defines the error detection and recovery mechanisms for fabrics that guarantee in-order frame delivery. However, if continuously increasing sequence count is used and if support for recovery qualifiers is fully implemented as defined in FC-FS, the same recovery mechanisms may be used for fabrics that do not guarantee in-order frame delivery, as shown in the examples in Annex C.

Examples of error detection and error recovery are provided in Annex C.

12.2 FCP Error Detection

12.2.1 Overview of FCP-2 Error Detection

The subclauses of 12.2 describe the initial events that indicate an error may have occurred. The error may be recovered at the Exchange level or at the Sequence level.

12.2.2 FCP-2 Error Detection using protocol errors for all classes of service.

The Exchange originator (initiator) shall detect any of the following errors:

a) a Sequence error is detected in a Sequence transmitted from a target to an initiator;

b) a read command completed with the data count smaller than FCP_DL and FCP_RESID_UNDER is set to zero;

c) a read-type command completed with the data count smaller than FCP_DL, FCP_RESID_UNDER is set to one, and the data count plus FCP_RESID is not equal to FCP_DL; or

d) an ABTS is received.

It may optionally further identify and recover additional errors as described in 12.4.

The Exchange Responder shall also initiate error detection and recovery after a Sequence error is detected in a Sequence transmitted from an initiator to a target. (See 12.3.5).

The Exchange Responder (target) may detect that REC_TOV times out and an expected FCP_CONF IU has not been received. The Exchange Responder may then use the methods described in 12.4 to determine the presence of an error, regardless of whether exchange level or sequence level error recovery is to be used.

12.2.3 Error Detection mechanisms for acknowledged classes of Service

Acknowledged classes of service provide the additional error detection mechanisms described below.

The Exchange originator (initiator) shall detect any of the following errors:

a) after E_D_TOV times out, no ACK has been received for the FCP_CMND IU;

b) after E_D_TOV times out, no ACK has been received for FCP_DATA IU(s) (see example in figure C.22);

c) after E_D_TOV times out and no ACK has been received for the FCP_CONF IU; or

d) an ACK with the F_CTL Abort Sequence Condition bits set to Abort Sequence, Perform ABTS is received. (See FC-FS.)

It may optionally further identify and recover additional errors as described in 12.4.

The Exchange Responder (target) shall detect any of the following errors:

a) after E_D_TOV times out and no ACK has been received for the FCP_XFER_RDY IU (see example in figure C.6);

b) after E_D_TOV times out and no ACK has been received for FCP_DATA IU(s) (see example in figure C.21); or

c) after E_D_TOV times out and no ACK has been received for the FCP_RSP IU.

It may optionally further identify and recover additional errors as described in 12.4.

If an ABTS is transmitted by a Sequence Initiator because it had detected a missing ACK and the BA_ACC response to the ABTS indicates the Sequence was correctly received by the Sequence Recipient, no error detection or recovery is required.

12.3 Exchange level recovery using recovery abort

12.3.1 Recovery abort requirements

The recovery abort is an FC-FS protocol that recovers FCP_Port resources associated with an exchange that is being terminated, either because of a task management request or because of an error.

Recovery abort may be used whether or not the FCP devices have agreed to Sequence level error recovery.

All FCP initiators shall be capable of invoking the recovery abort protocol to terminate failing commands for later retry. (See 9.1.2.4). All FCP targets shall be capable of accepting and completing the recovery abort protocol.

12.3.2 Initiator invocation of recovery abort

The initiator terminating the exchange sends an ABTS sequence to the D_ID of the target of the exchange being terminated. The ABTS sequence shall have the PARAMETER field set to ABORT EXCHANGE. The ABTS sequence shall be generated using the OX_ID and RX_ID of the exchange to be aborted. FC-FS allows ABTS to be generated by an FCP_Port regardless of whether or not it has sequence initiative. 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 set to one is received from the target.

Recovery abort may not take effect immediately. For example, if ABTS is sent following transmission of a READ command, the initiator may receive some or all of the requested read data before receiving the BA_ACC to the ABTS. The initiator shall be capable of receiving this data and providing BB_Credit in order for the target to send the BA_ACC.

After the execution of a task management function that clears tasks, recovery abort shall be invoked for all Exchanges not successfully terminated with an FCP_RSP IU status set to COMMAND CLEARED. (See 9.1.2.4).

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

If a BA_ACC, BA_RJT, LOGO, or PRLO is not received from the target within 2 times $R\_A\_TOV_{ELS}$, second level error recovery as described in 12.5 shall be performed.

12.3.3 Target response to recovery abort

When an ABTS (Abort Exchange) is received at the target, it shall abort the designated Exchange and return one of the following responses:

a) the target shall discard the ABTS and return LOGO if the N_Port or NL_Port issuing the ABTS is not currently logged in (i.e., no PLOGI);

b) the target shall return BA_RJT with Last Sequence of Exchange bit set to one if the received ABTS contains an assigned RX_ID and a FQXID that is unknown to the target; or

c) the target shall return BA_ACC with Last Sequence of Exchange bit set to one.

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

If the RX_ID is FFFFh, targets shall qualify the FQXID of the ABTS based only upon the combined values of D_ID, S_ID, and OX_ID, not RX_ID.

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.

When an RRQ is received at the target, it shall return one of the following responses:

a) the target shall discard the RRQ and return LOGO if the N_Port or NL_Port issuing the RRQ is not currently logged in (i.e., no PLOGI);

b) the target shall return LS_RJT with Last Sequence of Exchange bit set to one if the received RRQ contains an RX_ID, other than FFFFh, that is unknown to the target. The reason code shall be “Logical error” with a reason code explanation set to “Invalid OX_ID-RX_ID combination”; or

c) the target shall return ACC with Last Sequence of Exchange bit set to one.

12.3.4 Additional error recovery by initiator

This procedure may be used whether or not the FCP devices have agreed to Sequence level recovery.

If ULP_TOV times out and the Exchange is not complete, the application client shall clear the exchange resources using the ABORT TASK task management request or the initiator shall clear the exchange resources using the recovery abort protocol. (See 9.1.3.)

12.3.5 Additional error recovery by target

This procedure may be used whether or not the FCP devices have agreed to Sequence level recovery.

If a target detects a Sequence error, it shall discard the Sequence(s) based on the Exchange error policy specified by the F_CTL Abort Sequence Condition bits in the first frame of the Exchange. (See FC-FS.)

For acknowledged classes of service, if a target detects a Sequence error, it may abort the sequence by sending an ABTS with the PARAMETER field to ABORT SEQUENCE. See FC-FS. If a Recovery Qualifier range is returned in the BA_ACC for the ABTS the target shall send a RRQ ELS after $R\_A\_TOV_{SEQ\_QUAL}$ times out after receipt of the BA_ACC.

For unacknowledged classes of service, the target shall not attempt recovery for Sequence errors. The target shall depend on initiator time-outs for recovery.

Targets shall implement RR_TOV as described in 11.4 to facilitate recovery of resources allocated to an initiator that is no longer responding. The target may send a LOGO to the initiator and terminate all open Exchanges for that initiator upon detection of the following:

a) The initiator has failed to perform target Exchange authentication within RR_TOV (see FC-PLDA); or

b) RR_TOV times out without the initiator transmitting any expected Sequence for any open Exchange at this target (e.g., FCP write data in response to an FCP_XFER_RDY IU).

12.4 Sequence level error detection and recovery

12.4.1 Using information from REC to perform Sequence level recovery

12.4.1.1 Polling Exchange state with REC

REC is periodically transmitted by the initiator to poll each outstanding Exchange to determine if a SCSI task is progressing properly and if any Sequences have been received incorrectly. Timing of polling with the REC Extended Link Service is controlled by REC_TOV. REC_TOV is normally selected to be long enough that processing the transfers of initiative in the exchange and completing the exchange occur before REC_TOV times out. If REC_TOV times out, then an REC Extended Link Service is performed. The information returned in the REC ACC payload is compared with the expected state information known by the initiator and target. If the information is inconsistent, indicating that a link error occurred, optional error recovery actions may be performed to complete the Exchange. Subclauses 12.4.1.2 through 12.4.1.8 define optional error detection and recovery procedures for acknowledged and unacknowledged classes of service.

12.4.1.2 Detection of errors while polling with REC

If an Exchange Originator receiving an acknowledged service Sequence detects a Sequence error, it shall send an ACK Frame with the F_CTL Abort Sequence Condition bits set to “Abort Sequence, Perform ABTS” before issuing the REC. See FC-FS. The REC for the Exchange containing the FCP_CMND IU shall be issued in a new Exchange.

If the response to the new Exchange issuing the REC is an LS_RJT with a reason code of “command not supported”, the initiator shall assume the target is an FCP device not supporting error detection using REC. The device shall perform recovery using recovery abort as documented in 12.3.

If an ACC, LS_RJT, LOGO, or PRLO is not received from the target within 2 times $R\_A\_TOV_{ELS}$, second level error recovery as described in 12.5 shall be performed.

12.4.1.3 FCP_CMND IU Recovery using information from REC

This procedure may be used whether or not the FCP devices have agreed to Sequence level recovery.

If the FCP_CMND IU was not received by the target (i.e., the initiator receives an LS_RJT for the REC with the reason code of “Logical error” and reason code explanation set to “Invalid OX_ID-RX_ID combination”), retransmit the FCP_CMND IU using a new Exchange. If the precise delivery function is enabled, the CRN value shall remain the same in the retransmitted FCP_CMND IU.

If the ACC for the REC indicates that the FCP_CMND IU was received by the target and that no reply Sequence has been sent (i.e., by indicating that the initiator does not hold Sequence Initiative, and that the Exchange is not complete), the command is in process and no recovery is needed at this time. At a minimum interval of REC_TOV, the REC shall be retransmitted to more quickly determine if a reply Sequence has been lost.

For examples of such recoveries, see figure C.1 and figure C.2

12.4.1.4 FCP_XFER_RDY IU Recovery

This procedure shall be used only by FCP devices that have agreed to Sequence level recovery.

If the ACC for an REC indicates that an FCP_XFER_RDY IU was sent by the target (i.e., by indicating that the initiator holds Sequence Initiative, that all bytes were not transferred, and that the Exchange is not complete), but not received by the initiator, issue an SRR in a new Exchange to request retransmission of the FCP_XFER_RDY IU. The target shall first transmit the ACC for the SRR and then shall retransmit the FCP_XFER_RDY IU in a new Sequence containing the same Relative Offset as the originally transmitted FCP_XFER_RDY IU. After the FCP_XFER_RDY IU is successfully received, the FCP I/O operation continues normally.

For examples of this type of recovery, see figure C.5 and figure C.6.

12.4.1.5 FCP_RSP IU Recovery

This procedure shall be used only by FCP devices that have agreed to Sequence level recovery.

An error in transmitting an FCP_RSP IU is detected if:

a) the ACC for the REC Extended Link Service indicates that an FCP_RSP IU was sent by the target and no FCP_CONF IU was requested (i.e., E_STAT indicates that the Exchange is complete), but the initiator has not yet received the FCP_RSP IU; or

b) the ACC for the REC Extended Link Service indicates that an FCP_RSP IU Sequence was sent by the target and an FCP_CONF IU was requested (i.e., E_STAT indicates that the Exchange is not complete, that the initiator has initiative, and that, if the data transfer was from the initiator to the target, the data transfer indicates that all of the bytes expected to be transferred by the command have been transferred.)

When an error in transmitting an FCP_RSP IU is detected, the initiator shall issue an SRR FC-4 Link Service frame in a new Exchange to request retransmission of the FCP_RSP IU. The target shall first transmit the ACC for the SRR, then shall retransmit the FCP_RSP IU in a new Sequence.

An Exchange carrying a command that was terminated by a CHECK CONDITION requesting an FCP_CONF IU prior to transferring data may have the same REC values as an Exchange carrying a command having an FCP_XFER_RDY IU not received by the initiator. For a command transferring data from the initiator to the target with a non-zero FCP_DL, the parameters for the SRR shall indicate that an FCP_XFER_RDY IU is expected from the target. The target is aware of the actual present state of the transfer and response and shall either retry the FCP_XFER_RDY IU or, if the actual data transfer length for the command was zero, retry the FCP_RSP IU.

For non-tagged command queuing operations, the target shall retain the Exchange information until:

a) the next FCP_CMND IU has been received for that LUN from the same initiator;

b) an FCP_CONF IU is received for the Exchange; or

c) after RR_TOV times out.

For tagged command queuing operations, the target shall retain Exchange information until:

a) an FCP_CONF IU is received for the Exchange; or

b) after RR_TOV times out.

The Exchange information retained shall include data transfer information, data descriptors, and FCP_RSP IU information.

If retransmission is enabled between the initiator and target, FCP_RSP IU information shall be:

a) discarded RR_TOV after the FCP_RSP IU was transmitted to the initiator; or,

b) discarded after a new Exchange with the same OX_ID and S_ID is received.

If retransmission is not enabled between the initiator and target, FCP_RSP information may be discarded immediately after the FCP_RSP IU has been transmitted to the initiator.

The value of RR_TOV is set using the Fibre Channel Port Control page. See 10.4.10.

If task retry identification has been agreed to by both the initiator and target, the same value of task retry identifier shall not be used within RR_TOV.

Examples of FCP_RSP IU recoveries are provided in figure C.8 through figure C.12.

12.4.1.6 FCP_DATA IU Recovery - Write

This procedure shall be used only by FCP devices that have agreed to Sequence level recovery.

If the ACC for an REC indicates that an FCP_DATA IU was sent by the initiator, but not received by the target (i.e., the data received count in the REC response is smaller than what the initiator sent, and the target indicates it does not hold Sequence Initiative) then the initiator shall send a SRR FC-4 Link Service frame in a new Exchange to request retransmission of an FCP_XFER_RDY IU to request the missing data. The target discards the Sequence in error, but does not initiate any recovery action for Class 3. (See 12.3.5.) After first transmitting the ACC for the SRR, the target transmits an FCP_XFER_RDY IU in a new Sequence with the Relative Offset parameter specified by the SRR. The initiator responds with the requested data.

FCP_DATA IU shall be retransmitted in a new Sequence. For acknowledged classes, the SEQ_CNT shall be one greater than that used to transmit the last Sequence, usually the ABTS. For unacknowledged classes, the SEQ_CNT may start at zero, even if continuously increasing sequence count is being used.

Examples of data recovery during write operations are provided in figure C.13 through figure C.16.

12.4.1.7 FCP_DATA IU Recovery - Read

This procedure shall be used only by FCP devices that have agreed to Sequence level recovery.

If the ACC for the REC indicates that data was sent by the target but not successfully received by the initiator (i.e., by indicating a data sent count greater than the initiator has successfully received), then the initiator shall send a SRR FC-4 Link Service frame in a new Exchange to request retransmission of the FCP_DATA IU that was not successfully received. The initiator shall set the RELATIVE OFFSET field in the SRR to that of the next data requested. If the initiator is unable to determine the Relative Offset of the next data requested, the initiator shall set the RELATIVE OFFSET field to zero. The target shall first transmit the ACC for the SRR, then shall retransmit the requested data specified by the SRR in a new Sequence, and then complete the Exchange in the normal manner, including transmitting or retransmitting the FCP_RSP IU. If the target responds to the SRR with an FCP_RJT and an FCP_RSP IU has not yet been sent or is again requested, the target shall send an FCP_RSP IU with CHECK CONDITION status and sense data containing a sense key of HARDWARE ERROR and an ASC/ASCQ of INITIATOR DETECTED ERROR MESSAGE RECEIVED.

The FCP_DATA IU shall be retransmitted in a new Sequence. For acknowledged classes, the SEQ_CNT shall be one greater than that used to transmit the last Sequence, usually the ABTS. For unacknowledged classes, the SEQ_CNT may start at zero, even if continuously increasing sequence count is being used.

It is the responsibility of the initiator to determine the action (e.g., retry, allow ULP time-out, or return status to ULP) based on the information determined by REC and other internal states. The target does not initiate error recovery for Class 3. (See 12.3.5.)

Examples of data recovery during read operations are provided in figure C.17 through figure C.20.

12.4.1.8 FCP_CONF IU Recovery

This procedure may be used whether or not an FCP device has agreed to Sequence level recovery.

This recovery procedure is used by target devices using all service classes.

Target devices that implement confirmed completion shall set the RX_ID to a unique value other than FFFFh for each Exchange to enable unambiguous recovery.

If the target has requested that the initiator transmit an FCP_CONF IU by setting the FCP_CONF_REQ in the FCP_RSP IU, then the target may periodically poll the initiator by transmitting REC to the initiator to determine if the FCP_CONF has been transmitted. Timing of polling with the REC Extended Link Service is controlled by REC_TOV.

If the initiator has sent the FCP_CONF IU, the reply to the REC from the target shall be a LS_RJT with the reason code of “Logical error” and reason code explanation set to “Invalid OX_ID-RX_ID combination”. The target shall assume that the FCP_CONF IU was sent and release the Exchange.

If the initiator has received the FCP_RSP IU with the FCP_CONF_REQ bit set to one and has not sent the FCP_CONF IU before the REC is received, the REC reply shall be an ACC indicating the Exchange is still open. In this case the target shall wait REC_TOV and, if the FCP_CONF IU has not been received, send another REC. The target shall repeat this process until the FCP_CONF IU is received, a new FCP_CMND IU is received with the same OX_ID as the Exchange waiting for the FCP_CONF IU, or until the Exchange is aborted.

If another FCP_CMND IU is received by the target with the same OX_ID as an Exchange waiting for an FCP_CONF IU and with the RX_ID unassigned, the target shall assume that the FCP_CONF IU was sent and release the Exchange.

Examples of recovery of FCP_CONF IUs are provided in figure C.23 through figure C.25.

12.4.2 Additional error recovery requirements

12.4.2.1 Error indicated in ACK

If an ACK is received with the F_CTL Abort Sequence Condition bits set to Abort Sequence, Perform ABTS, the Sequence Initiator shall send an ABTS for the Sequence. After R_A_TOV times out, an RRQ shall be sent by the Sequence Initiator.

12.4.2.2 Missing ACK

FC-FS requires that an ABTS(Sequence) be transmitted by a Sequence Initiator detecting a missing ACK. If no ACK has been received within E_D_TOV, the target shall abort the sequence by sending an ABTS request with the PARAMETER field set to ABORT SEQUENCE. If a Recovery Qualifier range is returned in the BA_ACC for the ABTS the target shall send an RRQ at least $R\_A\_TOV_{SEQ\_QUAL}$ after receipt of the BA_ACC. Adjustment of subsequent sequence counts may be required as specified by FC-FS.

12.4.2.3 Distinguishing exchange to be aborted

When OX_ID values are reused within R_A_TOV and RX_ID values are not used, and if there is a missing ACK to an FCP_RSP IU, a target may attempt to abort a more recent Exchange that used the same OX_ID. To prevent that, a target using acknowledged service behavior and performing error recovery shall:

a) set RX_ID to a value other than FFFFh to distinguish outstanding Exchanges as described in FC-FS; or

b) always request FCP_CONF IU.

If a Sequence error is detected for an FCP_DATA IU performing a Data Out action, the target shall send an ACK Frame with the Abort Sequence Condition bits set to “Abort Perform ABTS”.

Examples of data recovery for acknowledged services are shown in Annex C.

Recovery abort shall be invoked for Exchanges that were not successfully recovered by the specified error recovery procedures.

12.5 Second-level error recovery

12.5.1 ABTS

If a response to an ABTS is not received within 2 times $R\_A\_TOV_{ELS}$, the initiator may send the ABTS again, attempt other retry operations allowed by FC-FS, or explicitly logout the target. If those retry operations attempted are unsuccessful, the initiator shall explicitly logout (i.e., use FC-FS Logout, LOGO) the target. All outstanding Exchanges with that target are terminated at the initiator.

12.5.2 REC

If a response to an REC is not received within 2 times $R\_A\_TOV_{ELS}$, the initiator shall:

1) send an ABTS(Exchange) for the REC followed by an RRQ if a BA_ACC is received for the ABTS; and

2) send another REC in a new Exchange.

If the response to the second REC is not received within 2 times $R\_A\_TOV_{ELS}$, the initiator should send an ABTS(Exchange) for the REC followed by an RRQ if a BA_ACC is received for the ABTS;

Other retry mechanisms after the second REC fails are optional and, if implemented, shall comply with FC-FS.

ABTS(Exchange) may be required to clear resources associated with the original failing Exchange if the retry mechanisms are not successful.

See figure C.26 through figure C.29.

