Working Draft

T10 T10/1557-D

Revision 14
18 November 2004

Information technology - Automation/Drive Interface - Transport Protocol (ADT)

This is an internal working document of T10, a Technical Committee of Accredited Standards Committee INCITS (InterNational Committee for Information Technology Standards). As such this is not a completed standard and has not been approved. The contents may be modified by the T10 Technical Committee. The contents are actively being modified by T10. This document is made available for review and comment only.

T10 Technical Editor:

Paul Entzel
Quantum Corporation
4001 Discovery Drive
Boulder, CO 80303
USA

Telephone: 720-406-5782
Facsimile: 720-406-5762
Email: paul.entzel@quantum.com

Reference number
ISO/IEC XXXXX-XXX : 200x
ANSI NCITS.*:200x**

Printed November 18, 2004

Points of Contact:

T10 Chair
John B. Lohmeyer
LSI Logic
4420 Arrows West Drive
Colorado Springs, CO 80907-3444
Tel: (719) 533-7560
Fax: (719) 533-7183
Email: lohmeyer@t10.org

T10 Vice-Chair
George O. Penokie
IBM
3605 Highway 52 N
MS: 2C6
Rochester, MN 55901
Tel: (507) 253-5208
Fax: (507) 253-2880
Email: gop@us.ibm.com

INCITS Secretariat
INCITS Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005
Telephone: 202-737-8888
Facsimile: 202-638-4922
Email: INCITS@itic.org

T10 Web Site www.t10.org

T10 Reflector To subscribe send e-mail to majordomo@T10.org with 'subscribe' in message body
To unsubscribe send e-mail to majordomo@T10.org with 'unsubscribe' in message body
Internet address for distribution via T10 reflector: T10@T10.org

Document Distribution

INCITS Online Store
managed by Techstreet
1327 Jones Drive
Ann Arbor, MI 48105
http://www.techstreet.com/INCITS.html
Telephone: 1-734-302-7801 or
1-800-699-9277
Facsimile: 1-734-302-7811

or

Global Engineering
15 Inverness Way East
Englewood, CO 80112-5704
http://global.ihs.com/
Telephone: 1-303-792-2181 or
1-800-854-7179
Facsimile: 1-303-792-2192

ANSI (r)
NCITS.***:200x

Draft

American National Standards for Information Systems -

Automation/Drive Interface - Transport Protocol (ADT)

Secretariat
InterNational Committee for Information Technology Standards

Approved mm dd yy

American National Standards Institute, Inc.

Abstract

This standard specifies the transport requirements for the SCSI automation drive interface device. This standard permits the SCSI automation drive interface devices to attach to application clients and provides the definitions for their use.

This standard does not contain material related to any command structure for automation drive interface devices, that is used in conjunction with this standard. For reference to command structure, refer to ADC.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered and that effort be made toward their resolution.

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

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

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

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 patent holders have, 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.

Published by
American National Standards Institute
11 West 42nd Street, New York, NY 10036

Printed in the United States of America

Contents

Tables

Figures

Foreword

This foreword is not part of American National Standard NCITS.***:200x.

This draft specification covers the Automation Drive Interface - Transport Protocol. The ADI working group addressed the interface between removable media library controllers and the physical drives resident in those libraries. This specific document covers the transport mechanisms of that interface, specifically the encapsulation, logical transmission, and end-point delivery and reception of the commands associated with the ADI effort.

With any technical document there may arise questions of interpretation as new products are implemented. INCITS has established procedures to issue technical opinions concerning the standards developed by INCITS. These procedures may result in SCSI Technical Information Bulletins being published by INCITS.

These Bulletins, while reflecting the opinion of the Technical Committee that developed the standard, are intended solely as supplementary information to other users of the standard. This standard, ANSI NCITS.***:200x, as approved through the publication and voting procedures of the American National Standards Institute, is not altered by these bulletins. Any subsequent revision to this standard may or may not reflect the contents of these Technical Information Bulletins.

Current INCITS practice is to make Technical Information Bulletins available through:

INCITS Online Store http://www.techstreet.com/INCITS.html
managed by Techstreet Telephone: 1-734-302-7801 or
1327 Jones Drive 1-800-699-9277
Ann Arbor, MI 48105 Facsimile: 1-734-302-7811

or

Global Engineering http://global.ihs.com/
15 Inverness Way East Telephone: 1-303-792-2181 or
Englewood, CO 80112-5704 1-800-854-7179
Facsimile: 1-303-792-2192

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, National Committee for Information Technology Standards, Information Technology Institute, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

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

(Editor's Note: Insert INCITS member list)

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
Dan Colegrove
Roger Cummings
Zane Daggett
Claudio DeSanti
Rob Elliott
Paul Entzel
Mark Evans
Ashlie Fan
Mike Fitzpatrick
Bill Galloway
Robert Griswold
Nathan Hastad
David Hawks
Emily Hill
Gerald Houlder
Skip Jones
James A. Lott, Jr.
Kevin Marks
Ron Mathews
William P. McFerrin
Jay Neer
Terence J. Nelson
Robert H. Nixon
Vit Novak
Erich Oetting
Gary S. Robinson
Robert Sheffield
Robert Snively
Hiroshi Suzuki
Tim Symons
Pat Thaler
Douglas Wagner
Neil Wanamaker
Michael Wingard

I. Dal Allan (Alt)
Dexter Anderson (Alt)
Dennis Appleyard (Alt)
Charles Binford (Alt)
David Black (Alt)
Craig W. Carlson (Alt)
Doug Cole (Alt)
Jim Coomes (Alt)
Martin Czekalski (Alt)
Joe Dambach (Alt)
Dan Dawiedczyk (Alt)
Marc Dupuis (Alt)
Werner Glinka (Alt)
William Ham (Alt)
Rob Haydt (Alt)
Randall C. Hines (Alt)
Kenneth Hirata (Alt)
Keith Holt (Alt)
Jim Jones (Alt)
Jerry Kachlic (Alt)
Robert Kando (Alt)
Tasuku Kasebayashi (Alt)
Jim Koser (Alt)
Ben-Koon Lin (Alt)
Tim Mackley (Alt)
Fabio Maino (Alt)
John Majernik (Alt)
Ron Martin (Alt)
Andy Nowak (Alt)
Nathan Obr (Alt)
Sumi Puri (Alt)
John P. Scheible (Alt)
Phil Shelton (Alt)
Cris Simpson (Alt)
Himmeler Themistocle (Alt)
John Tyndall (Alt)
Rudolf Vitti (Alt)
Dean Wallace (Alt)
Steven Wilson (Alt)
Steve Wong (Alt)

Introduction

The Automation/Drive Interface - Transport Protocol (ADT) standard is divided into eight clauses:

Clause 1 is the scope.
Clause 2 enumerates the normative references that apply to this standard.
Clause 3 describes the definitions, symbols, and abbreviations used in this standard.
Clause 4 describes the conceptual relationship between this document and the SAM-2. It also describes concepts that cross boundaries between protocols supported by this transport.
Clause 5 describes the physical layer including connectors and signal levels.
Clause 6 describes the link layer including encoding, frame format, and link services functions.
Clause 7 describes the transport layer and includes the method of encapsulating SCSI.
Clause 8 describes the mapping between SCSI protocol services defined in SAM-2 and the services provided by this protocol.

The annexes provide information to assist with implementation of this standard.

American National Standard **INCITS.***:200x**

American National Standard for Information Systems - Information Technology - Automation/Drive Interface - Transport Protocol (ADT)

1 Scope

This standard defines the protocol requirements of the Automation/Drive Interface - Transport Protocol to allow conforming ADI SCSI devices to inter-operate. The objectives of ADT are:

a) To provide a low cost interconnect method between an automation device and the data transfer devices that reside within the media changer (see SMC-2). To standardize this interface such that different disk drives, tape drives, optical media drives, and other SCSI devices may be added to conforming media changers without requiring modifications to generic system hardware; and
b) To provide for the addition of special features and functions through the use of vendor-specific options,. reserved areas are provided for future standardization.

The interface protocol includes provision for the connection of two SCSI ports. One of these ports is intended to be attached to a media changer device and may operate either as a SCSI initiator port or a SCSI initiator/target port. The other device is intended to be attached to a data transport type device (i.e. a disk drive, tape drive, or optical medium device) and may operate as either a SCSI target port or SCSI initiator/target port. No provision is made for connection of more than two ports.

This standard defines the transport attributes of an input/output Automation/Drive Interface for interconnecting a conforming media changer device to a conforming data transport device.

The set of SCSI standards specifies the interfaces, functions, and operations necessary to ensure interoperability between conforming SCSI implementations. This standard is a functional description. Conforming implementations may employ any design technique that does not violate interoperability.

Figure 1 is intended to show the general structure of SCSI standards. Figure 1 is not intended to imply a relationship such as a hierarchy, protocol stack, or system architecture.

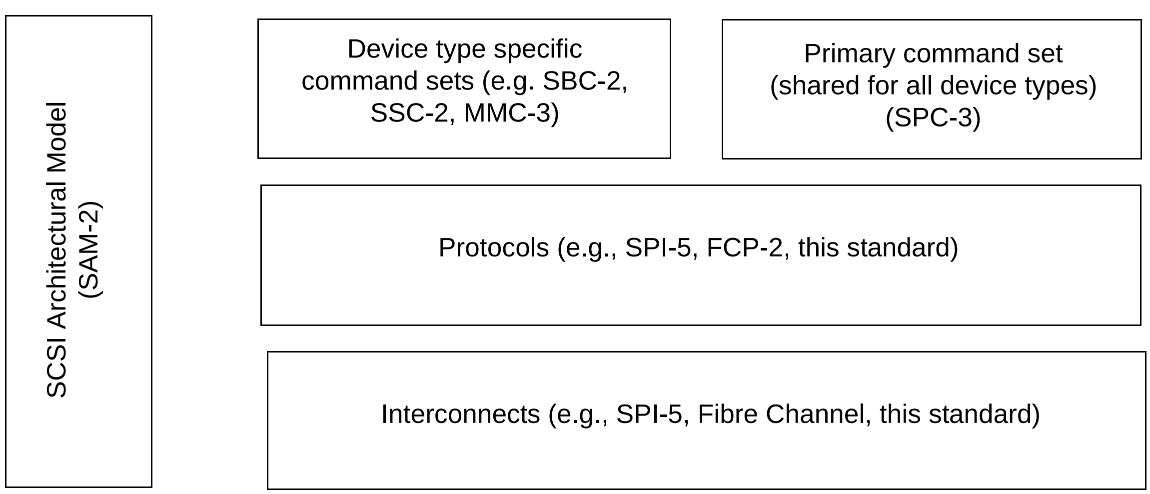


Figure 1 — General Document Structure of SCSI

These standards specify the interfaces, functions and operations necessary to ensure interoperability between conforming implementations. This standard is a functional description. Conforming implementations may employ any design technique that does not violate interoperability.

2 Normative references

2.1 Normative references

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

Copies of the following documents 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.

Additional availability contact information is provided below as needed.

2.2 Approved references

ISO/IEC 14776-412, SCSI Architecture Model - 2 (SAM-2) [ANSI INCITS.366-2003]

ISO/IEC 14776-113, SCSI Parallel Interface - 5 (SPI-5) [ANSI NCITS.367-2003]

ISO/IEC 14776-222, SCSI Fibre Channel Protocol - 2 (FCP-2) [ANSI INCITS.350:2003]

ISO/IEC 14776-351, SCSI-3 Medium Changer Commands (SMC) [ANSI NCITS.314:1998]

ISO/IEC 14776-352, SCSI Media Changer Commands - 2 (SMC-2) [T10/1383-D]

ISO/IEC 14776-332, SCSI Stream Commands - 2 (SSC-2) [T10/1434-D]

2.3 References under development

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

ISO/IEC 14776-413, SCSI Architecture Model - 3 (SAM-3) [T10/1561-D]

ISO/IEC 14776-453, SCSI Primary Commands - 3 (SPC-3) [T10/1416-D]

IISO/IEC 14776-356, Automation/Drive Interface, Commands (ADC) [T10/1558-D]

NOTE 1 For more information on the current status of the document, contact the INCITS Secretariat at 202-737-8888 (phone), 202-638-4922 (fax) or via Email at INCITS@itic.org. To obtain copies of this document, contact Global Engineering at 15 Inverness Way, East Englewood, CO 80112-5704 at 303-792-2181 (phone), 800-854-7179 (phone), or 303-792-2192 (fax).

2.4 Other references

For information on the current status of the listed document(s), or regarding availability, contact the indicated organization.

SFF-8054, Automation Drive Interface Connector

NOTE 2 For more information on the current status of the document, contact the SFF committee at 408-867-6630 (phone), or 408-867-2115 (fax). To obtain copies of this document, contact the SFF committee at 14426 Black Walnut Court, Saratoga, CA 95070 at 408-867-6630 (phone) or 408-741-1600 (fax).

ASTM D-4566, Standard Test Methods for Electrical Performance Properties of Insulations and Jackets for Telecommunications Wire and Cable

ANSI/TIA/EIA-422-B-1994 Electrical Characteristics of Balanced Voltage Digital Interface Circuits. (RS-422)

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 acknowledgement IU: An ACK information unit (IU), NAK IU, Initiate Recovery IU, or Initiate Recovery NAK IU (see 6.5.3).

3.1.2 ADT initiator port: A SCSI initiator port that implements ADT.

3.1.3 ADT port: An ADT initiator port, ADT target port, or ADT target/initiator port.

3.1.4 ADT target port: A SCSI target port that implements ADT.

3.1.5 ADT target/initiator port: A port that has all the characteristics of an ADT target port and an ADT initiator port.

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

3.1.7 auto-contingent allegiance: An optional condition of a task set following the return of a CHECK CONDITION status (see SAM-2).

3.1.8 automation device: A device containing one or more SMC device servers (see SMC-2) or equivalent, one or more automation application clients, and one or more ADT ports. An automation device may contain one or more automation device primary ports (see ADC).

3.1.9 automation device port: An ADT port on an automation device.

3.1.10 bridging manager: An application client within a DT device that initiates SCSI requests to an SMC device server within an automation device (see ADC).

3.1.11 byte: Indicates an 8-bit construct.

3.1.12 device server: An object within a logical unit that processes SCSI tasks according to the rules for task management (see SAM-2).

3.1.13 DT device: A device containing an RMC device server, an ADC device server, one or more ADT ports, and one or more DT device primary ports. A DT device may contain a bridging manager and local SMC device server (see ADC).

3.1.14 DT device port: An ADT port on a DT device.

3.1.15 exchange: The basic mechanism that transfers information consisting of one or more related information units that may flow in the same or opposite directions. An exchange is identified by its X_Origin and Exchange ID (see 6.3).

3.1.16 expected frame number: The value in the FRAME NUMBER field that a receiver port expects in the next frame (see 4.5.3).

3.1.17 field: A group of one or more contiguous bits.

3.1.18 hard reset: A target action in response to a reset event in which the target port performs the operations described in 4.7.

3.1.19 information unit: An ADT frame header and payload (see 6.1).

3.1.20 I_T nexus: A nexus that exists between a SCSI initiator port and a SCSI target port.

3.1.21 I_T_L nexus: A nexus that exists between a SCSI initiator port, a SCSI target port, and a logical unit.

3.1.22 I_T_L_Q nexus: A nexus between a SCSI initiator port, a SCSI target port, a logical unit, and a queue tag following the successful receipt of a queue tag.

3.1.23 link parameters: The parameters affecting the physical operation of the link, including but not limited to maximum ACK offset, maximum payload size, and baud rate.

3.1.24 logical unit: A SCSI target device object, containing a device server and task manager, that implements a device model and manages tasks to process SCSI commands sent by an application client.

3.1.25 logical unit number: An identifier for a logical unit.

3.1.26 logical unit reset: A logical unit action in response to a logical unit reset event in which the logical unit performs the operations described in SAM–2.

3.1.27 logical unit reset event: An event that triggers a logical unit reset from a logical unit as described in SAM-2.

3.1.28 login process: The process of negotiating operating parameters for the transport using Port Login IUs (see 4.3.3).

3.1.29 negotiated parameters: The link parameter values agreed upon through negotiation. The negotiated parameters are an element of the intersection of the supported parameters for each port.

3.1.30 nexus: A relationship between two SCSI devices, and the SCSI initiator port and SCSI target port objects within those SCSI devices.

3.1.31 object: A container that encapsulates data types, services, or other objects that are related in some way.

3.1.32 operating parameters: The current link parameter values under which the port is operating.

3.1.33 port: Synonymous with ADT port

3.1.34 queue: The arrangement of tasks within a task set usually according to the temporal order that they were created.

3.1.35 reset event: A protocol specific event that triggers a hard reset from a device (see 4.7).

3.1.36 SCSI device: A device that contains one or more SCSI ports that are connected to a service delivery subsystem and supports a SCSI application protocol.

3.1.37 SCSI initiator device: A SCSI device containing application clients and SCSI initiator ports that originates device service and task management requests to be processed by a SCSI target device and receives device service and task management responses from SCSI target devices. When used, this term refers to SCSI initiator devices or SCSI target/initiator devices that are using the SCSI target/initiator port as a SCSI initiator port.

3.1.38 SCSI initiator port: A SCSI initiator device object that acts as the connection between application clients and the service delivery subsystem through which requests and confirmations are routed. In all cases when this term is used, it refers to an initiator port or a SCSI target/initiator port operating as a SCSI initiator port.

3.1.39 SCSI port: A SCSI device resident object that connects the application client, device server or task manager to the service delivery subsystem through which requests and responses are routed. SCSI port is synonymous with port. A SCSI port is either a SCSI initiator port (see 3.1.38) or a SCSI target port (see 3.1.41).

3.1.40 SCSI target device: A SCSI device containing logical units and SCSI target ports that receives device service and task management requests for processing and sends device service and task management responses to SCSI initiator devices. When used, this term refers to SCSI target devices or SCSI target/initiator devices that are using the SCSI target/initiator port as a SCSI target port.

3.1.41 SCSI target port: A SCSI target device object that contains a task router and acts as the connection between device servers and task managers and the service delivery subsystem through which indications and responses are routed. When this term is used it refers to a SCSI target port or a SCSI target/initiator port operating as a SCSI target port.

3.1.42 simple exchange: An exchange consisting of two information units, an IU with a type other than acknowledgement and the corresponding acknowledgement IU.

3.1.43 starting parameters: The maximum values of link parameters of which the port is capable.

3.1.44 supported parameters: The set of link parameter values under which the port is capable of operating.

3.1.45 symbol framing error: An error that occurs if the receiver of an asynchronously transmitted symbol detects a Start or Stop bit with an incorrect value.

3.1.46 task: An object within the logical unit representing the work associated with a command or group of linked commands.

3.1.47 task manager: A server within a logical unit that controls the sequencing of one or more tasks and processes task management functions.

3.1.48 task management function: A task manager service that may be invoked by sending a SCSI Task Management IU with the requested task management function to affect the processing of one or more tasks.

3.1.49 task set: A group of tasks within a logical unit, whose interaction is dependent on the task management (e.g., queuing) and ACA requirements.

3.1.50 vendor-specific: Something (e.g., a bit, field, code value) that is not defined by this standard and may be used differently in various implementations.

3.1.51 zero: A false signal value or a false condition of a variable.

3.2 Symbols and abbreviations

x	multiply
+	add
-	subtract
< or LT	less than
= or EQ	equal
> or GT	greater than
ACA	auto-contingent allegiance
ADC	Automation/Drive Interface - Commands
ADT	Automation/Drive Interface - Transport Protocol

AER	asynchronous event report
CDB	command descriptor block
CRN	command reference number
DT	data transfer (e.g. DT device)
FCP-2	Fibre Channel Protocol for SCSI, Second Version
IU	information unit
LSB	Least significant bit
LUN	Logical unit number
MSB	Most significant bit
SCSI	Small Computer System Interface
SAM-2	SCSI Architecture Model-2
SAM-3	SCSI Architecture Model-3
SCSI-3	Small Computer System Interface - 3
SMC	SCSI-3 Medium Changer Commands
SMC-2	SCSI Media Changer Commands-2
SPC-2	SCSI Primary Commands-2
SPC-3	SCSI Primary Commands-3
SPI-5	SCSI Parallel Interface-5
SSC-2	SCSI Stream Commands-2

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 an error.

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

3.3.4 may: A keyword that indicates flexibility of choice with no implied preference (synonymous with "may or may not").

3.3.5 may not: A keyword that indicates flexibility of choice with no implied preference (synonymous with "may or may not").

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

3.3.7 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, it shall be implemented as defined in this standard.

3.3.8 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. 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.

3.3.9 shall: A keyword indicating a mandatory requirement. Designers are required to implement all such requirements to ensure interoperability with other products that conform to this standard.

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

3.4 Conventions

Certain words and terms used in this American National Standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in clause 3 or in the text where they first appear. Names of signals, phases, messages, commands, statuses, sense keys, additional sense codes, and additional sense code qualifiers are in all uppercase (e.g., REQUEST SENSE), names of fields are in small uppercase (e.g., STATE OF SPARE), lower case is used for words having the normal English meaning.

Fields containing only one bit are usually referred to as the name bit instead of the name field.

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

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

Numbers immediately followed by lower-case h (xxh) are hexadecimal values.

Decimals are indicated with a comma (e.g., two and one half is represented as 2,5).

Decimal numbers having a value exceeding 999 are represented with a space (e.g., 24 255).

An alphanumeric list (e.g., a,b,c or A,B,C) of items indicates the items in the list are unordered.

A numeric list (e.g., 1,2,3) of items indicate the items in the list are ordered (i.e., item 1 shall occur or complete before item 2).

In the event of conflicting information the precedence for requirements defined in this standard is:

1) text,
2) tables, then
3) figures.

3.5 Notation for Procedures and Functions

Procedure Name ([input:1a|input:1b|input:1c][,input:2a+input:2b]...[input:n]||
[output:1][,output:2]...[output:n])

Where

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

"(...)": Parentheses enclosing the lists of input and output arguments.

input:1a|input:1b|... A number of arguments of which only one shall be used in any single procedure

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.

"||": A separator providing the demarcation between inputs and outputs. Inputs are listed to the left of the separator; outputs, if any, are listed to the right.

"[...]": Brackets enclosing optional or conditional parameters and arguments.

"|": A separator providing the demarcation between a number of arguments of which only one shall be used in any single procedure.

"+": A collection of objects presented to a single object. No ordering is implied.

3.6 State machine conventions

3.6.1 State machine conventions overview

Figure 2 shows how state machines are described.

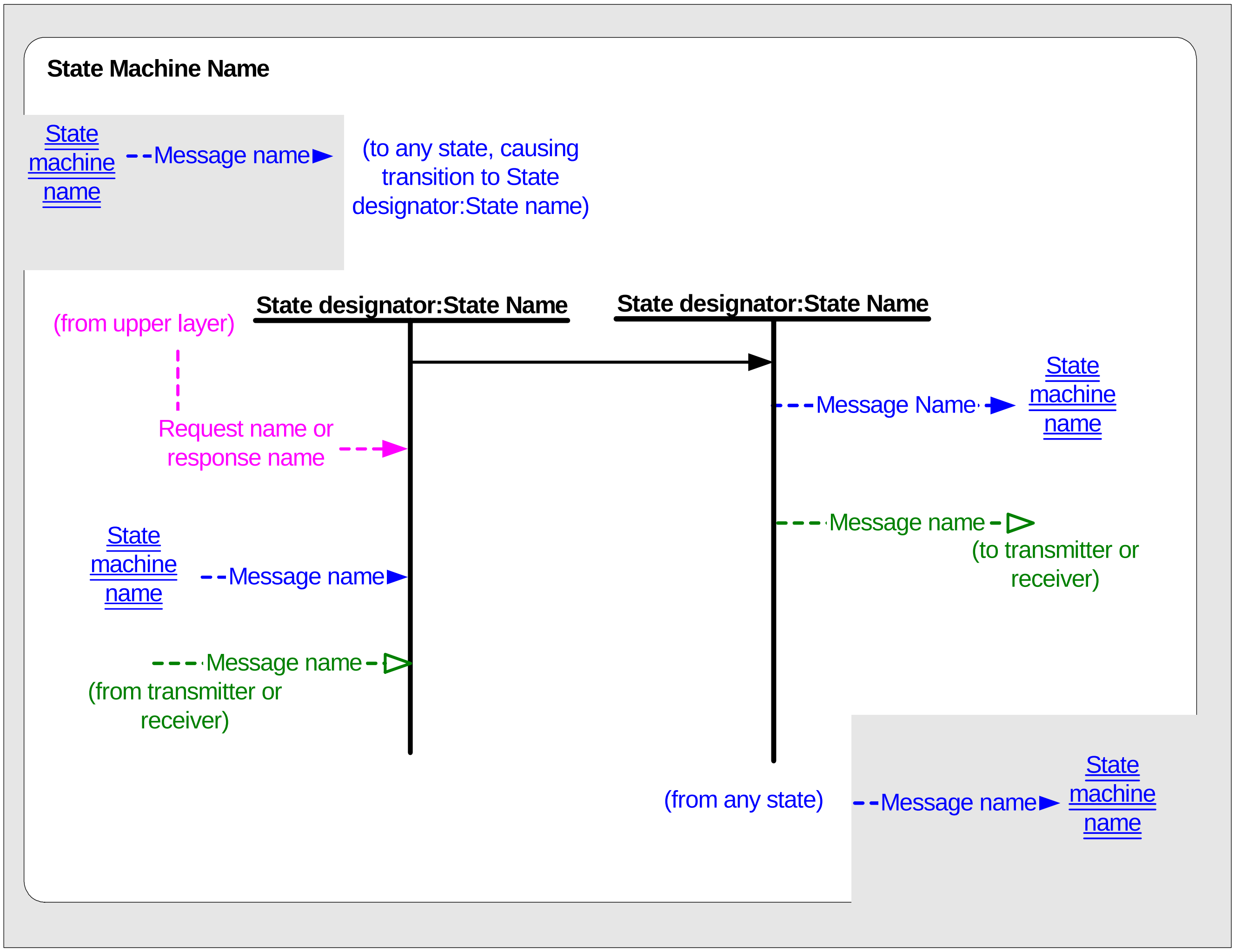


Figure 2 — State machine conventions

State machines are enclosed in boxes with rounded corners. Each state is identified by a state designator and a state name. The state designator (e.g., P0) is unique among all state machines in this standard. The state name (e.g., Idle) is a brief description of the primary action taken during the state, and the same state name may be used by other state machines. Actions taken while in each state are described in the state description text.

3.6.2 sub-state machines

Some states have sub-state machines associated with them. A sub-state machine only exists while the associated super-state is active. Activation of an associated super-state shall cause all sub-state machines of that state to enter their initial states.

