dpANS T10
T10/BSR INCITS 542

Revision 1a
5 November 2015

Information technology -
Automation/Drive Interface - Transport Protocol -3 (ADT-3)

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:

Curtis Ballard
Hewlett Packard Enterprise
3404 E. Harmony Road
B6U MS-16
Fort Collins, CO 80528
USA

Telephone: 970-898-3013
Email: Curtis dot Ballard at HPE dot com

Reference number
ISO/IEC XXXXX-XXX : 201x
ANSI NCITS 542:201x

Printed November 5, 2015

Points of Contact:

T10 Chair
Ralph Weber
Western Digital Technologies
18484 Preston Road, Suite 102, PMB 178
Dallas, TX 75252
Tel: 214-912-1373
Email: Ralph dot Weber at WDC dot com

T10 Vice-Chair
William Martin
Samsung Semiconductor Inc. (SSI)
7213 Marblethorpe Drive
Roseville, CA 95747-5925
Tel: 916-765-6875
Email: bill dot martin at ssi dot samsung dot com

INCITS Secretariat
INCITS Secretariat
1101 K Street NW, Suite 610
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

Purchase INCITS Standards
http://www.incits.org/standards-information/purchase-standards

Revision Information

Changes in the SCSI standards family list, clause 1, are never marked with change bars. Changes in other clauses may be marked with change bars in minor (e.g., letter revisions such as ADT3r01c) revisions but are never marked with changes bars in major revisions.

1 Approved Documents Included

No T10 approved proposals are included in ADT-3 r0. ADT-3 r0 is identical to ADT-2 r09 except that editorial changes made by the ANSI Editor as part of the INCITS Public Review for ADT-2 appear in ADT-3 r0.

The following T10 approved proposals have been incorporated ADT-3 up to and including this revision:

Table 1 — Incorporated T10 Approved Documents (in document number order)

Doc	In Rev	Document Title	Document Author
15-070r1	01	ADC-4 PCL in Separate Parameters	Kevin Butt (IBM)
15-179r2	01a	ADT-3 SAM-5 Updates	Curtis Ballard (HPE)

2 Revision History

2.1 Revision 0 (14 May, 2015)

Revision 0 of ADT-3 is substantially equal to revision 9 of ADT-2. The only differences arise from changes made in ADT-2 by the ANSI Editor during the INCITS Public Review process.

2.2 Revision 1 (05 June, 2015)

The following T10 approved proposals were incorporated in ADT-3 revision 1:

15-070r1 ADC-4 PCL in Separate Parameters [Kevin Butt, IBM]

Updated references to other standards.

2.3 Revision 1a (03 November, 2015)

The following T10 approved proposals were incorporated in ADT-3 revision 1:

15-079r2 ADT-3 SAM-5 Updates [Curtis Ballard, HPE]
Edits accepted for incorporation during working group review of 15-077r5

Updated references to other standards.

ANSI (r)
NCITS 542:201x

Draft

American National Standard for Information Technology -

Automation/Drive Interface - Transport Protocol - 3 (ADT-3)

Secretariat
Information Technology Industry Council

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-4.

American National Standard

Approval of an American National Standard requires review 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 a concerted effort be made towards their resolution.

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

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

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

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 such 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 holder(s) has (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 publisher in respect to any standard it processes. No representation is made or implied that this is the only license that may be required to avoid infringement in the use of this standard.

Published by
American National Standards Institute, Inc.
25 West 43rd Street, 4 floor, 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 542:201x.

This standard 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 542:201x, 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:

Ihttp://www.incits.org/standards-information/purchase-standards-or-download-dpans

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, InterNational Committee for Information Technology Standards, Information Technology Institute, 1101 K Street, NW, Suite 610, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by the InterNational 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 SCSI Storage Interfaces, which developed and reviewed this standard, had the following members:

Ralph Weber, Chair
William Martin, Vice-Chair
John Geldman, Secretary

Organization Represented	*Name of Representative*
TBD	

Introduction

The Automation/Drive Interface - Transport Protocol - 3 (ADT-3) 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-5. 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 connection layer including connection services.
Clause 7 describes the link layer including encoding, frame format, and link services functions.
Clause 8 describes the transport layer and includes the method of encapsulating SCSI.
Clause 9 describes the mapping between SCSI protocol services defined in SAM-5 and the services provided by this protocol.