12.5.3 SRR

If a response to an SRR is not received within 2 times $R\_A\_TOV_{ELS}$, the initiator shall send an ABTS(Exchange) for the SRR followed by an RRQ if a BA_ACC is received for the ABTS. The initiator shall then perform an ABTS(Exchange) for the original Exchange.

See figure C.30 through figure C.33.

12.6 Responses to FCP type frames before PLOGI or PRLI

If a target receives an FCP_CMND IU from an FCP_Port that is not successfully logged on to the target using either an implicit or explicit login (i.e. PLOGI), it shall discard the FCP_CMND IU and, in a new Exchange, send LOGO to that FCP_Port. No Exchange is created in the 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 target sending the LOGO.

If a target receives an FCP_CMND IU from an FCP_Port that has not successfully completed either implicit or explicit Process Login (i.e. PRLI) with the target, it shall discard the FCP_CMND IU and send PRLO to the initiator. No Exchange is created in the recipient FCP_Port for the discarded request, and the Originator of the discarded request terminates the Exchange associated with the discarded request.

If an FCP device receives a frame of category 0001b or 0011b (solicited data or solicited control) and the FCP device has not performed a successful implicit or explicit PLOGI and PRLI with the source of the frame, the FCP device shall discard and ignore the content of the frame. If the PLOGI is not completed, the FCP device may transmit a LOGO extended link service request to the source of the unexpected frame. If the PLOGI is completed, but the PRLI is not completed, the FCP device may transmit a PRLO extended link service request to the source of the unexpected frame.

Annex A

(normative)

FCP mapping to SAM-2 (Fibre Channel Protocol Service mapping to SCSI Architectural Model - 2 (SAM-2))

A.1 Definition of procedure terms

FCP services are provided to the application client by the initiator to request and manage tasks as described by the SAM-2 standard. SAM-2 further defines how the target enables the device server to receive and process the tasks addressed to a logical unit. The Fibre Channel protocol is described in terms of the services provided by the initiator and target.

See table A.1 for the mapping of objects and identifiers used in this standard to the equivalent remote procedure call terms and definitions used in the SCSI Architecture Model-2 standard.

Table A.1 - FCP procedure terms mapped to terms from SAM-2 standard

FCP standard procedure terms	Equivalent SCSI Architecture Model-2 standard terms
address identifier of initiator port	initiator identifier
address identifier of target port	target identifier
fully qualified exchange identifier	I_T_L_Q Nexus
fully qualified exchange identifier + logical unit number	I_T_L_Q Nexus

See table A.2 for the definitions of the terms used by this standard and the equivalent SCSI Architecture Model-2 standard names of the terms, the name of the standard where the procedure terms are defined, the standard where the binary contents of the terms are defined, and the routing of the terms. The routing shows:

a) the originating object of the term,
b) the object that is the final destination of the term, and
c) the objects that the term moves through to reach the final destination object.

Table A.2 - Procedure Terms

FCP terms	Standard where term defined	Standard where binary contents of term defined	Term routing
application client buffer offset	SAM-2	SAM-2	DS → targ → init
data buffer size	SAM-2	SAM-2	AC → init
command descriptor block	SAM-2	SAM-2/cmd (note 1)	AC → init → targ → DS
data-in buffer	SAM-2	cmd (note 2)	DS → targ → init → AC
data-out buffer	SAM-2	cmd (note 2)	AC → init → targ → DS
device server buffer	SAM-2	cmd (note 2)	DS → targ → init
initiator SCSI ID	SAM-2	this standard	DS → targ **or** TM → targ
link control function	this standard	this standard	AC → init → targ
logical unit number	SAM-2	this standard	AC → init → targ → DS **or** AC → init → targ → TM **or** DS → targ → init
request byte count	SAM-2	SAM-2	DS → targ
service response	SAM-2	this standard	DS → targ → init → AC **or** targ → DS
status	SAM-2	SAM-2	DS → targ → init → AC
tag	SAM-2	this standard	AC → init → targ → DS **or** AC → init → targ → TM **or** DS → targ → init
target identifier	SAM-2	this standard	AC → init → targ → DS **or** AC → init → targ → TM **or** DS → targ
target identifier + initiator identifier	this standard	this standard	targ → DS **or** targ → TM
task attribute	SAM-2	this standard	AC → init → targ → DS

Key: AC = application client, cmd = SCSI command standards, DS = device server, init = initiator, SAM-2 = SCSI Architecture Model-2 standard, TM = task manager, targ = target

Notes
1) The portions not defined in SAM-2 are defined in the SCSI command standards (e.g. SPC-2).
2) Parameter lists are defined within one of the SCSI command standards (e.g., SPC-2). SCSI standards do not define non-parameter list information.

A.2 Notation for procedures and functions

In this standard, the model for functional interfaces between objects is the callable procedure as defined in SAM-2. The interfaces are specified using the following notation:

[Result =] Procedure Name (IN ([input-1] [,input-2] …), OUT ([output-1] [,output-2] …))

Where:

Result: A single value representing the outcome of the procedure or function.

Procedure Name: A descriptive name for the function to be performed.

input-1, input-2, …: A comma-separated list of names identifying caller-supplied input data objects.

output-1, output-2, …: A comma-separated list of names identifying output data objects to be returned by the procedure.

“[…]”: Brackets enclosing optional or conditional parameters and arguments.

This notation is explained in more detail in SAM-2.

A.3 Application client SCSI command services

The SCSI command services shall be requested by the application client using a procedure call defined as:

Service response =Execute Command (IN (I_T_L_x Nexus, CDB, [Task Attribute], [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], Autosense Request, [Command Reference Number]), OUT ([Data-In Buffer], [Sense Data], Status))

A.4 Send SCSI command service

The send SCSI command service is a four step confirmed service that provides the means to transfer a command data block to a device server.

Processing the execute command procedure call for a send SCSI command service shall be composed of the 4 step confirmed service shown in table A.3.

Table A.3 - Processing of send SCSI command service procedure

Step	Protocol service name Fibre Channel Protocol Service Interface procedure call
request	Send SCSI Command request
	Send SCSI Command (IN (I_T_L_x Nexus, CDB, [Task Attribute], [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], Autosense Request, [Command Reference Number]))
indication	SCSI Command Received indication
	SCSI Command Received (IN (I_T_L_x Nexus, [Task Attribute], CDB, Autosense Request, [Command Reference Number]))
response	Send Command Complete response
	Send Command Complete (IN (I_T_L_x Nexus, [Sense Data], Status, Service Response))
confirmation	Command Complete Received confirmation
	Command Complete Received (IN (I_T_L_x Nexus, [Data-In Buffer], [Sense Data], Status, Service Response))

A.5 Data Transfer Protocol Services

A.5.1 Overview of data buffer movement services

The SCSI data buffer movement services shall be requested by the device server using either the data-in delivery service or the data-out delivery service.

A.5.2 Data-in delivery service

The data-in delivery service is a two step confirmed service (see table A.4) that provides the means to transfer a parameter list or data from a device server to an initiator..

Table A.4 - Processing of data-in delivery service procedure

Step	Protocol service name Fibre Channel Protocol Service Interface procedure call
request	Send Data-In request
	Send Data-In (IN (I_T_L_x Nexus, Device Server Buffer, Application Client Buffer Offset, Request Byte Count))
confirmation	Data-In Delivered confirmation
	Data-In Delivered (IN (I_T_L_x Nexus))

A.5.3 Data-out delivery service

The data-out delivery service is a two step confirmed service (see table A.5) that provides the means to transfer a parameter list or data from an initiator to a device server.

Table A.5 - Processing of data-out delivery service procedure

Step	Protocol service name Fibre Channel Protocol Service Interface procedure call
request	Receive Data-Out request
	Receive Data-Out (IN (I_T_L_x Nexus, Application Client Buffer Offset, Request Byte Count, Device Server Buffer))
confirmation	Data-Out Received confirmation
	Data-Out Received (IN (I_T_L_x Nexus))

A.6 Task management services

The task management function shall be requested from the application client using a procedure call defined as:

Service response = Function name (IN (I_T_x_x nexus))

SAM-2 defines the allowed service responses and the allowed function names. The task management functions for FCP are defined in 9.1.2.4 and 9.1.3.

Annex B

(informative)

FCP examples

B.1 Examples of the use of FCP Information Units (IUs)

B.1.1 Overview of examples

This annex provides examples of the use of FCP IUs. The functions enclosed in square brackets summarize actions that are not specified by this standard, but are typically executed by SCSI initiators or targets. Sequence streaming may optionally be performed between any two IUs that do not transfer Sequence Initiative.

B.1.2 SCSI FCP read operation

A typical SCSI FCP read operation with a single data IU is shown in table B.1.

Table B.1 - FCP read operation, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND →	
		[Prepare data transfer]
	← I3, FCP_DATA	Data in action
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		

B.1.3 SCSI FCP write operation

A typical SCSI FCP write operation with three data IUs and using FCP_XFER_RDY is shown in table B.2.

Table B.2 - FCP write operation, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND →	
		[Prepare data transfer buffer]
	← I1, FCP_XFER_RDY	First data delivery request
First Data Out Action	T6, FCP_DATA →	
	← I1, FCP_XFER_RDY	Second data delivery request
Second Data Out Action	T6, FCP_DATA →	
	← I1, FCP_XFER_RDY	Last data delivery request
Last Data Out Action	T6, FCP_DATA →	
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		

B.1.4 SCSI FCP operation with no data transfer or with check condition

A typical SCSI FCP operation terminating without data transfer, either because of an error or because the SCSI command does not require any data transfer, is shown in table B.3.

Table B.3 - FCP operation without data transfer, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND →	
		[perform command]
	← I4, FCP_RSP	Response
[Indicate command completion]		

B.1.5 SCSI FCP read operation with multiple FCP_DATA IUs

A typical SCSI read operation with multiple FCP_DATA IUs is shown in table B.4.

Table B.4 - FCP read operation, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND →	
		[Prepare data transfer]
	← I3, FCP_DATA	Data in action
	← I3, FCP_DATA	Data in action
	← I3, FCP_DATA	Data in action
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		

B.1.6 SCSI FCP write operation with FCP_XFR_RDY disabled

A typical SCSI write operation performed with FCP_XFER_RDY disabled is shown in table B.5. Only the first transfer is performed without a requesting FCP_XFER_RDY.

Table B.5 - FCP write operation with FCP_XFER_RDY disabled, example

Initiator function	IU	Target function
Command request	T2, FCP_CMND →	
Data Out Action	T6, FCP_DATA →	
	← I1, FCP_XFER_RDY	Second data delivery request
Data Out Action	T6, FCP_DATA →	
	← I1, FCP_XFER_RDY	Last data delivery request
Data Out Action	T6, FCP_DATA →	
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		

B.1.7 SCSI linked commands

A SCSI WRITE command linked after a SCSI READ command is shown in table B.6. The WRITE command is using the FCP_XFER_RDY IU. INTERMEDIATE Status in the FCP_RSP, together with the link control bits present in the CDB of the FCP_CMND indicate that the second operation is linked to the first.

Table B.6 - FCP linked commands, example

Initiator function	IU	Target function
Command request (READ)	T1, FCP_CMND →	
		[Prepare data transfer]
	← I3, FCP_DATA	Data in action
		[Prepare response message]
	← I5, FCP_RSP	Response (INTERMEDIATE or INTERMEDIATE CONDITION MET status)
[Perform command linking]		
Command request (WRITE)	T3, FCP_CMND →	
		[Prepare data transfer buffer]
	← I1, FCP_XFER_RDY	Data delivery request
Data Out Action	T6, FCP_DATA →	
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		

B.1.8 SCSI WRITE command with confirmed completion

A SCSI WRITE command with confirmed completion is shown in table B.7.

Table B.7 - FCP write command with confirmed completion

Initiator function	IU	Target function
Command request (WRITE)	T1, FCP_CMND →	
		[Prepare data transfer]
	← I1, FCP_XFER_RDY	Data delivery request
Data Out action	T6, FCP_DATA →	
		[Prepare response message]
	← I5, FCP_RSP	Response, with FCP_CONF_REQ
[indicate command completion]		
Confirm completion	T12, FCP_CONF →	[Accept confirmation]

B.1.9 SCSI FCP task management function

An example of a SCSI Task Management function is shown in table B.8. Additional link services may be required in some cases to complete the activities initiated by the Task Management function.

Table B.8 - FCP task management function, example

Initiator function	IU	Target function
Command request, no CDB	T1, FCP_CMND →	
		[Do Task Management]
	← I4, FCP_RSP	Response
[Indicate task management complete]		

B.2 FCP write example, frame level

A chart of the Sequences and frames typically transmitted to perform an FCP write is shown in figure B.1. All frames of a Sequence have a frame level FC-FS acknowledgment returned automatically as part of the link control.

Figure B.1 - Example of class 2 FCP write I/O operation

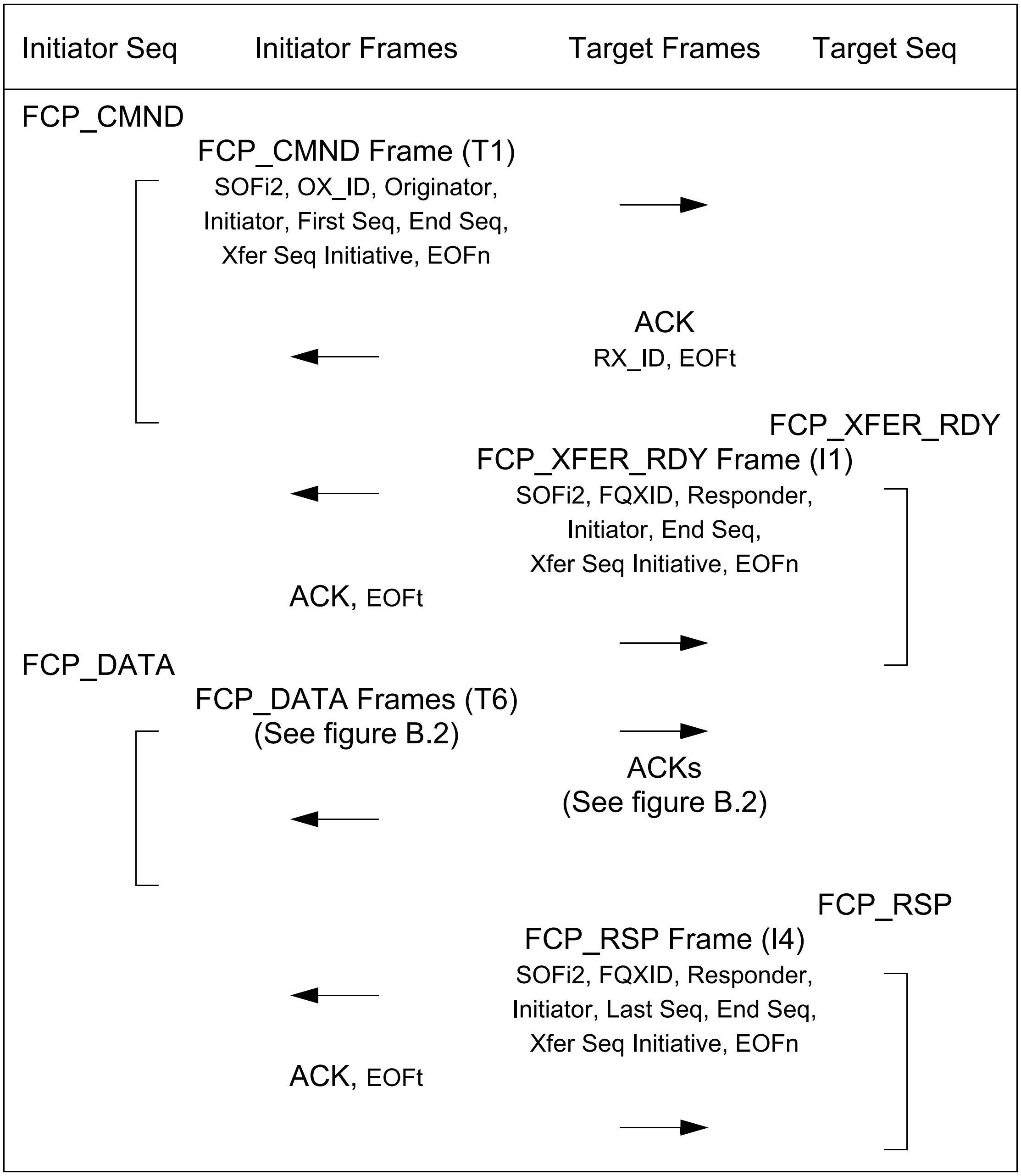
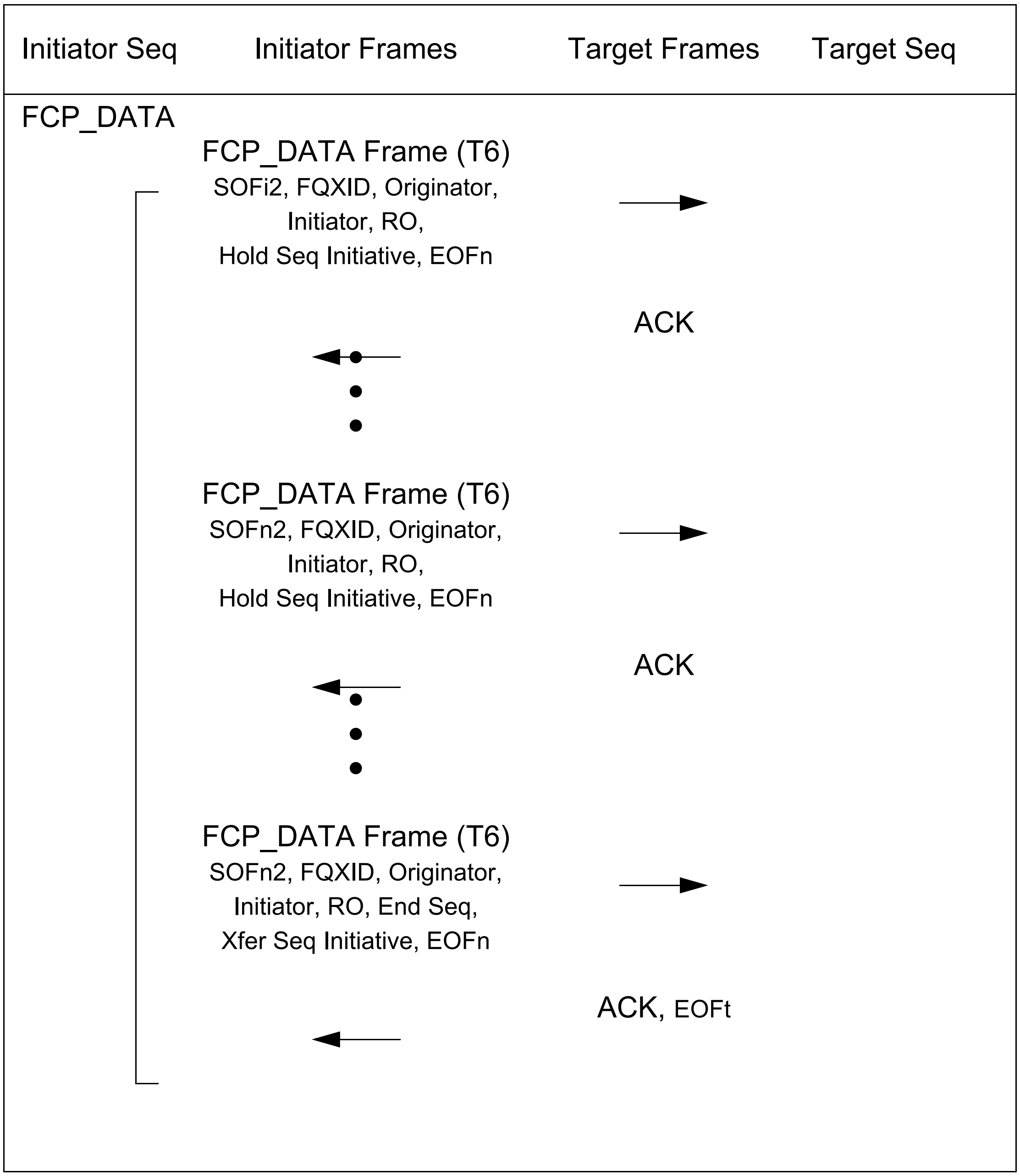


Figure B.2 - Example of class 2 FCP_DATA write



B.3 FCP read example, frame level

A chart of the Sequences typically transmitted to perform an FCP read is shown in figure B.3. All frames of a Sequence have a frame level FC-FS acknowledgment returned automatically as part of the link control.