3.6.3 Transitions

Transitions between states are shown with solid lines with an arrow pointing to the destination state. The conditions that cause a transition are fully described in the transition description text for each state.

Transitions between states are instantaneous.

3.6.4 Messages, requests, and event notifications

Messages passed between state machines are shown with dashed lines labeled with a message name. If messages are passed between state machines, they are identified by either a dashed line to or from a state machine name label with double underlines.

The meaning of each message is described in the state description text.

Requests and event notifications are shown with curved dashed lines originating from or going to the top or bottom of the figure. Each request and event notification is labeled. The meaning of each request and event notification is described in the state description text where it is used.

Messages with unfilled arrowheads are passed to or from the state machine's transmitter or receiver, not shown in the state machine figures, and are directly related to data being transmitted on or received from the service delivery subsystem.

Messages, requests and event notifications that affect all states in the state machine are shown as touching the edge of the state machine enclosure. In this case, the meaning is described in the general state machine description subclause. Similarly, those that originate from all states are shown as exiting from the state machine enclosure.

4 General

4.1 Architecture

Figure 3 shows an example of an ADT interface within a media changer containing two DT devices. Other common components of a media changer are also shown for reference.

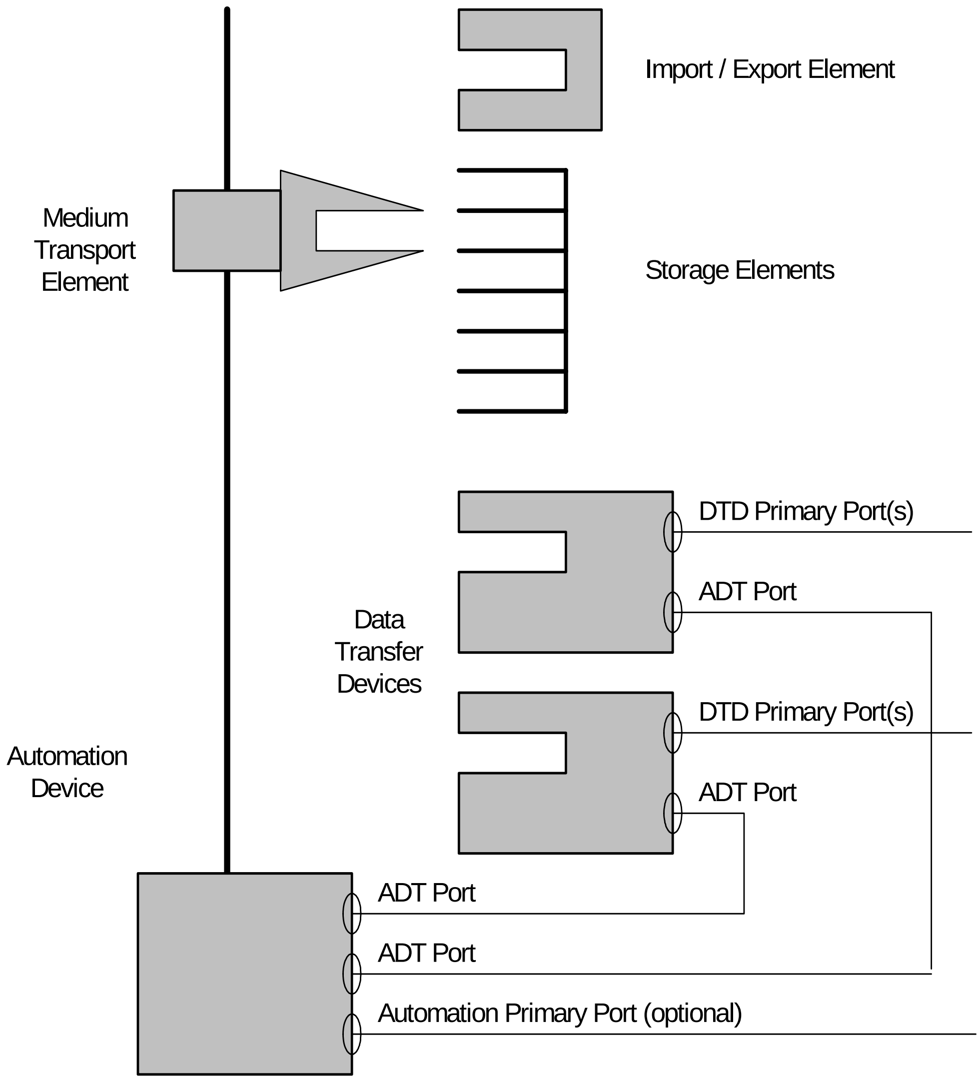


Figure 3 — Example Media Changer application of ADT

If ADI Bridging is enabled (see ADC), each ADT port in the DT device and automation device acts as a SCSI target/initiator port. If ADI Bridging is disabled, the DT device port acts as a SCSI target port and the automation device port acts as a SCSI initiator port.

4.2 Default operating parameters

The default operating parameters for a port are as follows:

a) The baud rate shall be set to 9 600;
b) the ACK offset shall be set to 1; and
c) the Maximum Payload size shall be 256 bytes.

These values shall remain in effect until the login process is complete, at which time the negotiated values shall take effect.

4.3 ADT state machines

4.3.1 Introduction

The ADT transport layer contains five state machines to manage a connection between two ADT ports. The state machines are as follows:

a) Port;
b) Link negotiation;
c) Transmitter;
d) Transmitter error recovery; and
e) Receiver error recovery.

The port state machine is the primary machine and always active. The other state machines are only active to manage specific operations (i.e they are sub-state machines of a state in the port state machine).

4.3.2 Port state machine

4.3.2.1 Port state machine overview

The port state machine consists of the following port states:

a) P0:Initial;
b) P1:Login;
c) P2:Logged-In; and
d) P3:Logged-Out.

This state machine shall start in P0:Initial state after a hard reset event.

Figure 4 shows the port state machine. The following subclauses describe the transitions and the actions taken in each state.

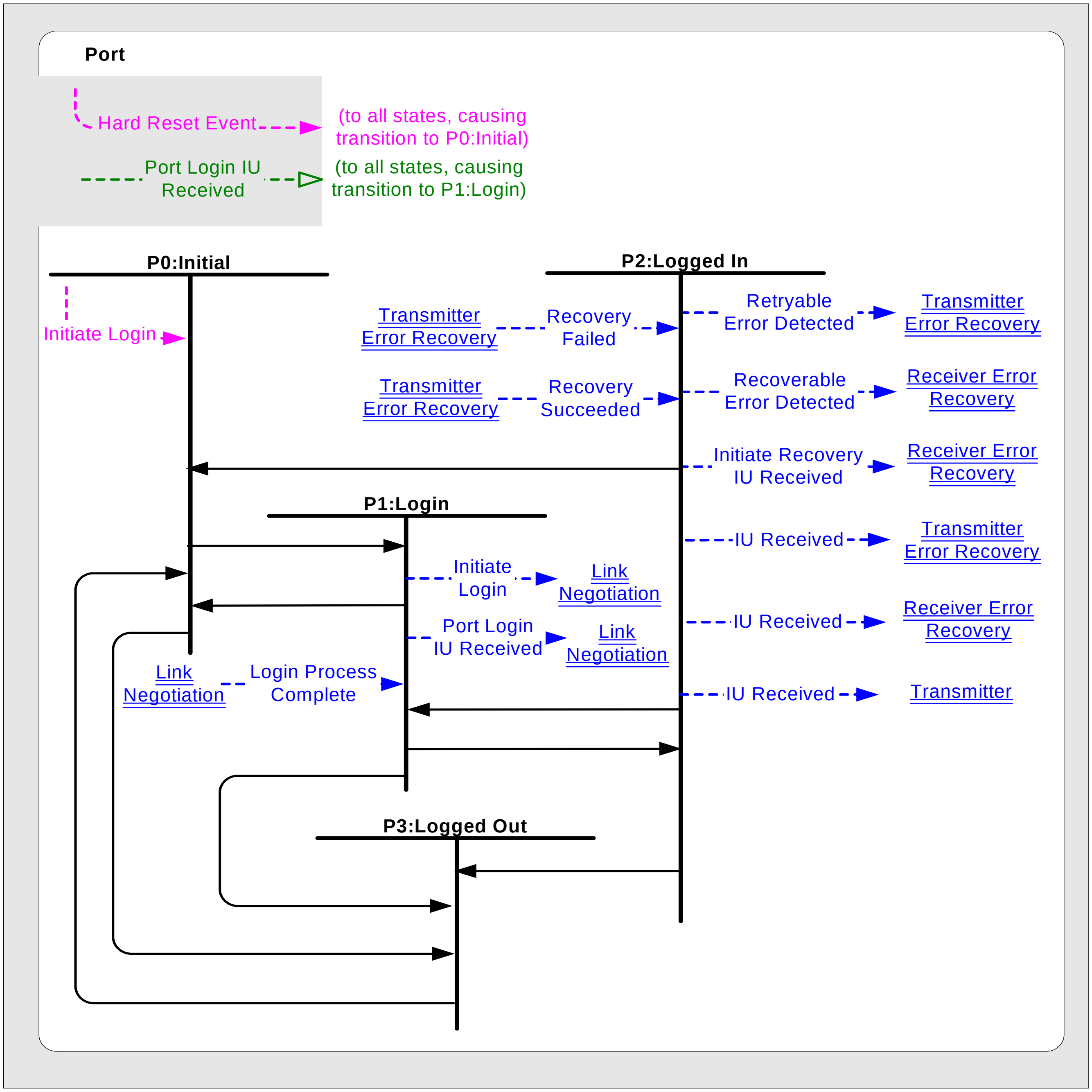


Figure 4 — Port State Machine Diagram

4.3.2.2 P0:Initial state

4.3.2.2.1 State description

This is the initial state of the port state machine.

A port in P0:Initial state shall send a NAK IU with status code of REJECTED, PORT IS LOGGED OUT (see table 14) in response to any frame other than Port Login IU, Port Logout IU, NOP IU or acknowledgement IU. Upon entering this state, all other state machines shall be set to their initial states.

4.3.2.2.2 Transition P0:Initial to P1:Login

The port shall transition to P1:Login after receiving a Port Login IU (see 6.5.4) or an Initiate Login request.

4.3.2.2.3 Transition P0:Initital to P3:Logged-Out

A port shall transition to P3:Logged-Out state after it receives a Port Logout IU and sends the corresponding ACK IU.

4.3.2.3 P1:Login state

4.3.2.3.1 State description

While in the P1:Login state, Port Login IUs are used to establish or change link parameters used by both ports on the service delivery subsystem. The login process is a negotiation between the ports that shall result in establishment of negotiated parameters. Following a hard reset or a transition to P3:Logged Out state, the port shall set its operating parameters to default parameters before sending the Port Login IU (see 4.2). If the port is already logged in, the operating parameters shall not be changed before sending the Port Login IU, unless otherwise specified. The login process consists of a series of Port Login IUs all within a single exchange (i.e the same X-Origin and Exchange ID values are used in all information units throughout the process (see 6.3)).

A port in this state shall send a NAK IU with a status code of LOGIN IN PROGRESS (see table 14) in response to any frame other than Port Login IU, Port Logout IU, NOP IU or acknowledgement IU.

While in this state, the port shall send a Port Login IU received message to the link negotiation state machine each time it receives a Port Login IU.

If the port enters this state as a result of an Initiate Login request, it shall send an Initiate Login message to the link negotiation state machine.

After acknowledging a Port Login IU, transmission of frames for other exchanges shall either be suspended or aborted based on the setting of the AOE bit in the Port Login IU (see 6.5.4).

4.3.2.3.2 Transition P1:Login to P0:Initial

A port shall transition to P0:Initial state after it sends a Port Logout IU and receives the corresponding ACK IU.

4.3.2.3.3 Transition P1:Login to P2:Logged-In

A port shall set its operating parameters to the negotiated values and transition to P2:Logged-In state after receiving a Login Process Complete message.

4.3.2.3.4 Transition P1:Login to P3:Logged-Out

A port shall transition to P3:Logged-Out state after it receives a Port Logout IU and sends the corresponding ACK IU.

4.3.2.4 P2:Logged-In

4.3.2.4.1 State description

While in this state, the port's permission to transmit is managed through the use of the transmitter state machine.

While in this state, error recovery is managed through the use of the transmitter error recovery and receiver error recovery state machines.

The port shall send a Retryable Error Detected message to the transmitter error recovery state machine if the port detects an error as defined in 4.6.1.2. In addition, the port shall suspend the transmission of all frames other than Port Login IU, Port Logout IU, Initiate Recovery IU, NOP IU or acknowledgment IU.

The port shall send a Recoverable Error Detected message to the receiver error recovery state machine if the port detects an error as defined in 4.6.1.3.

The port shall send an Initiate Recovery IU Received message to the receiver error recovery state machine if the port receives an Initiate Recovery IU.

If a frame other than Port Login IU, Port Logout IU or NOP IU is received and the receiver error recovery state machine is not in R0:Idle, the port shall send an IU Received message to the receiver error recovery state machine.

If a frame other than Port Login IU, Port Logout IU or NOP IU is received and the transmitter error recovery state machine is not in TE0:Idle, the port shall send an IU Received message to the transmitter error recovery state machine.

If the port receives a Recovery Succeeded message, the port shall resume the transmission of frames.

4.3.2.4.2 Transition P2:Logged-In to P0:Initial

A port shall transition to P0:Initial after receiving an ACK IU for a Port Logout IU.

4.3.2.4.3 Transition P2:Logged-In to P1:Login

A port shall transition to P1:Login and initiate a port login exchange after receiving a Recovery Failed message (see 4.3.5.4.2) or upon receiving a Port Login IU. If the transition is due to a Recovery Failed message, the port shall abort all exchanges, set its operating parameters to default, and initiate a Port Login exchange with the AOE bit set to one.

4.3.2.4.4 Transition P2:Logged-In to P3:Logged-Out

A port shall transition to P3:Logged-Out state after it receives a Port Logout IU and sends the corresponding ACK IU.

4.3.2.5 P3:Logged-Out state

4.3.2.5.1 State description

A port in P3:Logged-Out state shall not initiate an exchange. While in this state, upon receiving any frame other than a Port Login IU, the port shall send a NAK IU with a status code of REJECTED, PORT IS LOGGED OUT (see table 14).

4.3.2.5.2 Transition P3:Logged-out to P0:Initial

A port shall transition to P0:Initial after the logout duration time specified in the Port Logout IU (see 6.5.5) has expired.

4.3.3 Link negotiation state machine

4.3.3.1 Link negotiation state machine overview

The link negotiation state machine is used to manage the login process. It is a sub-state machine of the port state P1:Login. The states are as follows:

a) N0:Idle;
b) N1:Negotiating;
c) N2:Accept Sent;
d) N3:Accept ACK Sent; and
e) N4:Agreed; and

This state machine becomes active when the port enters the P1:Login state.

If a port receives a Port Login IU with the ACCEPT bit set to one and with parameter values that are different from the last Port Login IU sent, the port shall send a NAK IU with a status code of NEGOTIATION ERROR (see table 14) and transition to N1:Negotiating to initiate a new login exchange.

Figure 5 shows the link negotiation state machine. The following subclauses describe the transitions and the actions taken in each state.

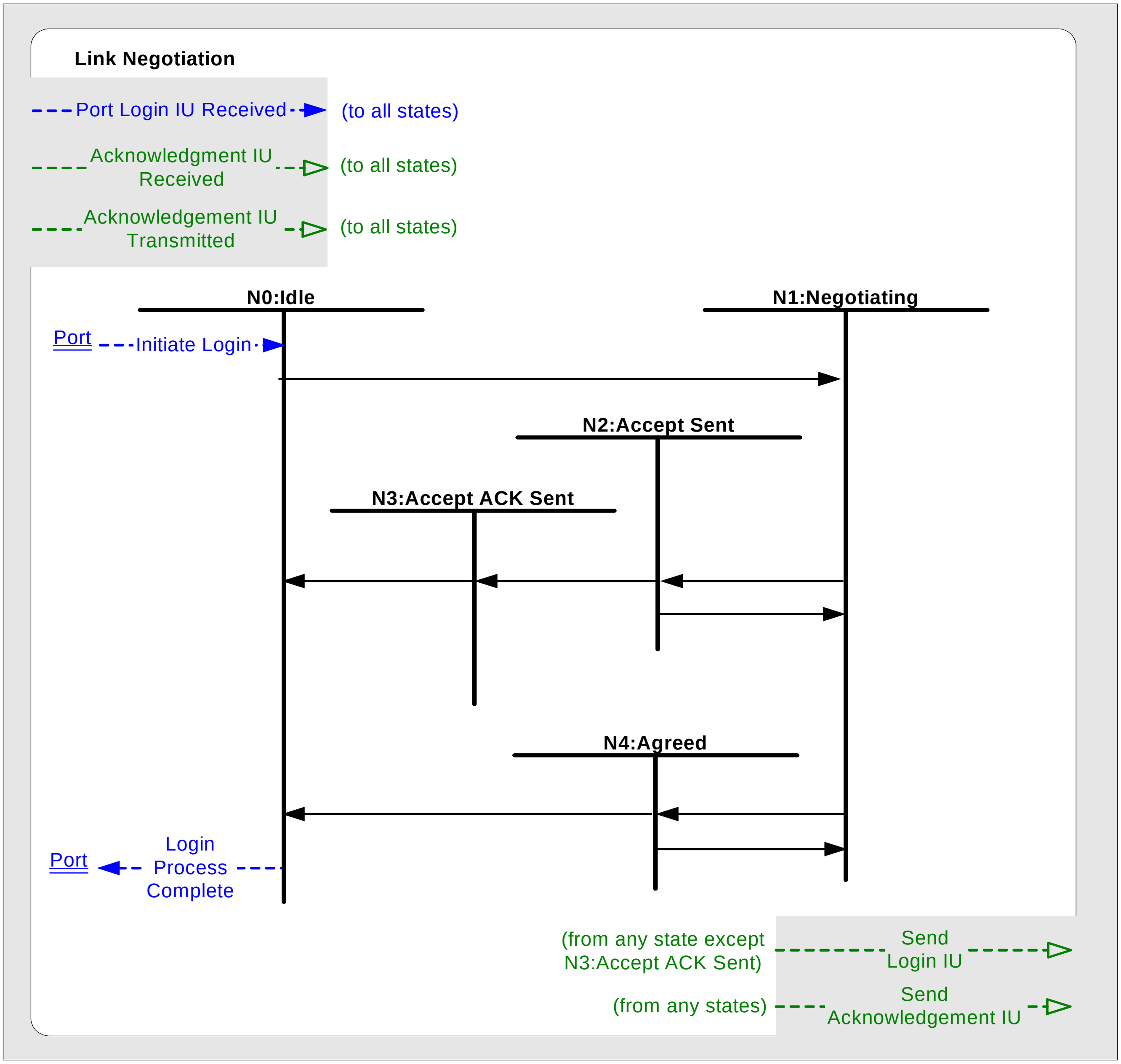


Figure 5 — Link Negotiation State Diagram

4.3.3.2 Precedence of port login exchanges

To avoid a deadlock condition if both ports are attempting to initiate a Port Login exchange at the same time, the following rules shall apply. An automation device port that receives a Port Login IU with an exchange originated by the other port shall perform one of the following actions:

a) send an ACK IU and discard the Port Login IU from the other port if the automation device port has initiated a Port Login exchange that has not yet completed;

b) send an ACK IU and discard the Port Login IU and initiate a Port Login exchange; or
c) complete the port login process using the exchange originated by the other port if no other Port Login exchange is open.

A DT device that receives a Port Login IU in a new exchange shall abort all other Port Login exchanges, transition to N2:Negotiating, and process the Port Login IU.

4.3.3.3 N0:Idle state

4.3.3.3.1 State description

The N0:Idle state waits for the port to receive a Port Login IU.

4.3.3.3.2 Transition N0:Idle to N1:Negotiating

A port shall transition from N0:Idle to N1:Negotiating when:

a) a Port Login IU Received message is received and the link parameter (see 3.1.23) values within the Port Login IU are not in the supported parameters (see 3.1.44) of the port; or
b) an Initiate Login message is received.

4.3.3.3.3 Transition N0:Idle to N2:Accept Sent

If the received Port Login IU has the ACCEPT bit set to zero and the link parameter (see 3.1.23) values are in the supported parameters (see 3.1.44) of the port, the port shall transition to N2:Accept Sent and send a Port Login IU with the link parameter values unchanged and the ACCEPT bit set to one.

4.3.3.4 N1:Negotiating state

4.3.3.4.1 State description

If the port transitioned to this state as the result of an Initiate Login message, the port shall send a Port Login IU in a new exchange with the ACCEPT bit set to zero and the link parameter values set to the starting parameters.

If the port transitioned to this state as a result of a negotiation error, the port shall send a Port Login IU in a new exchange. The Port Login IU shall contain starting parameters. The ACCEPT bit shall be set to zero.

When a Port Login IU message is received, the parameters shall be inspected. If the value of any of the link parameters (see 3.1.23) specified in Port Login IU are not supported parameters (see 3.1.44), the port shall adjust all link parameters values that are not supported by the port and respond with a Port Login IU that contains these adjusted link parameter values. The ACCEPT bit shall be set to zero. The method of adjusting link parameter values in a Port Login IU is described in 6.5.4.

If a port has not received a Port Login IU within 15 seconds after receiving the ACK IU for a Port Login IU that it has sent, the port shall consider this condition an error. The port shall abort the Port Login exchange, set the port operating parameters to default, and initiate a new Port Login exchange.

4.3.3.4.2 Transition N1:Negotiating to N0:Idle

If a port sends a NAK IU in response to a Port Login IU it shall transition to N0:Idle.

4.3.3.4.3 Transition N1:Negotiating to N2:Accept Sent

If the ACCEPT bit is set to zero and the link parameters (see 3.1.23) in the received Port Login IU are supported parameters (see 3.1.44), the port shall send a Port Login IU with the link parameters unchanged and the ACCEPT bit set to one and transition to N2:Accept Sent.

4.3.3.4.4 Transition N1:Negotiating to N4:Agreed

If the ACCEPT bit is set to one and the link parameters in the received Port Login IU are unchanged from the values sent in the last Port Login IU, the port shall send a Port Login IU with the same values and the ACCEPT bit set to one and transition to N4:Agreed.

4.3.3.5 N2:Accept Sent state

4.3.3.5.1 State description

A port enters this state if it has sent a Port Login IU with the ACCEPT bit set to one and with parameters unchanged before it received a Port Login IU with the ACCEPT bit set to one (i.e., it is the first port to send a Port Login IU with the ACCEPT bit set to one).

4.3.3.5.2 Transition N2:Accept Sent to N0:Idle

If a port receives a Port Login IU with a protocol error or resource limitation error, the port shall send a NAK IU and transition to N0:Idle.

4.3.3.5.3 Transition N2:Accept Sent to N1:Negotiating

If a port receives a Port Login IU with the ACCEPT bit set to zero or with parameter values that are different from the last Port Login IU sent, the port shall send a NAK IU with a status code of NEGOTIATION ERROR (see table 14) and transition to N1:Negotiating to initiate a new login exchange.

4.3.3.5.4 Transition N2:Accept Sent to N3:Accept ACK Sent

If a port receives a Port Login IU with the ACCEPT bit set to one and with parameters unchanged, the port shall send an ACK IU and transition to N3:Accept ACK Sent.

4.3.3.6 N3:Accept ACK Sent state

4.3.3.6.1 State description

A port enters this state if it has sent a Port Login IU with the ACCEPT bit set to one and with parameters unchanged, and then received a Port Login IU with the ACCEPT bit set to one.

4.3.3.6.2 Transition N3:Accept ACK Sent to N0:Idle

When the ACK IU has finished transmitting, the port shall send a Login Process Complete message to the port state machine and transition to N0:Idle.

4.3.3.7 N4:Agreed state

4.3.3.7.1 State description

A port enters this state if it is the second port to accept the operating parameters. A port in this state is waiting for the ACK IU in response to the Port Login IU it sent (see clause 4.3.3.4.4).

4.3.3.7.2 Transition N4:Agreed to N1:Negotiating

If a port receives a NAK IU it shall transition to N1:Negotiating to initiate a new login exchange.

4.3.3.7.3 Transition N4:Agreed to N0:Idle

After receiving an ACK IU for the Port Login IU it sent, the port shall send a Login Process Complete message to the port state machine and transition to N0:Idle.

4.3.4 Transmitter state machine

4.3.4.1 Transmitter state machine overview

The transmitter state machine manages the port's permission to transmit. It is a sub-state machine of the port state P2:Logged-In. The transmitter state machine consists of the following states:

a) T0:Entering;
b) T1:Active; and
c) T2:Paused.

This state machine becomes active when the port enters P2:Logged-In state.

Figure 6 shows the transmitter state machine. The following subclauses describe the transitions and the actions taken in each state.

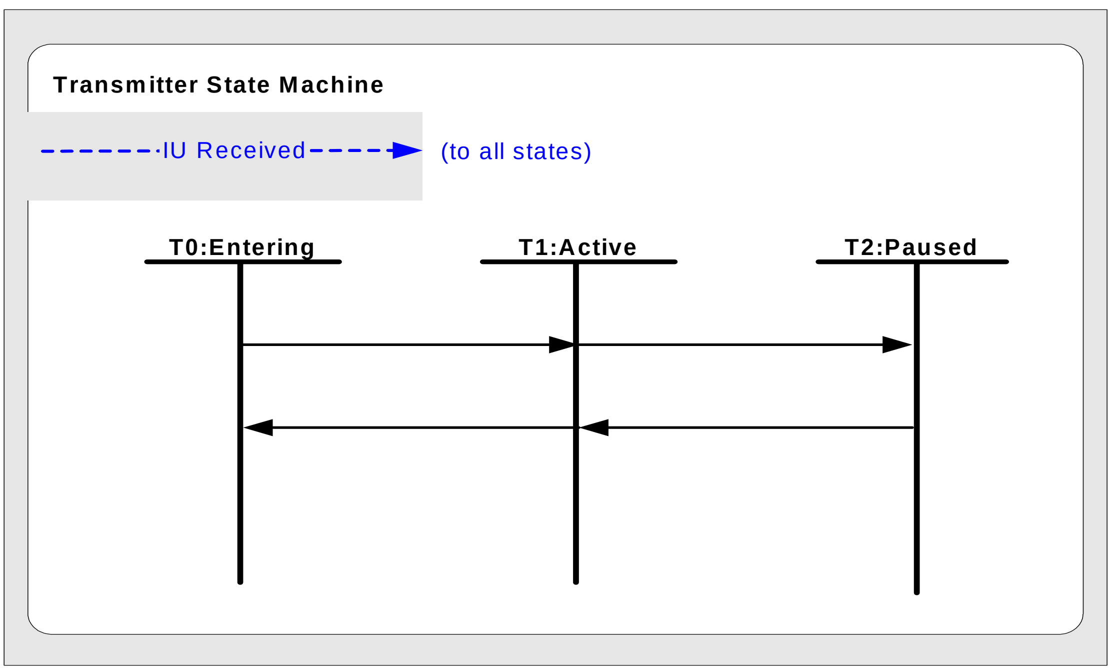


Figure 6 — Transmitter State Diagram

4.3.4.2 T0:Entering state

4.3.4.2.1 State description

On entry to this state the port shall start a 100 millisecond timer.