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

American National Standard **INCITS 542:201x**

American National Standard
for Information Technology -

Automation/Drive Interface -
Transport Protocol - 3
(ADT-3)

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-3 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-3). 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 operates as a SCSI initiator port or as both a SCSI initiator port and a SCSI 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 operates as a SCSI target port or as both a SCSI initiator port and a SCSI target port.

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.

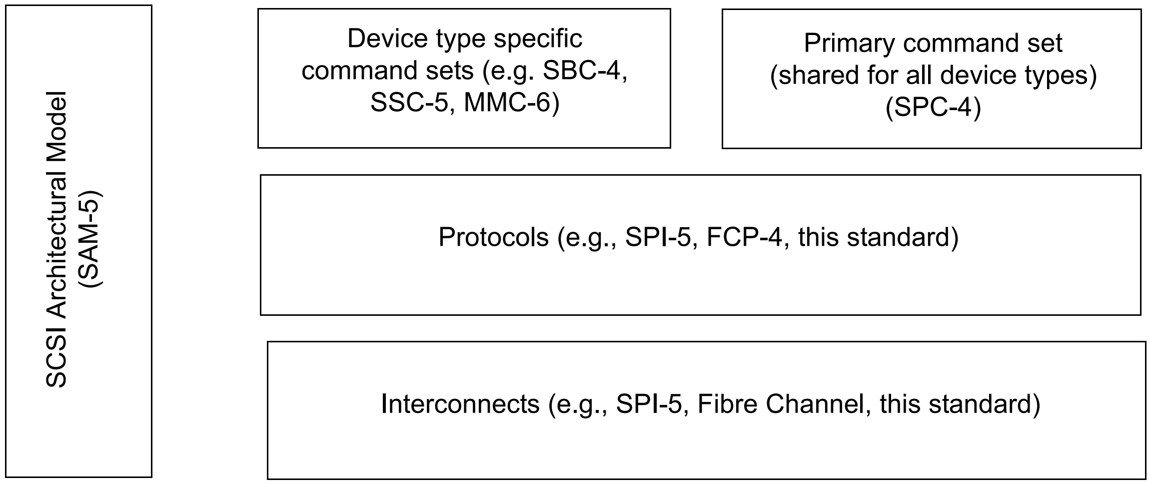


Figure 1 — General Document Structure of SCSI

2 References

2.1 Normative references

2.1.1 Normative references overview

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 ANS, an ISO member organization:

a) approved ANSI standards;
b) approved and draft international and regional standards (ISO, IEC); and
c) approved and draft foreign standards (including JIS, and DIN).

For further information, contact the ANSI Customer Service Department:

Phone: +1 212-642-4980

Fax: +1 212-302-1286

Web: http://www.ansi.org

E-mail: info@ansi.org

or the InterNational Committee for Information Technology Standards (INCITS):

Phone: +1 202-626-5739

Web: http://www.incits.org

E-mail: incits@itic.org

Additional availability contact information is provided below as needed.

2.1.2 Approved references

ISO/IEC 14776-414, *SCSI Architecture Model - 4* (SAM-4) [ANSI INCITS 447-2008]

ISO/IEC 14776-115, *SCSI Parallel Interface - 5* (SPI-5) [ANSI INCITS 367-2003]

ISO/IEC 14776-224, *SCSI Fibre Channel Protocol - 4 (FCP-4)* [ANSI INCITS 481-2012]

ISO/IEC 14776-353, *SCSI Media Changer Commands - 3* (SMC-3) [ANSI INCITS 484-2012]

ANSI/INCITS 497-2012, *Automation/Drive Interface Commands - 3* (ADC-3)

ANSI/INCITS 468-2010, MultiMedia Command Set - 6 (MMC-6)