Figure B.3 - Example of class 2 FCP read I/O operation

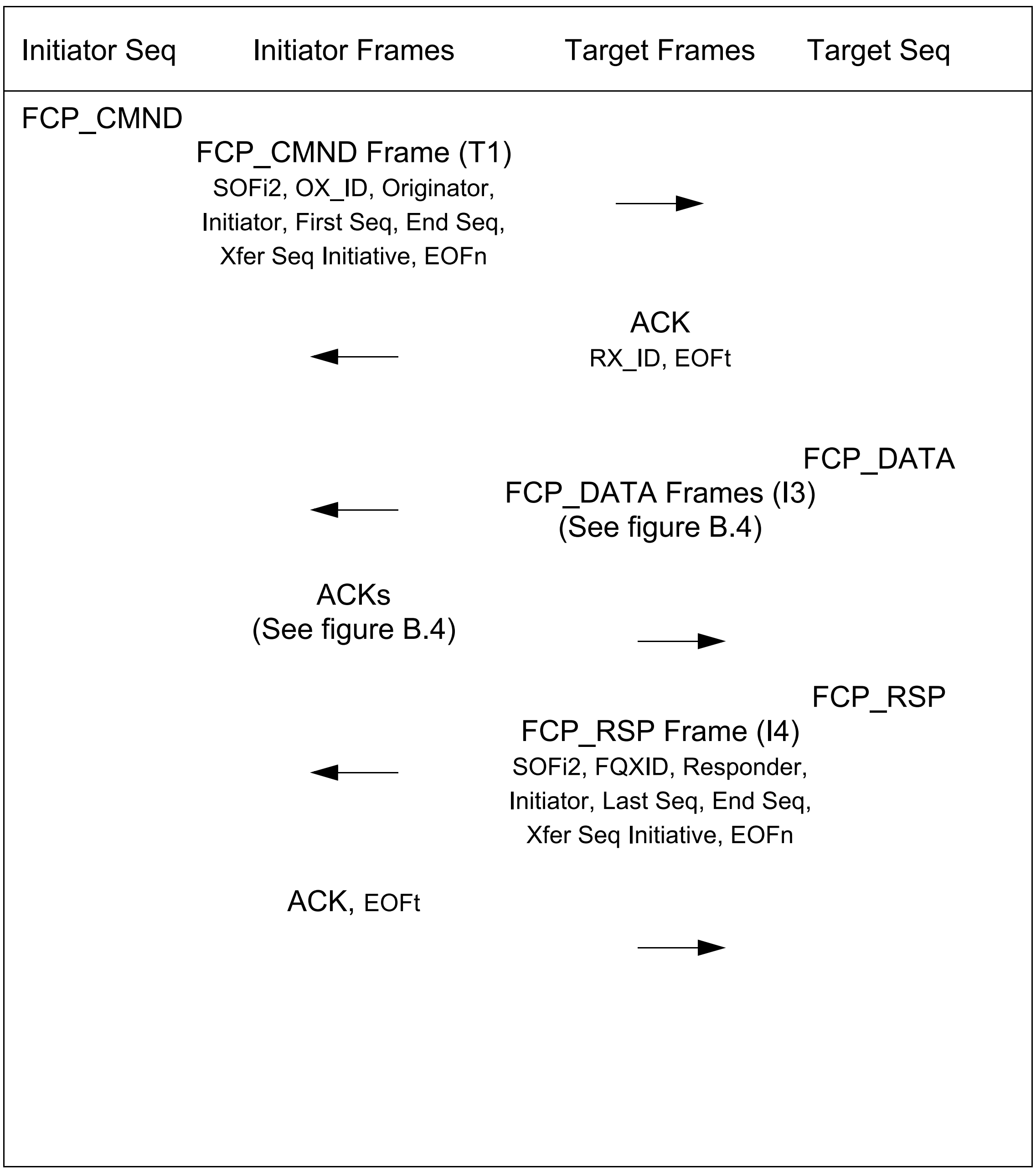
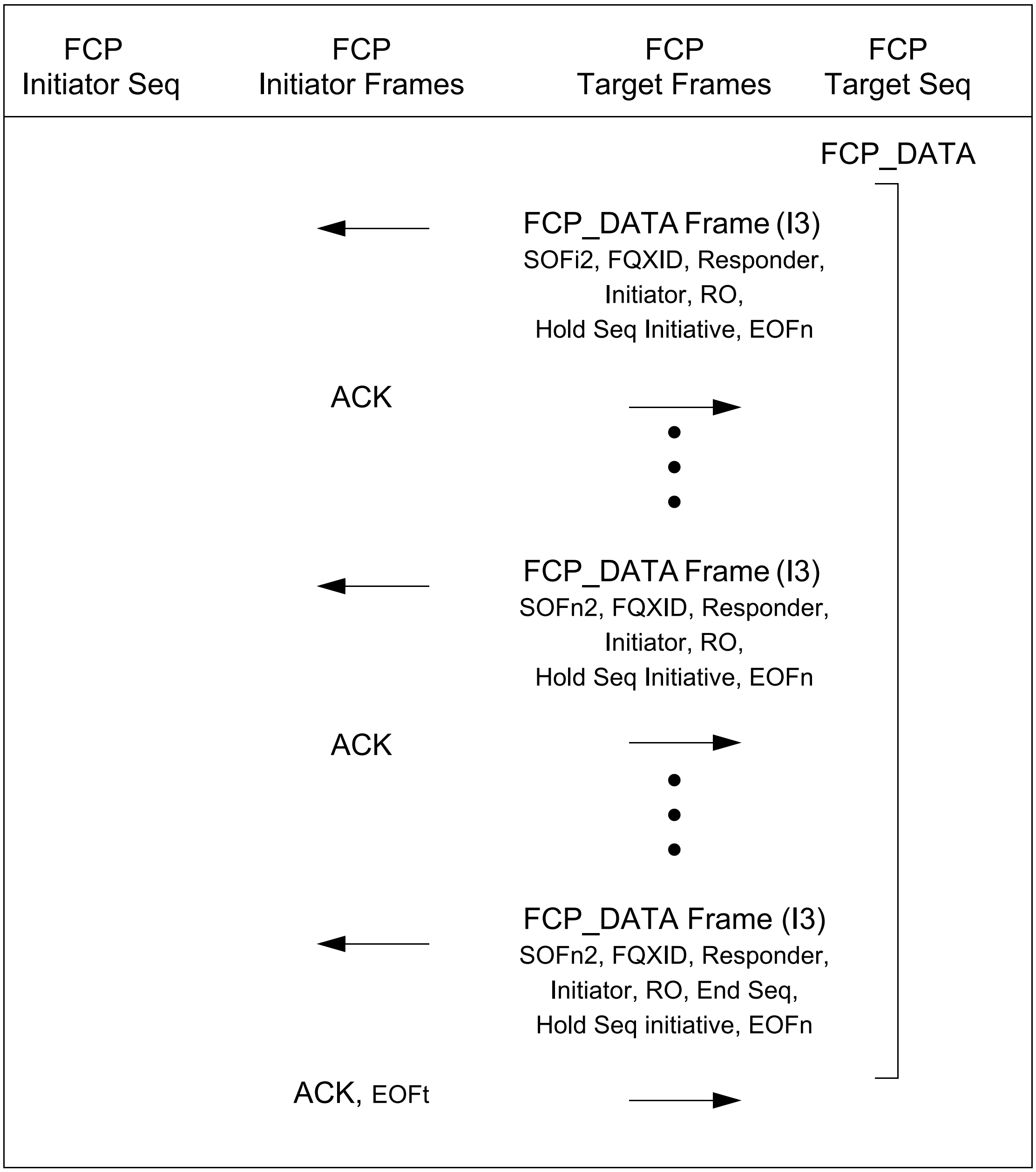


Figure B.4 - Example of class 2 FCP_DATA read



Annex C

(informative)

Error detection and recovery action examples

C.1 Introduction

This informative annex diagrams various error detection and recovery procedures for FCP devices conforming to this standard. These examples include cases where recovery mechanisms specified by FC-FS interact with recovery mechanisms specified by this standard. The conventions for the diagrams are shown in table C.1.

Table C.1 - Diagram Drawing Conventions

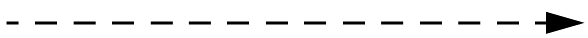
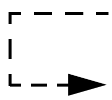
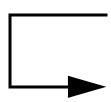
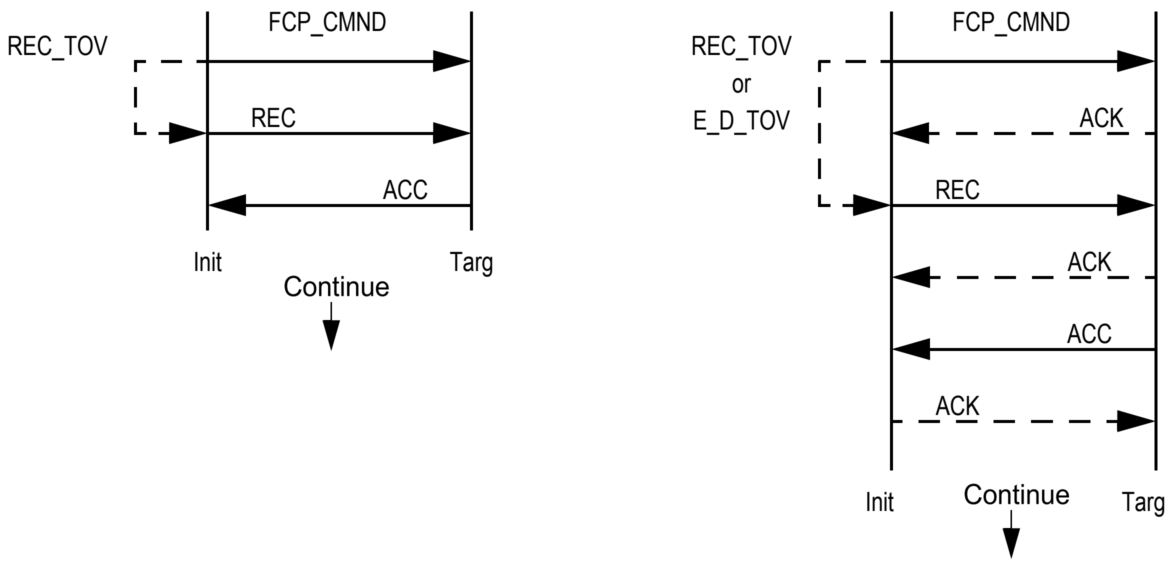
Drawing Convention	Meaning
	Acknowledged or Unacknowledged Frame
	Acknowledgement Frame
	Time-out value exceeded, caused transmission of IU, FC-4 Link Service, or ELS
	IU or ELS received is processed to transmit IU, FC-4 Link Service, or ELS
X	Frame lost or dropped
CI Continue ↓	Error detection complete. Operation continues with specified Error Recovery if continuously increasing Sequence count prerequisites are met.
Continue ↓	Error detection complete. Operation continues with specified Error Recovery if continuously increasing Sequence count prerequisites are not met.

Figure C.1 - Lengthy FCP_CMND or Lost ACK

Status Detection

Unacknowledged Classes **Acknowledged Classes**



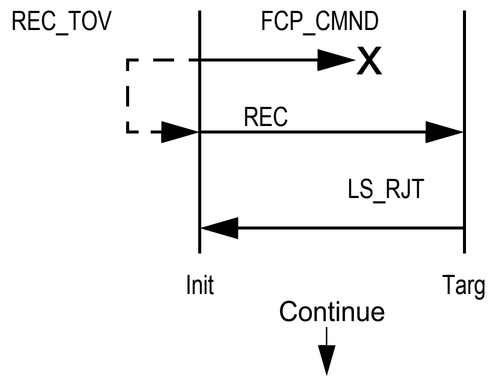
REC can be optionally used at any time to ascertain status of an Exchange. It can also be used in conjunction with ABTS(Sequence) to obtain additional information useful in the Error Recovery process.

No Error Recovery

The ACC for the REC indicates the Exchange is open and the target holds Sequence Initiative. No error recovery is required.

Figure C.2 - FCP_CMND Lost, Unacknowledged Classes

Error Detection



Error Recovery

The LS_RJT (Logical Error, Invalid OX_ID - RX_ID combination) for the REC indicates the Exchange is unknown. The FCP_CMND is retransmitted in a new Exchange using the same CRN and in the same class of service.

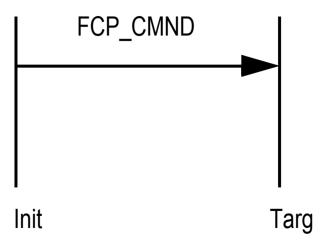
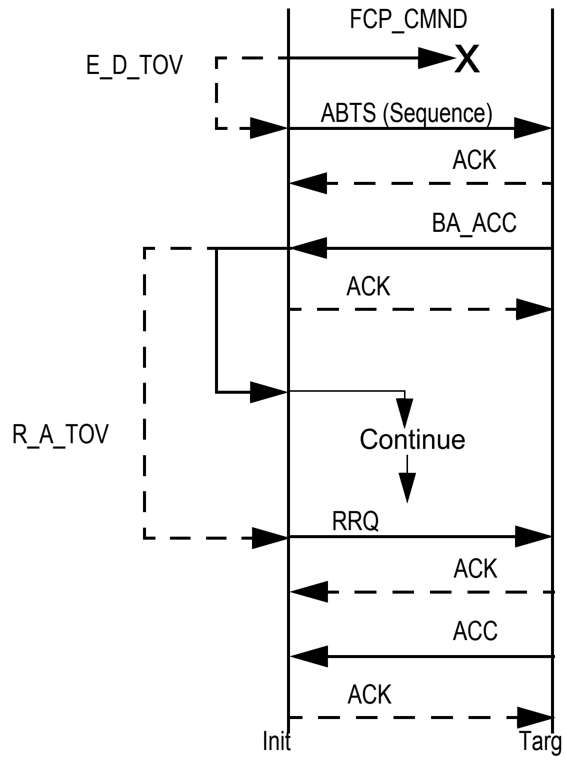


Figure C.3 - FCP_CMND Lost, Acknowledged Classes

Error Detection



The BA_ACC indicates that the FCP_CMND was never received. The BA_ACC payload is SEQ_ID Validity = invalid, Low SEQ_CNT = 0, High SEQ_CNT = SEQ_CNT of ABTS frame. Both the initiator and target establish Recovery Qualifiers. The value of R_A_TOV for in-order delivery is zero. The use of REC to determine status for error recovery is optional.
The FCP_CMND is retransmitted in a new Exchange using the same CRN and in the same class of service.

Error Recovery

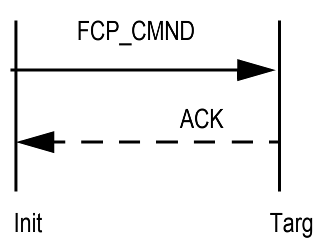
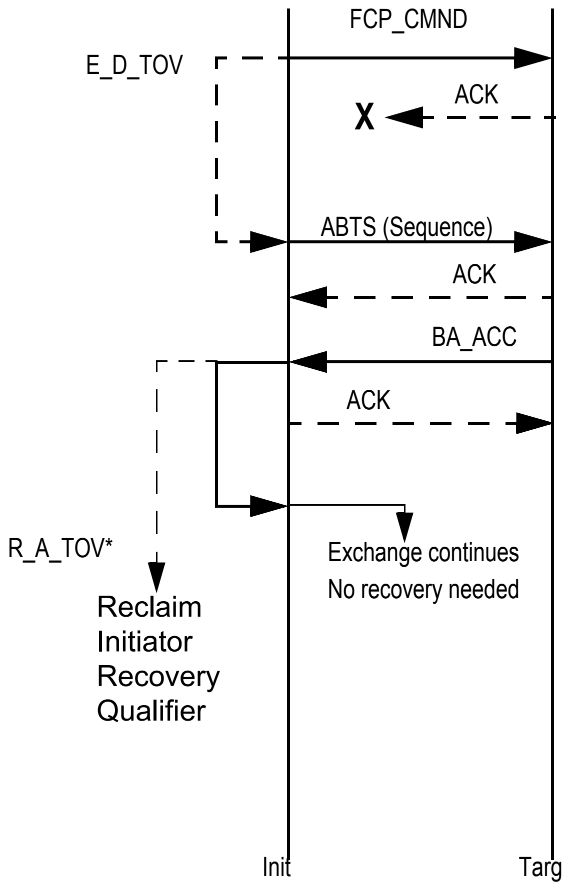


Figure C.4 - FCP_CMND Acknowledgement Lost, Acknowledged Classes

Error Detection

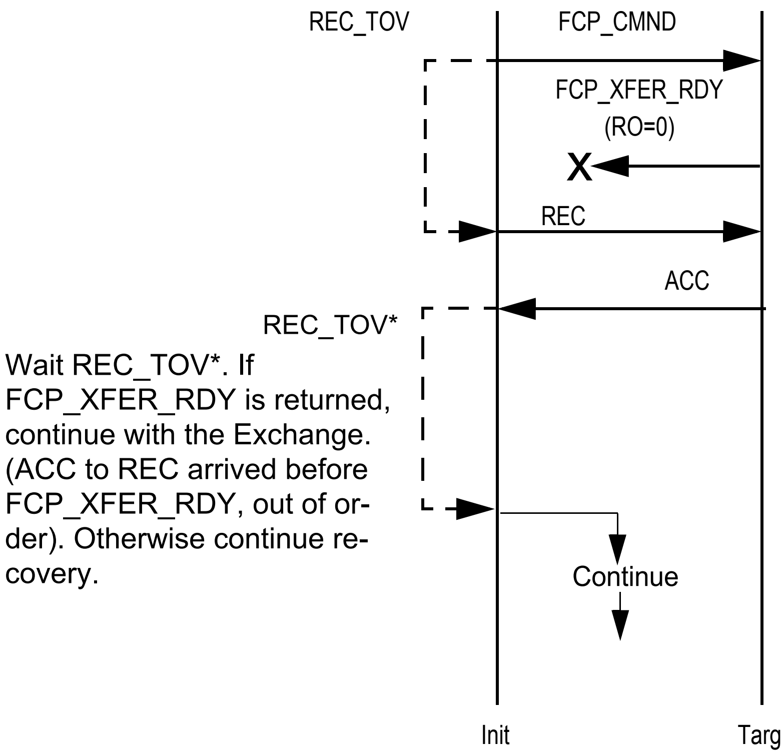


No Error Recovery

The BA_ACC Payload indicates that FCP_CMND was received. The BA_ACC payload is SEQ_ID Valid, SEQ_ID value of FCP_CMND, low SEQ_CNT = high SEQ_CNT = SEQ_CNT of FCP_CMND. Note that the issuance of RRQ is not necessary in this case, since the target has not established a Recovery Qualifier. However, the initiator cannot reclaim the resources associated with its Recovery Qualifier until R_A_TOV time-out expires. The value for R_A_TOV is zero for in-order delivery. The use of REC to determine status for error recovery is optional.

Figure C.5 - FCP_XFER_RDY Lost, Unacknowledged Classes

Error Detection



Error Recovery

The ACC for the REC indicates the initiator holds Sequence Initiative and the Exchange is open. The initiator sends an SRR requesting the FCP_XFER_RDY be resent. The target resends the FCP_XFER_RDY using the same Relative Offset. A new SEQ_ID is used for retransmission of the FCP_XFER_RDY. For in-order delivery, the value of REC_TOV* is zero.