While in this state, a port shall not transmit.

4.3.4.2.2 Transition T0:Entering to T1:Active

A port shall transition to T1:Active state when it receives a frame that is not corrupted (see 4.6.1.3) or after a period of 100 milliseconds.

4.3.4.3 T1:Active state

4.3.4.3.1 State description

A port in T1:Active state may transmit and receive all types of information units.

4.3.4.3.2 Transition T1:Active to T2:Paused

A port shall transition to T2:Paused state after it receives a Pause IU and sends the corresponding ACK IU.

4.3.4.4 T2:Paused state

4.3.4.4.1 State description

A port in T2:Paused state shall not transmit a frame.

4.3.4.4.2 Transition T2:Paused to T1:Active

A port shall transition to T1:Active state after receiving any frame other than a Port Login IU, Port Logout IU, Pause IU, or acknowledgment IU.

4.3.5 Transmitter error recovery state machine

4.3.5.1 Transmitter error recovery state machine overview

The transmitter error recovery state machine manages error recovery in the transmitting port. It is a sub-state machine of the port state machine's P2:Logged-In state. The transmitter error recovery state machine consists of the following states:

a) TE0:Idle;
b) TE1:Initiating Recovery; and
c) TE2:Retry Initiate Recovery.

This state machine becomes active when the port enters P2:Logged-In state.

Figure 7 shows the transmitter error recovery state machine. The following subclauses describe the transitions and the actions taken in each state.

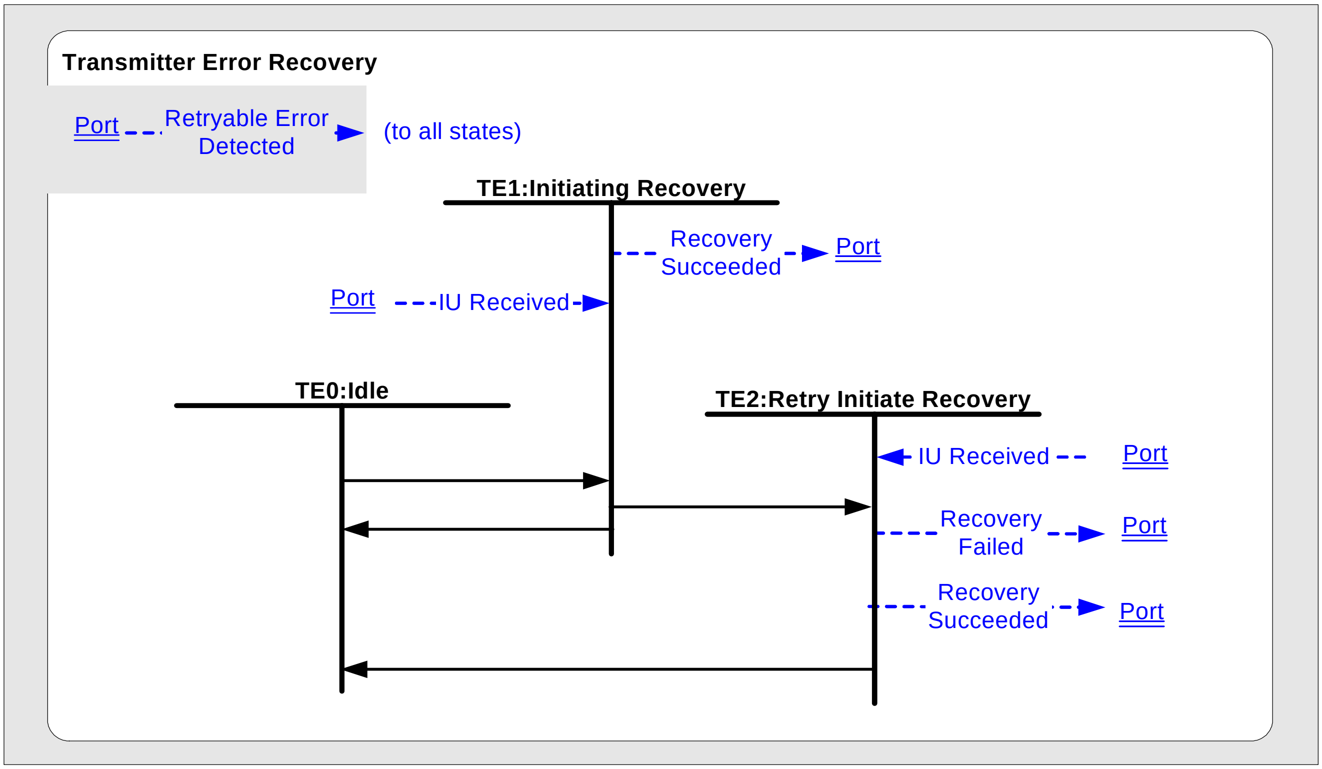


Figure 7 — Transmitter Error Recovery State Diagram

4.3.5.2 TE0:Idle state

4.3.5.2.1 State description

The TE0:Idle state waits for the port to detect a retryable error.

4.3.5.2.2 Transition TE0:Idle to TE1:Initiating Recovery

Upon receiving a Retryable Error Detected message, the port shall send an Initiate Recovery IU and transition to TE1:Initiating Recovery.

4.3.5.3 TE1:Initiating Recovery state

4.3.5.3.1 State description

A port in TE1:Initiating Recovery state shall not send any frames other than acknowledgement IUs, Initiate Recovery IUs, Port Login IUs, NOP IUs, Pause IUs, or Port Logout IUs.

A port in TE1:Initiating Recovery state shall discard ACK IUs and NAK IUs for frames other than Port Login IUs, NOP IUs, Pause IUs, and Port Logout IUs.

4.3.5.3.2 Transition TE1:Initiating Recovery to TE0:Idle

If an Initiate Recovery ACK IU for the Initiate Recovery IU is received, the port shall send a Recovery Succeeded message to the port state machine and transition to TE0:Idle. This shall cause the transmission of the frame that had failed.

4.3.5.3.3 Transition TE1:Initiating Recovery to TE2:Retry Initiate Recovery

If a Retryable Error Detected message or Initiate Recovery NAK IU for the Initiate Recovery IU is received, the port shall re-send the Initiate Recovery IU and transition to TE2:Retry Initiate Recovery.

4.3.5.4 TE2:Retry Initiate Recovery state

4.3.5.4.1 State description

A port in TE2:Retry Initiate Recovery state shall not send any frames other than acknowledgement IUs, Port Login IUs, NOP IUs, Pause IUs, or Port Logout IUs.

A port in TE2:Retry Initiate Recovery state shall discard ACK IUs and NAK IUs for frames other than Port Login IUs, NOP IUs, Pause IUs, and Port Logout IUs.

4.3.5.4.2 Transition TE2:Retry Initiate Recovery to TE0:Idle

If an Initiate Recovery ACK IU for the Initiate Recovery IU is received, the port shall send a Recovery Succeeded message to the port state machine and transition to TE0:Idle. This shall cause the transmission of the frame that had failed.

If a Retryable Error Detected message or Initiate Recovery NAK IU for the Initiate Recovery IU is received, the port shall send a Recovery Failed message to the port state machine and transition to TE0:Idle.

4.3.6 Receiver error recovery state machine

4.3.6.1 Receiver error recovery state machine overview

The receiver error recovery state machine manages error recovery in the receiving port. It is a sub-state machine of the port state P2:Logged-In. The state machine consists of the following states:

a) R0:Idle;
b) R1:Pending Recovery; and
c) R2:Recovering.

This state machine becomes active when the port state machine enters P2:Logged-In state.

Figure 8 shows the receiver error recovery state machine. The following subclauses describe the transitions and the actions taken in each state.

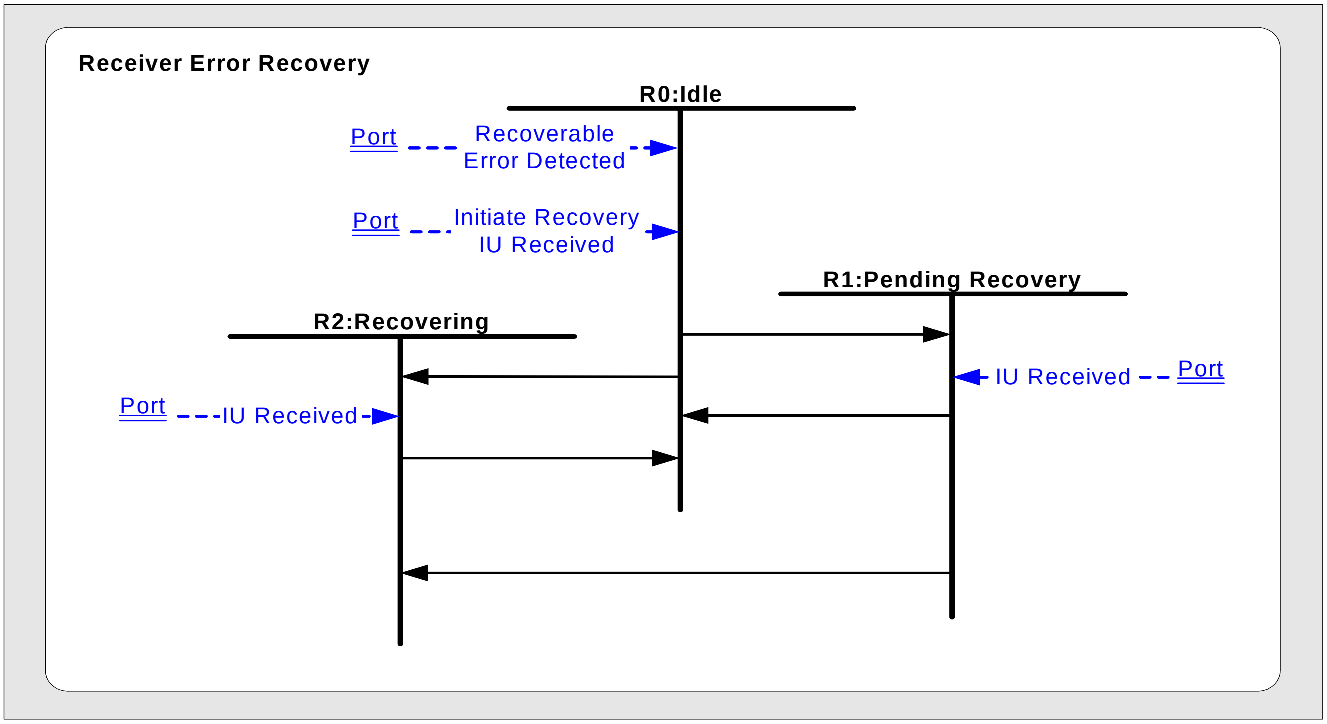


Figure 8 — Receiver Error Recovery State Diagram

4.3.6.2 R0:Idle state

4.3.6.2.1 State description

The R0:Idle state waits for the port to detect a recoverable error or receive an Initiate Recovery IU indicating that the transmitter has detected a retryable error.

4.3.6.2.2 Transition R0:Idle to R1:Pending Recovery

The port shall transition to R1:Pending Recovery upon receiving a Recoverable Error Detected message.

4.3.6.2.3 Transition R0:Idle to R2:Recovering

The port shall transition to R2:Recovering upon receiving an Initiate Recovery IU Received message and the FRAME NUMBER field value does not match the Expected Frame Number counter.

4.3.6.3 R1:Pending Recovery state

4.3.6.3.1 State description

While a port is in R1:Pending Recovery state, receipt of a frame other than an Initiate Recovery IU, NOP IU, Port Login IU, Port Logout IU, Pause IU, or acknowledgment IU is an error and the port shall send a NAK IU with a status code of AWAITING INITIATE RECOVERY IU (see table 14) and PR bit set to one.

4.3.6.3.2 Transition R1:Pending Recovery to R0:Idle

A port shall transition to R0:Idle state after it receives a valid Initiate Recovery IU with a FRAME NUMBER field value that matches the Expected Frame Number counter.

4.3.6.3.3 Transition R1:Pending Recovery to R2:Recovering

A port shall transition to R2:Recovering state after it receives a valid Initiate Recovery IU with a FRAME NUMBER field value that does not match the Expected Frame Number counter.

4.3.6.4 R2:Recovering state

4.3.6.4.1 State description

A port in this state shall acknowledge and discard all frames that were previously processed. This is all frames with a FRAME NUMBER field value that does not match the Expected Frame Number counter.

4.3.6.4.2 Transition R2:Recovering to R0:Idle

When a frame other than an Initiate Recovery IU with a FRAME NUMBER field value that matches the Expected Frame Number counter is received, the port shall transition to the R0:Idle state.

4.4 ACK Offset

The receiving port explicitly acknowledges all frames that are transmitted. By default, a port waits for an acknowledgement IU for every frame it sends before sending another frame, except for acknowledgement IUs. This mode of operation requires a minimal amount of resources in the port to buffer incoming frames. However, it imposes a throughput reduction caused by delays waiting for acknowledgement.

The MAXIMUM ACK OFFSET field in the Port Login IU allows the ports to negotiate the number of frames that may be sent without waiting for acknowledgement, based on the resources available to the ports. Regardless of the setting of this field, all frames shall be acknowledged in the order they are received.

Each port shall keep a counter to track the unacknowledged frames that it has sent, called the Unacknowledged Frame Counter. This counter shall be set to zero at hard reset events. It shall also be set to zero before sending a Port Login IU or upon receiving an ACK IU for an Initiate Recovery IU. The counter shall be incremented by one for each frame that is sent except acknowledgement frames. It shall be decremented by one for each acknowledgement IU that is received. A port shall not transmit frames when this counter is equal to the maximum ACK offset value except for acknowledgement IUs, Port Login IUs, and Initiate Recovery IUs. If the port has not successfully completed the Port Login process, it shall not transmit a frame if the unacknowledged frame count is one.

Upon receipt of a frame, a port may detect that the maximum ACK offset value has been exceeded. A port that detects receipt of a frame that exceeds the maximum ACK offset value shall either:

a) send a NAK IU with a status code of MAXIMUM ACK OFFSET EXCEEDED (see table 14); or
b) discard the frame without sending an acknowledgment IU.

4.5 Frame Number Counters

4.5.1 Frame Number Counters overview

The ADT frame header contains a field called FRAME NUMBER that serves three purposes:

a) it allows an acknowledgement IU to be associated with a specific frame;
b) it allows a receiving port to detect missing frames; and
c) it allows a port to identify a frame in order to retry transmission of frames that fail.

To accomplish these, each port shall keep two counters, one to keep track of the frame number in the next frame to send, and one to track the next expected frame number to be received.

4.5.2 Next Frame To Send counter

The Next Frame To Send counter value shall be calculated as follows:

1) It shall be set to one after sending or receiving a Port Login IU with the AOE bit set to one;
2) it shall be set to one after sending or receiving a Port Logout IU;
3) it shall be set to the value in the FRAME NUMBER field of an Initiate Recovery IU that is sent by the port;
4) it shall not be adjusted after sending an acknowledgement IU, a Port Login IU with the AOE bit set to zero, a Pause IU, or a NOP IU; and
5) after sending all other frame types, it shall be set to the frame number of the last frame sent plus one. If this value is greater than seven, it shall be set to one.

4.5.3 Expected Frame Number counter

The Expected Frame Number counter shall be used to detect missing frames in the receive stream. It shall be calculated as follows:

1) It shall be set to one after sending or receiving a Port Login IU with the AOE bit set to one;
2) it shall be set to one after receiving a Port Logout IU;
3) it shall be set to one after receiving an ACK IU for a Port Logout IU;
4) it shall not be adjusted when receiving an acknowledgement IU, a Port Login IU with the AOE bit set to zero, a Pause IU, a NOP IU, an Initiate Recovery IU, or a frame with a receiver detected error (see 4.6.1.3);
5) it shall not be adjusted when the port is operating in R2:Recovering state; and
6) if the port is operating in P2:Logged-In state, the port shall compare the FRAME NUMBER field in each received frame with the Expected Frame Number counter, and:
   a) If they do not match, the port shall send a NAK IU in response to the frame with a status code of UNEXPECTED FRAME NUMBER (see table 14) and the Expected Frame Number counter shall not be adjusted; or
   b) If they do match, the Expected Frame Number counter shall be incremented by one. If this value is greater than seven, it shall be set to one.

A port that receives a Port Login IU, Port Logout IU, Pause IU, or NOP IU shall verify the FRAME NUMBER field in the ADT frame header is set to zero. If the FRAME NUMBER field is not zero, the port shall respond with a NAK IU with a status code of INVALID OR ILLEGAL IU RECEIVED (see table 14).

4.6 Link layer error recovery

4.6.1 Error detection

4.6.1.1 Error detection overview

Errors in the transport layer may be detected by either the sender of a frame, the receiver of a frame, or by both.

4.6.1.2 Error detection by the frame sender

4.6.1.2.1 Errors detected by the frame sender overview

Retryable errors are defined as errors detected by the sending port by either:

a) a timeout without receipt of an acknowledgement IU;
b) receipt of a NAK IU with the PR bit set to one; or
c) acknowledgement IU received out of order.

4.6.1.2.2 Acknowledgement IU time-out

The sender of a frame, other than an acknowledgement IU, shall time-out the resulting acknowledgement. It shall be considered an error condition if a corresponding acknowledgement IU is not received within the time-out period. The time-out period shall start after the EOF of the frame has been sent. When operating with a maximum ACK offset greater than one, a port may start the time-out period for a frame that has completed transmission after the acknowledgement IU for a previously sent frame has been received. The minimum acknowledgement IU time-out period shall be calculated using the formula in figure 9.

$$Timeout_{ACK} = (Period * Size_{Max} * 2) + (Period * (Offset_{Max} * Size_{NAK} * 2)) + 0,1 \text{ seconds}$$

Where:

$Timeout_{ACK}$ is the minimum time-out period in seconds.

Period is the time per byte calculated as (10 / Baud Rate) and is expressed in seconds per byte.

$Size_{Max}$ is the Maximum Payload Size negotiated with the Port Login process, plus SOF, EOF, ADT Header, and checksum bytes (see 6.1).

$Offset_{Max}$ is the maximum ACK offset negotiated with the Port Login process (see 4.4).

$Size_{NAK}$ is the size in bytes of the NAK IU including SOF, EOF, and checksum bytes (see 6.5.3.3).

For example, at 9 600 Baud with a negotiated Maximum Payload Size of 1 024 and Maximum ACK Offset of 2, the minimum timeout period would be approximately 2,28 seconds.

Figure 9 — Minimum acknowledgement time-out period

4.6.1.2.3 NAK acknowledgement

It shall be considered an error condition if a port receives a NAK IU (see 6.5.3.3).

4.6.1.2.4 Out of order acknowledgement IU received

An acknowledgement IU that is received with a frame number that does not match the frame number of the oldest frame waiting for an acknowledgment shall be considered out of order. A port that detects an out of order acknowledgement shall consider the oldest frame waiting for an acknowledgment to be the frame in error.

4.6.1.3 Error detection by the frame receiver

There are four types of errors detectable by the frame receiver:

a) corrupted frame;
b) protocol error;
c) resource limitation error; and
d) recoverable error.

Corruption of a received frame is indicated by:

a) an incorrect checksum;
b) the occurrence of a hardware framing error;
c) the occurrence of a hardware over-run; or
d) receiving an SOF character before receiving an EOF character.

Protocol errors are detectable errors for which no retry process is defined by this standard.

Resource limitation errors are due to lack of resources sufficient to process the request, and retransmission may succeed when resource usage has changed.

Recoverable errors are those that may be recovered by retransmission of one or more frames (e.g., if the port receives a frame that is not a link service frame and the frame number does not match the Expected Frame Number counter).

4.6.2 Error recovery

4.6.2.1 Corrupted frame

If a port detects corruption of a received frame, it shall discard the frame and shall not send an acknowledgement IU.

NOTE 3 The sender of the frame detects a retryable error upon a timeout without receipt of an acknowledgement IU and performs error recovery as defined in 4.6.2.4.4.

4.6.2.2 Error recovery for symbol framing errors

After detecting four or more symbol framing errors without the receipt of a frame, a port shall abort all exchanges, set its operating parameters to default, transition to P1:Login state, and initiate a Port Login exchange with the AOE bit set to one.

4.6.2.3 Recoverable error

If a port detects a recoverable error with a frame it receives it shall send a NAK IU to the other port with the appropriate status code (see table 14) and the Pending Recovery (PR) bit set to one so that the port that sent the frame in error is able to initiate recovery steps. The FRAME NUMBER field of the NAK IU shall be set to the Expected Frame Number counter value (see 4.5.3) when the error was detected. The port shall send a Recoverable Error Detected message to the Receiver Error Recovery state machine causing it to transition to the R1:Pending Recovery state.

If a port receives an Initiate Recovery IU it is an indication that the other port is attempting to recover from a retryable error. The following steps shall be taken by the receiving port to accommodate the recovery process:

a) an Initiate Recovery ACK IU shall be sent to acknowledge receipt of the Initiate Recovery IU;
b) the FRAME NUMBER field in the Initiate Recovery IU shall be compared to the Expected Frame Number counter (see 4.5.3). If the frame numbers match and the Receiver Error Recovery state machine is in R0:Idle state, the port shall remain in its current state. If the frame numbers match and the Receiver Error Recovery state machine is in R1:Pending Recovery state, the Receiver Error Recovery state machine shall transition to R0:Idle; and
c) if the frame number does not match, this is an indication that an ACK IU was lost in transmission. The Receiver Error Recovery state machine shall transition into R2:Recovering state. While in this state, frames that are received by the port shall be acknowledged and discarded. Once a frame is received with a frame number that matches the Expected Frame Number counter, the Receiver Error Recovery state machine shall transition to the R0:Idle state.

4.6.2.4 Retryable error

4.6.2.4.1 Port login IUs

A port that is in P1:Login that receives a NAK IU or detects an acknowledgment IU time-out shall restart the negotiation by transitioning to N1:Negotiating, setting operating parameters to default (see 4.2), and initiating a new login exchange using starting parameters.

4.6.2.4.2 Port Logout, NOP, Initiate Recovery and Pause IUs

If the port detects an acknowledgement IU time-out, the port may resend the IU. If sent, the IU shall be within a new exchange.

If the port receives a NAK IU due to a resource limitation, the port may resend the IU. If sent, the IU shall be within a new exchange.

If the port receives a NAK IU due to any other error condition, the behavior is not specified.

4.6.2.4.3 Initiate Recovery IU

For error recovery on Initiate Recovery IUs, see 4.6.2.4.4.

4.6.2.4.4 Non link service IUs

After detecting that a retryable error has occurred with a frame that it sent (see 4.6.1.2), a port shall initiate the following error recovery process. A port that detects a retryable error on a frame that it sent shall retry sending the frame. The frame retry sequence is:

1) the port that sent the frame in error sets the Next Frame To Send counter to the frame number that was detected in error, sends a Retryable Error Detected message to the Transmitter Error Recovery state machine causing it to transition to TE1:Initiating Recovery state, and sends an Initiate Recovery IU. The Initiate Recovery IU contains the Next Frame To Send counter value in the FRAME NUMBER field;
2) while in TE1:Initiating Recovery state, the port waits for an Initiate Recovery ACK IU for that frame. No other frames shall be sent by that port except acknowledgement IUs for frames it receives until an Initiate Recovery ACK IU is received for the Initiate Recovery IU, a time-out occurs on the ACK IU, or a Port Login IU is received;
3) if an Initiate Recovery ACK IU is received for the Initiate Recovery IU, the error port shall resume normal operation by entering TE0:Idle and re-sending the frame in error and all frames sent after it before the error

was detected, with the exception of acknowledgement IUs. The FRAME NUMBER field values for re-transmitted frames shall not be changed from the values used when they were originally transmitted;

4) if no Initiate Recovery ACK IU is received for the Initiate Recovery IU before the ACK time-out, or an Initiate Recovery NAK IU is received indicating an error on the Initiate Recovery IU, and the Initiate Recovery IU has not been retried, the port in error shall re-send the Initiate Recovery IU; and
5) if the Initiate Recovery IU has been sent twice with no Initiate Recovery ACK IU returned, or an Initiate Recovery NAK IU is received indicating an error on the Initiate Recovery IU, the port shall send a Recovery Failed message to the port state machine (see 4.3.5.4.2).

4.6.2.5 Protocol error

4.6.2.5.1 Port Login IU

If a protocol error is detected on a Port Login IU, the port shall send a NAK IU with the PR bit set to zero and the status code of NEGOTIATION ERROR (see table 14). The port shall set the operating parameters to default (see 4.2), the port state machine shall transition to P1:Login, and the link negotiation state machine shall transition to N0:Idle.

4.6.2.5.2 Port Logout, NOP and Pause IUs

If a protocol error is detected on a Port Logout IU, NOP IU or Pause IU, the port shall send a NAK IU with PR bit set to zero and the appropriate status code (see table 14) then discard the frame.

4.6.2.5.3 Initiate Recovery IU

If a protocol error is detected on an Initiate Recovery IU, the port shall send an Initiate Recovery NAK IU with PR bit set to zero and the appropriate status code (see table 14) then discard the frame.

4.6.2.5.4 Non link service IUs

If a port detects a protocol error on a frame it receives it shall send a NAK IU with PR bit set to zero and the appropriate status code (see table 14) then discard the frame. The FRAME NUMBER field of the NAK IU shall be set to the Expected Frame Number counter value (see 4.5.3) when the error was detected.

4.6.2.6 Resource limitation error

4.6.2.6.1 Port Login IU

If a resource limitation error is detected on a Port Login IU, the port shall send a NAK IU with a status code of OUT OF RESOURCES (see table 14), set the operating parameters to default (see 4.2), the Port state machine shall transition to P1:Login, and the Link Negotiation state machine shall transition to N0:Idle.

4.6.2.6.2 Port Logout, NOP and Pause IUs

If a resource limitation error is detected on a Port Logout IU, NOP IU or Pause IU, the port shall send a NAK IU with the PR bit set to zero and the appropriate status code (see table 14) then discard the frame.

If the port is unable to send an acknowledgment IU due to a resource limitation, it shall discard the frame.

4.6.2.6.3 Initiate Recovery IU

If a resource limitation error is detected on an Initiate Recovery IU, the port shall send an Initiate Recovery NAK IU with the PR bit set to zero and a status code of OUT OF RESOURCES (see table 14) then discard the frame.