ANSI/TIA 422-B-1994 (R2005) Electrical Characteristics of Balanced Voltage Digital Interface Circuits. (RS-422)

2.1.3 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 14776-335 *SCSI Stream Commands - 5* (SBC-5) [T10/BSR INCITS 503]

T10/BSR INCITS 541, *Automation/Drive Interface Commands - 4* (ADC-4)

ISO/IEC 14776-415, *SCSI Architecture Model - 5* (SAM-5) [T10/BSR INCITS 515]

ISO/IEC 14776-454, *SCSI Primary Commands - 4* (SPC-4) [T10/BSR INCITS 513]

2.1.4 IETF References

Copies of the following approved IETF standards may be obtained through the Internet Engineering Task Force

(IETF) at www.ietf.org.

RFC 768, User Datagram Protocol

RFC 791, Internet Protocol – DARPA Internet Program – Protocol Specification

RFC 793, Transmission Control Protocol (TCP) – DARPA Internet Program – Protocol Specification

RFC 1112, Host Extensions for IP Multicasting

RFC 2365, Administratively Scoped IP Multicast

RFC 2460, Internet Protocol, Version 6 (IPv6) Specification

RFC 3170, IP Multicast Applications: Challenges and Solutions

RFC 3493, Basic Socket Interface Extensions for IPv6

RFC 5246, The Transport Layer Security (TLS) Protocol, version 1.2

2.1.5 IEEE references

IEEE 802.3-2005, Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications

2.2 Informative 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 1 For more information on the current status of the document, or to obtain copies, contact the SFF committee at 14426 Black Walnut Court, Saratoga, CA 95070 at 408-867-6630 (phone), or 408-867-2115 (fax), or http://www.sffcommittee.org (web).

ANSI/ASTM D4566-1994, Standard Test Methods for Electrical Performance Properties of Insulations and Jackets for Telecommunications Wire and Cable

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 acknowledgement IU
ACK information unit (IU), NAK IU, Initiate Recovery ACK IU, or Initiate Recovery NAK IU

NOTE 1 to entry: see 7.5.3.

3.1.2 ADT initiator port
SCSI initiator port that implements ADT

3.1.3 ADT port
ADT initiator port, ADT target port, or ADT target/initiator port

3.1.4 ADT target port
SCSI target port that implements ADT

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

3.1.6 application client
object that is the source of commands and task management function requests

NOTE 1 to entry: see SAM-5.

3.1.7 automation device:
device containing one or more SMC device servers (see SMC-3) or equivalent, one or more automation application clients, and one or more ADT ports

NOTE 1 to entry: see ADC-4

3.1.8 automation device port
ADT port on an automation device

3.1.9 bridging manager
application client within a DT device that initiates SCSI requests to an SMC device server (see SMC-3) within an automation device

NOTE 1 to entry: see ADC-4.

3.1.10 byte
sequence of eight contiguous bits considered as a unit

3.1.11 command
request describing a unit of work to be performed by a device server

NOTE 1 to entry: see SAM-5.

3.1.12 command identifier
numerical identifier of a command (see 3.1.11)

NOTE 1 to entry: see SAM-5.

3.1.13 connection
link between two ADT ports by which the ports are able to exchange encoded characters

NOTE 1 to entry: see 6.1.

3.1.14 device server
object within a logical unit that processes SCSI tasks according to the rules for task management

NOTE 1 to entry: see SAM-5.

3.1.15 DT device:
device containing an RMC device server, an ADC-4 device server, one or more ADT ports, and one or more DT device primary ports

NOTE 1 to entry: see ADC-4.

3.1.16 DT device port
ADT port on a DT device

3.1.17 exchange:
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

NOTE 1 to entry: see 7.3.

3.1.18 expected frame number
value in the FRAME NUMBER field that a receiver ADT port expects in the next frame

NOTE 1 to entry: see 4.5.3.

3.1.19 field
group of one or more contiguous bits

3.1.20 hard reset
target action in response to a reset event in which the target ADT port performs the operations described in 4.7

3.1.21 information unit
ADT frame header and payload

NOTE 1 to entry: see 7.1.