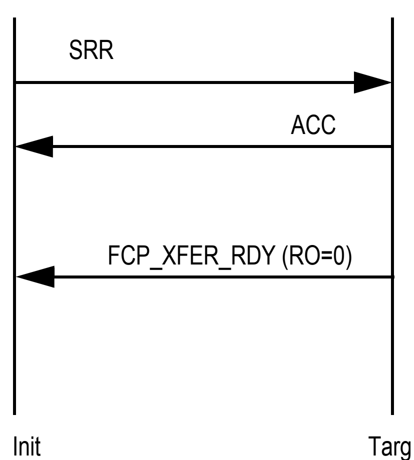
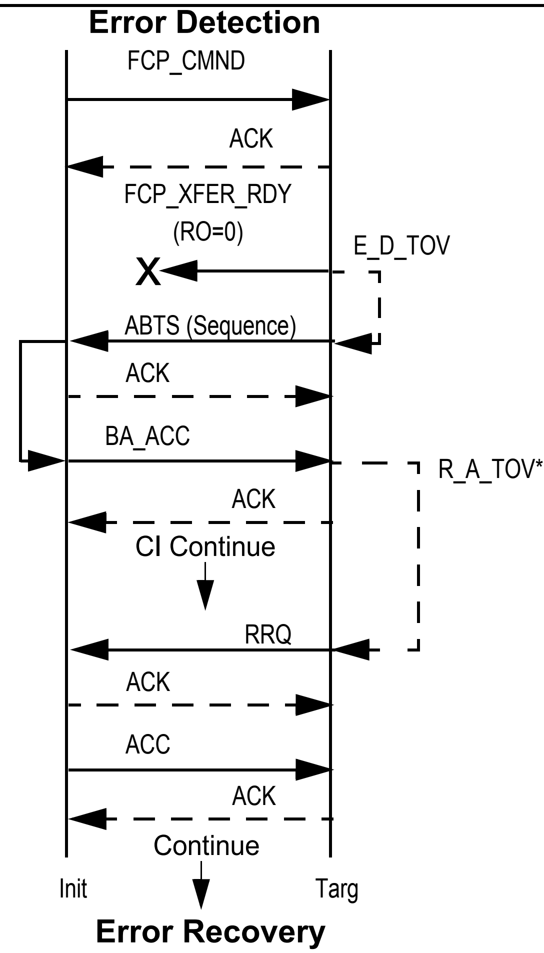


Figure C.6 - FCP_XFER_RDY Lost, Acknowledged Classes



For acknowledged classes, the BA_ACC indicates that the FCP_XFER_RDY was never received by the initiator. The BA_ACC payload is SEQ_ID invalid, low SEQ_CNT = 0, high SEQ_CNT = SEQ_CNT in ABTS frame = 1. Both target and initiator establish Recovery Qualifiers. The value for R_A_TOV* for in-order delivery is zero. A new SEQ_ID is used in the retransmission of FCP_XFER_RDY. The use of REC to determine status for error recovery is optional.

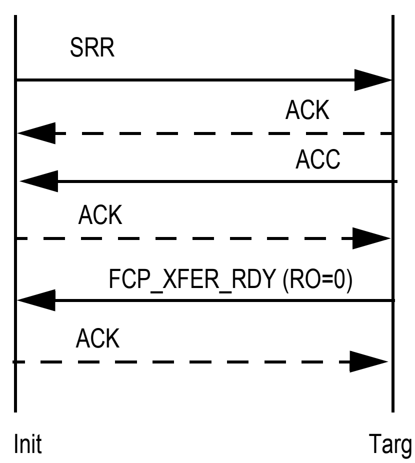
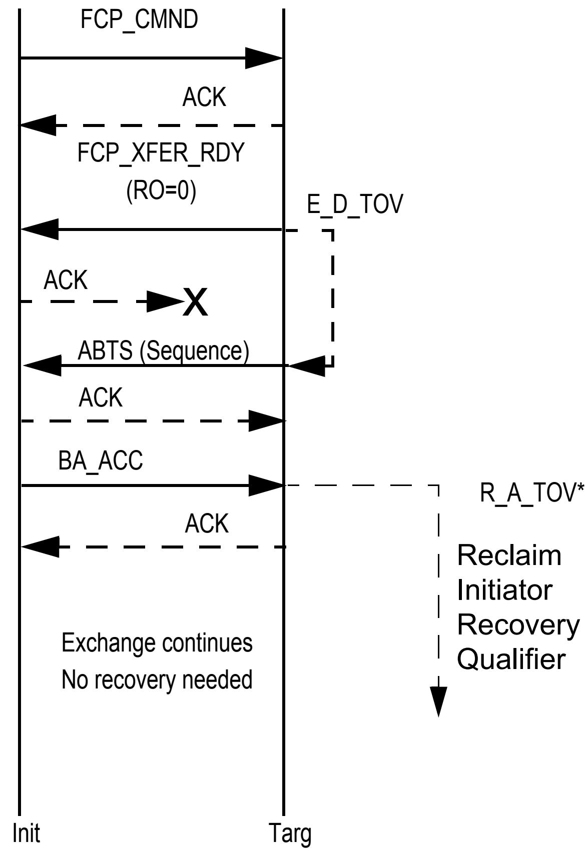


Figure C.7 - FCP_XFER_RDY Received, ACK Lost, Acknowledged Classes

Error Detection

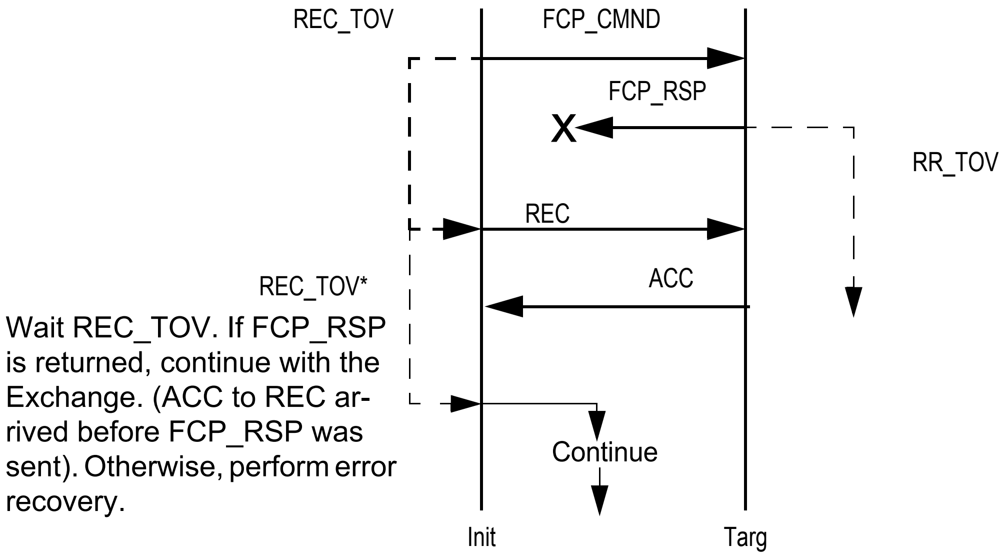


No Error Recovery

The BA_ACC indicates that the FCP_XFER_RDY was received by the initiator. The BA_ACC payload is SEQ_ID valid, SEQ_ID = SEQ_ID of the FCP_XFER_RDY frame, low SEQ_CNT = high SEQ_CNT = SEQ_CNT of the ABTS frame. No error recovery is required. There is no need for the target to issue the RRQ since no Recovery Qualifier was established by the initiator in this case. It still lets R_A_TOV* expire before reclaiming the resources associated with its Recovery Qualifier. The value of R_A_TOV* for in-order delivery is zero, The use of REC to determine status for error recovery is optional.

Figure C.8 - FCP_RSP Lost, FCP_CONF not requested, Unacknowledged Classes

Error Detection



Error Recovery

The ACC for the REC indicates the initiator holds Sequence Initiative and the Exchange is complete. The initiator sends an SRR requesting the FCP_RSP be resent. The target retransmits the FCP_RSP. For in-order delivery, the value of REC_TOV* is zero.
For Sequence level recovery, the target keeps the context of this Exchange as defined in 12.4.1.5.
For Exchange level recovery, the context of the Exchange can be purged by the target after FCP_RSP is transmitted.

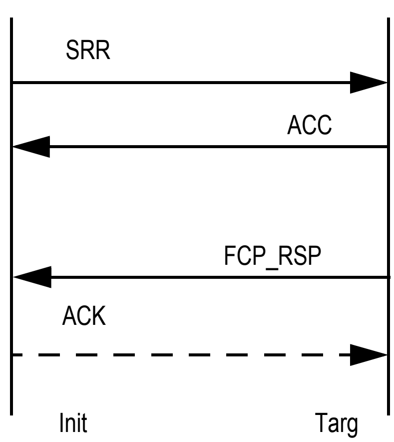
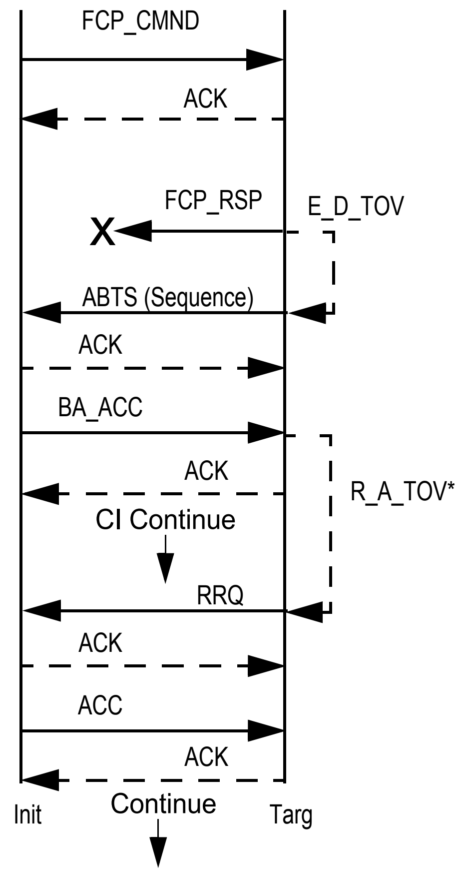


Figure C.9 - FCP_RSP Lost, FCP_CONF not requested, Acknowledged Classes

Error Detection



Error Recovery

BA_ACC to ABTS indicates that the FCP_RSP was never received by the initiator. The payload is SEQ_ID invalid, low SEQ_CNT = 0, high SEQ_CNT = SEQ_CNT in ABTS frame = 1. Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order delivery is zero. A new SEQ_ID is used in the retransmission of FCP_RSP.
The use of REC to determine status for error recovery is optional.

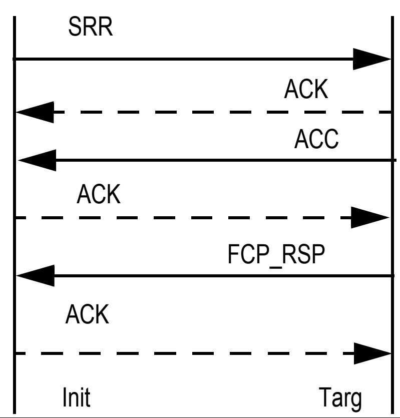
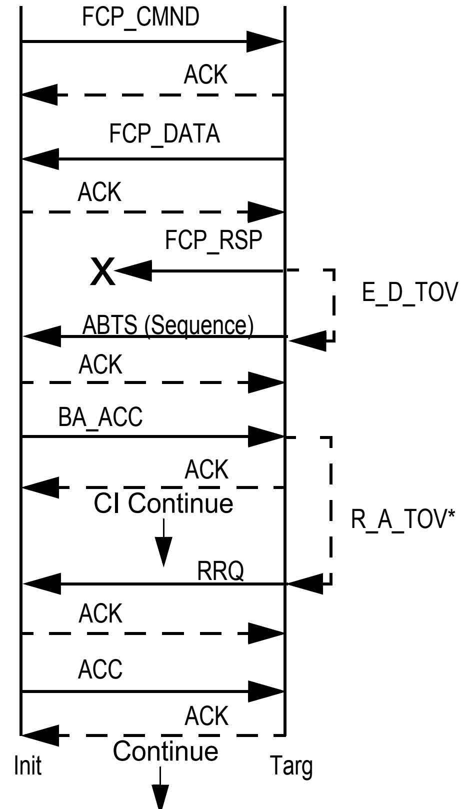


Figure C.10 - FCP_RSP Lost Read Command, no FCP_CONF, Acknowledged Classes

Error Detection



Error Recovery

BA_ACC to ABTS indicates that the FCP_RSP was never received by the initiator. The payload is SEQ_ID valid, SEQ_ID = SEQ_ID of FCP_DATA Sequence, low SEQ_CNT = x, high SEQ_CNT = SEQ_CNT in ABTS frame. Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order delivery is zero. A new SEQ_ID is used in the retransmission of FCP_RSP.

The use of REC to determine status for error recovery is optional.

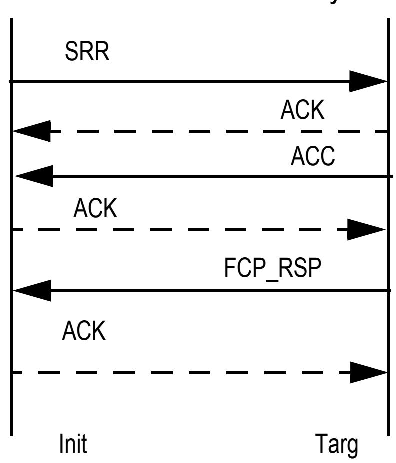
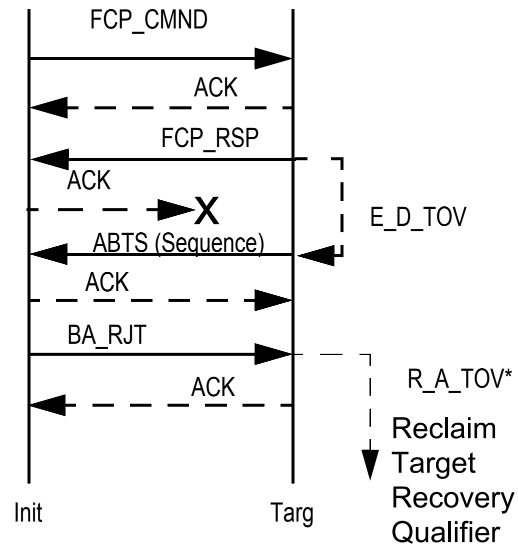


Figure C.11 - FCP_RSP Received, ACK Lost, Acknowledged Classes, Example 1

Error Detection

Acknowledged Classes

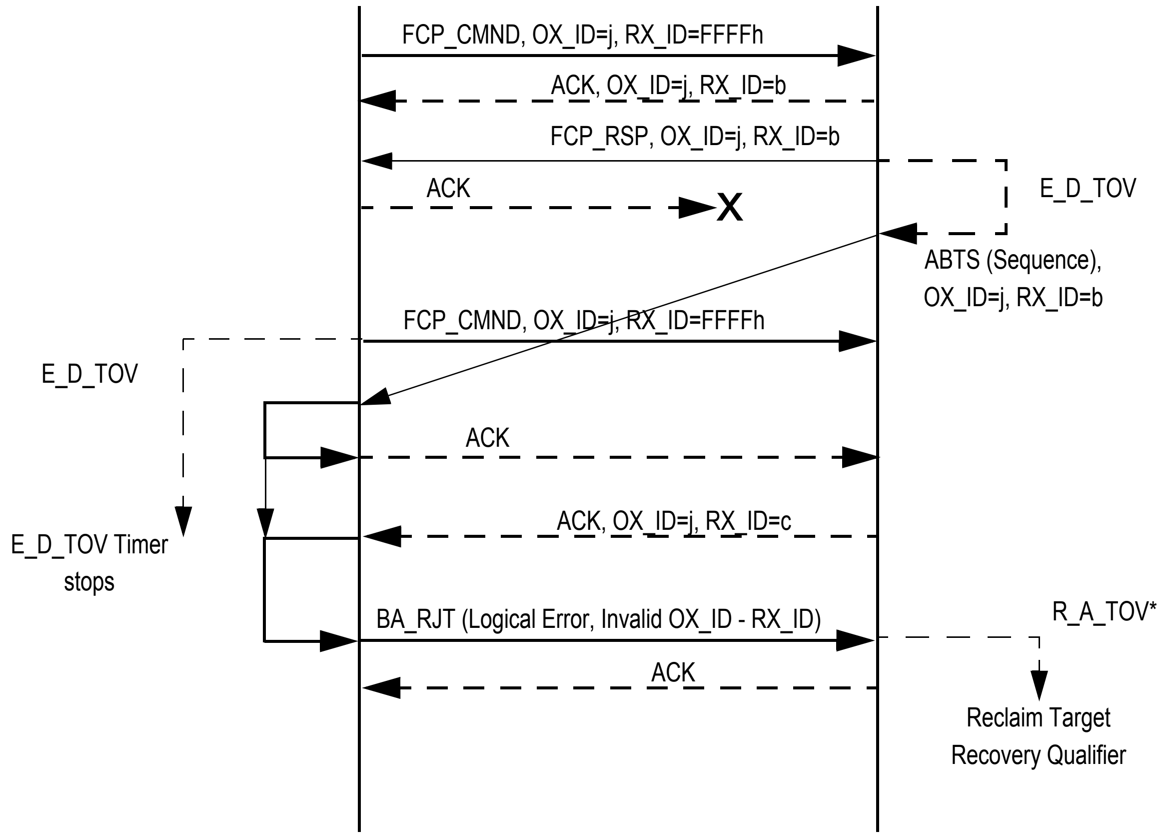


No Error Recovery

The BA_RJT for the ABTS indicates with reason code of Logical Error and the reason explanation code of Invalid OX_ID - RX_ID Combination that the Exchange is unknown and therefore complete. No error recovery is required. The target establishes a Recovery Qualifier. The associated resources cannot be reused for a period of R_A_TOV. For in-order delivery, the value of R_A_TOV* is zero. The target does not need to issue RRQ as no Recovery Qualifier was established by the initiator.

Figure C.12 - FCP_RSP Received, ACK Lost, Acknowledged Classes, Example 2

Error Detection

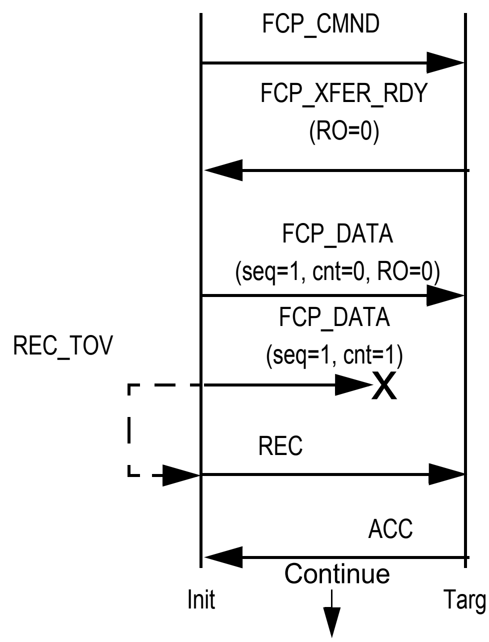


No Error Recovery

The BA_RJT for the ABTS indicates that the Exchange is unknown and therefore complete and no error recovery is required. The target establishes a Recovery Qualifier. The associated resources cannot be reused for a period of R_A_TOV.
No action is taken on ABTS until the ACK to the outstanding Sequence has been received, allowing the analysis to take into consideration the RX_ID, to eliminate ambiguity.
For in-order delivery, the value of R_A_TOV* is zero.
The target need not issue RRQ, as no Recovery Qualifier was established by the initiator.

Figure C.13 - Lost Write Data, Last Frame of Sequence, Unacknowledged Classes

Error Detection



Error Recovery

The ACC for the REC indicates the target does not hold Sequence Initiative and the Exchange is open. The initiator sends an SRR requesting an FCP_XFER_RDY be resent using the specified Relative Offset. The target retransmits the FCP_XFER_RDY using the specified Relative Offset. New SEQ_IDs are used for FCP_XFER_RDY and FCP_DATA retransmission.