If the port is unable to send an acknowledgment IU due to a resource limitation, it shall discard the frame

4.6.2.6.4 Non link service IUs

If a port detects a resource limitation error on a frame it receives, it should send a NAK IU with the PR bit set to one and a status code of OUT OF RESOURCES (see table 14). A port that receives a NAK IU indicating a resource limitation error shall consider the error retryable and perform the actions described in 4.6.2.4.4.

If the port is unable to send an acknowledgment IU due to a resource limitation, it shall discard the frame.

4.7 Hard reset

A hard reset is a response to a power on condition or optionally a $Reset_a$ event (see table 7). The target port's response to a hard reset shall include initiating the equivalent of a logical unit reset for all logical units as described in SAM-2.

The effect of the hard reset on tasks that have not completed, SCSI device reservations, and SCSI device operating modes is defined in SAM-2.

4.8 I_T nexus loss

An I_T nexus loss event shall occur if an ADT port:

a) sends a Port Login IU with the AOE bit set to one;
b) receives a Port Login IU with the AOE bit set to one;
c) detects the change of state of the Sense line from presence to absence (i.e., $Sense_a$ for DT device port and $Sense_b$ for automation device port (see figure 11); or
d) detects the assertion of the $Reset_a$ line (see table 7).

If an ADT port detects an I_T nexus loss event it shall send a Nexus Loss event notification indication (see SAM-3) to the SCSI Application layer.

4.9 Transport protocol variations from SAM-2

The ADT transport protocol provides all of the services mandated by SAM-2. In addition to the mandatory protocol services, ADT provides several extensions for the initiator port. This subclause provides an overview of these extensions. See clause 8 for details.

The ADT transport protocol provides the capability of bridging SCSI traffic from the primary interface of the DT device to the automation device. To facilitate this function, the DT device contains both a device server and an application client (Bridging Manager) from the perspective of SAM-2 (See ADC). The extensions of the ADT protocol services allow this bridging function to be provided within a device with limited resources dedicated to this feature.

SAM-2 requires that an initiator that invokes the **Execute Command** remote procedure call have available the data and buffer space required to transfer all of the data associated with the command. The extensions of the ADT protocol services allow the application client to invoke the **Execute Command** remote procedure call with only part

of the data or buffer space available, then invoke the additional protocol services functions to transfer the remaining data.

An application client that does not make use of the transport protocol service extensions should set the **Data-In Buffer Size** and **Data-Out Buffer Size** arguments to the **Execute Command** remote procedure call per SAM-2, and ignore the **Data-In Received** and **Data-Out Delivered** confirmations from the transport protocol layer.

An application client that makes use of the transport protocol service extensions may set the **Data-In Buffer Size** and/or the **Data-Out Buffer Size** arguments to the **Execute Command** remote procedure call to a value less than the total amount of data required by the command. The transport layer shall assert the **Data-In Received** confirmation when all of the data requested by the **Data-In Buffer Size** has been transferred. The transport layer shall assert the **Data-Out Delivered** confirmation when all of the data indicated by the **Data-Out Buffer Size** has been transferred.

An application client shall invoke the **Send Data-Out** transport protocol service after receiving the **Data-Out Delivered** confirmation if there is more data to be sent to the device server. An application client shall not invoke the **Send Data-Out** transport protocol service for an I_T_L_Q nexus after it has invoked the **Send Data-Out** transport protocol service for that I_T_L_Q nexus until it receives a **Data-Out Delivered** confirmation for that I_T_L_Q nexus. An application client shall invoke the **Send Data-Out** transport protocol service after performing the **Execute Command** remote procedure call if there is data to be sent to the device server and the **Data-Out Buffer Size** argument in the **Execute Command** remote procedure call was set to zero. An application client shall not invoke the **Send Data-Out** transport protocol service for a given I_T_L_Q nexus if the **Data-Out Buffer Size** argument in the **Execute Command** remote procedure call for that I_T_L_Q nexus was non-zero until it receives a **Data-Out Delivered** confirmation for that I_T_L_Q nexus.

An application client shall invoke the **Receive Data-In** transport protocol service after receiving the **Data-In Received** confirmation if there is more data expected from the device server. An application client shall not invoke the **Receive Data-In** transport protocol service for an I_T_L_Q nexus after it has invoked the **Receive Data-In** transport protocol service for that I_T_L_Q nexus until it receives a **Data-In Received** confirmation for that I_T_L_Q nexus. An application client shall invoke the **Receive Data-In** transport protocol service after performing the **Execute Command** remote procedure call if there is data expected from the device server and the **Data-In Buffer Size** argument in the **Execute Command** remote procedure call was set to zero. An application client shall not invoke the **Receive Data-In** transport protocol service for a given I_T_L_Q nexus if the **Data-In Buffer Size** argument in the **Execute Command** remote procedure call for that I_T_L_Q nexus was non-zero, until it receives a **Data-In Received** confirmation for that I_T_L_Q Nexus.

An application client shall not invoke the **Send Data-Out** or **Receive Data-In** transport protocol services for an I_T_L_Q nexus after it has received a **Command Complete Received** confirmation or after it has invoked a task management protocol service that aborts the task associated with the I_T_L_Q nexus.

An ADT transport layer shall not invoke the **Data-In Received** or **Data-Out Delivered** transport protocol service for an I_T_L_Q nexus after asserting the **Command Complete Received** confirmation for that I_T_L_Q nexus.

5 Physical layer

5.1 Electrical Characteristics

5.1.1 ADT compliance points

An ADT compliance point is a defined point in the ADT physical interconnection. At an ADT compliance point, a compliant device shall meet the ADT interoperability specifications. ADT compliance points always occur at separable connectors. Table 1 lists the ADT compliance points.

Table 1 — ADT Compliance Points

Compliance Point	Description
I_t	Initiator port connector; transmit serial port
I_r	Initiator port connector; receive serial port
T_t	Target port connector; transmit serial port
T_r	Target port connector; receive serial port

5.1.2 Cabling

All ADT connections shall have a length less than or equal to 25m.

NOTE 4 The connection specifications in sub clauses 5.1.3 through 5.1.5 assume cable with a R < 400 ohms/km, Z_0 = 100 ohms (nominal), and C = 50 pF/m (nominal).

5.1.3 Sense connection

A Sense connection is a complete uni-directional signal path from one ADT port to a second ADT port. A Sense connection includes:

a) a current generator connected to the output compliance point of one ADT port;
b) a transmission medium from the output compliance point of one ADT port to the input compliance point of a second ADT port; and
c) a current detector connected to the input compliance point of the second ADT port.

Table 2 describes the electrical characteristics of a Sense connection at the output compliance point.

Table 2 — Sense connection output characteristics

Signal State	Current	Voltage
Asserted	$-150\ \mu A < I_{OL}$	$-0{,}2\ V < V_{OL} < 0{,}4\ V$; $V_{OL} < 0{,}2V_{dd}$[a]
Negated	$I_{OH} < 100\ \mu A$	$0{,}7V_{dd}$[a] $< V_{OH} < 3{,}6\ V$
[a] V_{dd} is the positive supply voltage at the receiving end.		

5.1.4 Signal connection

A Signal connection is a complete uni-directional signal path from one ADT port to a second ADT port. A Signal connection includes:

a) a signal generator connected to the output compliance point of one ADT port;
b) a transmission medium from the output compliance point of one ADT port to the input compliance point of a second ADT port; and
c) a signal receiver connected to the input compliance point of the second ADT port.

A signal connection shall use single-ended signalling. An ADT port shall include termination for Signal connection inputs.

Single-ended signals always exist in one of two states: true (i.e., asserted) or false (i.e., negated). The device that asserts a signal shall actively drive the signal to the true state. A device that negates a signal shall not drive the signal to either state. A non-driven signal goes to the false state because the bias of the terminator pulls the signal false.

Table 3 describes the electrical characteristics of a Signal connection at the output compliance point.

Table 3 — Signal connection output characteristics

Signal State	Current	Voltage
Asserted	-12 mA < I_{OL}	-0,2 V < V_{OL} < 0,4 V; V_{OL} < 0,2 V_{dd}[a]
Negated		V_{OH} <= 3,6 V

[a] V_{dd} is the positive supply voltage at the receiving end.

Table 4 describes the electrical characteristics of a Signal connection at the input compliance point.

Table 4 — Signal connection input characteristics

Signal State	Current	Voltage
Asserted	-12 mA < I_{IL} at 0 V	-0,2 V < V_{IL} < 0,3 V_{dd}[a]
Negated		0,7 V_{dd}[a] < V_{IH} <= 3,6 V; 400 mV < $V_{hysteresis}$

[a] V_{dd} is the positive supply voltage at the receiving end.

Table 5 describes the timing characteristics of a Signal connection.

Table 5 — Signal connection timing characteristics

Characteristic	Timing	Timing
Duration	1,5 µs < t_L	1,5 µs < t_H
Transition: 0,3 V_{dd}[a] to 0,7 V_{dd} with a connection capacitance of 1250 pF.	t_r < 500 ns	t_f < 500 ns

[a] V_{dd} is the positive supply voltage at the receiving end.

Figure 10 defines $V_{hysteresis}$.

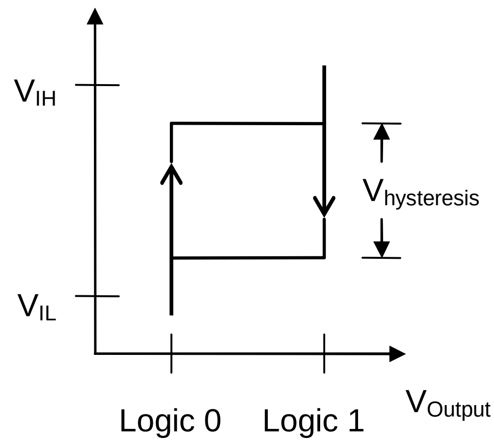


Figure 10 — $V_{hysteresis}$ definition

5.1.5 Transmit-receive connection

A Transmit-Receive (Tx-Rx) connection is a complete simplex signal path from one ADT port to a second ADT port. A Tx-Rx connection includes:

a) a signal generator connected to the output compliance point of one ADT port;
b) a pair of transmission media from the output compliance point of one ADT port to the input compliance point of a second ADT port; and
c) a signal receiver connected to the input compliance point of the second ADT port.

A Tx-Rx connection shall conform to TIA/EIA-422-B as measured at the associated compliance points.

A Tx-Rx connection shall support 9 600 baud and may support the Modulation Rates listed in table 6.

Table 6 — Optional Tx-Rx Modulation Rates

Modulation rate (baud)
19 200
38 400
57 600
76 800
115 200
153 600

A Tx-Rx connection shall use Non-return to Zero (NRZ) encoding of data bits to signaling elements. Hence, the data-signaling rate (in bps) equals the modulation rate (in baud).

A Tx-Rx connection shall transmit data bytes asynchronously adding one start bit, zero parity bits, and one stop bit to each data byte as depicted in figure 11 .

	start	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	stop

Figure 11 — Asynchronous Transmission Format

5.2 Bus composition

Table 7 defines the connections that make up the ADT bus. With the exception of $Sense_a$ and $Sense_d$ this standard defines the behavior of these connections only when an initiator port asserts $Sense_a$ and a target port asserts $Sense_d$.

Table 7 — ADT bus connections

Connection Name	O/M[a]	Connection Type	Driven By	Connection Definition
$Reset_a$	O	Signal	automation device port	An automation device may use this connection to signal a reset request to a DT device. A DT device shall treat the receipt of a signal on this connection either: a) as a port logout (see 6.5.5); or b) as a hard reset (see 4.7).
$Sense_a$	M	Sense	automation device port	A DT device shall use this connection to sense the presence or absence of an automation device on the ADT bus.
$Sense_{aux}$	O	Sense		This standard does not define the use of this connection.
$Sense_d$	M	Sense	DT device port	An automation device shall use this connection to sense the presence or absence of a DT device on the ADT bus.
$Signal_{aux}$	O	Signal		This standard does not define the use of this connection.
Tx_a - Rx_d	M	Tx-Rx	automation device port	An automation device shall use this connection to send serialized data. A DT device shall receive serialized data on this connection.
Tx_d - Rx_a	M	Tx-Rx	DT device port	A DT device shall use this connection to send serialized data. An automation device shall receive serialized data on this connection.

[a] O indicates support is optional, M indicates support is mandatory

5.3 Connector pin-out

ADT ports shall use the plug connector defined in SFF-8054. Table 8 defines the pinout for the ADT port connector on the DT device.

Table 8 — DT device ADT port connector pinout

Pin Number	Connection Name
1	$+Tx_a$ - Rx_d
2	$-Tx_a$ - Rx_d
3	Ground
4	$-Tx_d$ - Rx_a
5	$+Tx_d$ - Rx_a
6	$Sense_d$
7	$Sense_a$
8	$Reset_a$
9	$Signal_{aux}$
10	$Sense_{aux}$

6 Link layer

6.1 Basic frame format

The general layout of an ADT frame is shown in figure 12. It consists of a Start of Frame (SOF) character, followed by an ADT frame header, a frame payload, a checksum field, and concludes with an End of Frame (EOF) character.

SOF	ADT frame header	Payload	Checksum	EOF

Figure 12 — Basic ADT frame format

6.2 Encoding

To guarantee that the Start of Frame (SOF) and End of Frame (EOF) characters are unique to the data stream, special characters are reserved to represent them. To ensure that these are unique to the data stream, a technique known as byte stuffing is utilized to encode any other occurrence of these values other than the start or end of a frame. This is accomplished by using an escape character to indicate that the very next byte in the stream has been modified from its original value.

Table 9 — Special characters

Character	Description
5Bh	Start of frame
5Dh	End of frame
7Fh	Escape

Occurrences of the Escape character value are also encoded. When a data byte having the value that matches the code assigned to SOF, EOF, or Escape is encountered in the data stream, an Escape character is inserted before it and the data byte itself is modified by an XOR operation with 80h.

Byte stuffing shall not affect the actual usable header or payload sizes, as the Escape encoding and decoding shall be performed as the data is being sent and received. The checksum shall be calculated before the encoding occurs and after the decoding occurs.

6.3 ADT frame header

An ADT frame header shall be included in every frame. The ADT frame header contains the information needed to validate and route the frame to the proper protocol handler. Table 10 defines the ADT frame header.

Table 10 — ADT frame header

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved	PROTOCOL			FRAME TYPE			
1	X_ORIGIN	EXCHANGE ID			Reserved	FRAME NUMBER		
2	(MSB)	PAYLOAD SIZE						
3								(LSB)

The PROTOCOL field indicates the protocol that is carried in the payload. Table 11 defines the values for the PROTOCOL field. If a port that receives a frame does not support the value in the protocol field it shall return a NAK IU with a status code of UNSUPPORTED PROTOCOL (see table 14).

Table 11 — PROTOCOL field values

Value	Description	Reference
0	Link service	6.5
1	Encapsulated SCSI	7.1
2	Fast access	7.2
3	Vendor specific	
4 - 7	Reserved	

The FRAME TYPE field specifies the type of data contained in the frame and is defined by the protocol. See the reference sub clause from table 11 for a description of the values in this field.

The X_ORIGIN bit shall be set to zero if the automation device originates the exchange. The X_ORIGIN bit shall be set to one if the DT device originates the exchange. This bit shall remain constant for all frames associated with a given exchange.

The EXCHANGE ID field contains the identifier used to distinguish frames that are part of the same exchange. Some exchanges require more than one frame to complete, often involving frames originating in both ports. All frames that are associated with the same exchange shall have the same exchange ID and X_Origin values. The originator of a new exchange shall not re-use the exchange ID value of an existing exchange that it originated. A port may check for re-use of exchange ID values for exchanges initiated by the other port. If this check is performed and re-use is detected, the port shall send a NAK IU with status code of INVALID EXCHANGE ID (see table 14).

The FRAME NUMBER field is assigned by the transmitting port to uniquely identify a frame from other frames sent by that port over a small period of time. It ranges from zero to seven. ACK IUs return the FRAME NUMBER field value of the frame they acknowledge. The FRAME NUMBER field of a NAK IU shall contain the Expected Frame Number counter (see 4.5.3). The FRAME NUMBER field of a Port Login IU, Port Logout IU, Pause IU, or NOP IU shall be set to zero. A transmitting port shall assign all other types of frames the value in the Next Frame to Send counter (see 4.5.2).

The PAYLOAD SIZE field shall contain the number of bytes in the payload area of the frame. The number of bytes does not include the SOF, EOF, ADT frame header, checksum, or escape bytes within the payload. If a port receives a frame where the payload length does not match the value in the PAYLOAD SIZE field, the port shall return a NAK IU with the appropriate status code from table 14. If a port receives a frame with a value in the PAYLOAD SIZE field that exceeds the maximum payload size, the port shall return an NAK IU with a status code of MAXIMUM PAYLOAD SIZE EXCEEDED (see table 14).

A receiving port shall send a NAK IU in response to any frame, except an acknowledgement IU, that contains a reserved field in the ADT frame header that is not set to zero. The STATUS CODE field of the NAK IU shall be set to HEADER RESERVED BIT SET (see table 14).

Except for a Port Login IU, Port Logout IU, Pause IU, or NOP IU, a receiving port shall send a NAK IU in response to any frame with a FRAME NUMBER field set to zero. The STATUS CODE field of the NAK IU shall be set to INVALID OR ILLEGAL IU RECEIVED (see table 14).

6.4 Checksum

The CHECKSUM field shall be one byte. The value of this field shall be the XOR of the following bytes:

a) all bytes in the ADT header field;
b) all bytes (if any) in the ADT payload field; and
c) one byte of value FFh.

The CHECKSUM value shall be calculated before the encoding operation specified in 6.2.

NOTE 5 When verifying the checksum of a received frame, the XOR of all data received after the SOF character and before the EOF character after decoding should be equal to FFh on a frame with good checksum.

6.5 Link service information units

6.5.1 Link service frames overview

Either port may initiate link service frames. Link service frames are used to manage the transport layer. Table 12 defines the values for the FRAME TYPE field in the ADT frame header for link service protocol frames.

Table 12 — Link service information units

Frame Type	Description
0h	ACK (acknowledge)
1h	NAK (negative acknowledge)
2h	Port login
3h	Port logout
4h	Pause
5h	NOP (no operation)
6h	Initiate recovery
7h	Initiate recovery ACK (acknowledgement)
8h	Initiate recovery NAK (negative acknowledgement)
9h - Fh	Reserved

6.5.2 Payload size – type consistency

Unless otherwise specified in this standard, the receiver of a link service frame shall not consider it an error if the value of the PAYLOAD SIZE field does not match the specified size for those link service information units that have a specified size. If the size of the payload exceeds the specified size, the frame receiver shall ignore the excess payload bytes except with respect to the calculation of the Checksum field.

If the size of the payload is less than the specified size, the frame receiver shall not change the current setting(s) of the parameter(s) controlled by any missing field(s). All other fields in frame shall be treated normally.

6.5.3 Acknowledgement information units

6.5.3.1 Acknowledgement information units introduction

An acknowledgement IU is an ACK IU, NAK IU, Initiate Recovery ACK IU, or Initiate Recovery NAK IU. A port shall not send an acknowledgement IU in response to receiving a frame that does not contain valid SOF and EOF characters. See 4.4 for other rules governing the use of acknowledgement IUs.

In an acknowledgement IU, the X_ORIGIN and EXCHANGE ID fields are undefined and shall be ignored.

6.5.3.2 ACK information unit

An ACK IU shall be sent by a port that has received a frame without error. Except for acknowledgement IUs, a port shall send an ACK IU for every frame that it receives without error. An ACK IU shall contain zero bytes of payload.

6.5.3.3 NAK information unit

A NAK IU is sent by the transport layer to indicate that the port has detected an error during the reception of a frame. Except for acknowledgement IUs, a port shall send a NAK IU for every frame that it receives with a detected error, except corrupted frames (see 4.6.1.3). The FRAME NUMBER field in the ADT frame header of the NAK IU shall be set to the value in the Expected Frame Number counter (see 4.5.3). The Payload of the NAK IU is shown in table 13.

Table 13 — NAK IU payload contents

Bit / Byte	7	6	5	4	3	2	1	0
0	PR	STATUS CODE						

The Pending Recovery (PR) bit shall be set to one if the port is in R1:Pending Recovery state. The bit shall be set to zero otherwise.

The STATUS CODE field shall contain the status code value. Status code values are shown in table 14.

Table 14 — NAK frame STATUS CODE value (Sheet 1 of 2)

Status	Description
00h	Reserved
01h	OVER-LENGTH (i.e., more bytes received than PAYLOAD SIZE field indicated)
02h	UNDER-LENGTH (i.e., fewer bytes received than PAYLOAD SIZE field indicated)
03h	UNEXPECTED FRAME NUMBER [a]
04h	AWAITING INITIATE RECOVERY IU [a]
05h	HEADER RESERVED BIT SET. Applies to the version of ADT that the receiving port supports.
06h	INVALID EXCHANGE ID
07h - 2Fh	Reserved
30h - 3Fh	Vendor specific
40h	UNSUPPORTED PROTOCOL
41h	OUT OF RESOURCES (i.e., the receiving port has run out of buffers to store the frame)
[a] This error is recoverable (see 4.6.2.3)	

Table 14 — NAK frame STATUS CODE value (Sheet 2 of 2)

Status	Description
42h	LOGIN IN PROGRESS
43h	INVALID OR ILLEGAL IU RECEIVED
44h	Reserved
45h	REJECTED, PORT IS LOGGED OUT
46h	MAXIMUM ACK OFFSET EXCEEDED
47h	MAXIMUM PAYLOAD SIZE EXCEEDED
48h	UNSUPPORTED FRAME TYPE FOR SELECTED PROTOCOL
49h	NEGOTIATION ERROR
4Ah - 6Fh	Reserved
70h - 7Fh	Vendor specific protocol error
[a] This error is recoverable (see 4.6.2.3)	

6.5.3.4 Interleaving acknowledgement and other frame types

A port shall not terminate transmission of a frame to send an acknowledgement IU except in the case of receiving a Port Login IU or Port Logout IU. A port that receives a Port Login IU or Port Logout IU may terminate transmission of a frame in progress. A port shall acknowledge all frames that it has received before starting transmission of any other frame type, except Port Login IUs, Port Logout IUs, or Pause IUs.

6.5.4 Port login information unit

See 4.3.3 for a description of the use of the Port Login IU.

A port that receives a Port Login IU with a payload containing fewer than 8 bytes shall send a NAK IU with a status code of NEGOTIATION ERROR (see table 14).

Table 15 defines the payload of the Port Login IU.

Table 15 — Port Login IU payload contents

<table>
<tr><th>Bit
Byte</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>ACCEPT</td><td colspan="5">Reserved</td><td colspan="2">Vendor Specific</td></tr>
<tr><td>1</td><td colspan="3">MAJOR REVISION</td><td colspan="5">MINOR REVISION</td></tr>
<tr><td>2</td><td colspan="8">Reserved</td></tr>
<tr><td>3</td><td>AOE</td><td colspan="5">Reserved</td><td colspan="2">MAXIMUM ACK OFFSET</td></tr>
<tr><td>4</td><td>(MSB)</td><td colspan="7" rowspan="2">MAXIMUM PAYLOAD SIZE</td></tr>
<tr><td>5</td><td>(LSB)</td></tr>
<tr><td>6</td><td>(MSB)</td><td colspan="7" rowspan="2">BAUD RATE</td></tr>
<tr><td>7</td><td>(LSB)</td></tr>
</table>

The ACCEPT bit shall be set to zero on the first Port Login IU of a negotiation exchange and all subsequent Port Login IUs sent by a port until the Port Login IU parameters it is sending matches the parameters of the last Port Login IU received. If the Port Login IU parameters sent by a port matches the parameters of the last Port Login IU it received, the ACCEPT bit shall be set to one.

Ports claiming compliance with a draft revision of this standard shall set the MAJOR REVISION field to 0. Ports claiming compliance with the INCITS approved version of this standard shall set the MAJOR REVISION field to 1.

Ports claiming compliance with a draft version of this standard shall set the MINOR REVISION field to the revision of the draft standard. Ports claiming compliance with the INCITS approved version of this standard shall set the MINOR REVISION field to 0. When initiating a Port Login exchange, a port shall set the MAJOR REVISION field to the highest value it supports and the MINOR REVISION field to the highest draft standard revision it supports for the reported major revision. A port that was designed to support the INCITS approved version of this standard shall initiate a Port Login exchange with the MAJOR REVISION field set to 1 and the MINOR REVISION field set to 0. During the negotiation process, a port that is not capable of supporting the revision in a Port Login IU that it receives shall reply with a reduced revision level. The revision level shall be reduced as follows:

a) if the port supports the major revision level in the received Port Login IU, but not the minor revision level in the IU, it shall respond with the same major revision level. The minor revision level shall be set to the highest it is does support that is lower than the minor revision in the received Port Login IU. If the device does not support the major revision with a lower minor revision value, it shall respond as if it does not support the major revision level; or
b) if the port does not support the major revision in the Port Login IU it receives, it shall respond with the highest major revision it does support that is lower than the major revision in the received Port Login IU. The minor revision shall be set to the highest level supported at that major revision level.

All fields in the Port Login IU sent by a port shall comply with the revision level specified by the MAJOR REVISION and MINOR REVISION fields. A port that receives a Port Login IU with supported MAJOR REVISION and MINOR REVISION field values and a payload that does not comply with the indicated revision shall send a NAK IU with a status code of NEGOTIATION ERROR (see table 14). Once the Port Login process has completed, both ports should operate as defined by the major revision and minor revision values in the accepted Port Login IU.

The Abort Other Exchanges (AOE) bit shall be set to one in a Port Login IU sent by a port under the following conditions:

a) the port has experienced a hard reset condition;
b) the port has experienced an error condition that may have led to loss of data or state on one or more exchanges; or
c) the port has received a Port Login IU with the AOE bit set to one.

The AOE bit shall not affect Port Login exchanges. See 4.3.3.2 for Port Login exchange precedence.

A port that receives a valid Port Login IU with the AOE bit set to one shall abort all other exchanges, other than Port Login exchanges. No frames shall be sent for exchanges other than the Port Login exchange after a Port Login IU with the AOE bit set to one has been acknowledged.

A Port that receives a valid Port Login IU with the AOE bit set to zero shall only send frames associated with the Port Login exchange after acknowledging the Port Login IU until the Port Login exchange is complete. Frames from exchanges other than Port Login exchanges shall not be sent until all port login exchanges are complete.

The MAXIMUM ACK OFFSET field indicates the number of frames that may be sent to the port without receiving an acknowledgement IU in response (see 4.4). A value of zero indicates the port is disabled for all but link service traffic.