3.1.22 IP
Internet protocol

NOTE 1 to entry: see RFC 791

NOTE 2 to entry: see RFC 2460

3.1.23 I_T nexu
nexus that exists between a SCSI initiator port and a SCSI target port.

3.1.24 I_T_L nexus
nexus between a SCSI initiator port, a SCSI target port, and a logical unit

3.1.25 link parameters
parameters affecting the physical operation of the link (e.g., maximum ACK offset, maximum payload size, and baud rate)

3.1.26 LLC
Ethernet link layer control

3.1.27 logical unit
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

3.1.28 logical unit number:
identifier for a logical unit

3.1.29 logical unit reset
logical unit action in response to a logical unit reset event in which the logical unit performs the operations described in SAM–3

3.1.30 logical unit reset event
event that triggers a logical unit reset from a logical unit as described in SAM-5

3.1.31 login process
process of negotiating operating parameters for the transport using Port Login IUs

NOTE 1 to entry: see 4.3.3.

3.1.32 logout duration time
length in seconds that an ADT port shall remain in P3:Logged-out state

NOTE 1 to entry: see 7.5.5.

3.1.33 MAC
Ethernet media access control

3.1.34 MDI
Ethernet medium dependent interface

3.1.35 negotiated parameters
link parameter values agreed upon through negotiation. The negotiated parameters are a set from the intersection of the supported parameters for each ADT port

3.1.36 negotiation error
condition that results in the ADT port sending a NAK IU with a status code of NEGOTIATION ERROR

NOTE 1 to entry: see 7.5.4.

3.1.37 nexus
relationship between two SCSI devices, and the SCSI initiator port and SCSI target port objects within those SCSI devices

3.1.38 object
container that encapsulates data types, services, or other objects that are related in some way

3.1.39 operating parameters
current link parameter values under which the ADT port is operating

3.1.40 PHY
Ethernet physical layer

3.1.41 PLS
Ethernet physical signaling sublayer

3.1.42 requested parameters
set of link parameter values received in a Port Login IU

3.1.43 reset event:
protocol specific event that triggers a hard reset from a device

NOTE 1 to entry: see 4.7.

3.1.44 SCSI device
device that contains one or more SCSI ports that are connected to a service delivery subsystem and supports a SCSI application protocol

3.1.45 SCSI initiator device
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

3.1.46 SCSI initiator port
SCSI initiator device object that acts as the connection between application clients and the service delivery subsystem through which requests and confirmations are routed

3.1.47 SCSI port
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. A SCSI port is either a SCSI initiator port (see 3.1.46) or a SCSI target port (see 3.1.49)

3.1.48 SCSI target device:
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

3.1.49 SCSI target port
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

3.1.50 session

association between two ADT ports

NOTE 1 to entry: see 7.6.

3.1.51 simple exchange:

exchange consisting of two information units, an IU with a type other than acknowledgement and the corresponding acknowledgement IU

3.1.52 simple link service exchange:

simple exchange (see 3.1.51) with the PROTOCOL field set to link service (see 7.3)

3.1.53 starting parameter set

set containing the maximum value for each link parameter under which the ADT port is capable of operating

3.1.54 supported parameter set

set containing one value for each link parameter under which the ADT port is capable of operating

3.1.55 supported parameters

set consisting of all of the ADT port’s supported parameter sets

3.1.56 symbol framing error:

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

3.1.57 task

object within the logical unit representing the work associated with a command

3.1.58 task manager

server within a logical unit that controls the sequencing of one or more tasks and processes task management functions

3.1.59 time-out discovery exchange

exchange that consists of a sequence of IUs as specified in 4.10.2.2

3.1.60 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

3.1.61 zero:

false signal value or a false condition of a variable

3.2 Symbols and abbreviations

x	multiply by
+	add to
-	subtract from
<	less than
ACA	auto-contingent allegiance
ADC-3	Automation/Drive Interface - Commands - 3