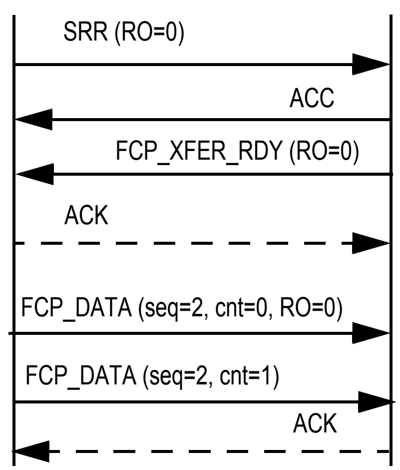
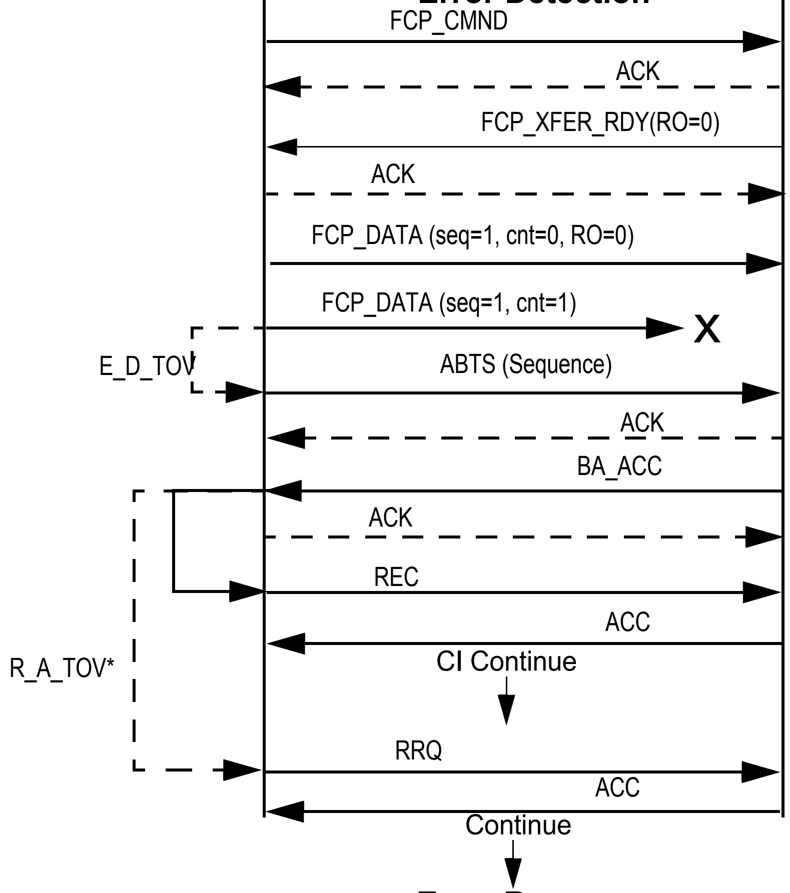


Figure C.14 - Lost Write Data, Last Frame of Sequence, Acknowledged Classes

Error Detection



Error Recovery

Using the information contained in the REC, the recovery may be performed as in figure C.13. The BA_ACC may also be used. The payload is SEQ_ID = valid, SEQ_ID = SEQ_ID of FCP_CMND, low SEQ_CNT of 0, high SEQ_CNT of 2.
New SEQ_IDs are used for retransmitting FCP_XFER_RDY and FCP_DATA. ACKs for REC/ACC and RRQ/ACC are not shown. The value of R_A_TOV* for in-order delivery is zero.

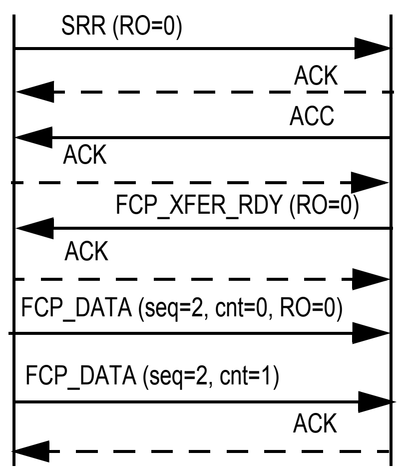
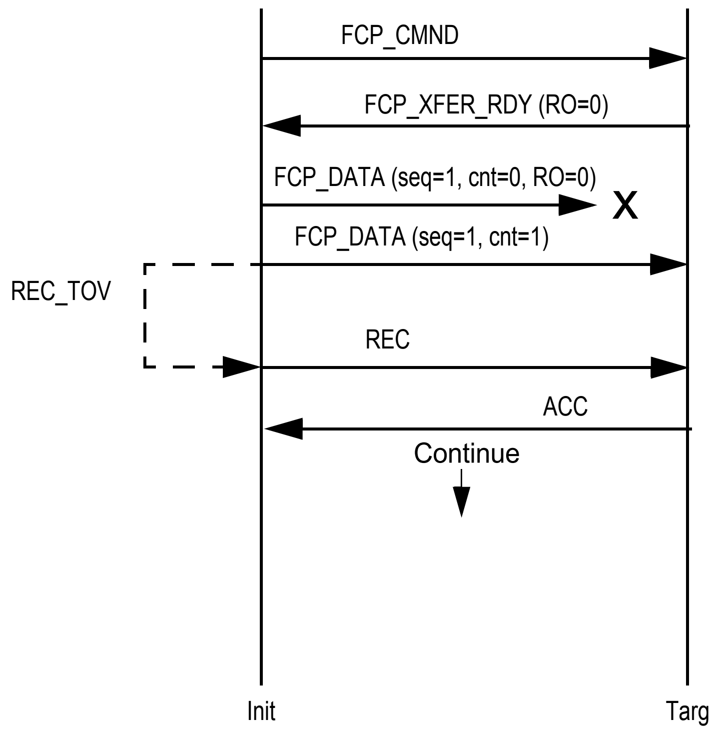


Figure C.15 - Lost Write Data, Not Last Frame of Sequence, Unacknowledged Classes

Error Detection



Error Recovery

The ACC for the REC indicates the target does not hold Sequence Initiative and the Exchange is open. The initiator sends an SRR requesting an FCP_XFER_RDY be resent using the specified Relative Offset. The target retransmits the FCP_XFER_RDY using the specified Relative Offset. New SEQ_IDs are used for FCP_XFER_RDY and FCP_DATA retransmission.

If the lost frame arrives at the target prior to the expiration of REC_TOV, the Exchange proceeds and no recovery is required.

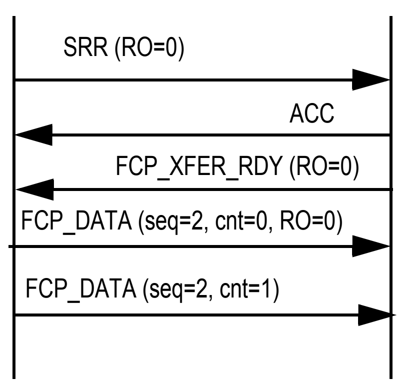
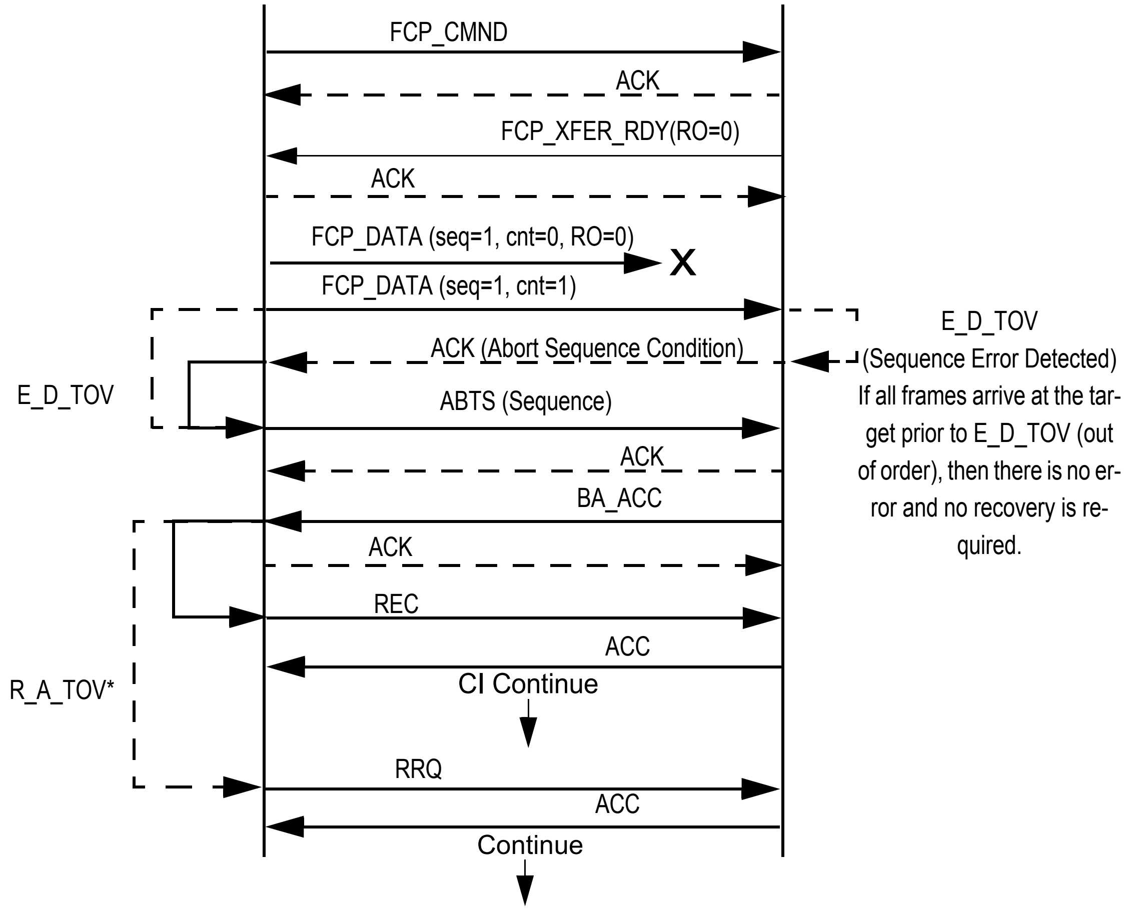


Figure C.16 - Lost Write Data, Not Last Frame of Sequence, Acknowledged Classes

Error Detection



Error Recovery

Using the information contained in the REC, the recovery may be performed as in figure C.15. The BA_ACC may also be used. The payload is SEQ_ID = valid, SEQ_ID = SEQ_ID of FCP_CMND, low SEQ_CNT of 0, high SEQ_CNT of 2.
New SEQ_IDs are used for retransmitting FCP_XFER_RDY and FCP_DATA. ACKs for REC/ACC and RRQ/ACC are not shown. The value of R_A_TOV* for in-order delivery is zero.

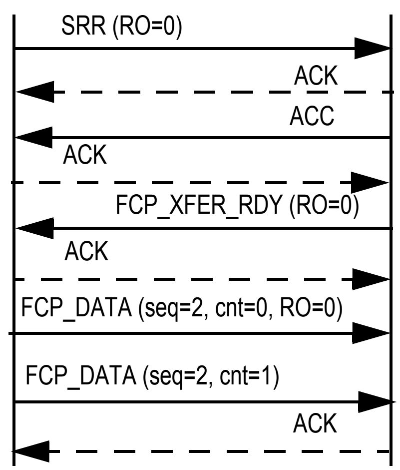
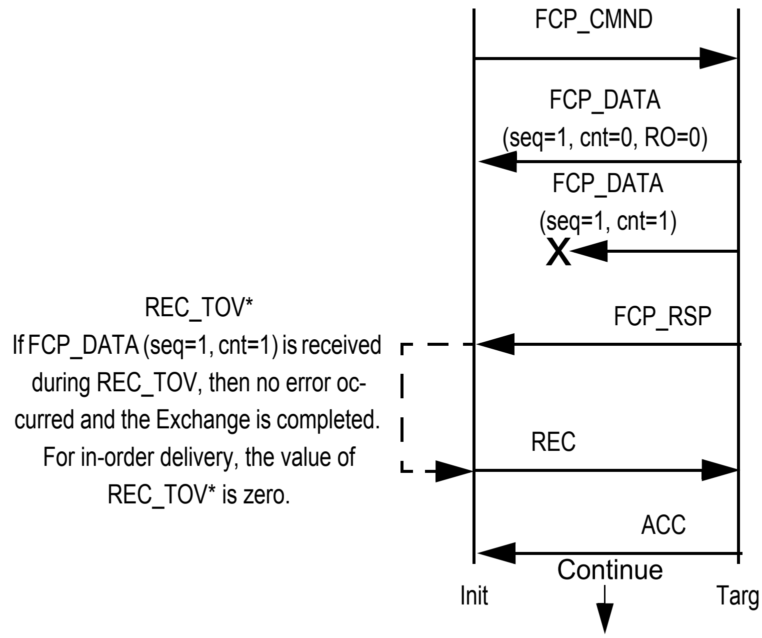


Figure C.17 - Lost Read Data, Last Frame of Sequence, Unacknowledged Classes

Error Detection



Error Recovery

The ACC for the REC indicates the target does not hold Sequence Initiative and since no FCP_CONF was requested, the Exchange is complete. The FC4VALUE specified in the REC ACC is less than the initiator's data transfer count, thus the initiator sends an SRR requesting an FCP_DATA be resent using the specified Relative Offset. FCP_DATA retransmission uses a new SEQ_ID.

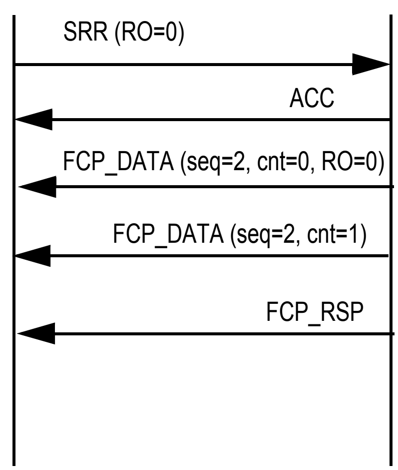
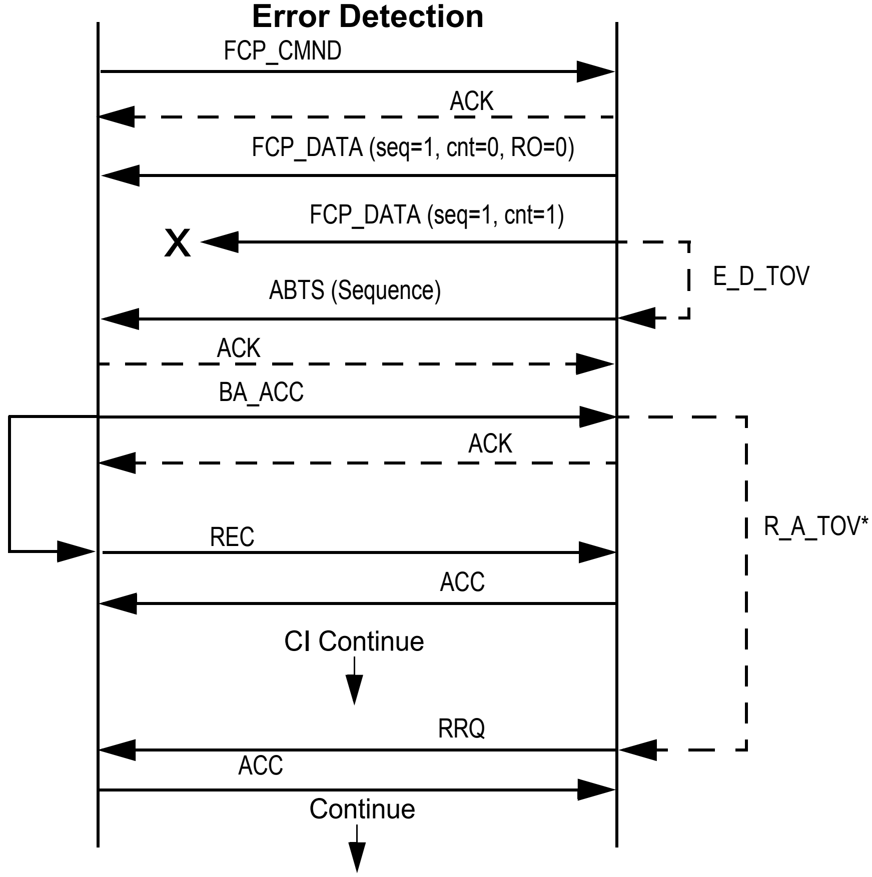


Figure C.18 - Lost Read Data, Last Frame of Sequence, Acknowledged Classes

Error Detection



Error Recovery

Using the information contained in the REC, the recovery may be performed as in figure C.17. The BA_ACC may also be used because it indicates that the FCP_DATA Sequence was not completely received. The payload is SEQ_ID = invalid, low SEQ_CNT of 0, high SEQ_CNT of ABTS frame.

New SEQ_IDs are used for retransmitting FCP_DATA. ACKs for REC/ACC and RRQ/ACC are not shown. The value of R_A_TOV* for in-order delivery is zero.

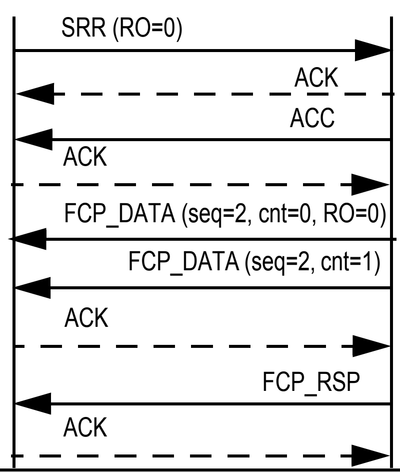
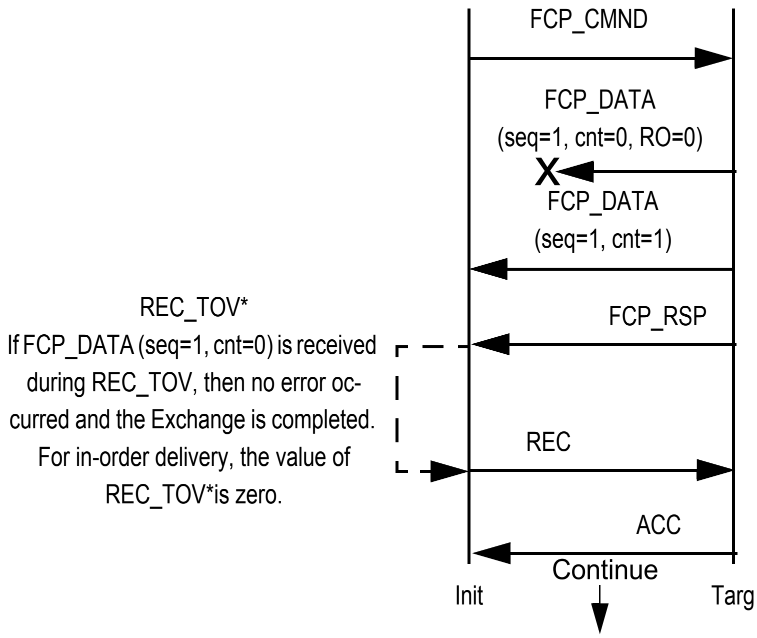


Figure C.19 - Lost Read Data, Not Last Frame of Sequence, Unacknowledged Classes

Error Detection



Error Recovery

The ACC for the REC indicates the target does not hold Sequence Initiative and since no FCP_CONF was requested, the Exchange is complete. The FC4VALUE specified in the REC ACC is less than the initiator's data transfer count, thus the initiator sends an SRR requesting an FCP_DATA be resent using the specified Relative Offset. FCP_DATA retransmission uses a new SEQ_ID.