The MAXIMUM PAYLOAD SIZE field indicates the maximum number of bytes in the payload of a frame that the port is able to accommodate. The MAXIMUM PAYLOAD SIZE field shall be set to at least 256 bytes to accommodate a SCSI Response IU with the maximum sense length of 252 bytes (see 7.1.4). If a port receives a Port Login IU containing a maximum payload size value less than 256 it shall respond with a NAK IU with a status code of NEGOTIATION ERROR (see table 14) and transition to N1:Negotiating to initiate a new login exchange.

The BAUD RATE field indicates the speed that the port's physical interface shall run after completion of negotiation. The BAUD RATE field contains the desired nominal Baud rate divided by 100. All ports shall default to operating at 9 600 Baud at power-up and following error conditions that require re-establishment of the operating parameters (see 4.6.2). If a port receives a Port Login IU containing a baud rate value less than 9 600 it shall respond with a NAK IU with a status code of NEGOTIATION ERROR (see table 14) and transition to N1:Negotiating to initiate a new login exchange

6.5.5 Port logout information unit

After sending a Port Logout IU and before receiving the corresponding acknowledgement IU, a port may discard without acknowledgement any frame, other than an acknowledgement IU, received.

Upon receiving a Port Logout IU, a DT Device port shall:

a) abort all open exchanges;
b) disable Asynchronous Event Reporting;
c) disable initiating Port Login exchanges; and
d) set port operating parameters to default following transmission of the ACK IU for the Port Logout IU (see 4.2).

Upon receiving a Port Logout IU, an automation port shall:

a) abort all open exchanges;
b) disable initiating Port Login exchanges; and
c) set port operating parameters to default following transmission of the ACK IU for the Port Logout IU (see 4.2).

If a DTD port sends a Port Logout IU to an automation port, it should send a Port Login IU to the automation port within the logout duration time.

Knowledge of the logged out state may be volatile, as a result of a hard reset condition in the logged out port may cause the port to become active again and attempt to log in to the attached port.

The payload of the Port Logout IU is shown in table 16.

Table 16 — Port Logout IU payload contents

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	LOGOUT DURATION						
1								(LSB)

The LOGOUT DURATION field contains the length in seconds that the port that receives the Port Logout IU shall remain in P3:Logged-out state. A value of zero indicates that the port that receives the Port Logout IU shall remain in the P3:Logged-out state until it receives a Port Login IU.

After a port sends an ACK IU in response to a Port Logout IU it shall set its operating parameters to default and enter the P3:Logged-Out state. Once the originator of a Port Logout IU receives an ACK IU for that exchange, it shall set its operating parameters to default and enter the P0:Initial state. See 4.3 for a definition of the port states.

6.5.6 Pause information unit

A Pause IU may be sent by an automation device port to temporarily stop traffic on the service delivery subsystem. DT device ports shall not initiate a Pause IU exchange. If a DT device port receives a Pause IU, it shall acknowledge the frame and, if the acknowledgement is an ACK, temporarily discontinue sending any more frames on the service delivery subsystem. Once in the T2:Paused state, receipt of any valid frame other than an acknowledgement IU shall place the port back into T1:Active state. The T2:Paused state is volatile, as a result a power cycle or other hard reset condition in the paused device may cause the port to become active again. The T2:Paused state only affects the sending of frames, a DT device port shall always be capable of receiving frames.

An automation device port shall be capable of receiving frames unless it has placed the attached DT device port into T2:Paused state. The automation device port should consider the DT device port in the T2:Paused state after it receives an ACK IU in response to a Pause IU. See 4.3.4.4 for a list of events that take the DT device port out of T2:Paused state. An automation device port shall not send a Pause IU unless it is in T1:Active state. It may send a Pause IU independent of any error recovery that is in progress.

An automation device port that receives a Pause IU shall respond with a NAK IU with a status code of INVALID OR ILLEGAL IU RECEIVED (see table 14).

The Pause IU shall contain zero bytes of payload.

6.5.7 NOP information unit

A NOP IU may be sent by a port to cause the other device's port to transition from the T2:Paused state to the T1:Active state (see 4.3.4.4.2). A port that receives a NOP IU and is not in the T2:Paused state shall send an acknowledgement IU and take no further action.

The NOP IU shall contain zero bytes of payload.

6.5.8 Initiate Recovery information unit

An initiate Recovery IU shall be sent by a port when it detects an error has occurred with a frame that it sent. The FRAME NUMBER field in the ADT frame header shall contain the frame number of the frame in error. The X_ORIGIN and EXCHANGE ID fields are undefined and shall be ignored. An Initiate Recovery IU shall contain zero bytes of payload. See 4.6 for a full explanation of the error recovery process.

6.5.9 Initiate Recovery ACK information unit

This information unit is identical to the ACK IU, but it is used exclusively as a response to the Initiate Recovery IU.

6.5.10 Initiate Recovery NAK information unit

This information unit is identical to the NAK IU, but it is used exclusively as a response to the Initiate Recovery IU.

6.5.11 Link service exchange lifetime

6.5.11.1 Link service exchange types

Link service exchanges may be negotiation exchanges, port logout exchanges, pause exchanges, or NOP exchanges.

6.5.11.2 Simple link service exchange lifetime

Port logout IUs, Pause IUs, and NOP IUs are sent in simple exchanges. A simple exchange begins in the sending port with the transmission of the IU and ends with the reception of the corresponding ACK IU or NAK IU with the PR bit set to zero. A simple exchange begins in the receiving port with the reception of a valid IU and ends with the transmission of the ACK IU or NAK IU with the PR bit set to zero.

6.5.11.3 Negotiation exchange lifetime

In a port initiating a negotiation exchange, the exchange begins when the port transmits a Port Login IU with the ACCEPT bit set to zero in a nonexistent exchange. In a port not initiating a negotiation exchange, the exchange begins when the port receives a Port Login IU with the ACCEPT bit set to zero in a nonexistent exchange. A negotiation exchange ends in a port when either:

a) the port has sent a Login IU with the ACCEPT bit set to one, received a Login IU with the ACCEPT bit set to one, and sent an ACK IU in response to it;
b) the port has received a Login IU with the ACCEPT bit set to one, sent a Login IU with the ACCEPT bit set to one, and received an ACK IU in response to it; or
c) the port has received a Login IU with a different exchange ID, indicating that negotiation has restarted.

If a port receives a Port Login IU with the ACCEPT bit set to one in a nonexistent exchange, it shall transmit a NAK IU with a status code of INVALID EXCHANGE ID (see table 14) and discard the Port Login IU.

7 Transport layer

7.1 SCSI Encapsulation

7.1.1 SCSI encapsulation overview

SCSI information units contain information required to implement the SCSI protocol. The X_ORIGIN bit in the ADT frame header conveys the SCSI initiator port and SCSI target port identities. The EXCHANGE ID value from the ADT frame header of an encapsulated SCSI protocol IU takes on the role of the task tag from SAM-2. The LUN is included in the SCSI Command IU and SCSI Task Management IU payload contents. See 4.9 for transport protocol variations from SAM-2. See clause 8 for the mapping of the IUs described in this clause to the SCSI transport protocol services.

Table 17 defines the values for the FRAME TYPE field in the ADT frame header for encapsulated SCSI protocol information units.

Table 17 — SCSI protocol information units

Frame Type	Description
0h	SCSI Command
1h	SCSI Response
2h	SCSI Transfer Ready
3h	SCSI Data
4h	SCSI Task Management
5h - Fh	Reserved

7.1.2 SCSI Command information unit

The SCSI Command IU payload shall contain the information described in Table 18.

Table 18 — SCSI Command IU payload contents

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	LUN						
1								(LSB)
2	Reserved				TASK ATTRIBUTE			
3	Reserved							
4	CDB							
19								
20	(MSB)	FIRST DATA-IN BURST LENGTH						
23								(LSB)

The LUN field indicates the Logical Unit Number to which the command shall be routed within the SCSI target device. If the addressed logical unit does not exist, the task manager shall follow the rules for selection of invalid logical units defined in SAM-2.

The TASK ATTRIBUTE field is defined in table 19.

Table 19 — TASK ATTRIBUTE field values

Value	Task Attribute	Description
0h	SIMPLE	Requests that the task be managed according to the rules for a simple task attribute (see SAM-2).
1h	HEAD OF QUEUE	Requests that the task be managed according to the rules for a head of queue task attribute (see SAM-2).
2h	ORDERED	Requests that the task be managed according to the rules for an ordered task attribute (see SAM-2).
3h	Reserved	
4h	ACA	Requests that the task be managed according to the rules for an automatic contingent allegiance task attribute (see SAM-2).
5h - Fh	Reserved	

The FIRST DATA-IN BURST LENGTH field indicates the size of the buffer that has been allocated to receive data within the initiator port. A non-zero value in the FIRST DATA-IN BURST LENGTH field requests that the target port transfer one or more IUs having a total data length that does not exceed first data-in burst length. This has the same effect as a SCSI Transfer Ready IU with a BUFFER OFFSET field of zero and a BURST LENGTH field of FIRST DATA-IN BURST LENGTH. A value of zero in the FIRST DATA-IN BURST LENGTH field indicates that no space has been allocated in the initiator port and no data shall be sent by the target port until it receives a SCSI Transfer Ready IU. An initiator port may put a non-zero value in the FIRST DATA-IN BURST LENGTH field of any SCSI Command IU. The target port shall ignore the value in the FIRST DATA-IN BURST LENGTH field for non-data or data-out commands.

NOTE 6 Putting a non-zero value in the FIRST DATA-IN BURST LENGTH field of a SCSI Command IU leaves the determination of data direction with the device server.

If an ADT target port receives a SCSI Command IU and the payload of the frame is not 24 bytes, the ADT target port shall return a SCSI Response IU with the RESPONSE CODE field set to INVALID FIELD IN ENCAPSULATED SCSI IU (EXCLUDES CDB).

The task manager is required to detect overlapped commands and handle them as described in SAM-2.

7.1.3 SCSI Task Management information unit

Table 20 defines the SCSI Task Management IU. The SCSI Task Management IU is sent by an initiator port to request that a task management function be processed by a task manager in a logical unit.

Table 20 — SCSI Task Management IU payload contents

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)	LOGICAL UNIT NUMBER						
1								(LSB)
2	TASK MANAGEMENT FUNCTION							
3	Reserved					TAG OF TASK TO BE MANAGED		

The LOGICAL UNIT NUMBER field contains the address of the logical unit. The structure of the logical unit number field shall be as defined in SAM-2. If the addressed logical unit does not exist, the task manager shall return a SCSI Response IU with its RESPONSE CODE field set to INVALID LOGICAL UNIT NUMBER IN SCSI TASK MANAGEMENT IU.

Table 21 defines the values for the TASK MANAGEMENT FUNCTION field. See SAM-2 and SAM-3 for a definition of the task management functions provided.

Table 21 — TASK MANAGEMENT FUNCTION values

Value	Description
00h	Reserved
01h	ABORT TASK
02h	ABORT TASK SET
03h	Reserved
04h	CLEAR TASK SET
05h - 07h	Reserved
08h	LOGICAL UNIT RESET
09h - 3Fh	Reserved
40h	CLEAR ACA
41h - 7Fh	Reserved
80h	QUERY TASK
81h - FFh	Reserved

If TASK MANAGEMENT FUNCTION is set to ABORT TASK or QUERY TASK, the TAG OF TASK TO BE MANAGED field specifies the tag value (see 7.1.1) from the SCSI Command IU that contained the task to be aborted or queried. For all other task management functions, the TAG OF TASK TO BE MANAGED field shall be ignored.

A LOGICAL UNIT RESET shall perform the logical unit reset actions specified in SAM-2 before returning a SCSI Response IU indicating function complete.

If an ADT target port receives a SCSI Task Management IU and the payload of the frame is not 4 bytes, it shall return a SCSI Response IU with the RESPONSE CODE field set to INVALID FIELD IN ENCAPSULATED SCSI IU (EXCLUDES CDB).

If an ADT target port receives a SCSI Task Management IU with an exchange ID that is already in use, it may return a NAK IU with a status code of INVALID EXCHANGE ID (see table 14).

7.1.4 SCSI Response information unit

A SCSI Response IU shall be returned to the exchange Initiator for every SCSI Command IU and SCSI Task Management IU that is received. Table 22 defines the payload of a SCSI Response IU.

Table 22 — SCSI Response IU payload contents

Bit / Byte	7	6	5	4	3	2	1	0
0	RESPONSE CODE							
1	SCSI STATUS							
2	(MSB)	SENSE LENGTH						
3								(LSB)
4	SCSI AUTOSENSE DATA							
n								

The RESPONSE CODE field indicates the results of the operation as an extension to the SCSI Status. Table 23 defines the values for this field.

Table 23 — RESPONSE CODE values

Value	Description
00h	COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE
01h	MORE DATA TRANSFERRED THAN REQUESTED
02h	INVALID FIELD IN ENCAPSULATED SCSI IU (EXCLUDES CDB)
03h	TASK MANAGEMENT FUNCTION NOT SUPPORTED
04h	TASK MANAGEMENT FUNCTION FAILED
05h	COMMAND COMPLETE WITH UNIT ATTENTION
06h	INVALID LOGICAL UNIT NUMBER IN SCSI TASK MANAGEMENT IU
07h	SERVICE DELIVERY FAILURE
08h - FFh	Reserved

The response code value of COMMAND COMPLETE WITH UNIT ATTENTION shall be sent by the remote SMC device server if bridging is enabled and a command completes with a SCSI status of GOOD that results in the generation of a unit attention to initiator ports other than the one that initiated the command. In this case, the SCSI STATUS field shall contain GOOD, INTERMEDIATE, or INTERMEDIATE-CONDITION MET and the SCSI AUTOSENSE

DATA field shall contain the sense data to be reported to those other initiator ports. Additionally, any data cached by the local SMC device server shall be invalidated (see ADC).

The SCSI STATUS field contains SCSI Status as defined in SAM-2. This is only valid if the RESPONSE CODE field is set to COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE or COMMAND COMPLETE WITH UNIT ATTENTION and the exchange was initiated by a SCSI Command IU.

The SENSE LENGTH field indicates the number of bytes of sense data in the SCSI Response IU. This field shall be set to zero if the response code is not COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE or COMMAND COMPLETE WITH UNIT ATTENTION, and no sense data shall be included in the IU. If the response code is COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE and the SCSI STATUS field contains Check Condition, or the response code is set to COMMAND COMPLETE WITH UNIT ATTENTION, autosense data shall be included in the IU as defined in SPC-3 and the SENSE LENGTH field shall be set to indicate how much sense data is included.

7.1.5 SCSI Transfer Ready information unit

A SCSI Transfer Ready IU shall be sent by one port to inform the other port that it is ready to receive data associated with the command. The sender of the SCSI Transfer Ready IU may request all of the data associated with a command with a single SCSI Transfer Ready, or it may use multiple SCSI Transfer Ready IUs within the same exchange to request the data. The contents of the SCSI Transfer Ready IU payload are described in Table 24.

Table 24 — SCSI Transfer Ready IU payload contents

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
				BUFFER OFFSET				
3								(LSB)
4	(MSB)							
				BURST LENGTH				
7								(LSB)

The BUFFER OFFSET field indicates the offset from the beginning of the buffer associated with the first byte of data that shall be sent. Data shall not be requested out of order.

The BURST LENGTH field indicates the size of the buffer that has been allocated to receive data within the sender of the SCSI Transfer Ready IU. The receiver of the SCSI Transfer Ready IU shall respond by transmitting data using one or more SCSI Data IUs until the number of bytes specified by the BURST LENGTH field have been sent.

If an ADT initiator port receives a SCSI Transfer Ready IU that is not 8 bytes long, it shall send an ACK IU and discard the frame. It may then abort the command.

If an ADT target port receives a SCSI Transfer Ready IU that is not 8 bytes long, it shall send an ACK IU, discard the frame and terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of INFORMATION UNIT TOO SHORT or INFORMATION UNIT TOO LONG.

If an ADT initiator port receives a SCSI Transfer Ready IU requesting zero bytes, it may abort the command.

If an ADT target port receives a SCSI Transfer Ready IU requesting zero bytes, it shall send an ACK IU, discard the frame and terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of DATA PHASE ERROR.

If an ADT initiator port receives a SCSI Transfer Ready IU with a requested offset that was not expected, it shall send an ACK IU and discard the frame, and it may abort the command.

If an ADT target port receives a SCSI Transfer Ready IU with a requested offset that was not expected, it shall terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of DATA OFFSET ERROR.

7.1.6 SCSI Data information unit

The SCSI Data IU is used to send data associated with SCSI Data In and Data Out operations. Table 25 describes the contents of a SCSI Data IU.

Table 25 — SCSI Data IU payload contents

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			BUFFER OFFSET				
3								(LSB)
4	(MSB)			DATA LENGTH				
7								(LSB
8				DATA				
n								

The BUFFER OFFSET field indicates the offset from the beginning of the buffer associated with the first byte sent. Data shall not be sent out of order.

The DATA LENGTH field indicates the number of bytes of data included in this IU.

The DATA field contains data.

If an ADT target port receives a SCSI Data IU for which there is no SCSI Transfer Ready IU outstanding or with a data offset that was not expected, it shall send an ACK IU and discard that frame and any subsequent SCSI Data IUs received for that command and, shall terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of DATA OFFSET ERROR.

If an ADT target port receives a SCSI Data IU with more write data than expected (i.e., the length of the SCSI Data IU extends past the end of the expected write data length), it shall send an ACK IU, discard the frame and terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of TOO MUCH WRITE DATA.

If an ADT target port receives a zero length SCSI Data IU, it shall send an ACK IU, discard the frame and terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of INFORMATION UNIT TOO SHORT.

If an ADT initiator port receives a SCSI Data IU with more read data than expected, it shall send an ACK IU, discard the frame, and it may abort the command. Due to a race condition, the ADT initiator port may receive a SCSI Response IU for the command before being able to abort the command.

If an ADT initiator port receives a SCSI Data IU with zero bytes, it shall send an ACK IU, discard the frame, and it may abort the command. Due to a race condition, the ADT initiator port may receive a SCSI Response IU for the command before being able to abort the command.

If an ADT initiator port receives a SCSI Data IU with a data offset that was not expected, it shall send an ACK IU and discard that frame and any subsequent SCSI Data IUs received for that command, and it may abort the command. Due to a race condition, the ADT initiator port may receive a SCSI Response IU for the command before being able to abort the command.

7.1.7 SCSI encapsulation exchange lifetime

A SCSI encapsulation exchange begins in an initiator port after the port transmits a SCSI Command IU or a SCSI Task Management IU. A SCSI encapsulation exchange begins in a target port after the port receives a SCSI Command IU or a SCSI Task Management IU.

A SCSI encapsulation exchange ends in an initiator port after:

a) the port receives a SCSI Response IU for that exchange and sends an ACK IU or NAK IU with the PR bit set to zero in response;
b) the port transmits a SCSI Task Management IU containing a task management request aborting the exchange and receives a SCSI Response IU with a response code value of COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE for the task management request; or
c) an I_T nexus loss occurs.

A SCSI encapsulation exchange ends in a target port after:

a) the port transmits a SCSI Response IU for that exchange and receives an ACK IU or NAK IU with the PR bit set to zero in response;
b) the port receives a SCSI Task Management IU containing a task management request aborting the exchange and transmits a SCSI Response IU with a response code value of COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE for the task management request;
c) a hard reset occurs; or
d) an I_T nexus loss occurs.

7.1.8 Reception of Encapsulated SCSI Information Units in exceptional circumstances

If a port receives a SCSI Response IU, SCSI Transfer Ready IU, or SCSI Data IU in a nonexistent exchange, it shall transmit a NAK IU with a status code of INVALID EXCHANGE ID (see table 14) and discard the SCSI IU.

Within a valid exchange, if a port receives a SCSI Transfer Ready IU for which it has no data to send, it shall transmit an ACK IU and take no further action.

7.2 Fast Access

7.2.1 Fast Access overview

This protocol is intended to provide a feature set beyond what is provided by SAM-2 to both take advantage of the features of the transport layer and work around its slower speed. The Fast Access protocol provides:

a) a simple method for accessing the Very High Frequency (VHF) Data defined in ADC;
b) an asynchronous event report, a method for a DT device to report asynchronous activity; and
c) a method to control these asynchronous reports.

Table 26 defines the values for the FRAME TYPE field in the ADT frame header for Fast Access protocol IUs.

Table 26 — Fast Access protocol IUs

Frame Type	Description
0h	Request for VHF Data IU
1h	VHF Data IU
2h	AER IU
3h	AER Control IU
4h - Fh	Reserved

7.2.2 Payload size – type consistency

Unless otherwise specified in this standard, the receiver of a Fast Access protocol IU shall not consider it an error if the value of the PAYLOAD SIZE field does not match the specified size for those Fast Access protocol IUs that have a specified size. If the size of the payload exceeds the specified size, the IU receiver shall ignore the excess payload bytes except with respect to the calculation of the Checksum field. If the size of the payload is less than the specified size, the IU receiver shall not change the current setting(s) of the parameter(s) controlled by any missing field(s).

7.2.3 Request for VHF Data information unit

Only automation device ports may initiate a Request for VHF Data IU. This IU has no payload.

7.2.4 VHF Data information unit

A VHF Data IU shall be returned by a DT device port in response to a Request for VHF Data IU. The VHF Data IU shall use the same exchange ID used by the Request for VHF Data IU. Only DT devices may initiate a VHF Data IU. The payload of the VHF Data IU shall contain the VHF data as defined in ADC.

7.2.5 AER information unit

Asynchronous Event Report (AER) IUs may be used to report that an event has occurred. Only a DT device port may initiate AER IUs. The payload of an AER IU shall contain the VHF Data as defined in ADC.

7.2.6 AER Control information unit

An AER Control IU may be sent by an automation device port to a DT device port to enable or disable AER reporting. The payload of an AER Control IU shall contain a VHF Data structure, with the bits set to one for each field that the device shall report a change. Multiple-bit fields shall have either all of the bits of the field set to one or all of the bits in the field set to zero. A DT device that receives a multi-bit field containing at least one bit set to zero shall treat that entire field as set to zero. A DT device shall consider reserved bits as not supported for AER notification.

DT devices that do not support AER shall send a NAK IU in response with a status code of UNSUPPORTED FRAME TYPE FOR SELECTED PROTOCOL (see table 14).

Except as noted in this sub clause, DT devices that support AER shall respond to the receipt of an AER Control IU by sending an AER Control IU back to the automation device with the same X_ORIGIN and EXCHANGE ID values. The payload of the IU shall contain a VHF Data IU data structure. Each field that has been enabled for AER notification and is supported by the device shall have all bits in the field set to one. Each field that has been either disabled for

AER notification or is not supported for AER notification by the device shall be set to zero. The default setting for all AER events in a DT device shall be zero.

All AER control fields shall be set to zero by the DT device at the start of the port login process when the AOE bit is set to one.

DT devices ports shall only send an AER Control IU in response to receiving an AER Control IU from an automation device port.

7.2.7 Fast Access exchange lifetime

7.2.7.1 Fast Access exchange types

Fast Access exchanges may be either VHF Data exchanges, AER Control exchanges, or AER exchanges.

7.2.7.2 VHF Data exchange lifetime

A VHF data exchange begins in an automation device port after the port transmits a Request for VHF Data IU. The exchange begins in a DT device port after the port receives a Request for VHF Data IU.

A VHF data exchange ends in an automation device port after the port receives a VHF Data IU for the exchange and sends an ACK IU or a NAK IU with the PR bit set to zero in response to it. The exchange ends in a DT device port after the port transmits a VHF Data IU for the exchange and receives an ACK IU or NAK IU with the PR bit set to zero in response to it.

If an automation device port receives a VHF Data IU for a nonexistent exchange, it shall transmit a NAK IU with a status code of INVALID EXCHANGE ID (see table 14) and discard the VHF Data IU. If a DT device port receives a VHF Data IU, it shall transmit a NAK IU with a status code of UNSUPPORTED FRAME TYPE FOR SELECTED PROTOCOL (see table 14) and discard the VHF Data IU.

7.2.7.3 AER Control exchange lifetime

An AER control exchange begins in an automation device port after the port transmits an AER Control IU. The exchange begins in a DT device port after the port receives an AER Control IU.

An AER control exchange ends in an automation device port after the port receives an AER Control IU and sends an ACK IU or a NAK IU with the PR bit set to zero in response to it. The exchange ends in a DT device port after the port transmits an AER Control IU and receives an ACK IU or NAK IU with the PR bit set to zero in response to it.

If an automation device port receives an AER Control IU in a nonexistent exchange, it shall transmit a NAK IU with a status code of INVALID EXHANGE ID (see table 14) and discard the AER Control IU.

7.2.7.4 AER exchange lifetime

AER exchanges are simple exchanges. The exchange begins in a DT Device port with the transmission of an AER IU and ends with the reception of the corresponding ACK IU or NAK IU with the PR bit set to zero. An AER exchange begins in an automation device port with the reception of a valid AER IU and ends with the transmission of the corresponding ACK IU or NAK IU with the PR bit set to zero.

If a DT device port receives an AER Data IU, it shall transmit a NAK IU with a status code of UNSUPPORTED FRAME TYPE FOR SELECTED PROTOCOL (see table 14) and discard the AER Data IU.

8 SCSI Application layer

8.1 SCSI Transport protocol services overview

An application client requests the processing of a SCSI command by invoking SCSI transport protocol services, the collective operation of which is conceptually modeled in the following remote procedure call (see SAM-2):

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))

ADT defines the transport protocol services required by SAM-2 in support of this remote procedure call (see 8.2).

An application client requests the processing of a SCSI task management function by invoking SCSI transport protocol services, the collective operation of which is conceptually modeled in the following remote procedure calls (see SAM-2):

a) Service Response = **ABORT TASK (IN (I_T_L_Q Nexus));**
b) Service Response = **ABORT TASK SET (IN (I_T_L Nexus));**
c) Service Response = **CLEAR ACA (IN (I_T_L Nexus));**
d) Service Response = **CLEAR TASK SET (IN (I_T_L Nexus));**
e) Service Response = **LOGICAL UNIT RESET (IN (I_T_L Nexus));**
f) Service Response = **QUERY TASK (IN (Nexus));** and

ADT defines the transport protocol services required by SAM-2 in support of these remote procedure calls (see 8.3).

See Annex A for specific examples that illustrate the use of the SCSI transport protocol services and the interaction between the ADT ports, the application client, and the device server.

Table 27 describes the mapping of the remote procedure calls to transport protocol services and the ADT implementation of each transport protocol service.