ADC-4	Automation/Drive Interface - Commands - 4
ADT	Automation/Drive Interface - Transport Protocol
ADT-3	Automation/Drive Interface - Transport Protocol - 3
AER	asynchronous event report
CDB	command descriptor block
CRN	command reference number
DT	data transfer (e.g. DT device)
FCP-4	Fibre Channel Protocol - 4
IU	information unit
LSB	Least significant bit
LUN	Logical unit number
MSB	Most significant bit
SAM-4	SCSI Architecture Model - 4
SAM-5	SCSI Architecture Model - 5
SCSI	Small Computer System Interface
SMC	SCSI Medium Changer Commands
SMC-3	SCSI Media Changer Commands - 3
SPC-4	SCSI Primary Commands - 4
SPC-5	SCSI Primary Commands - 5
SPI-5	SCSI Parallel Interface - 5
SSC-5	SCSI Stream Commands - 5

3.3 Keywords

3.3.1 expected

keyword used to describe the behavior of the hardware or software in the design models assumed by this standard

NOTE 1 to entry: Other hardware and software design models may also be implemented.

3.3.2 invalid

keyword used to describe an illegal or unsupported bit, byte, word, field or code value

NOTE 1 to entry: Receipt of an invalid bit, byte, word, field or code value shall be reported as an error.

3.3.3 mandatory

keyword indicating an item that is required to be implemented as defined in this standard

3.3.4 may

keyword that indicates flexibility of choice with no implied preference

NOTE 1 to entry: “May” is equivalent to "may or may not".

3.3.5 may not

keyword that indicates flexibility of choice with no implied preference

NOTE 1 to entry: “May” is equivalent to "may or may not".

3.3.6 obsolete

keyword indicating that an item was defined in prior SCSI standards but has been removed from this standard

3.3.7 optional
keyword that describes features that are not required to be implemented by this standard; however, if any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard

3.3.8 reserved
keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization

NOTE 1 to entry: A reserved bit, byte, word of field shall be set to zero, or in accordance with a future extension to this standard.

NOTE 2 to entry: Recipients are not required to check reserved bits, bytes, words or fields for zero values.

NOTE 3 to entry: Recipt of reserved code values in defined fields shall be reported as an error.

3.3.9 shall
keyword indicating a mandatory requirement

NOTE 1 to entry: Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

3.3.10 should
keyword indicating flexibility of choice with a strongly preferred alternative

NOTE 1 to entry: “Should” is equivalent to the phrase "it is strongly recommended".

3.4 Editorial conventions

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

If there is more than one CDB length for a particular command (e.g., MODE SENSE(6) and MODE SENSE(10)) and the name of the command is used in a sentence without any CDB length descriptor (e.g., MODE SENSE), then the condition specified in the sentence applies to all CDB lengths for that command.

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

A binary number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0_0101_1010b).

A hexadecimal number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 through 9 and/or the upper-case English letters A through F immediately followed by a lower-case h (e.g., FA23h). Underscores or spaces may be included in hexadecimal number representations to increase readability or delineate field boundaries (e.g., B FD8C FA23h or B_FD8C_FA23h).

A decimal number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 through 9 not immediately followed by a lower-case b or lower-case h (e.g., 25).

A range of numeric values is represented in this standard in the form "a to z", where a is the first value included in the range, all values between a and z are included in the range, and z is the last value included in the range (e.g., the representation "0h to 3h" includes the values 0h, 1h, 2h, and 3h).

When the value of the bit or field is not relevant, x or xx appears in place of a specific value.

This standard uses the following conventions for representing decimal numbers:

a) the decimal separator (i.e., separating the integer and fractional portions of the number) is a period;
b) the thousands separator (i.e., separating groups of three digits in a portion of the number) is a space; and
c) the thousands separator is used in both the integer portion and the fraction portion of a number.

Table 1 shows some examples of decimal numbers represented using various conventions.

Table 1 — Numbering conventions examples

French	English	This Standard
0,6	0.6	0.6
3,141 592 65	3.14159265	3.141 592 65
1 000	1,000	1 000
1 323 462,95	1,323,462.95	1 323 462.95

A decimal number represented in this standard with an overline over one or more digits following the decimal point is a number where the overlined digits are infinitely repeating (e.g., $666.\overline{6}$ means 666.666 666. or 666 2/3 and $12.\overline{142\ 857}$ means 12.142 857 142 857. or 12 1/7).