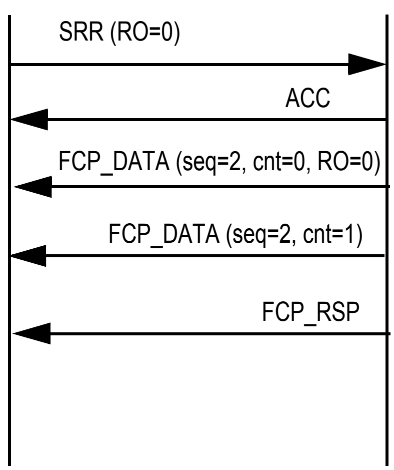
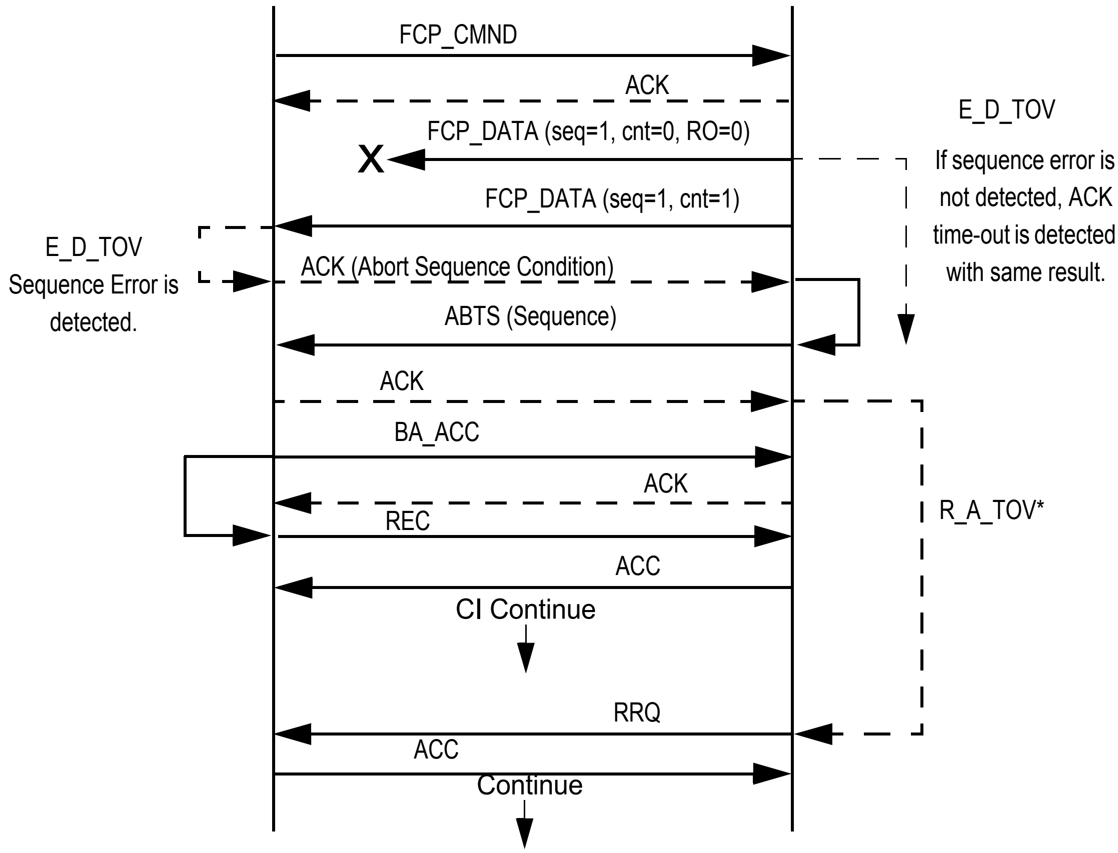


Figure C.20 - Lost Read Data, Not Last Frame of Sequence, Acknowledged Classes

Error Detection



Error Recovery

Using the information contained in the REC, the recovery may be performed as in figure C.17. The BA_ACC may also be used because it indicates that the FCP_DATA Sequence was not completely received. The payload is SEQ_ID = invalid, low SEQ_CNT of 0, high SEQ_CNT of ABTS frame.

New SEQ_IDs are used for retransmitting FCP_DATA. ACKs for REC/ACC and RRQ/ACC are not shown. The value of R_A_TOV* for in-order delivery is zero.

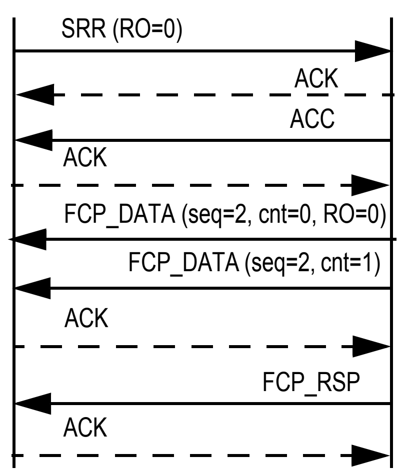
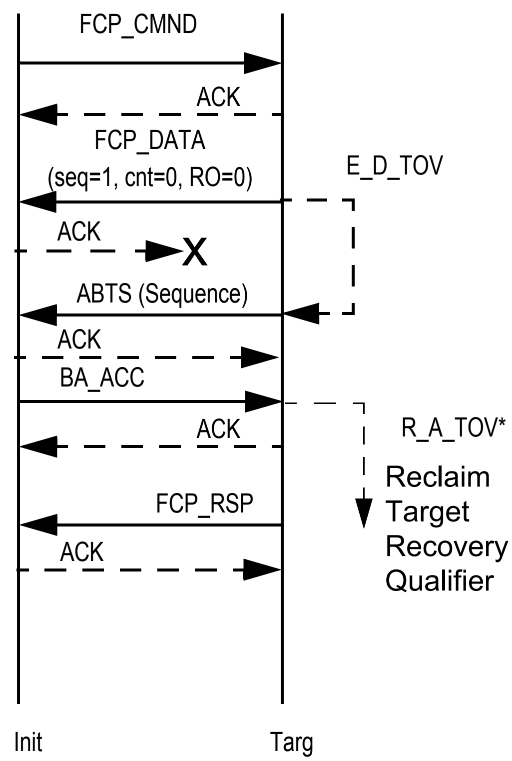


Figure C.21 - ACK Lost on Read (Acknowledged Classes)

Error Detection

Acknowledged Classes



No Error Recovery

The initiator has received the FCP_DATA frame or Sequence. No error recovery is required.

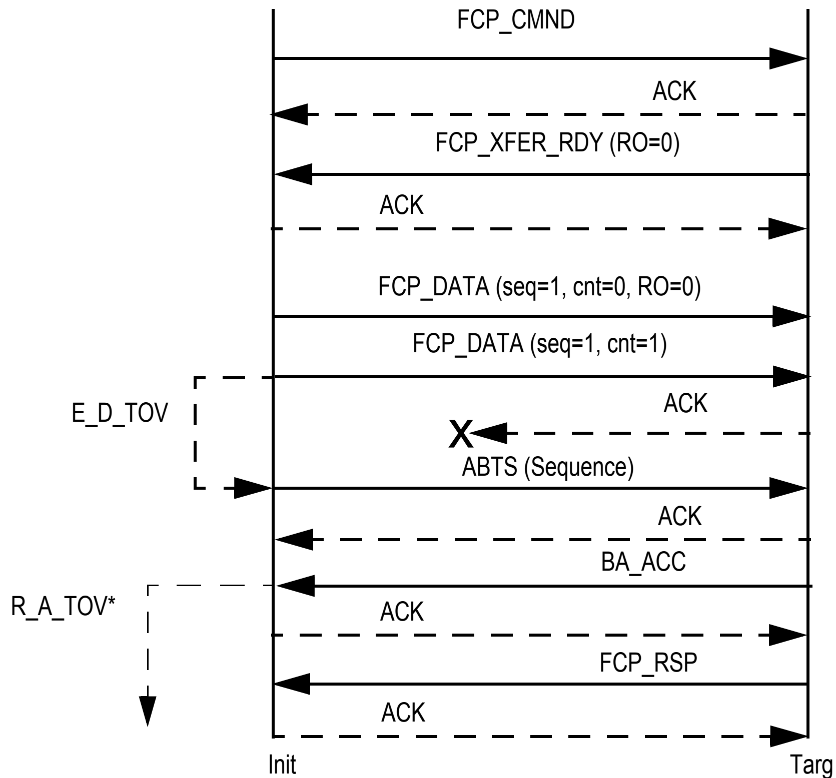
The BA_ACC indicates the FCP_DATA Sequence was received (Payload is SEQ_ID valid, SEQ_ID value = SEQ_ID value of FCP_DATA Sequence, low SEQ_CNT = high SEQ_CNT = SEQ_CNT of ABTS frame).

The target establishes its Recovery Qualifier. The resources associated with the Recovery Qualifier can be reclaimed after R_A_TOV. For in-order delivery, the value of R_A_TOV* is zero.

The issuance of RRQ is optional as no Recovery Qualifier was established by the initiator in this case. FCP_RSP can be received anytime after the transmission of FCP_CMND due to out of order delivery.

Figure C.22 - ACK Lost on Write (Acknowledged Classes)

Error Detection

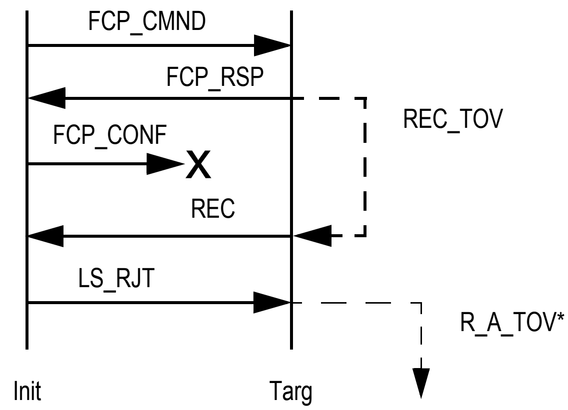


No Error Recovery

The target has received the FCP_DATA Sequence. No error recovery is required.
The BA_ACC indicates the data Sequence was received. The payload is set to SEQ_ID valid, SEQ_ID value = SEQ_ID value of the FCP_DATA Sequence, low SEQ_CNT = high SEQ_CNT = SEQ_CNT of ABTS frame. The target and initiator continue the Exchange.
The initiator establishes its Recovery Qualifier. The resources associated with the Recovery Qualifier can be reclaimed after R_A_TOV. For in-order delivery, the value of R_A_TOV* is zero. The issuance of the RRQ is optional, as no Recovery Qualifier was established by the target. FCP_RSP can be received at any time after the last FCP_DATA frame has been transmitted.

Figure C.23 - FCP_CONF Lost, Unacknowledged Classes

Error Detection

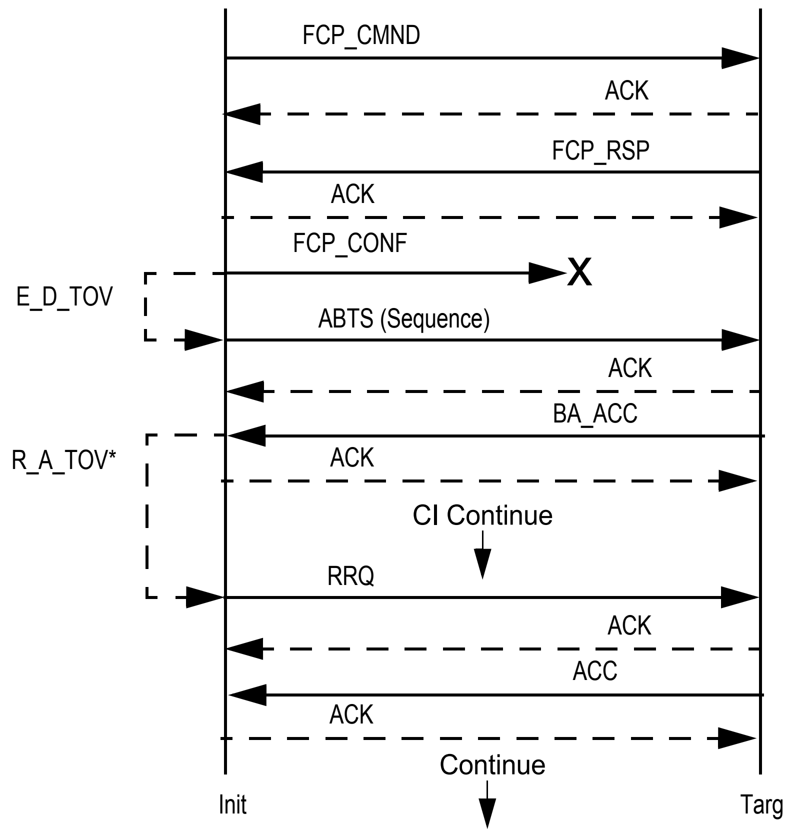


No Error Recovery

The LS_RJT for the REC indicates the Exchange is unknown. That implicitly indicates that the initiator received FCP_RSP and sent FCP_CONF.
The context for the Exchange in the target is preserved for another R_A_TOV to prevent possible aliasing. If FCP_CONF is received after REC is transmitted, it is accepted and discarded and the context for the Exchange can be purged. For in-order delivery, the value of R_A_TOV* is zero.

Figure C.24 - FCP_CONF Lost, Acknowledged Classes

Error Detection



Error Recovery

The BA_ACC indicates that the FCP_CONF was not received. The BA_ACC payload is SEQ_ID valid, SEQ_ID value = SEQ_ID of FCP_CMND, Low SEQ_CNT = 0, High SEQ_CNT = 1. The value of R_A_TOV* for in-order delivery is zero. FCP_CONF is retransmitted with a different SEQ_ID.

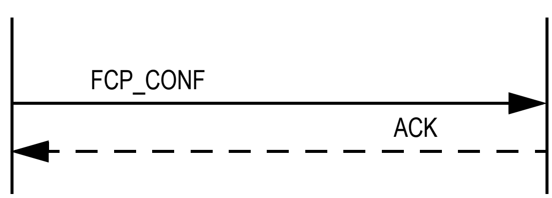
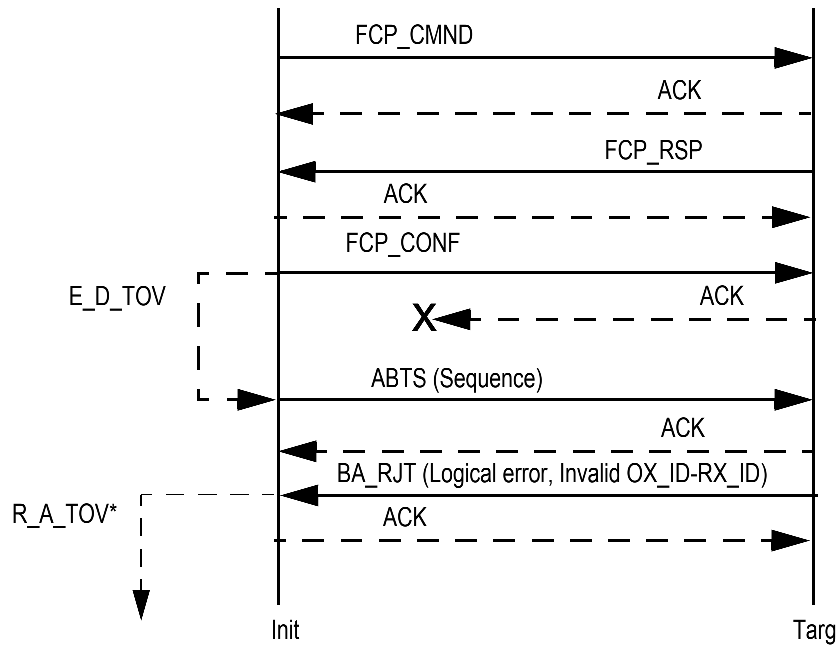


Figure C.25 - ACK lost on FCP_CONF, Acknowledged Classes

Error Detection



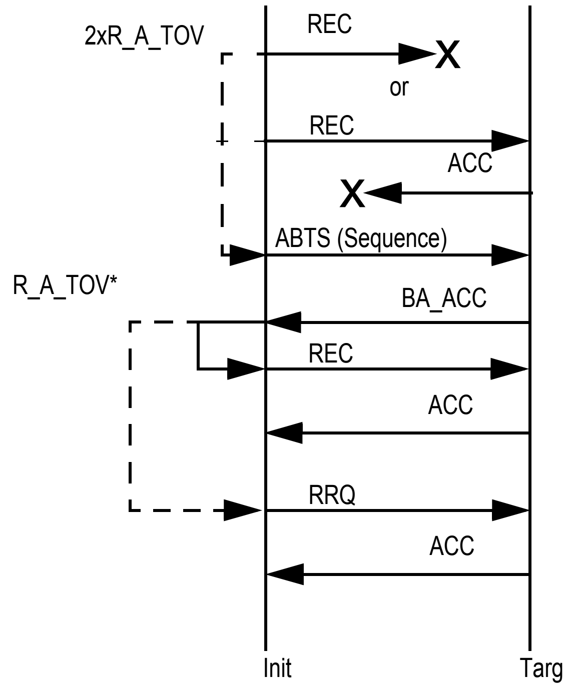
No Error Recovery

BA_RJT is the response to the ABTS, since no context exists for this Exchange and the ABTS was not issued on the first Sequence of a new Exchange. The initiator establishes a Recovery Qualifier on receipt of the BA_RJT. The resources associated with the Recovery Qualifier can be reclaimed when R_A_TOV expires.

Issuance of the RRQ is optional, as no Recovery Qualifier was established by the target.

For in-order delivery, the value of R_A_TOV* is zero.

Figure C.26 - REC or REC Response Lost, Unacknowledged Classes



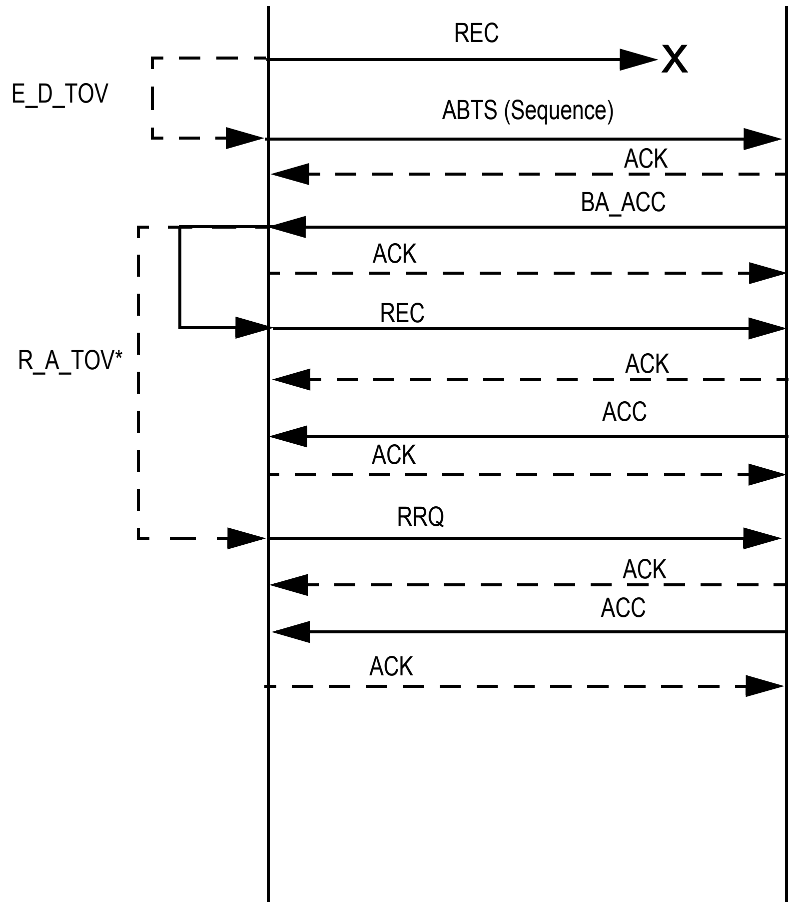
Error Recovery Explanation

For the case of the REC never having been received, the BA_ACC payload is SEQ_ID invalid, low SEQ_CNT = 0, high SEQ_CNT = SEQ_CNT of ABTS = 1.

For the case of the ACC response to REC never having been received, the target would view the ABTS as having been issued on a new Exchange. The BA_ACC payload is SEQ_ID invalid, low SEQ_CNT = high SEQ_CNT = SEQ_CNT of ABTS.

In both cases, a Recovery Qualifier is established. The second REC is issued in a new Exchange. For in-order delivery, the value of R_A_TOV* is zero.

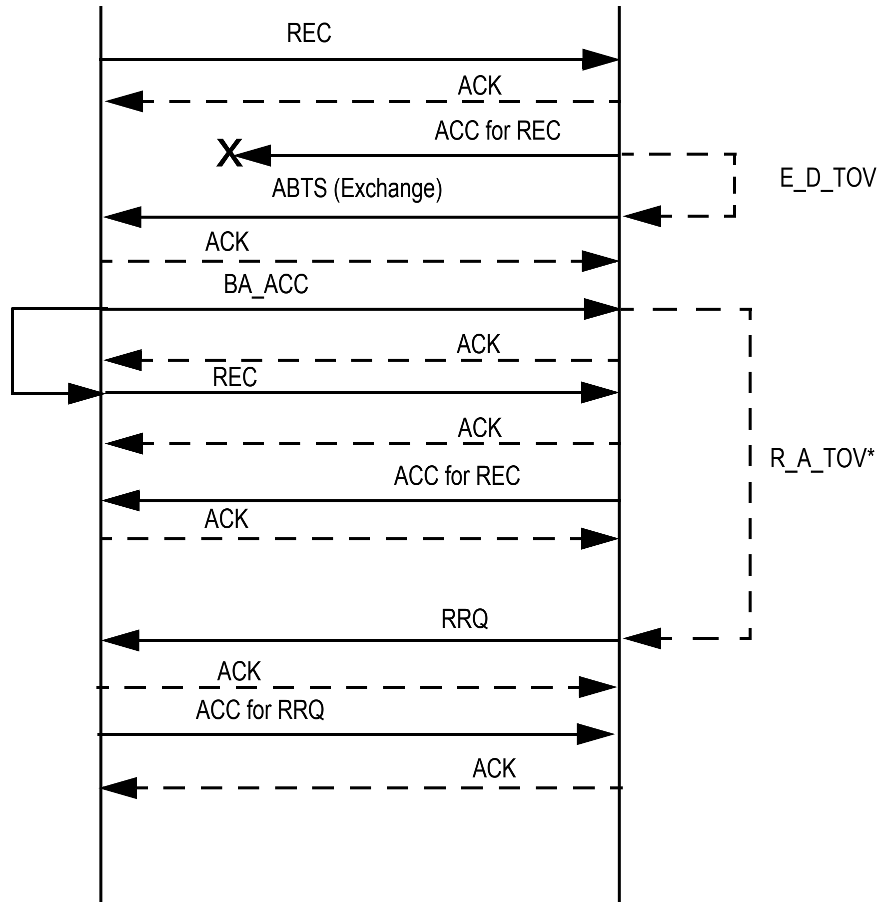
Figure C.27 - REC Lost, Acknowledged Classes



Error Recovery Explanation

The BA_ACC payload indicates that the REC was never received by the target. The payload is SEQ_ID invalid, low SEQ_CNT = 0, high SEQ_CNT = SEQ_CNT in ABTS frame. Recovery Qualifiers are established on each side.
For in-order delivery, the value of R_A_TOV* is zero.
The second REC is issued using a new Exchange.