Table 27 — Remote procedure call mapping

Remote procedure call	Type of transport protocol service	Transport protocol service interaction	Transport protocol service	I/T[a]	ADT implementation	Sub-clause
Execute Command	Request/ Confirmation	Request	**Send SCSI Command**	I	SCSI Command IU	8.2.1
		Indication	**SCSI Command Received**	T	Receipt of SCSI Command IU	8.2.2
		Response	**Send Command Complete**	T	SCSI Response IU	
		Confirmation	**Command Complete Received**	I	Receipt of SCSI Response IU	8.2.4
	Data Transfer	Request	**Send Data-In**	T	SCSI Data IU	8.2.5
		Confirmation	**Data-In Delivered**	T	Positive acknowledgement of the last SCSI Data IU	8.2.6
		Request	**Receive Data-Out**	T	SCSI Transfer Ready IU	8.2.7
		Confirmation	**Data-Out Received**	T	Receipt of last SCSI Data IU	8.2.8
ABORT TASK, ABORT TASK SET, CLEAR ACA, CLEAR TASK SET, LOGICAL UNIT RESET and QUERY TASK	Request/ Confirmation	Request	**Send Task Management Request**	I	SCSI Task Management IU	8.3.1
		Indication	**Task Management Request Received**	T	Receipt of SCSI Task Management IU	8.3.2
		Response	**Task Management Function Executed**	T	SCSI Response IU	8.3.3
		Confirmation	**Receive Task Management Function-Executed**	I	Receipt of SCSI Response IU	8.3.4

[a] I/T indicates whether the initiator port (I) or the target port (T) implements the transport protocol service.

Table 26 describes the transport protocol services provided by ADT that are extensions of the services required by SAM-2. See 4.9 for details of the use of these transport protocol services.

Table 28 — Extended remote procedure call mapping

Remote procedure call	Type of transport protocol service	Transport protocol service interaction	Transport protocol service	I/T[a]	ADT implementation	Sub-Clause
Execute Command	Data Transfer	Request	**Send Data-Out**	I	SCSI Data IU	8.2.9
		Confirmation	**Data-Out Delivered**	I	Positive acknowledgement of the last SCSI Data IU	8.2.10
		Request	**Receive Data-In**	I	SCSI Transfer Ready IU	8.2.11
		Confirmation	**Data-In Received**	I	Positive acknowledgement of the last SCSI Data IU containing data to satisfy the SCSI Transfer Ready IU	8.2.12

[a] I/T indicates whether the initiator port (I) or the target port (T) implements the transport protocol service.

8.2 Transport layer protocol services to support Execute Command

8.2.1 Send SCSI Command transport protocol service

An application client uses the **Send SCSI Command** transport protocol service to request that an ADT initiator port transmit a SCSI Command IU containing a SCSI command.

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]))

Table 29 shows how the arguments to the **Send SCSI Command** transport protocol service are used in the generation of a SCSI Command IU.

Table 29 — Send SCSI Command transport layer protocol service arguments (Sheet 1 of 2)

Argument	ADT Implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T is used to set the X_ORIGIN field; b) L is used to set the LUN field; and c) Q is set by the transport layer.
CDB	Used to set the CDB field.
[Task Attribute]	Used to set the TASK ATTRIBUTE field.
[Data-In Buffer Size]	Used to set the FIRST DATA-IN BURST LENGTH field.

Table 29 — Send SCSI Command transport layer protocol service arguments (Sheet 2 of 2)

Argument	ADT Implementation
[Data-Out Buffer]	Buffer of data to send.
[Data-Out Buffer Size]	Maximum of 2^{32}-1
[Autosense Request]	True
[CRN]	Not used.

8.2.2 SCSI Command Received transport protocol service

An ADT target port uses the **SCSI Command Received** transport protocol service to notify a device server that it has received a SCSI Command IU.

SCSI Command Received (IN (I_T_L_x Nexus, CDB, [Task Attribute], [Autosense Request], [Command Reference Number]))

Table 30 shows how the arguments to the **SCSI Command Received** transport protocol service are used.

Table 30 — SCSI Command Received transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T is indicated by the X-ORIGIN field; b) L is indicated by the LUN field; and c) Q is indicated by the EXCHANGE ID field in the ADT frame header.
CDB	From the CDB field in the SCSI Command IU
[Task Attribute]	From the TASK ATTRIBUTE field.
[Autosense Request]	True
[CRN]	Not used.

8.2.3 Send Command Complete transport protocol service

A device server uses the **Send Command Complete** transport protocol service to request an ADT target port to transmit a SCSI Response IU.

Send Command Complete (IN (I_T_L_x Nexus, [Sense Data], Status, Service Response))

Table 31 shows how the arguments to the **Send Command Complete** transport protocol service are used in the generation of a SCSI Response IU.

Table 31 — Send Command Complete transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	From the **SCSI Command Received** transport protocol service call that established the task.
[Sense Data]	Used to set the AUTOSENSE DATA field.
Status	Used to set the SCSI STATUS field.
Service Response	Used to set the RESPONSE CODE and STATUS fields: a) TASK COMPLETE: the RESPONSE CODE field is set to COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE or COMMAND COMPLETE WITH UNIT ATTENTION and the STATUS field is set to a value other than INTERMEDIATE or INTERMEDIATE-CONDITION MET; b) LINKED COMMAND COMPLETE: The RESPONSE CODE field is set to COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE or COMMAND COMPLETE WITH UNIT ATTENTION and the STATUS field is set to INTERMEDIATE or INTERMEDIATE-CONDITION MET; or c) SERVICE DELIVERY OR TARGET FAILURE: The RESPONSE CODE field is set to SERVICE DELIVERY FAILURE.

8.2.4 Command Complete Received transport protocol service

An ADT initiator port uses the **Command Complete Received** transport protocol service to notify an application client that it has received a response for a SCSI Command IU initiated by a **Send SCSI Command** transport protocol service (e.g., a SCSI Response IU or a NAK IU).

Command Complete Received (IN (I_T_L_x Nexus, [Data-In Buffer], [Sense Data], Status, Service Response))

Table 32 shows how the arguments to the **Command Complete Received** transport protocol service are determined.

Table 32 — Command Complete Received transport layer protocol service arguments (Sheet 1 of 2)

Argument	ADT Implementation
I_T_L_x nexus	I_T_L_Q nexus, where: a) I_T is indicated by the X-ORIGIN field; b) L is indicated by the LUN field; and c) Q is indicated by the EXCHANGE ID field in the ADT frame header.
[Data-In Buffer]	Pointer to a buffer containing command specific information returned by the logical unit on command completion.
[Sense Data]	From the SCSI Response IU SCSI AUTOSENSE DATA field.

Table 32 — Command Complete Received transport layer protocol service arguments (Sheet 2 of 2)

Argument	ADT Implementation
Status	From the SCSI Response IU SCSI STATUS field.
Service Response	From the SCSI Response IU RESPONSE CODE and STATUS field, or from a NAK on the SCSI Command IU: a) TASK COMPLETE: The RESPONSE CODE field is set to COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE or COMMAND COMPLETE WITH UNIT ATTENTION and the STATUS field is set to a value other than INTERMEDIATE or INTERMEDIATE-CONDITION MET; b) LINKED COMMAND COMPLETE: The RESPONSE CODE field is set to COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE or COMMAND COMPLETE WITH UNIT ATTENTION and the STATUS field is set to INTERMEDIATE or INTERMEDIATE-CONDITION MET; or c) SERVICE DELIVERY OR TARGET FAILURE: RESPONSE CODE field is set to Service delivery failure.

8.2.5 Send Data-In transport protocol service

A device server uses the **Send Data-In** transport protocol service to request that an ADT target port transmit data to an initiator port using one or more SCSI Data IUs. A target port shall send one or more SCSI Data IUs as a result of a **Send Data-In** transport protocol service invocation if:

a) it has received a SCSI Transfer Ready IU requesting the data (see 7.1.5); or
b) the SCSI Command IU contained a non-zero value in the FIRST DATA-IN BURST LENGTH field (see 7.1.2).

Send Data-In (IN (I_T_L_x Nexus, Device Server Buffer, Application Client Buffer Offset, Request Byte Count))

A device server shall only call **Send Data-In** during a read or bi-directional command.

A device server shall not call **Send Data-In** for a given I_T_L_Q nexus after it has called **Send Command Complete** for that I_T_L_Q nexus (e.g., a SCSI Response IU with that I_T_L_Q nexus) or called **Task Management Function Executed** for a task management function that terminates that task (e.g., an ABORT TASK).

Table 33 shows how the arguments to the **Send Data-In** transport protocol service are used.

Table 33 — Send Data-In transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	From the **SCSI Command Received** transport protocol service call that established the task.
Device Server Buffer	Pointer to a buffer where the data is located.
Application Client Buffer Offset	Used to set the BUFFER OFFSET field in the first SCSI Data IU. The transport layer may use more than one SCSI Data IU to transmit the data. If it does, the BUFFER OFFSET field in each subsequent SCSI Data IU shall be set adjusted by the number of bytes in the previous SCSI Data IU.
Request Byte Count	Total number of bytes to transmit. If multiple SCSI Data IUs are used to transmit the data, the total bytes transmitted shall equal the **Request Byte Count** value.

8.2.6 Data-In Delivered transport protocol service

An ADT target port uses the **Data-In Delivered** transport protocol service to notify a device server of the results of transmitting the data associated with a **Send Data-In** transport protocol service.

Data-In Delivered (IN (I_T_L_x Nexus))

Table 34 shows how the arguments to the **Data-In Delivered** transport protocol service are determined.

Table 34 — Data-In Delivered transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T_L_x nexus value passed to the **Send Data-In** transport layer protocol service request that initiated the transfer.

8.2.7 Receive Data-Out transport protocol service

A device server uses the **Receive Data-Out** transport protocol service to request that an ADT target port transmit a SCSI Transfer Ready IU.

Receive Data-Out (IN (I_T_L_x Nexus, Application Client Buffer Offset, Request Byte Count, Device Server Buffer))

A device server shall only call **Receive Data-Out** during a write or bi-directional command.

A device server shall not call **Receive Data-Out** for a given I_T_L_Q nexus until **Data-Out Received** has completed successfully for the previous **Receive Data-Out** call (i.e., no SCSI Transfer Ready IU until all write SCSI Data IUs for the previous SCSI Transfer Ready IU have completed, if any, and has provided link layer acknowledgement for all of the previous SCSI Data IUs for that I_T_L_Q nexus).

A device server shall not call **Receive Data-Out** for a given I_T_L_Q nexus after a **Send Command Complete** has been called for that I_T_L_Q nexus or after a **Task Management Function Executed** has been called for a task management function that terminates that task (e.g., an ABORT TASK).

Table 35 shows how the arguments to the **Receive Data-Out** transport protocol service are used.

Table 35 — Receive Data-Out transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	From the **SCSI Command Received** transport protocol service call that established the task.
Application Client Buffer Offset	Used to set the BUFFER OFFSET field in the SCSI Transfer Ready IU.
Request Byte Count	Used to set the BURST LENGTH field in the SCSI Transfer Ready IU.
Device Server Buffer	The buffer in the device server to which data is to be transferred.

8.2.8 Data-Out Received transport protocol service

An ADT target port uses the **Data-Out Received** transport protocol service to notify a device server of the result of the request to receive data initiated by a call to **Receive Data-Out** transport layer protocol service request.

Data-Out Received (IN (I_T_L_x Nexus))

Table 36 shows how the arguments to the **Data-Out Received** transport protocol service are determined.

Table 36 — Data-Out Received transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T_L_x nexus value passed to the **Receive Data-Out** transport layer protocol service request that initiated the transfer.

8.2.9 Send Data-Out transport protocol service

An application client uses the **Send Data-Out** transport protocol services to request that an ADT initiator port transmit data to a target port using one or more SCSI Data IUs. An initiator port shall not send a SCSI Data IU as a result of a **Send Data-Out** transport protocol service invocation until it has received a SCSI Transfer Ready IU requesting the data.

Send Data-Out (IN (I_T_L_x Nexus, Application Client Buffer, Device Server Buffer Offset, Request Byte Count))

An application client shall only call **Send Data-Out** during a write or bi-directional command.

An application client shall not call **Send Data-Out** for a given I_T_L_Q nexus after it has received a **Command Complete Received** confirmation for that I_T_L_Q nexus (e.g., a SCSI Response IU has been received for that I_T_L_Q nexus) or called a task management function that terminates that task (e.g., an ABORT TASK).

Table 37 shows how the arguments to the **Send Data-Out** transport protocol service are used.

Table 37 — Send Data-Out transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	Used to set the X_ORIGIN and EXCHANGE ID fields in the ADT frame(s) header.
Device Server Buffer	Pointer to a buffer where the data is located.
Application Client Buffer Offset	Used to set the BUFFER OFFSET field in the first SCSI Data IU. The transport layer may use more than one SCSI Data IU to transmit the data. If it does, the BUFFER OFFSET field in each subsequent SCSI Data IU shall be set adjusted by the number of bytes in the previous SCSI Data IU.
Request Byte Count	Total number of bytes to transmit. If multiple SCSI Data IUs are used to transmit the data, the total bytes transmitted shall equal the **Request Byte Count** value.

8.2.10 Data-Out Delivered transport protocol service

An ADT Initiator port uses the **Data-Out Delivered** transport protocol service to notify an application client of the results of transmitting the data associated with a **Send Data-Out** transport protocol service.

Data-Out Delivered (IN (I_T_L_x Nexus))

Table 38 shows how the arguments to the **Data-Out Delivered** transport protocol service are determined.

Table 38 — Data-Out Delivered transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T_L_x nexus value passed to the **Send Data-Out** transport layer protocol service request that initiated the transfer.

8.2.11 Receive Data-In transport protocol service

An application client uses the **Receive Data-In** transport protocol service to request that an ADT Initiator port transmit a SCSI Transfer Ready IU.

Receive Data-In (IN (I_T_L_x Nexus, Device Server Buffer Offset, Request Byte Count, Application Client Buffer))

An application client shall only call **Receive Data-In** during a read or bi-directional command.

An application client shall not call **Receive Data-In** for a given I_T_L_Q nexus until it receives a **Data-In Received** confirmation for a previous **Receive Data-In** call (i.e., no SCSI Transfer Ready IU until all SCSI Data IUs for the previous SCSI Transfer Ready IU have completed, if any, and has provided link layer acknowledgement for all of the previous SCSI Data IUs for that I_T_L_Q nexus).

Table 39 shows how the arguments to the **Receive Data-In** transport protocol service are used.

Table 39 — Receive Data-In transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	Used to set the X_ORIGIN and EXCHANGE ID fields in the ADT frame(s) header.
Device Server Buffer Offset	Used to set the BUFFER OFFSET field in the SCSI Transfer Ready IU.
Request Byte Count	Used to set the BURST LENGTH field in the SCSI Transfer Ready IU.
Application Client Buffer	The buffer in the application client to which data is to be transferred.

8.2.12 Data-In Received transport protocol service

An ADT initiator port uses the **Data-In Received** transport protocol service to notify an application client of the result of the request to receive data. The request may be initiated by a call to the **Receive Data-In** transport protocol service or the result of receiving data initiated by a call to the **Send SCSI Command** transport protocol service where the **Data-In Buffer Size** is non-zero.

Data-In Received (IN (I_T_L_x Nexus))

An ADT initiator port shall notify an application client that has called the **Receive Data-In** transport protocol service by calling the **Data-In Received** confirmation when the number of bytes received matches the **Request Byte Count** parameter, or when a SCSI Response IU for the I_T_L_x nexus is received. If a SCSI Response IU for the I_T_L_x nexus is received after **Receive Data-In** has been called but before **Data-In Received** has been called, the initiator port shall call **Data-In Received** before it calls **Command Complete Received**.

Table 40 shows how the arguments to the **Data-In Received** transport protocol service are determined.

Table 40 — Data-In Received transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T_L_x nexus value passed to the **Receive Data-I**n transport protocol service that initiated the transfer.

8.3 Task management protocol services

8.3.1 Send Task Management Request transport protocol service

An application client uses the **Send Task Management Request** transport protocol service to request that an ADT initiator port transmit a SCSI Task Management IU requesting a task management function.

Send Task Management Request (IN (Nexus, Function Identifier))

Table 41 shows how the arguments to the **Send Task Management Request** transport protocol service are used.

Table 41 — Send Task Management Request transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T, I_T_L, or I_T_L_Q nexus identifier used to set the TAG OF THE TASK TO BE MANAGED and LUN fields in the SCSI Task Management IU.
Function Identifier	Used to set the TASK MANAGEMENT FUNCTION field in the SCSI Task Management IU. Only these task management functions are supported: a) ABORT TASK (Nexus argument specifies an I_T_L_Q Nexus); b) ABORT TASK SET (Nexus argument specifies an I_T_L Nexus); c) CLEAR ACA (Nexus argument specifies an I_T_L Nexus); d) CLEAR TASK SET (Nexus argument specifies an I_T_L Nexus); e) LOGICAL UNIT RESET (Nexus argument specifies an I_T_L Nexus); f) QUERY TASK (Nexus argument specifies an I_T_L_Q Nexus)

8.3.2 Task Management Request Received transport protocol service

An ADT target port uses the **Task Management Request Received** transport protocol service to notify a device server that it has received a SCSI Task Management IU containing a task management request.

Task Management Request Received (IN (Nexus, Function Identifier))

Table 42 shows how the arguments to the **Task Management Request Received** transport protocol service are determined.

Table 42 — Task Management Request Received transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T, I_T_L, or I_T_L_Q nexus identifier created from the TAG OF THE TASK TO BE MANAGED and LUN fields in the SCSI Task Management IU.
Function Identifier	From the TASK MANAGEMENT FUNCTION field in the SCSI Task Management IU.

8.3.3 Task Management Function Executed transport protocol service

A device server uses the **Task Management Function Executed** transport protocol service to request that an ADT target port transmit a SCSI Response IU with the results of the task management function.

Task Management Function Executed (IN (Nexus, Service Response))

A device server shall only call **Task Management Function Executed** after receiving **Task Management Request Received**.

Table 43 shows how the arguments to the **Task Management Function Executed** transport protocol service are used.

Table 43 — Task Management Function Executed transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T, I_T_L, or I_T_L_Q nexus from the Task Management Request Received argument list.
Service Response	Used to set the RESPONSE CODE field in the SCSI Response IU: a) FUNCTION REJECTED: The RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION NOT SUPPORTED or INVALID FIELD IN ENCAPSULATED SCSI IU (EXCLUDES CDB); b) FUNCTION COMPLETE: The RESPONSE CODE field is set to COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE; or c) SERVICE DELIVERY OR SUBSYSTEM FAILURE: The RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION FAILED. d) INCORRECT LOGICAL UNIT NUMBER: The RESPONSE CODE field is set to INVALID LOGICAL UNIT NUMBER IN SCSI TASK MANAGEMENT IU (see SAM-3).

8.3.4 Received Task Management Function-Executed transport protocol service

An ADT initiator port uses the **Received Task Management Function-Executed** transport protocol service to notify an application client that it has received a response to **Send Task Management Request** transport protocol service request (e.g., received a SCSI Response IU or a NAK IU).

Received Task Management Function-Executed (IN (Nexus, Service Response))

Table 44 shows how the arguments to the **Received Task Management Function-Executed** transport protocol service are determined.

Table 44 — Received Task Management Function-Executed transport layer protocol service arguments

Argument	ADT Implementation
I_T_L_x nexus	I_T, I_T_L, or I_T_L_Q nexus from the Send Task Management Request argument list.
Service Response	Determined from the RESPONSE CODE field in the SCSI Response IU: a) FUNCTION REJECTED: The RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION NOT SUPPORTED or INVALID FIELD IN ENCAPSULATED SCSI IU (EXCLUDES CDB); b) FUNCTION COMPLETE: The RESPONSE CODE field is set to COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE; or c) SERVICE DELIVERY OR SUBSYSTEM FAILURE: The RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION FAILED. d) INCORRECT LOGICAL UNIT NUMBER: The RESPONSE CODE field is set to INVALID LOGICAL UNIT NUMBER IN SCSI TASK MANAGEMENT IU (see SAM-3).

8.4 SCSI mode parameters

8.4.1 Disconnect-Reconnect mode page

An ADT target port shall not support the Disconnect-Reconnect mode page.

8.4.2 Protocol-Specific Port mode page

An ADT target port shall not support the Protocol-Specific Port mode page.

8.4.3 Protocol-Specific Logical Unit mode page

An ADT target port shall not support the Protocol-Specific Logical Unit mode page.

Annex A
(informative)

SCSI transport protocol service examples

A.1 Introduction

This annex provides specific examples to illustrate the use of the SCSI transport protocol services.

All of these examples assume an error free exchange.

A.2 SCSI command with no data phase

Figure A.1 shows how SCSI transport protocol services are be used to process a SCSI command that has no data phase.

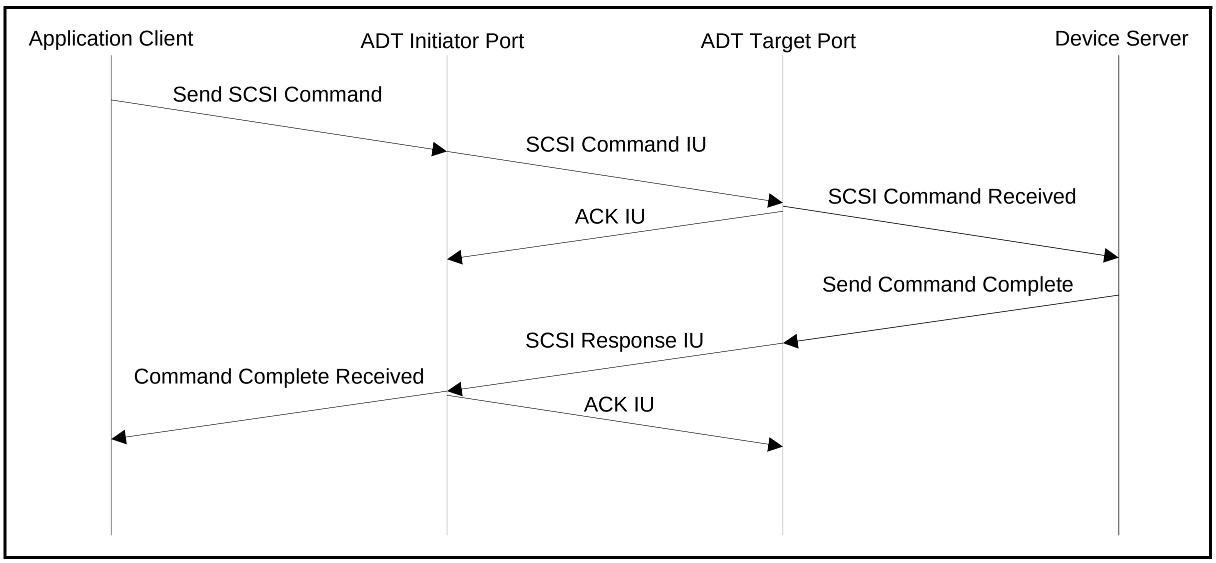


Figure A.1 — SCSI command with no data phase

A.3 SCSI Command with data in

Figure A.2 shows how SCSI transport protocol services may be used to process a SCSI command with a data-in phase. All IUs transferred between the ADT ports are acknowledged using ACK IUs. Most of the ACK IUs are not shown in figure A.2 in an effort to make it more readable. The ACK IUs that are shown are those that have a direct impact on the communication between the ADT target port and device server. There are many possible variations of the order and number of protocol service calls and SCSI Data IUs. This is one example of how a SCSI command with data-in may be accomplished. This example shows a SCSI command that has an overall data-in length of 8192, but the data-in buffer size in the Send SCSI Command request is 4096. The SCSI Command IU has a FIRST DATA-IN BURST LENGTH field value of 4096 which has the effect of a SCSI Transfer Ready IU with a BUFFER OFFSET field of zero and a BURST LENGTH field of 4096 (see 7.1.2).

Once the device server receives the SCSI Command Received indication, it requests 4096 bytes of data to be transferred to the application client. The ADT target port uses multiple SCSI Data IUs to transfer the data to the ADT initiator port. Once it has received the number of bytes specified by the FIRST DATA-IN BURST LENGTH field, it sends a Data-In Received confirmation to the application client. Once the last SCSI Data IU has been acknowledged, the ADT target port notifies the device server using the Data-In Delivered confirmation.

The device server may send more data at any time using the Send Data-In request, but the ADT target port waits until it has received a SCSI Transfer Ready IU (see 7.1.5) before transmitting a SCSI Data IU. When the application client issues a Receive Data-In request, the ADT initiator port transmits a SCSI Transfer Ready IU. The ADT target port responds by transmitting data using one or more SCSI Data IUs until the number of bytes specified by the BURST LENGTH field (see 7.1.5) have been sent.

When the device server has completed sending all the data-in for the command, it uses the Send Command Complete protocol service to request the ADT target port transmit a SCSI Response IU. Upon receiving the SCSI Response IU, the ADT initiator port notifies the application client using the Command Complete protocol service.