Lists sequenced by lowercase or uppercase letters show no ordering relationship between the listed items.

EXAMPLE 1 - The following list shows no relationship between the colors named:

a) red, specificity one of the following colors:
   A) crimson; or
   B) amber;
b) blue; or
c) green.

Lists sequenced by numbers show an ordering relationship between the listed items.

EXAMPLE 2 - The following list shows the order in which a page is meant to be read:

1) top;
2) middle; and
3) bottom.

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

3.5 Notation conventions

3.5.1 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.

3.5.2 Notation for state machines

3.5.2.1 Notation for state machines overview

Figure 2 shows how state machines are described.

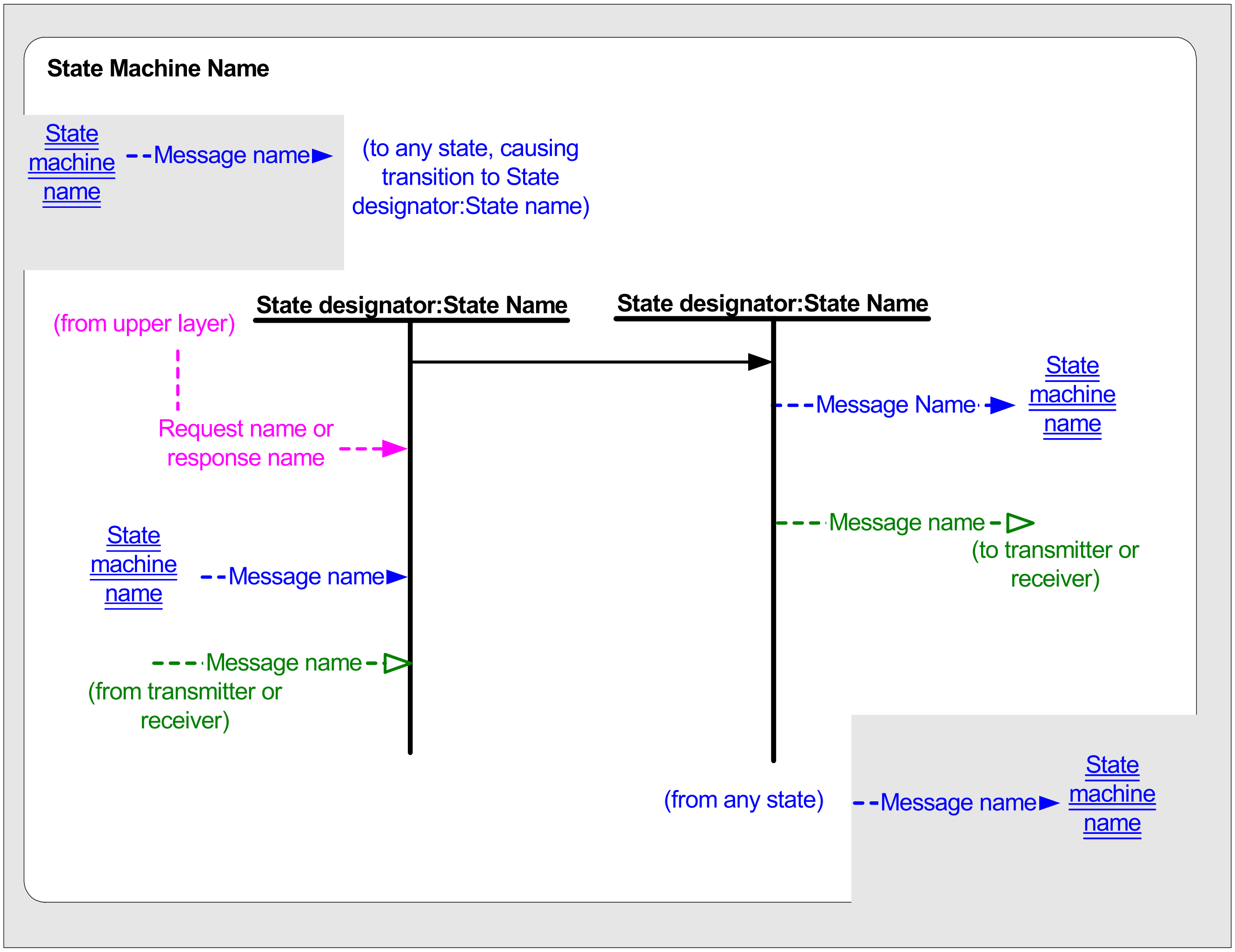


Figure 2 — State machine conventions

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.

3.5.2.2 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.

3.5.2.3 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.

3.5.2.4 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.

3.5.3 Notation for communication sequence diagrams