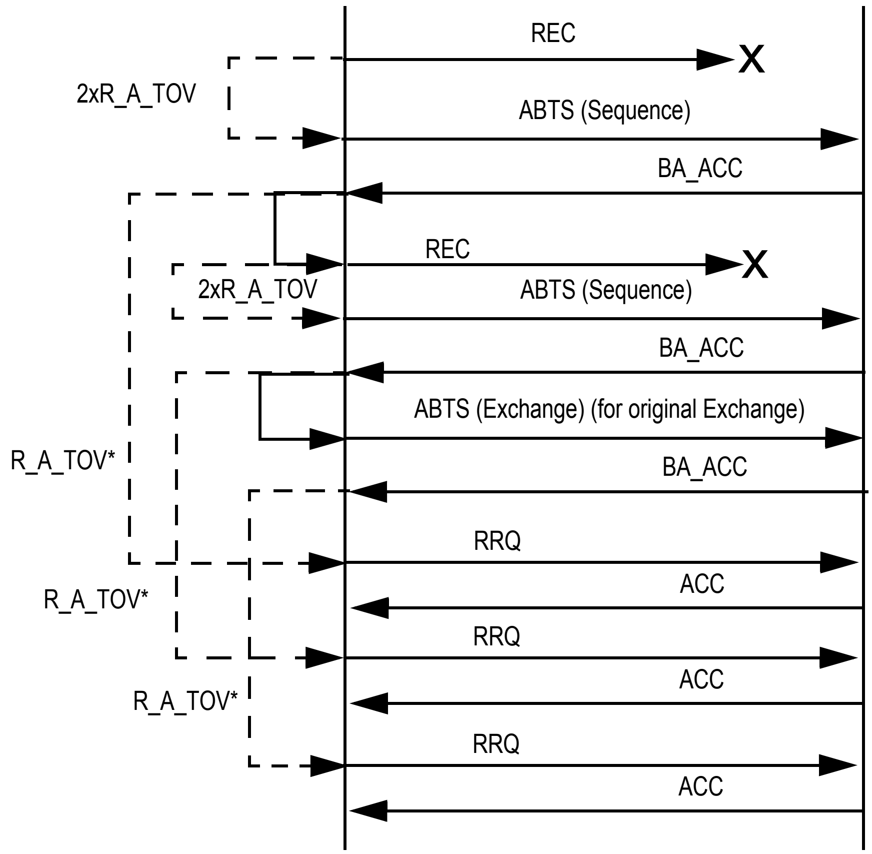
Figure C.28 - REC Response Lost, Acknowledged Classes



Error Recovery Explanation

The BA_ACC payload indicates that the ACC was never received by the initiator. The payload is SEQ_ID invalid, low SEQ_CNT = 0, high SEQ_CNT = SEQ_CNT in ABTS frame.
After responding to the ABTS, the initiator reissues the REC in a new Exchange.
Recovery Qualifiers are established on each side.
For in-order delivery, the value of R_A_TOV* is zero.

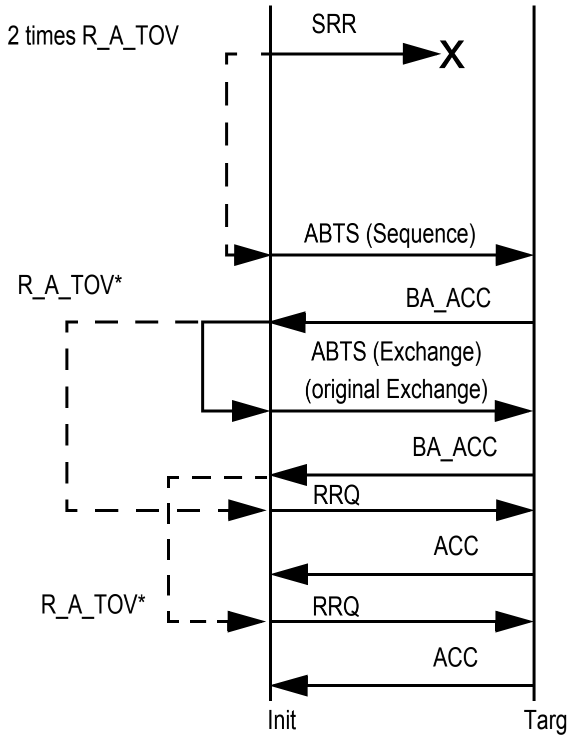
Figure C.29 - Two RECs Lost, Unacknowledged Classes, Abort the original Exchange



Error Recovery Explanation

The failure of two RECs issued against the same Exchange causes all associated Exchanges to be aborted. The ABTS for the original Exchange uses the previous SEQ_ID and a SEQ_CNT one greater than the count used in the previous Sequence and Bit 0 = 0 set in the PARAMETER field. The payload for the BA_ACC is SEQ_ID valid, SEQ_ID = SEQ_ID of last deliverable Sequence received, low SEQ_CNT = 0 and high SEQ_CNT = FFFFh.
Recovery qualifiers are established on both sides for each Exchange.
For in-order delivery, the value of R_A_TOV* is zero.

Figure C.30 - SRR Lost, Unacknowledged Classes, Abort original Exchange

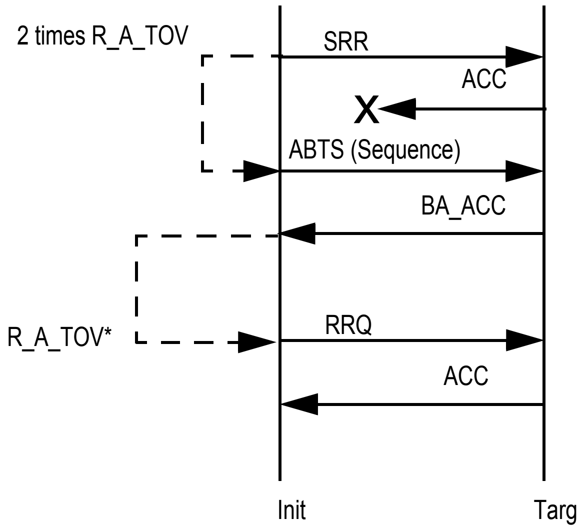


Error Recovery Explanation

The payload for the BA_ACC associated with the ABTS of the SRR is SEQ_ID invalid, low SEQ_CNT = 0, high SEQ_CNT = SEQ_CNT of the ABTS frame.

The ABTS for the original Exchange uses the previous SEQ_ID and a SEQ_CNT one greater than the count used in the previous Sequence and Bit 0 = 0 in the PARAMETER field. The payload for the BA_ACC associated with the ABTS for the original Exchange is SEQ_ID valid, the SEQ_ID = SEQ_ID of the last deliverable Sequence of the original Exchange received, low SEQ_CNT = 0, and high SEQ_CNT = FFFFh.
Recovery Qualifiers are established on both sides for each Exchange. For in-order delivery, the value of R_A_TOV* is zero.

Figure C.31 - SRR Response Lost, Unacknowledged Classes



Error Recovery Explanation

If the SRR Exchange is unknown to the Recipient, the Exchange was completed and the context purged. The payload for the BA_ACC is SEQ_ID invalid, low SEQ_CNT = 0, high SEQ_CNT = FFFFh. Recovery Qualifiers are established on both sides.

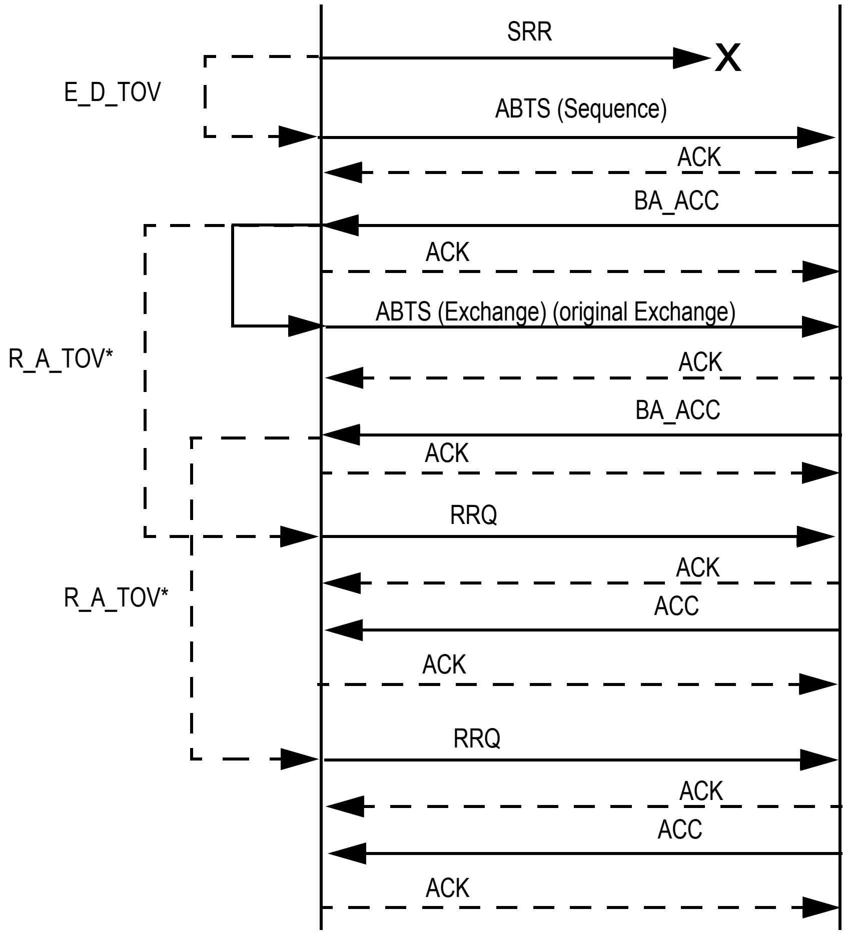
If the SRR Exchange is still known to the Recipient, the payload for the BA_ACC is SEQ_ID valid, SEQ_ID = SEQ_ID of the SRR, low SEQ_CNT = high SEQ_CNT = SEQ_CNT of the ABTS frame. Since no Recovery Qualifier is established, RRQ need not be issued. The Recovery Qualifier is established on the initiator side and is timed out for R_A_TOV.

For in-order delivery, the value of R_A_TOV* is zero.

In either case, the original Exchange need not be aborted.

The RRQ references the Exchange of the SRR.

Figure C.32 - SRR Lost, Acknowledged Classes



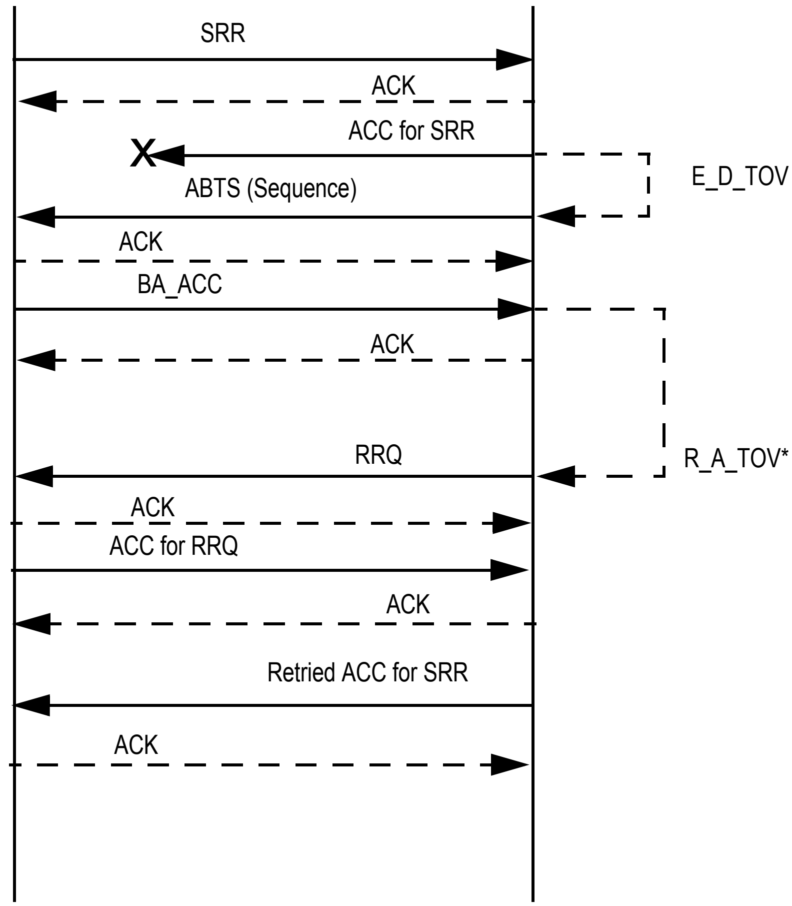
Error Recovery Explanation

Since the ABTS on the SRR is on a new Exchange, Recovery Qualifiers are established. The BA_ACC payload indicates SEQ_ID invalid, low SEQ_CNT = 0, and high SEQ_CNT = SEQ_CNT of the ABTS.

An error on an SRR is a second error and the original Exchange is also aborted. SRR is not retried.

For in-order delivery, the value of R_A_TOV* is zero.

Figure C.33 - SRR Response Lost, Acknowledged Classes



Error Recovery Explanation

The BA_ACC of the ABTS associated with the SRR indicates that the ACC for the SRR was not received and is discarded if it is later received. The BA_ACC payload indicates SEQ_ID invalid, low SEQ_CNT = 0, and high SEQ_CNT = SEQ_CNT of the ABTS.
The retry of the ACC for SRR is issued with a new SEQ_ID.
Recovery Qualifiers are established on each side.
For in-order delivery, the value of R_A_TOV* is zero.

Annex D

(informative)

FCP Device Discovery Procedure

D.1 FCP Device Discovery Procedure

D.1.1 Initiator discovery of Fabric-attached targets

The following procedure may be used by initiators for discovering and authenticating FCP devices in a Fabric switch environment. The FCP device discovery procedure also applies to NL_Ports attached to an FL_Port that provides the Name Server capability described in FC-GS-3. The Node_Name and Port_Name should be Name_Identifiers that are Worldwide_Names, assuring that they are uniquely identifiable. Logical units are also assigned a Worldwide_Name that may be examined using the INQUIRY command with the EVPD bit set to one and the PAGE CODE set to device identification page.

The following steps are all optional. Depending on the specific configuration and the management requirements, any step may be omitted and any step may be performed using actions outside this standard or the referenced standards.

1) Perform FLOGI.
2) Login with the Name Server.
3) Register information with Name Server:
    a) FC-4 TYPEs object (see FC-GS-3);
    b) FC-4 Features object (see 7.2); and
    c) FC-4 Descriptor object (see 7.3).
4) Register for State Change Notification with the Fabric Controller. See FC-FS.
5) Issue a GID_FT query to the Name Server with the Domain_ID Scope and Area_ID Scope fields set to zero, and the FC-TYPE Code set to 08h. This query obtains a list of the Port Identifiers of devices that support the FCP protocol. See FC-GS-3.
6) For each Port Identifier returned in the Accept CT_IU for the GID_FT, perform PLOGI/PRLI and, if the device is determined to be an FCP target, issue an INQUIRY command to LUN 0 to identify the type of target. See 6.2 and SPC-2.
7) If the INQUIRY succeeds, issue a REPORT LUNS command to LUN 0 to obtain a list of the logical units accessible through the target. See SPC-2.
8) Issue an INQUIRY command for each reported LUN to determine the type of device and supported command set for the logical unit. See SPC-2.
9) Issue an INQUIRY command with the EVPD bit set to one and the PAGE CODE set to device identification page for each reported LUN to obtain the logical unit's Worldwide_Name. This allows higher level programs to identify possible redundant paths to a logical unit. See SPC-2.

D.1.2 Initiator discovery of loop-attached targets

The following procedure may be used by initiators for discovering and authenticating FCP devices in a loop environment where no fabric switch is attached to the loop.

The following steps are all optional. Depending on the specific configuration and the management requirements, any step may be omitted and any step may be performed using actions outside this standard or the referenced standards.

1) Obtain a map of the loop or poll all possible addresses if a loop map is not available to identify those devices that are present on the loop. See FC-AL-2.
2) For each loop ID found in step 1, perform PLOGI/PRLI and, if the device is determined to be an FCP target, issue an INQUIRY command to LUN 0. See 6.2 and SPC-2.
3) If the INQUIRY command succeeds, issue a REPORT LUNS command to LUN 0 to identify the logical units supported by the target. See SPC-2.
4) Issue an INQUIRY command with the EVPD bit set to one and the PAGE CODE set to device identification page for each reported LUN to obtain the logical unit's Worldwide_Name. This allows higher level programs to identify possible redundant paths to a logical unit. See SPC-2.

D.2 Fabric and Device Authentication

The following mechanisms are used by any Fibre Channel device to verify its relationship with other devices attached to the fabric. Such verification may be required after initialization or other temporary fabric disturbances. The following steps are all optional. Depending on the specific configuration and the management requirements, any step may be omitted by any of the attached devices and any step may be performed using actions outside this standard or the referenced standards.

1) N_Ports or NL_Ports retain the Fabric Port Name and Fabric Name of the Fabric from the information exchanged during FLOGI and associate that information with the Loop Fabric Address of that Fabric. This information is retained by the N_Port or NL_Port for as long as the login with the Fabric is active. See FC-FS.
2) All N_Ports and NL_Ports, including initiators and targets, validate the current Fabric login following every Loop Initialization by comparing the Loop Fabric Address, Fabric Port Name, and Fabric Name received during FLOGI with those reported by the FAN performed during the initialization. If all three identifiers reported by the FAN do not match the values reported during FLOGI, a configuration change has occurred and an explicit LOGO is performed and all open Exchanges are terminated. See 4.9 and FC-FLA.
3) N_Ports and NL_Ports retain the Node Name and Port Name of the other port from each PLOGI and associate that information with the Address Identifier of that port. This information is retained for as long as the PLOGI with the other port is active. See FC-FS.
4) Initiators and targets validate N_Port and NL_Port logins following every Loop Initialization by comparing the Port Name, Node Name, and Address Identifier received during the PLOGI with those reported by the Name Server (see FC-GS-3) or the ADISC/ACC (see FC-AL-2) that follows loop initialization. If all three identifiers reported by the Name Server or ADISC/ACC do not match the values reported during PLOGI, a configuration change has occurred and a LOGO is performed and all open Exchanges with that initiator or target are terminated. See 4.9 and FC-FLA.

D.3 Logical unit authentication

A logical unit's identity is optionally verified and monitored by performing an INQUIRY command with the EVPD bit set to one and the PAGE CODE set to device identification page to obtain the logical unit's Worldwide_Name. The same Worldwide_Name is presented by a logical unit regardless of the port by which it is accessed and the value of the LUN field used to access it.

Annex E

(informative)

FCP-2 examples of link service usage

E.1 Formats for recovery link services

Examples of the formats for recovery ELSs are described in this annex.

E.2 Abort Sequence (ABTS) Request

E.2.1 Abort Sequence (ABTS) Request fields

The initiator or target may transmit an ABTS Frame. When it does so, the specified fields should be set as shown in table E.1.