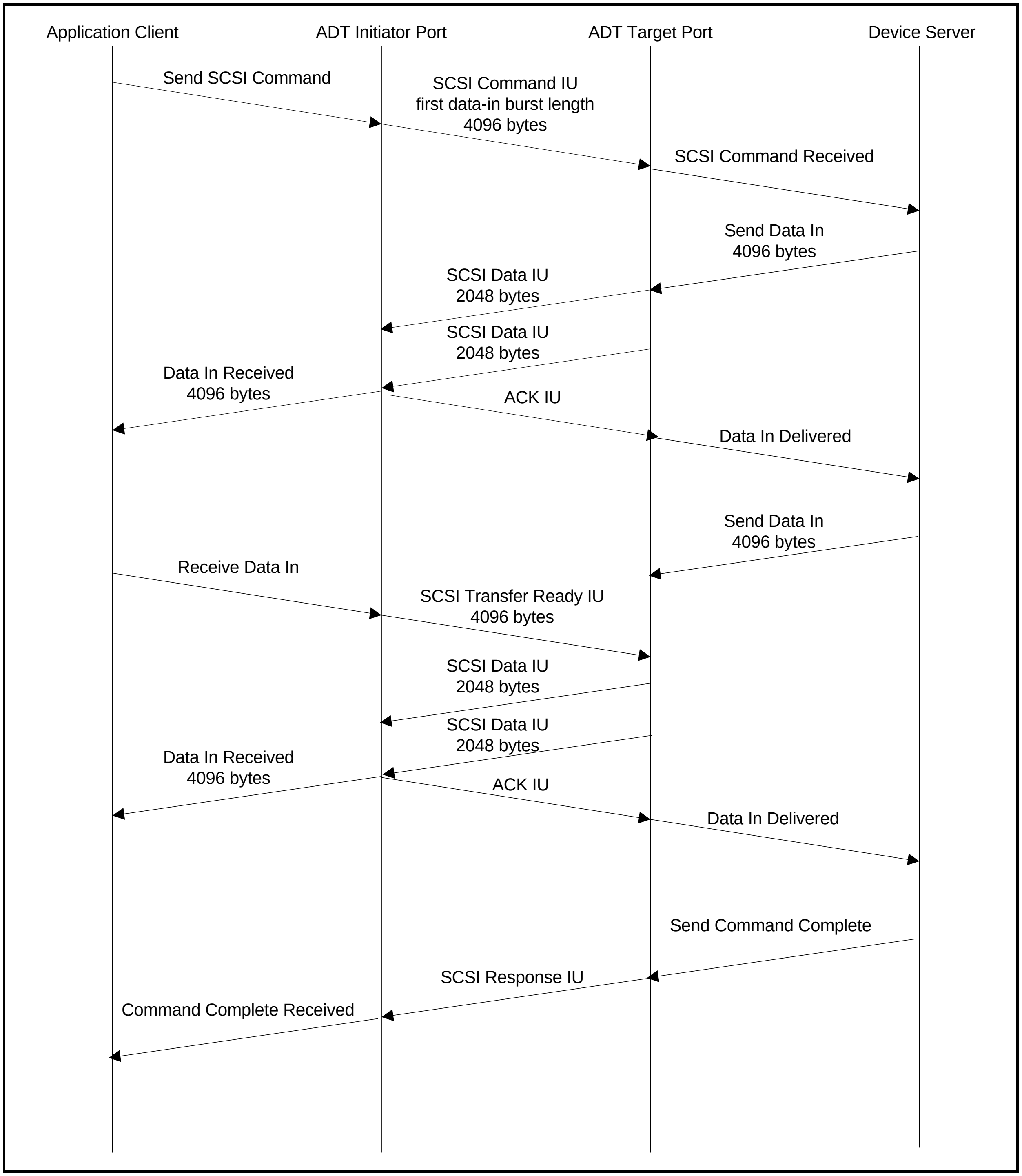


Figure A.2 — SCSI command with data-in

A.4 SCSI Command with data out

Figure A.3 shows how SCSI transport protocol services may be used to process a SCSI command with a data-out phase. All IUs transferred between the ADT ports are acknowledged. Most of the ACK IUs are not shown in figure A.3 in an effort to make it more readable. The ACK IUs that are shown are those that have a direct impact on the communication between the ADT initiator port and application client.

There are many possible variations of the order and number of protocol service calls and SCSI Data IUs. This is one example of how a SCSI command with data-out may be accomplished.

This example shows a SCSI command that has an overall data-out length of 8k, but the data-out buffer size in the Send SCSI Command request is 0k. Instead it uses the Send Data-Out protocol service to request the data to be transferred.

The application client requests 4k of data to be transferred to the device server, using the Send Data-Out protocol service. The ADT initiator port waits to transfer the data until it has received a SCSI Transfer Ready IU (see 7.1.5). The device server uses the Receive Data-Out protocol service to request a SCSI Transfer Ready IU be transmitted. The ADT initiator port uses multiple SCSI Data IUs to transmit the data. Once the last SCSI Data IU has been acknowledged, the ADT initiator port notifies the application client using the Data-Out Delivered confirmation.

Once the ADT target port has received the number of bytes requested it notifies the device server using the Data-Out Received confirmation. This process repeats a second time to transfer the remaining 4k of data.

As shown in figure A.3 the Send Data-Out request may arrive in the ADT Initiator port before or after the SCSI Transfer Ready IU.

When the device server has received all of the data for the command, it uses the Send Command Complete protocol service to request the ADT target port transmit a SCSI Response IU. Upon receiving the SCSI Response IU, the ADT initiator port notifies the application client using the Command Complete protocol service.

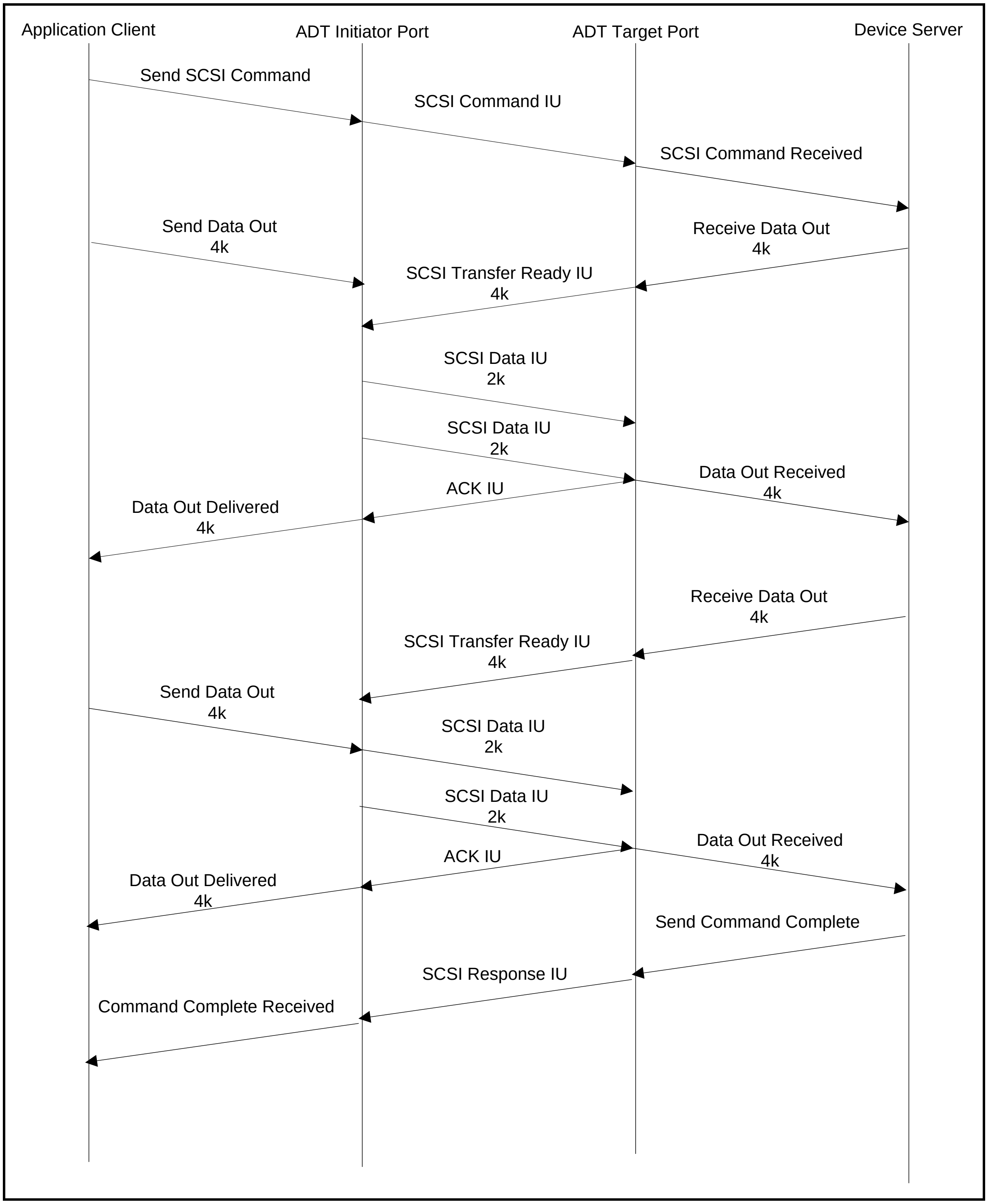


Figure A.3 — SCSI command with data-out

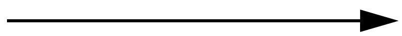
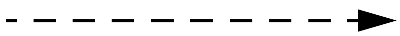
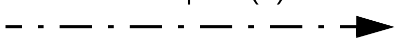
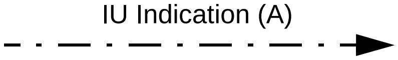
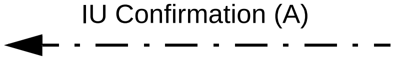
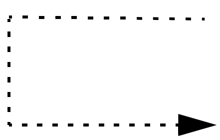
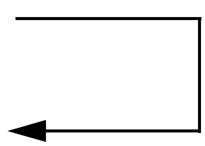
Annex B
(informative)

Error detection and recovery action examples

B.1 Introduction

This annex diagrams various error detection and recovery procedures for ADT ports conforming to this standard. The conventions for the diagrams are shown in table B.1.

Table B.1 — Diagram drawing conventions

Drawing Convention	Meaning
	Acknowledged or Unacknowledged IU
	IU with error other than corruption
	Acknowledgement IU
IU Request (A) 	Service request from upper layer of sender
IU Indication (A) 	Service indication to upper layer of receiver
IU Confirmation (A) 	Service confirmation to upper layer of sender
	Time-out value exceeded
	IU received is processed to transmit IU
X	IU lost or corrupted
FN	Frame Number in the IU
PR	Value of Pending Recovery (PR) bit in the NAK IU
EFN	Value of the port’s Expected Frame Number counter
NFTS	Value of the Next Frame To Send counter
a = b	Counter a is set to expression b
(a == b)	Expression a equals expression b
(a != b)	Expression a does not equal expression b
(condition) => action	Because the condition is true, the action is performed

B.2 Receiver-detected retryable error

Figure B.1 shows the detection of a retryable error by the receiver and the subsequent recovery.

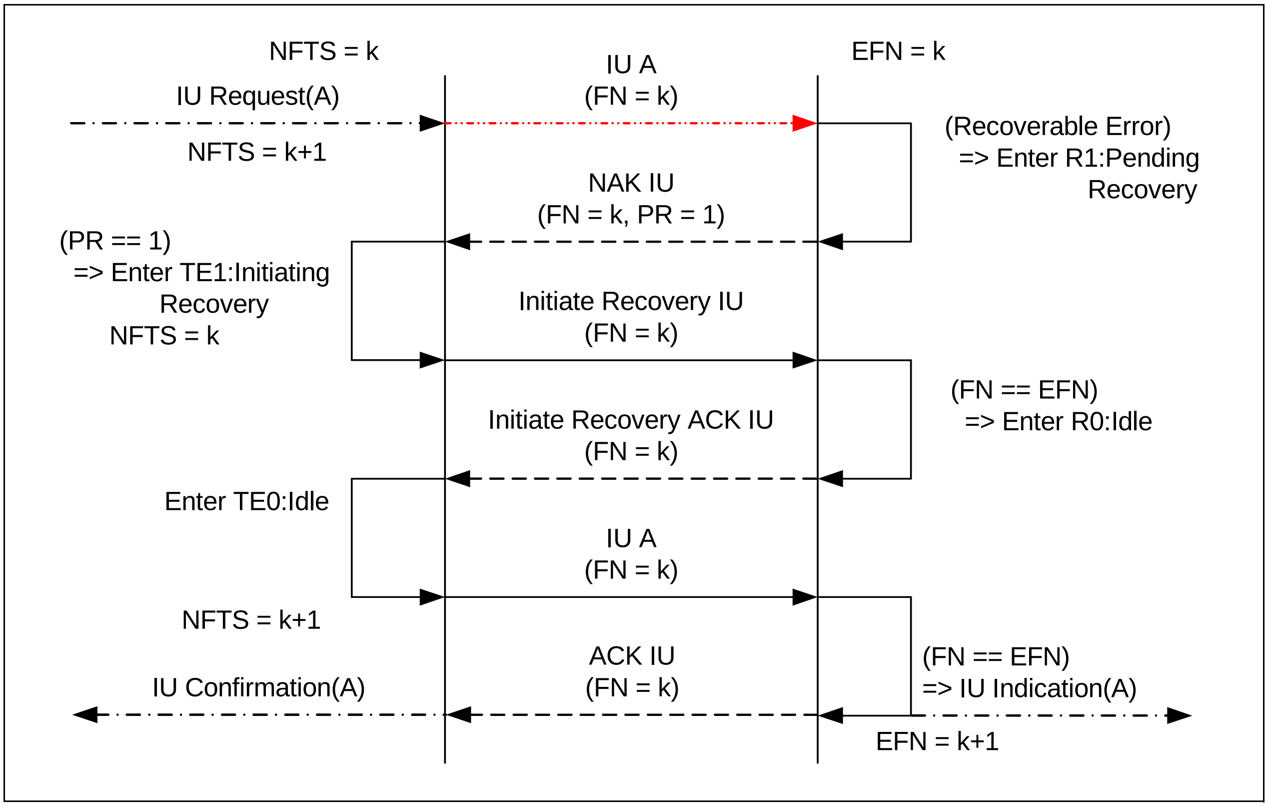


Figure B.1 — Receiver-detected retryable error

B.3 Receiver-detected retryable error with multiple active IUs

Figure B.2 shows the detection of a retryable error by the receiver, when the IU in error is followed by a good IU.

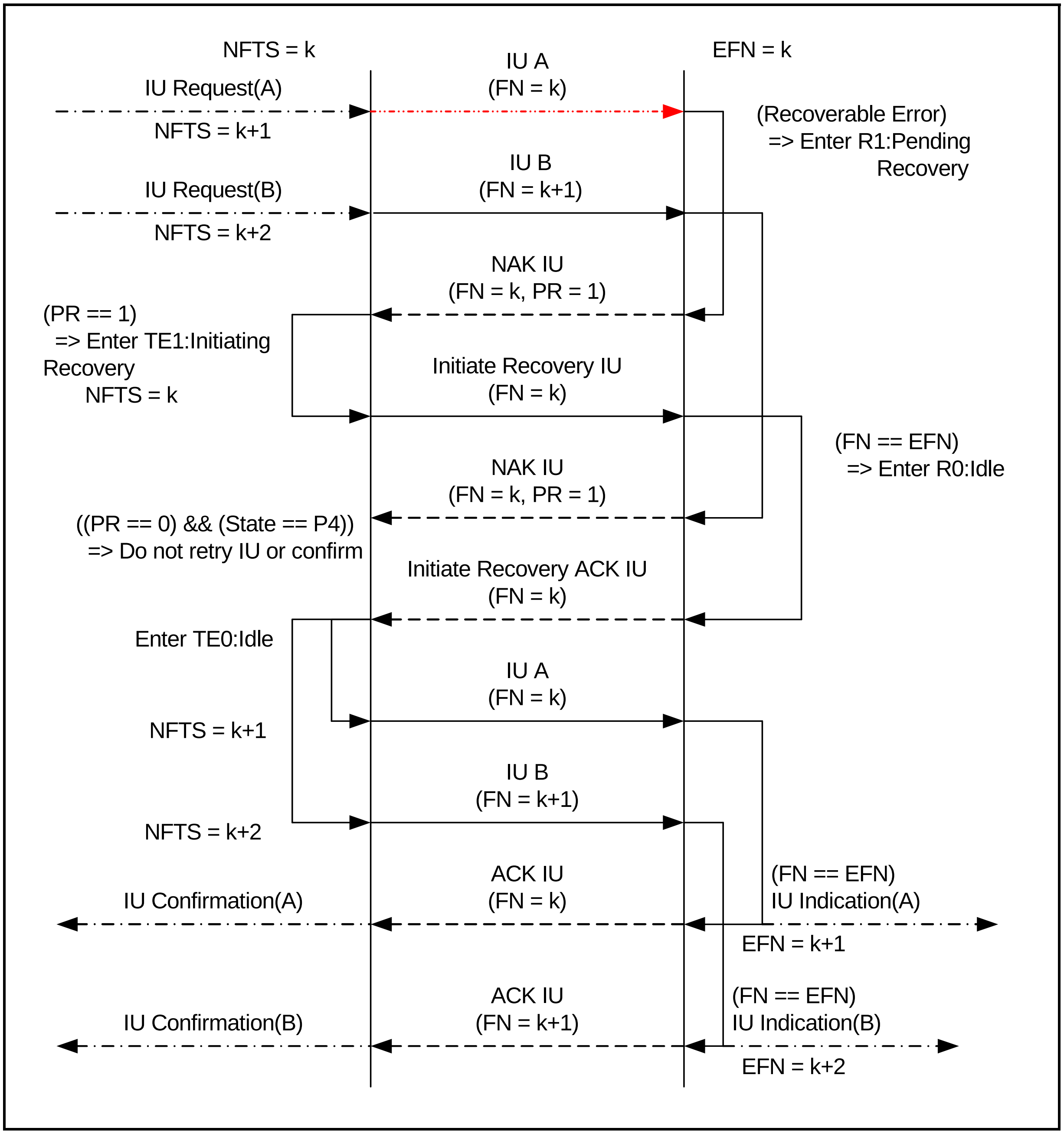


Figure B.2 — Receiver-detected retryable error with multiple active IUs

B.4 Lost IU with no further traffic

Figure B.3 shows a lost or corrupted IU, in which there is no further traffic from the sender. The sender detects the error when a timeout occurs without receipt of an Acknowledgement frame.

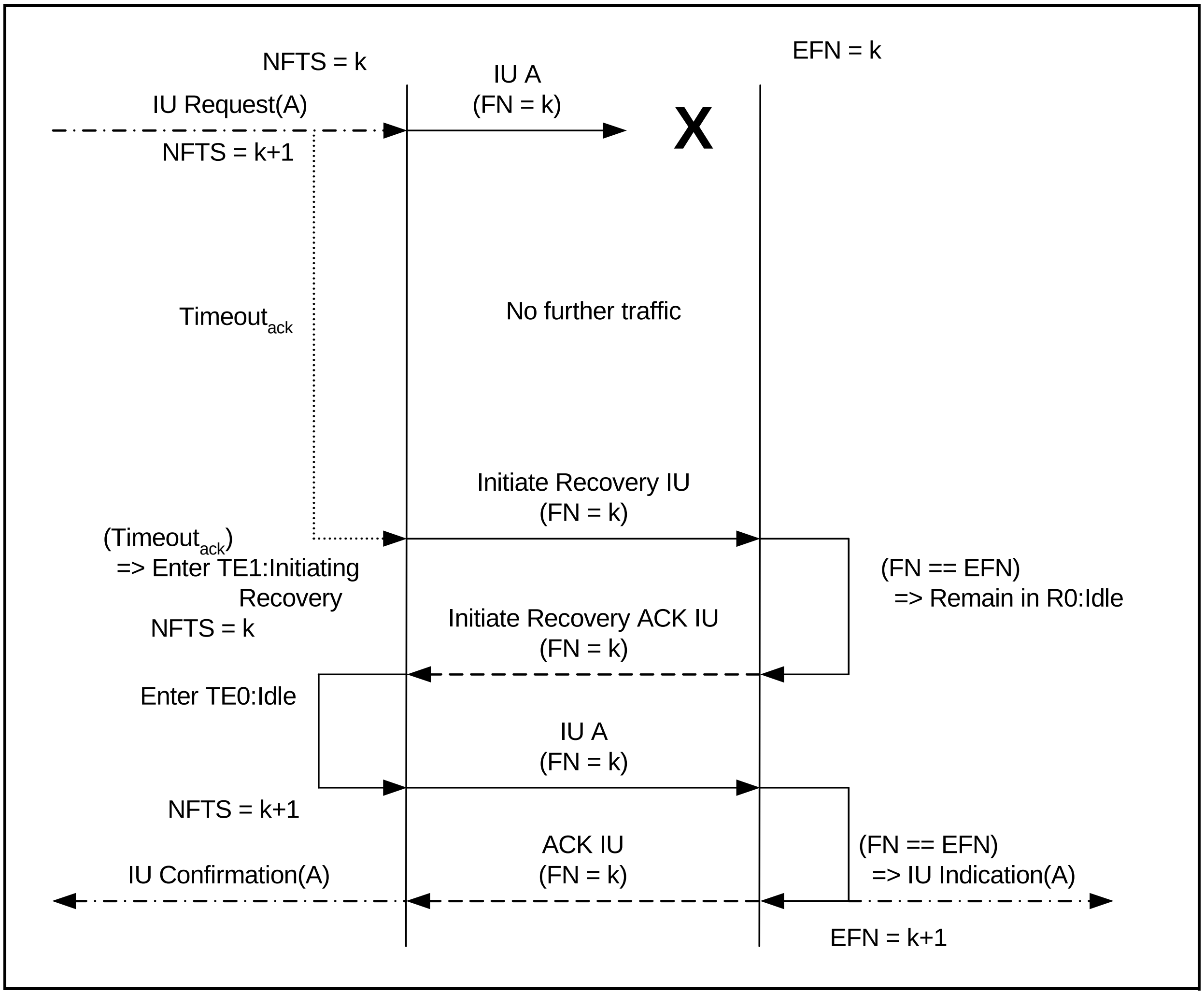


Figure B.3 — Lost IU with no further traffic

B.5 Lost ACK with recovery driven by out-of-order ACK

Figure B.4 illustrates how the requirement for acknowledgement IUs to be sent in the same order that the original IUs are received allows detection of the lost or corrupted acknowledgement for FN k. There is no need to wait for timer expiration.

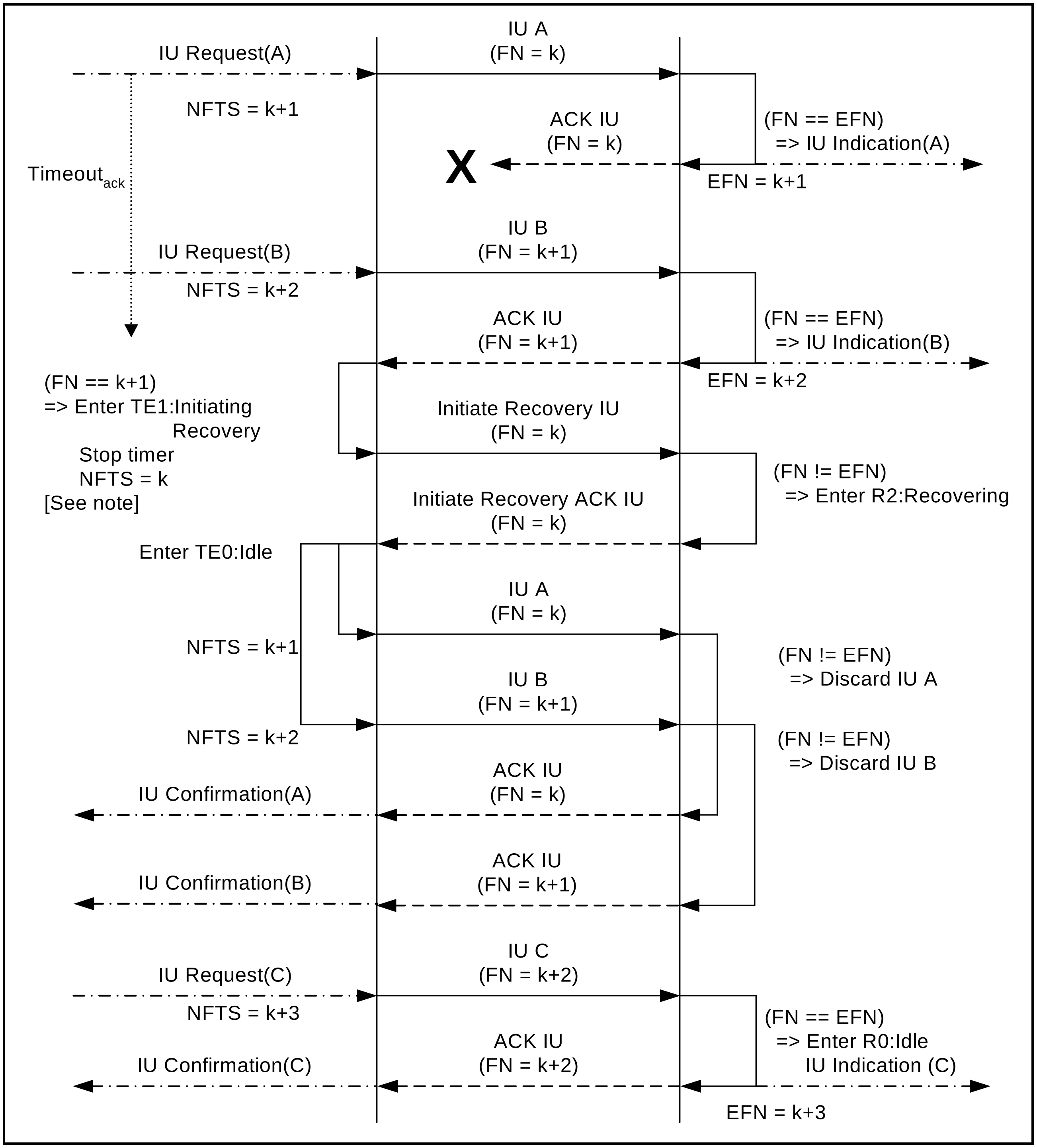


Figure B.4 — Lost ACK with recovery driven by out-of-order ACK

B.6 Lost IU with recovery driven by out-of-order NAK

Figure B.5 is similar to the previous one, but the second IU receives a NAK instead of an ACK. Again, there is no need to wait for timer expiration.

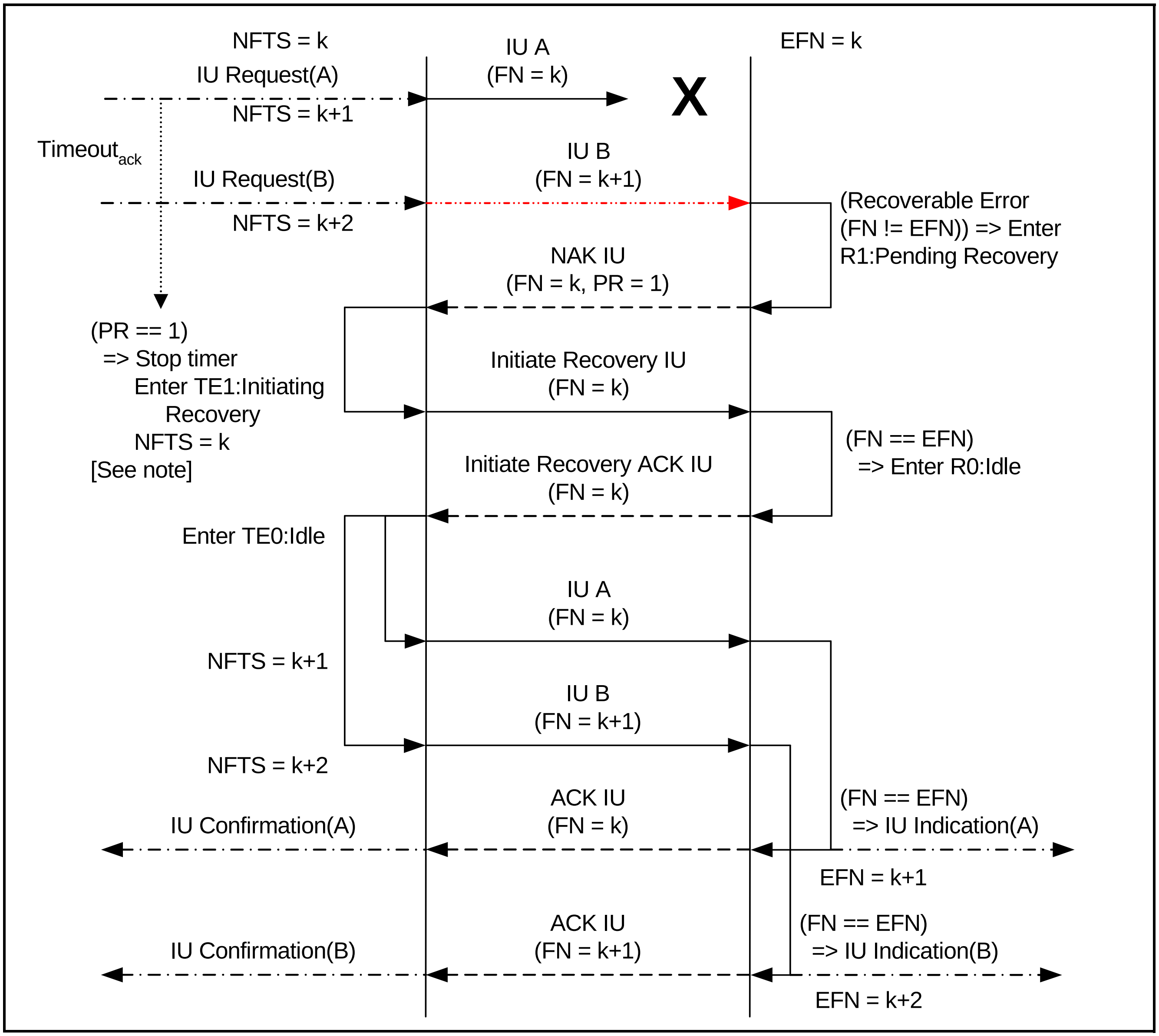


Figure B.5 — Lost IU with recovery driven by out-of-order NAK

B.7 Lost NAK with recovery driven by timeout

In the example in figure B.6, unlike the previous ones, the sender does not use an out-of-order acknowledgement IU to infer that an earlier Acknowledgement IU was lost. Instead, it waits for the $Timeout_{ack}$ on the earlier Acknowledgement IU. This diagram would also apply similarly if IU A received an ACK instead of a non-retryable NAK

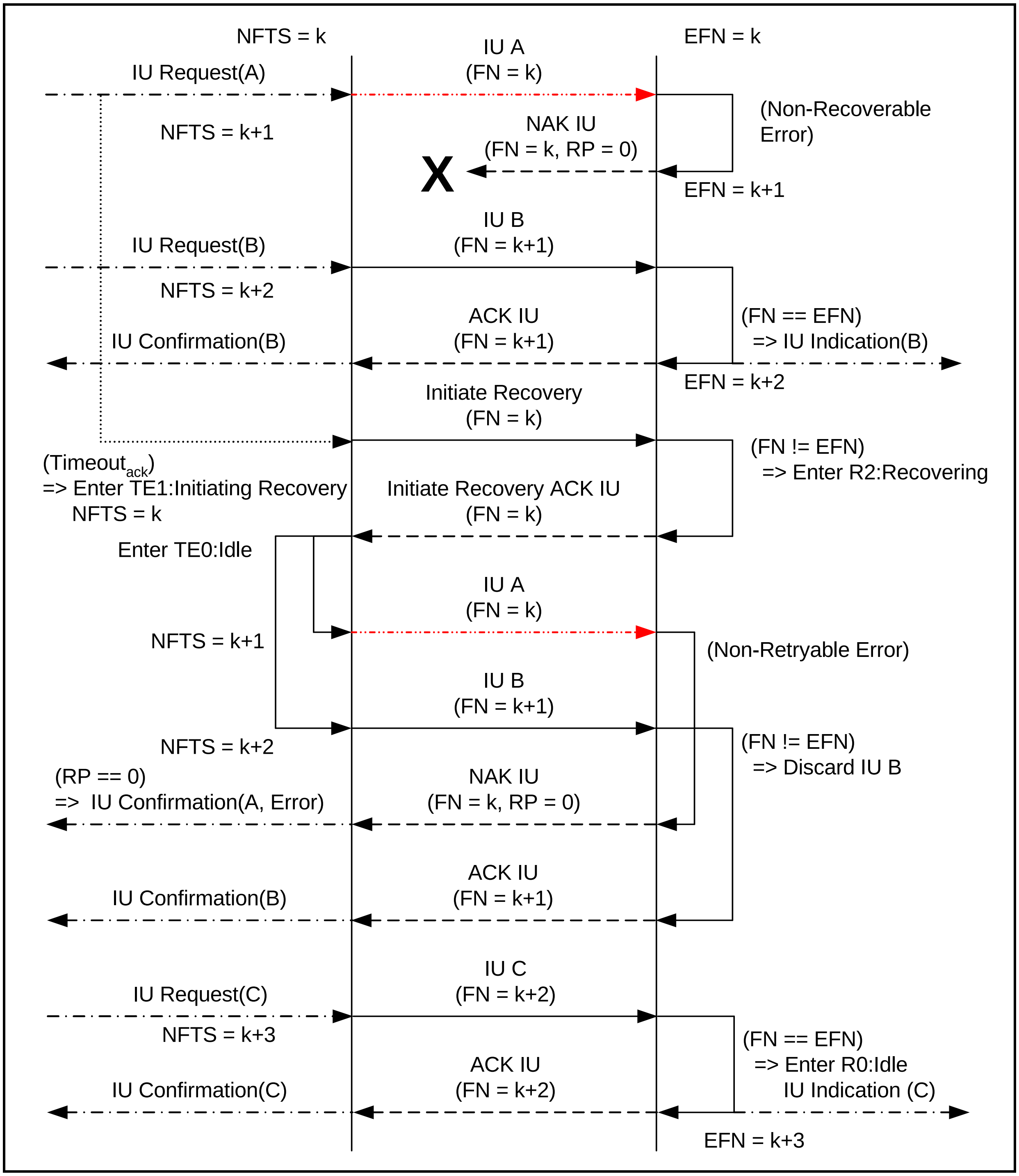


Figure B.6 — Lost NAK with recovery driven by timeout

B.8 Non-retryable error

In figure B.7, the receiver detects a non-retryable error and sends a NAK IU with a value of zero in the PENDING RECOVERY (PR) field. The error is reported to the sender's upper layer and when transmission of the next IU is requested, it is sent with the next frame number in sequence.

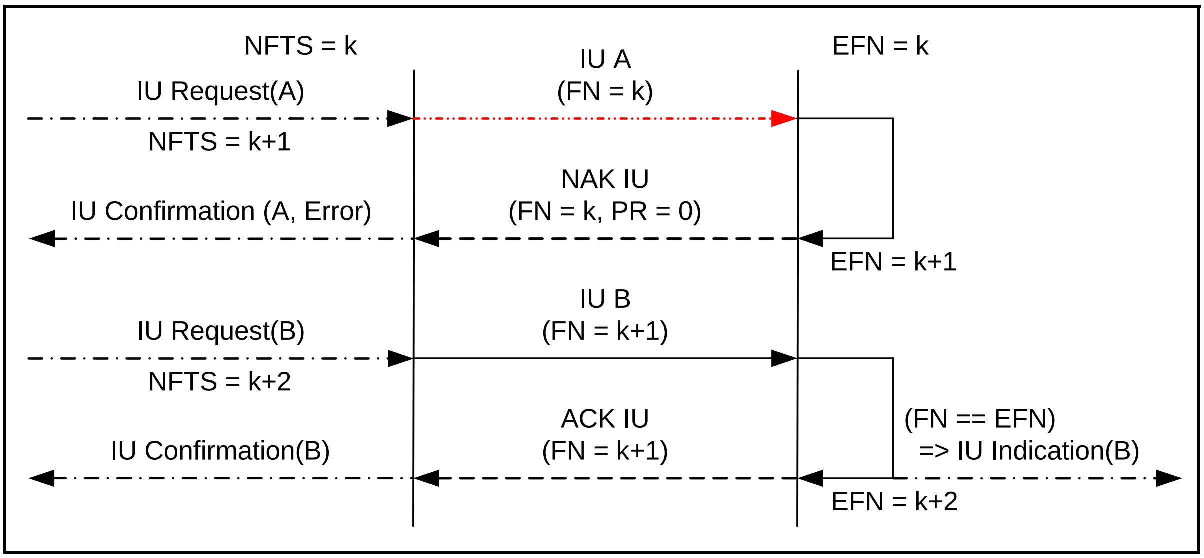


Figure B.7 — Non-retryable error

B.9 Lost ACK with errors on next IU

Figure B.8 shows a succession of three errors: a lost ACK, a retryable error, and a lost NAK for the retryable error. It is the timeout on the original lost ACK which begins the error recovery sequence.

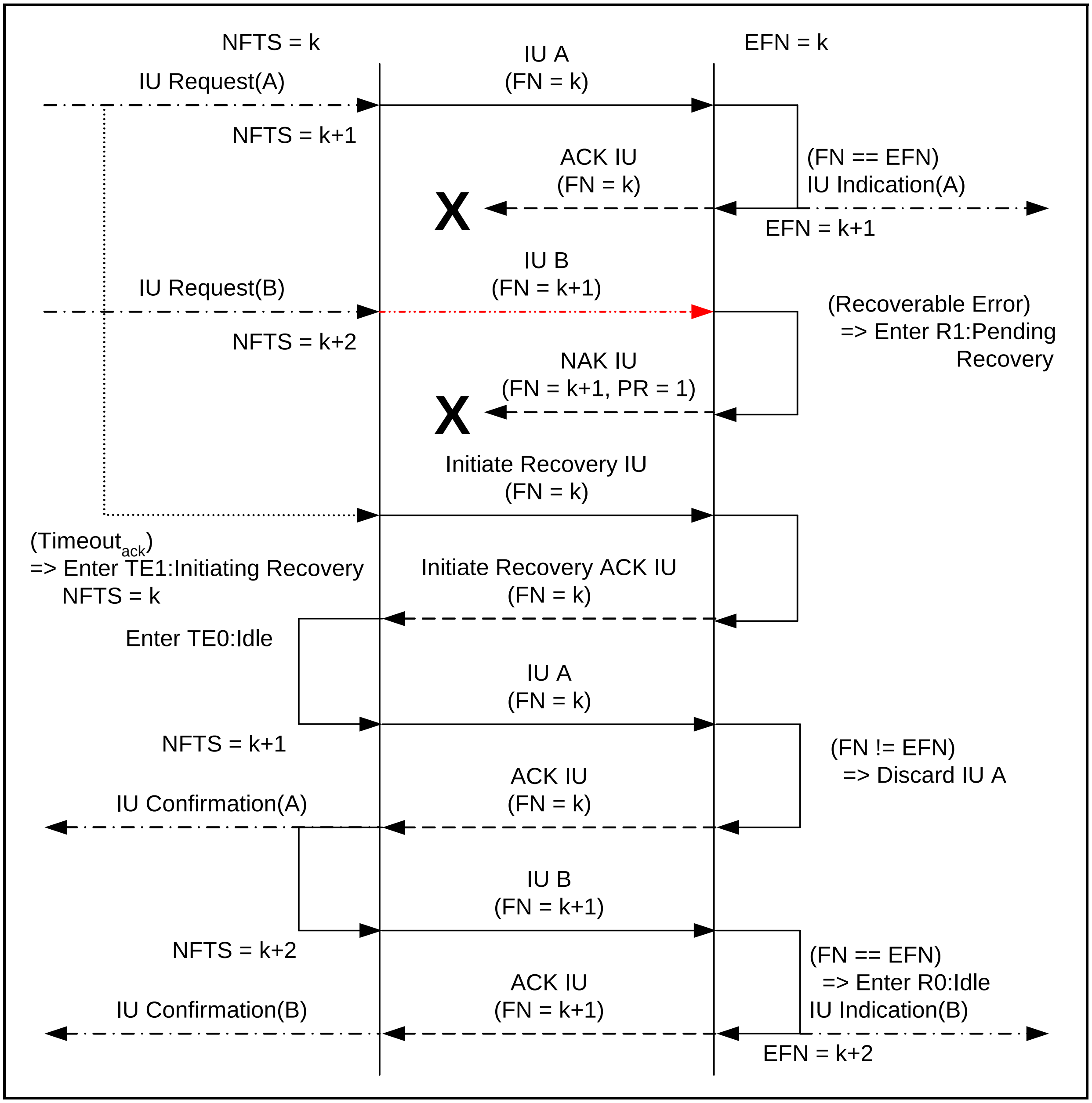


Figure B.8 — Lost ACK with errors on next IU

B.10 Delayed response with recovery driven by timeout

Figure B.11 shows a scenario in which the recipient of a frame is delayed in processing the frame by some other factor in the device. It is the timeout on the ACK which begins the error recovery sequence.

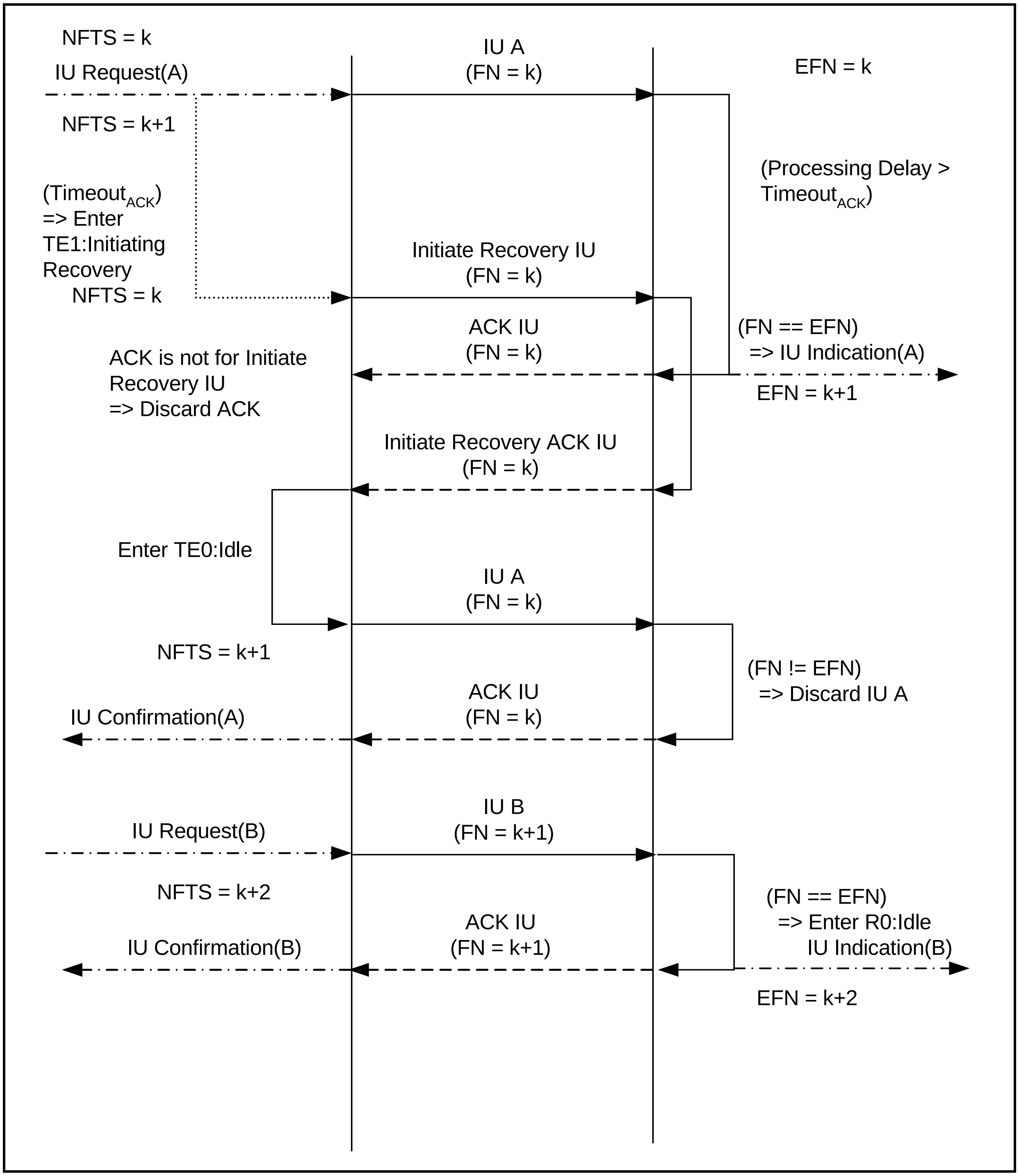


Figure B.9 — Lost ACK with errors on next IU

Annex C
(informative)

Link Negotiation Examples

C.1 Introduction

This annex provides an example link negotiation sequence as described in 4.3.3 and 6.5.4. This example does not attempt to cover all of the possible cases of negotiation, only a select few that are likely to occur.

For this example, the automation device port has the following capabilities:

a) support for the INCITS approved revision of this standard;
b) support for up to 3 frames of ACK offset;
c) support for frame payload sizes up to 1 024 bytes;
d) supports Baud rates of 115K, 38,4K, 19,2K, and 9 600; and
e) always responds to a Port Login Exchange initiated by a DT device port with a Port Login exchange initiated by the automation device port (see 4.3.3.2)

The DT device port has the following capabilities:

a) support for the INCITS approved revision of this standard as well as draft revision 9.
b) support for up to 2 frames of ACK offset;
c) support for frame payload sizes up to 512 bytes; and
d) supports Baud rates of 57,6K, 19,2K, and 9 600.

C.2 Field values common to all frames

These examples contain tables that list fields in the Port Login IUs and the values these fields contain. Table C.1 lists the fields that contain the same value for all frames described in the example and that are not negotiated.

Table C.1 — Field values common to all Port Login IUs in these examples

Field	Value	Description
PROTOCOL	0	Link service.
FRAME TYPE	2	Port Login IU.
FRAME NUMBER	0	Port Login IU always 0.
PAYLOAD SIZE	8	Payload size for Port Login IU.
VENDOR SPECIFIC	0	Not used in this example.

C.3 DT device initiates a login after power-up

In this example, the DT device port is ready to negotiate first. The automation device port is not ready to communicate on the service delivery subsystem until some time after the DT device port.

1) The DT device port sends a Port Login IU at 9 600 Baud with the parameters listed in table C.2. After sending the frame, the port starts a timer for 650 ms for the acknowledgement IU (see 4.6.1.2.2).

Table C.2 — Field values for initial Port Login IU from the DT device

Field	Value	Description
X_ORIGIN	1	DT device originated.
EXCHANGE ID	0	New exchange.
ACCEPT	0	Is zero on the first IU of an exchange.
MAJOR REVISION	1	ADT revision 1.
MINOR REVISION	0	Approved revision.
AOE	1	Abort other exchanges.
MAXIMUM ACK OFFSET	2	Maximum ACK offset that this port supports.
MAXIMUM PAYLOAD SIZE	512	Maximum payload size that this port supports.
BAUD RATE	576	Maximum baud rate that this port supports.

2) The automation device port has not been configured yet, so the DT device port receives no acknowledgement.

3) After 650 ms the acknowledgement timer expires in the DT device port. The port aborts the first exchange internally and sends a new Port Login IU at 9 600 Baud with a different exchange id value as shown in table C.3.

Table C.3 — Field values for second Port Login IU from the DT device

Field	Value	Description
X_ORIGIN	1	DT device originated.
EXCHANGE ID	1	New exchange.
ACCEPT	0	Is zero on the first IU of an exchange.
MAJOR REVISION	1	ADT revision 1.
MINOR REVISION	0	Approved revision.
AOE	1	Abort other exchanges.
MAXIMUM ACK OFFSET	2	Maximum ACK offset that this port supports.
MAXIMUM PAYLOAD SIZE	512	Maximum payload size that this port supports.
BAUD RATE	576	Maximum baud rate that this port supports.

4) This sequence repeats with a new exchange ID each time until the automation device port responds.

C.4 Automation device initiates login after power-up

In this example, the DT device port has initiated the port login process. The automation device port is initialized after missing a Port Login IU from the DT device port. This demonstrates the effects of the Port Login precedence described in 4.3.3.2.

1) The automation device port sends a Port Login IU at 9 600 Baud with the parameters listed in table C.4. After sending the frame, the port starts a timer for 650 ms for the acknowledgement IU (see 4.6.1.2.2);

Table C.4 — Field values for initial Port Login IU from the automation device

Field	Value	Description
X_ORIGIN	0	Automation originated.
EXCHANGE ID	0	New exchange.
ACCEPT	0	Is zero on the first IU of an exchange.
MAJOR REVISION	1	ADT revision 1.
MINOR REVISION	0	Approved revision.
AOE	1	Abort other exchanges.
MAXIMUM ACK OFFSET	3	Maximum ACK offset that this port supports.
MAXIMUM PAYLOAD SIZE	1 024	Maximum payload size that this port supports.
BAUD RATE	1 152	Maximum baud rate that this port supports.

2) Upon receiving the Port Login IU, the DT device port sends an ACK IU with X_ORIGIN, EXCHANGE ID, and FRAME NUMBER fields that match the Port Login IU it received. The DT device port then inspects the Port Login IU it received. Since it is in a new exchange and the AOE bit is set to one, the DT device port aborts all other exchanges in progress, including the Port Login IU it had sent and for which it is awaiting an acknowledgement. After sending the ACK IU, the DT device port sends a Port Login IU with the parameter values shown in table C.5;

Table C.5 — Field values for first reply Port Login IU from the DT device

Field	Value	Description
X_ORIGIN	0	Automation originated.
EXCHANGE ID	0	Exchange ID assigned by the automation device.
ACCEPT	0	Zero indicates that at least one field value has changed.
MAJOR REVISION	1	This value has stabilized.
MINOR REVISION	0	This value has stabilized.
AOE	1	This value has stabilized.
MAXIMUM ACK OFFSET	2	Maximum ACK offset that the DT device port supports.
MAXIMUM PAYLOAD SIZE	512	Maximum payload size that the DT device port supports.
BAUD RATE	576	Highest baud rate supported by the DT device port that is less than or equal to the value from the automation device.

3) Upon receiving the Port Login IU, the automation device port sends an ACK IU with X_ORIGIN, EXCHANGE ID, and FRAME NUMBER fields that match the Port Login IU it received. The automation device port then inspects the Port Login IU it received. Since it is part of the exchange the automation device had origi-

nated, it is a continuation of the negotiation already in progress. After sending the ACK IU, the automation device port sends a Port Login IU with the parameter values shown in table C.6;

Table C.6 — Field values for first reply Port Login IU from the automation device

Field	Value	Description
X_ORIGIN	0	Automation originated.
EXCHANGE ID	0	Exchange ID assigned by the automation device.
ACCEPT	0	Zero indicates that at least one field value has changed.
MAJOR REVISION	1	This value has stabilized.
MINOR REVISION	0	This value has stabilized.
AOE	1	This value has stabilized.
MAXIMUM ACK OFFSET	2	The automation device port is able to support this value. This value has now stabilized.
MAXIMUM PAYLOAD SIZE	512	The automation device port is able to support this value. This value has now stabilized.
BAUD RATE	384	Highest baud rate supported by the automation device port that is less than or equal to the value from the DT device.

4) Upon receiving the Port Login IU, the DT device port sends an ACK IU with X_ORIGIN, EXCHANGE ID, and FRAME NUMBER fields that match the Port Login IU it received. The DT device port then inspects the Port Login IU it received. Since it is part of the exchange that it is currently processing, it is a continuation of the negotiation already in progress. After sending the ACK IU, the DT device port sends a Port Login IU with the parameter values shown in table C.7;

Table C.7 — Field values for second reply Port Login IU from the DT device

Field	Value	Description
X_ORIGIN	0	Automation originated.
EXCHANGE ID	0	Exchange ID assigned by the automation device.
ACCEPT	0	Zero indicates that at least one field value has changed.
MAJOR REVISION	1	This value has stabilized.
MINOR REVISION	0	This value has stabilized.
AOE	1	This value has stabilized.
MAXIMUM ACK OFFSET	2	This value has stabilized.
MAXIMUM PAYLOAD SIZE	512	This value has stabilized.
BAUD RATE	192	Highest baud rate supported by the DT device port that is less than or equal to the value from the automation device.

5) Upon receiving the Port Login IU, the automation device port sends an ACK IU with X_ORIGIN, EXCHANGE ID, and FRAME NUMBER fields that match the Port Login IU it received. The automation device port then inspects the Port Login IU it received. Since it is part of the exchange the automation device had origi-

nated, it is a continuation of the negotiation already in progress. After sending the ACK IU, the automation device port sends a Port Login IU with the parameter values shown in table C.8;

Table C.8 — Field values for final reply Port Login IU from the automation device

Field	Value	Description
X_ORIGIN	0	Automation originated.
EXCHANGE ID	0	Exchange ID assigned by the automation device.
ACCEPT	1	One indicates all of the values in the payload are acceptable and none have been changed.
MAJOR REVISION	1	This value has stabilized.
MINOR REVISION	0	This value has stabilized.
AOE	1	This value has stabilized.
MAXIMUM ACK OFFSET	2	This value has stabilized.
MAXIMUM PAYLOAD SIZE	512	This value has stabilized.
BAUD RATE	192	This value has stabilized.

6) Upon receiving the Port Login IU, the DT device port sends an ACK IU with X_ORIGIN, EXCHANGE ID, and FRAME NUMBER fields that match the Port Login IU it received. The DT device port then inspects the Port Login IU it received. Since it is part of the exchange that it is currently processing, it is a continuation of the negotiation already in progress. After sending the ACK IU, the DT device port sends a Port Login IU with the parameter values shown in table C.9;

Table C.9 — Field values for final reply Port Login IU from the DT device

Field	Value	Description
X_ORIGIN	0	Automation originated.
EXCHANGE ID	0	Exchange ID assigned by the automation device.
ACCEPT	1	One indicates all of the values in the payload are acceptable and none have been changed.
MAJOR REVISION	1	This value has stabilized.
MINOR REVISION	0	This value has stabilized.
AOE	1	This value has stabilized.
MAXIMUM ACK OFFSET	2	This value has stabilized.
MAXIMUM PAYLOAD SIZE	512	This value has stabilized.
BAUD RATE	192	This value has stabilized.

7) Upon receiving the Port Login IU, the automation device port sends an ACK IU with X_ORIGIN, EXCHANGE ID, and FRAME NUMBER fields that match the Port Login IU it received. The automation device port then inspects the Port Login IU it received. Since it is part of the exchange the automation device had originated, it is a continuation of the negotiation already in progress. The ACCEPT bit set to one and no other parameters have changed indicates the negotiation process is complete. After it has successfully sent the ACK IU, the automation device port changes its operating parameters to match the negotiated values; and

8) Upon receiving the ACK IU for the final Port Login IU, the DT device port changes its operating parameters to match the negotiated values.