Table E.1 - ABTS Frame

	Field	Sub-field	Content
Frame Header	F_CTL	Sequence Context bit	Sequence Initiator (even though the ABTS Initiator may not have Sequence Initiative for the Sequence being aborted).
		Sequence Initiative bit	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 or SEQ_CNT of last Frame transmitted + 1.
	OX_ID		OX_ID = same as that assigned by the Exchange Originator (initiator) for the Exchange being aborted.
	RX_ID		Set to FFFFh or the same as that assigned by the Exchange Responder (target) for the task being aborted.
	PARAMETER	Bit 0 = 0 Bit 0 = 1	Abort Exchange Abort Sequence

E.2.2 Basic Accept (BA_ACC) Frame to ABTS

An initiator or target may accept ABTS with BA_ACC. When it does so, the BA_ACC should be as shown in table E.2.

Table E.2 - 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 bit	Set to one for Abort Exchange or Set to zero for Abort Sequence
		Sequence Context bit	Recipient
Payload	SEQ_ID validity		Set to 00h for Abort Exchange or Set to 80h for Abort Sequence
	SEQ_ID byte		Invalid (don’t care) for Abort Exchange or Set to SEQ_ID of last deliverable Sequence received from ABTS Initiator for Abort Sequence
	OX_ID		OX_ID from ABTS Frame
	RX_ID		RX_ID from ABTS Frame
	Lowest SEQ_CNT		Set to 0000h for Abort Exchange or Refer to FC-FS for Abort Sequence
	Highest SEQ_CNT		Set to FFFFh for Abort Exchange or Set to SEQ_CNT of ABTS Frame for Abort Sequence

E.2.3 Basic Reject (BA_RJT) Frame to ABTS

A target may reject ABTS with BA_RJT. When it does, the BA_RJT should be as shown in table E.3.

Table E.3 - 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 bit	1
		Sequence Context bit	Recipient
Payload	Byte 0	Reserved	00h
	Byte 1	Reason Code	refer to FC-FS
	Byte 2	Reason Explanation	refer to FC-FS
	Byte 3	Vendor Unique	00h

E.3 Reinstate Recovery Qualifier (RRQ)

E.3.1 RRQ request format

The format of the RRQ is shown in table E.4.

Table E.4 - Reinstate Recovery Qualifier

	Field	Content
Frame Header	OX_ID	Identifier of a new Exchange
	RX_ID	FFFFh
Payload	Originator S_ID	Source_ID of the 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 target responds with ACC.

Annex F

(informative)

Bidirectional operation support

F.1 Introduction

This annex describes how bidirectional commands are supported by FCP. Bidirectional commands transfer data in both read (data in) and write (data out) directions within a single command. The changes to the FCP_CMND IU and FCP_RSP IU necessary to support bidirectional commands are described in this annex. Additional information on FCP_DATA IU handling, error recovery, and some examples are also included. This annex is informative. Future versions of the Fibre Channel Protocol for SCSI may incorporate or modify the information in this annex.

F.2 Changes in the FCP device management model

F.2.1 Support of bidirectional operation

The description of the device management model contained in 4.2 is expanded to include bidirectional operation by including the following paragraph after the fourth paragraph of the present sub-clause:

When the device server for the command has completed the interpretation of the command and has determined that bidirectional transfer is required, it selects the first FCP_DATA IU to be transferred. The IU may be either a data in or a data out transfer. If the device server chooses to request a data out transfer first, it sends a data descriptor IU containing the FCP_XFER_RDY IU payload to the initiator to indicate which portion of the data is to be transferred. The FCP_Port that is the initiator then transmits the solicited data IU to the target containing the FCP_DATA IU payload requested by the FCP_XFER_RDY IU. The FCP_XFER_RDY IU and FCP_DATA IU payloads constitute the Receive Data-Out protocol service request and Data-Out Received service confirmation described in SAM-2. If the device server chooses to send a data in transfer first, the FCP_Port that is the target transmits a solicited data IU to the initiator containing the FCP_DATA IU payload. The FCP_DATA IU constitutes the Send Data-In protocol service request described in SAM-2. The device server then selects the next FCP_DATA IU to be transmitted and performs the appropriate procedure to transmit. Data deliveries continue until all data described by the SCSI command is transferred. This standard places no restrictions on the order in which the device server performs data in and data out transfer operations. If the system has mechanisms outside the scope of this standard for controlling the data transfer length, the transmission of the initial FCP_XFER_RDY IU for the data out transfers may be disabled.

F.2.2 Relationship between bidirectional and unidirectional operation

A new subclause is added after the device management model contained in 4.2 to indicate how bidirectional and unidirectional operation interoperate. The following text is added to document that information.

A device server that supports bidirectional operation implements many commands as unidirectional commands as well as some commands that provide for bidirectional transfer. Two FCP_RSP IU formats are defined. For commands that specify bidirectional transfer by setting both the RDDATA and WRDATA bits to one, the bidirectional FCP_RSP IU payload shall be used for presenting all status and error conditions. For commands that specify either no transfer or unidirectional transfer by setting either the RDDATA or WRDATA bit or both to zero, the unidirectional FCP_RSP IU payload shall be used for presenting all status and error conditions. The format of the FCP_RSP IU that is returned depends only on the state of the RDDATA and WRDATA bits and is not influenced by whether the command itself requests bidirectional behavior.

A device server that does not support bidirectional operation shall use the unidirectional FCP_RSP IU payload for presenting all status and error conditions. If a device server that does not support bidirectional operation receives a command that requests bidirectional operation by setting both the RDDATA and WRDATA bits to one, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of ILLEGAL FIELD IN PARAMETER LIST.

F.3 FCP_CMND IU changes

F.3.1 FCP_CMND IU payload

For a bidirectional command, the FCP_CMND IU is extended by four bytes to include an FCP_BIDIRECTIONAL_READ_DL field after the FCP_DL field. The format of the FCP_CMND IU for bidirectional commands is shown in table F.1.

Table F.1 - FCP_CMND payload for a bidirectional command

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)							
7	FCP_LUN							(LSB)
8	COMMAND REFERENCE NUMBER							
9	RESERVED					TASK ATTRIBUTE		
10	TASK MANAGEMENT FLAGS							
11	RESERVED	ADDITIONAL FCP_CDB LENGTH = (N-27)/4					RDDATA	WRDATA
12	(MSB)							
27	FCP_CDB							(LSB)
28	(MSB)							
n	ADDITIONAL FCP_CDB							(LSB)
n+1	(MSB)							
n+2	FCP_DL							
n+3								
n+4								(LSB)
n+5	(MSB)							
n+6	FCP_BIDIRECTIONAL_READ_DL							
n+7								
n+8								(LSB)

The fields contained in the FCP_CMND IU payload are unchanged in definition and usage from 9.1 except as noted in F.3.2 through F.3.5.

F.3.2 TASK MANAGEMENT FLAGS

If any bit in the TASK MANAGEMENT FLAG field is set to one, the FCP_BIDIRECTIONAL_READ_DL field is not included in the FCP_CMND IU payload. All other task management flag text remains unchanged. See 9.1.2.4.

F.3.3 RDDATA and WRDATA

For a bidirectional command, both RDDATA and WRDATA are set to one. This indicates that a bidirectional command is being performed and that an FCP_BIDIRECTIONAL_READ_DL field is included in the FCP_CMND payload.

All other text of 9.1.2.6 and 9.1.2.7 remains unchanged.

F.3.4 FCP_DL

For a bidirectional command, the FCP_DL field contains a count of the greatest number of data bytes expected to be transferred from the application client data buffer by the SCSI command. The parameter is the data-out buffer size defined by SAM-2.

F.3.5 FCP_BIDIRECTIONAL_READ_DL

If both the RDDATA and WRDATA bits are set to one, the FCP_BIDIRECTIONAL_READ_DL field follows the FCP_DL field. The FCP_BIDIRECTIONAL_READ_DL field contains a count of the greatest number of data bytes expected to be transferred to the application client data buffer by the command. The parameter is the data-in buffer size defined by SAM-2. An FCP_BIDIRECTIONAL_READ_DL value of zero indicates that no read data transfer is expected regardless of the state of the RDDATA bit and that no FCP_DATA IUs are transferred for read data.

If either RDDATA or WRDATA is set to zero, the FCP_BIDIRECTIONAL_READ_DL field is not included in the FCP_CMND IU payload.

F.4 FCP_DATA IU changes

During any write data transfer for a bidirectional command (i.e., an operation that uses Data Out actions, IUs T6 or T7), the initiator always has available a buffer of the length specified by the FCP_DL field containing data to be transferred to the target.

During any read data transfer for a bidirectional command (i.e., an operation that uses the Data In action, IU I3), the initiator always has available a buffer of the length specified by the FCP_BIDIRECTIONAL_READ_DL field that receives the data.

If a command requests that data beyond FCP_DL be transferred, the FCP_RESID_OVER bit is set to one in the FC_RSP IU. The command is completed normally except that data beyond the count in the FCP_DL field is not transferred and the appropriate overrun condition is presented. See 9.4.4.

If a command requests that data beyond the value specified in the FCP_BIDIRECTIONAL_READ_DL field be transferred, the FCP_BIDIRECTIONAL_READ_RESID_OVER bit is set to one in the FCP_RSP IU. The command is completed normally except that data beyond the FCP_BIDIRECTIONAL_READ_DL count is not transferred and the appropriate overrun condition is presented. See F.5.4.

If the amount of data transferred does not match the value specified by the FCP_DL field as modified by any residual values recorded in the FCP_RESID field for the write data transfer of a bidirectional command or the value specified by the FCP_BIDIRECTIONAL_READ_DL field as modified by any residual values recorded in the FCP_BIDIRECTIONAL_READ_RESID field for the read data transfer of a bidirectional command, the error detection and recovery procedure described in clause 12 may be invoked or the FCP I/O operation may be terminated with a recovery abort or other failure indication.

There are no other changes in the definition or usage of the FCP_DATA IU as described in 9.3.

F.5 FCP_RSP IU changes

F.5.1 FCP_RSP IU payload

For a bidirectional command, the FCP_RSP IU contains two new bits and one new field. The format of the FCP_RSP IU is shown in table F.2.

Table F.2 - FCP_RSP IU Payload

Byte \ Bit	7	6	5	4	3	2	1	0
0	RESERVED							
7								
8	RESERVED							
9	RESERVED							
10	FCP_BIDI_RSP	FCP_BIDI_READ_RESID_UNDER	FCP_BIDI_READ_RESID_OVER	FCP_CONF_REQ	FCP_RESID_UNDER	FCP_RESID_OVER	FCP_SNS_LEN_VALID	FCP_RSP_LEN_VALID
11	SCSI STATUS CODE							
12	(MSB)	FCP_RESID						
15								(LSB)
16	(MSB)	FCP_SNS_LEN (= N)						
19								(LSB)
20	(MSB)	FCP_RSP_LEN (= M)						
23								(LSB)
24	(MSB)	FCP_RSP_INFO (M BYTES LONG)						
23+m								(LSB)
24+m	(MSB)	FCP_SNS_INFO (N BYTES LONG)						
23+m+n								(LSB)
24+m+n	(MSB)	FCP_BIDIRECTIONAL_READ_RESID						
27+m+n								(LSB)

The fields contained in the FCP_RSP IU payload are unchanged in definition and usage from 9.4 except as noted in F.5.3 through F.5.6.

F.5.2 FCP_BIDI_RSP

An FCP_BIDI_RSP bit of one indicates that the FCP_RSP IU uses the format specified in table F.2. An FCP_BIDI_RSP bit of zero indicates that the FCP_RSP IU uses the format specified in table 23.

F.5.3 FCP_BIDI_READ_RESID_UNDER

An FCP_BIDI_READ_RESID_UNDER bit of one indicates that the FCP_BIDIRECTIONAL_READ_RESID field is valid and contains the count of bytes that were expected to be transferred, but were not transferred. The application client should examine the FCP_BIDIRECTIONAL_READ_RESID field in the context of the command to determine whether or not an error condition occurred.

F.5.4 FCP_BIDI_READ_RESID_OVER

An FCP_BIDI_READ_RESID_OVER bit of one indicates that the FCP_BIDIRECTIONAL_READ_RESID field is valid and contains the count of bytes that could not be transferred because the FCP_BIDIRECTIONAL_READ_DL was not sufficient. The application client should examine the FCP_BIDIRECTONAL_READ_RESID field in the context of the command to determine whether or not an error condition occurred.

F.5.5 FCP_RESID

If the FCP_RESID_UNDER bit or the FCP_RESID_OVER bit is set to one, the FCP_RESID field contains a count of the number of residual data bytes that were not transferred in the FCP_DATA IUs for this SCSI command. For bidirectional commands, the FCP_RESID field contains a count of the number of residual data bytes that were not transferred in the data out FCP_DATA IUs. For bidirectional commands, the FCP_BIDIRECTIONAL_READ_RESID field contains the corresponding count for data in FCP_DATA IUs. Upon successful completion of a FCP I/O operation, the residual value is normally zero and the FCP_RESID value is not valid. FCP devices having indeterminate data lengths may have a nonzero residual byte count after completing valid operations. Targets are not required to verify that the data length implied by the contents of the CDB cause an overrun or underrun before beginning execution of an SCSI command.

If the FCP_RESID_UNDER bit is set to one, a transfer that did not fill the buffer to the expected displacement FCP_DL was performed and the value of FCP_RESID is defined as follows:

FCP_RESID = FCP_DL - (highest offset of any byte transmitted + 1)

A condition of FCP_RESID_UNDER may not be an error for some FCP devices and some commands.

If the FCP_RESID_OVER bit is set to one, 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 be transferred. The FCP_RESID is defined as follows:

FCP_RESID = (Transfer length required by command) - FCP_DL

If a condition of FCP_RESID_OVER is detected, the termination state of the FCP I/O operation is not certain. Data may or may not have been transferred and the SCSI status byte may or may not provide correct command completion information.

If the FCP_RESID_UNDER and the FCP_RESID_OVER bits are set to zero, the FCP_RESID field is not meaningful and may have any value. The FCP_RESID field is always included in the FCP_RSP IU.

NOTE F.1 - Some early target implementations presented the FCP_RSP IU without the FCP_RESID, FCP_SNS_LEN, and FCP_RSP_LEN fields if the FCP_RESID_UNDER, FCP_RESID_OVER, FCP_SNS_LEN_VALID, and FCP_RSP_LEN_VALID bits were all set to zero. Initiators should be tolerant of this non-standard behavior.

F.5.6 FCP_BIDIRECTIONAL_READ_RESID

The FCP_BIDIRECTIONAL_READ_RESID field is included in the FCP_RSP IU for all bidirectional commands. If either the FCP_BIDI_READ_RESID_UNDER bit or the FCP_BIDI_READ_RESID_OVER bit is one, the FCP_BIDIRECTIONAL_READ_RESID field contains a count of the number of residual data bytes that were not transferred in the read FCP_DATA IUs for this bidirectional SCSI command. Upon successful completion of a FCP I/O operation, the residual value is normally zero and the FCP_BIDIRECTIONAL_READ_RESID value is not valid. FCP devices having indeterminate data lengths may have a nonzero residual byte count after completing valid

operations. Targets are not required to verify that the data length implied by the contents of the CDB cause an overrun or underrun before beginning execution of a SCSI command.

If the FCP_BIDI_READ_RESID_UNDER bit is set to one, a transfer that did not fill the buffer to the expected displacement FCP_BIDIRECTIONAL_READ_DL was performed and the value of FCP_BIDIRECTIONAL_READ_RESID is defined as follows:

FCP_BIDIRECTIONAL_READ_RESID = FCP_BIDIRECTIONAL_READ_DL - (highest offset of any byte written + 1)

A condition of FCP_BIDI_READ_RESID_UNDER may not be an error for some FCP devices and some commands.

If the FCP_BIDI_READ_RESID_OVER bit is set to one, the transfer was truncated because the data transfer required by the SCSI command extended beyond the displacement value of FCP_BIDIRECTIONAL_READ_DL. Those bytes that could be transferred without violating the FCP_DL value may be transferred. The FCP_BIDIRECTIONAL_READ_RESID is defined as follows:

FCP_BIDIRECTIONAL_RESID = (Read transfer length required by command) - FCP_BIDIRECTIONAL_READ_DL

If a condition of FCP_BIDI_READ_RESID_OVER is detected, the termination state of the FCP I/O operation is not certain. Data may or may not have been transferred and the SCSI status byte may or may not provide correct command completion information.

If both the FCP_BIDI_READ_RESID_UNDER and the FCP_BIDI_READ_RESID_OVER bits are zero, the FCP_BIDIRECTIONAL_READ_RESID field is not meaningful and may have any value.

F.6 Error recovery changes

F.6.1 Overview

The change proposed in F.6.2 is added to 12.1.2.

The changes proposed in F.6.3 is added to 12.2.2.

F.6.2 Sequence level error recovery

Sequence level recovery is not used for bidirectional commands.

F.6.3 FCP-2 Error Detection using protocol errors for all classes of service

The Exchange originator (initiator) detects the following errors for bidirectional commands.

a) a bidirectional command completed with the write data count smaller than FCP_DL and FCP_RESID_UNDER is set to zero;

b) a bidirectional command completed with the read data count smaller than FCP_BIDIRECTIONAL_READ_DL and FCP_BIDI_READ_RESID_UNDER is set to zero;

c) a bidirectional command completed with the write data count smaller than FCP_DL, FCP_RESID_UNDER is set to one, and the write data count plus FCP_RESID is not equal to FCP_DL; and,

d) a bidirectional command completed with the read data count smaller than FCP_BIDIRECTIONAL_READ_DL, FCP_BIDI_READ_RESID_UNDER is set to one, and the read data count plus FCP_BIDIRECTIONAL_READ_RESID is not equal to FCP_BIDIRECTIONAL_READ_DL.

F.7 FCP Examples

F.7.1 Overview

The text and tables in F.7.2 through F.7.5 are expected to be added to the examples in Annex B.

F.7.2 SCSI FCP bidirectional command with write before read

A typical SCSI FCP bidirectional command with a single data IU transferred in each direction is shown in table F.3. The command in the example accepts write data before returning read data.

Table F.3 - FCP bidirectional command with write before read, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND →	
		[Prepare data out transfer buffer]
	← I1, FCP_XFER_RDY	Data out delivery request
Data out action	T6, FCP_DATA →	
		[Prepare data in transfer]
	← I3, FCP_DATA	Data in action
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		

F.7.3 SCSI FCP bidirectional command with read before write

A typical SCSI FCP bidirectional command with a single data IU transferred in each direction is shown in table F.4. The example command returns read data before accepting write data.

Table F.4 - FCP bidirectional command with read before write, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND →	
		[Prepare data in transfer]
	← I3, FCP_DATA	Data in action
		[Prepare data out transfer buffer]
	← I1, FCP_XFER_RDY	Data out delivery request
Data out action	T6, FCP_DATA →	
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		

F.7.4 SCSI FCP bidirectional command, write first, write FCP_XFER_RDY disabled

A SCSI FCP bidirectional command with two write data IUs and one read data IU is shown in table F.5. The example command accepts write data before returning read data. The initial write FCP_XFER_RDY IU has been disabled during process login.

Table F.5 - FCP bidirectional command, write FCP_XFER_RDY disabled, example

Initiator function	IU	Target function
Command request	T2, FCP_CMND →	
Data out action	T6, FCP_DATA →	First data out
	← I1, FCP_XFER_RDY	Second data out delivery request
Data out action	T6, FCP_DATA →	
	← I1, FCP_XFER_RDY	Last Data out delivery request
Data out action	T6, FCP_DATA →	
		[Prepare data in transfer]
	← I3, FCP_DATA	Data in action
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		

F.7.5 SCSI FCP bidirectional command with intermixed writes and reads

A SCSI FCP bidirectional command with three data IUs transferred in each direction is shown in table F.6. The example command accepts some write data before returning read data, but intermixes writes and reads thereafter.

Table F.6 - FCP bidirectional command with intermixed writes and reads, example

Initiator function	IU	Target function
Command request	T1, FCP_CMND →	
		[Prepare data out buffer]
	← I1, FCP_XFER_RDY	First data out delivery request
Data out action	T6, FCP_DATA →	
		[Prepare data in transfer]
	← I3, FCP_DATA	First data in action
	← I1, FCP_XFER_RDY	Second data out delivery request
Data out action	T6, FCP_DATA →	
	← I1, FCP_XFER_RDY	Last data out delivery request
Data out action	T6, FCP_DATA →	
	← I3, FCP_DATA	Second data in action
	← I3, FCP_DATA	Last data in action
		[Prepare response message]
	← I4, FCP_RSP	Response
[Indicate command completion]		