Sequence diagrams are used to indicate communication among entities within a device and among devices. All communication sequence diagrams use the notation shown in Figure 3. Each entity is indicated by a horizontal bar with a label on top of a vertical bar. Entities within the same device are enclosed by a box with a label at the top of the box. Each communication is indicated by an arrow with an optional label. Solid arrows indicate mandatory communications and dashed arrows indicate optional communications. Time flows from the top of the diagram (i.e., first communication) to the bottom (i.e., last communication).

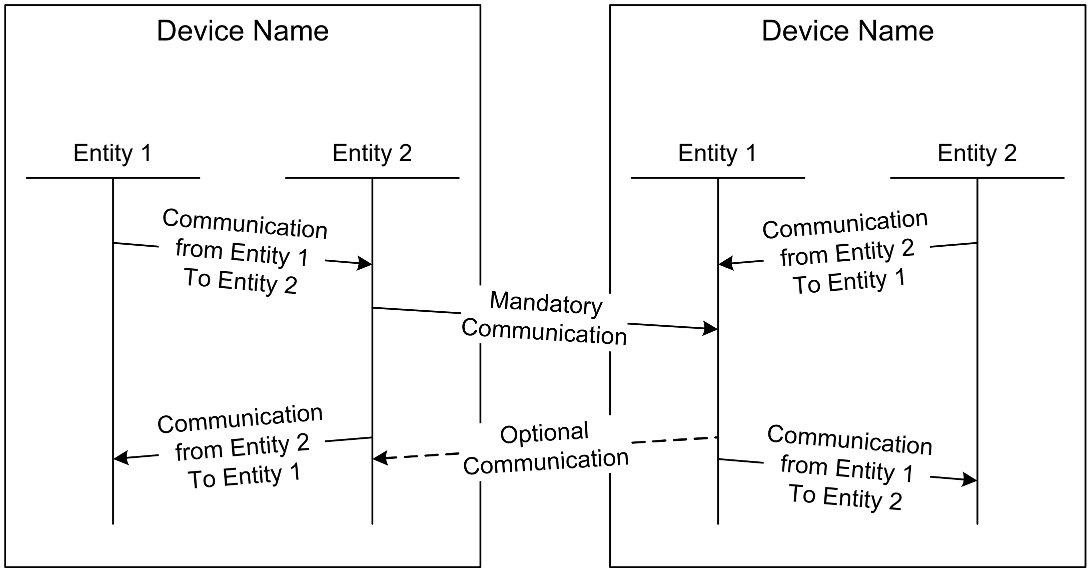


Figure 3 — Example communication sequence diagram

4 General

4.1 Architecture

4.1.1 Architecture introduction

Figure 4 shows examples of serial ADT (sADT) and internet ADT (iADT) interfaces (see 4.1.2) within a media changer containing two DT devices. Other common components of a media changer are also shown for reference. The components of an automation device are medium transport elements, data transfer (DT) devices, storage elements, and import/export elements (see SMC-3). The automation device may communicate with the DT devices through ADT ports, as defined in this standard. DT devices and automation devices communicate with initiator ports other than those in the automation device using primary ports.

NOTE 2 A media changer or DT device would not necessarily contain both sADT and iADT ports. They are both shown for comparison.

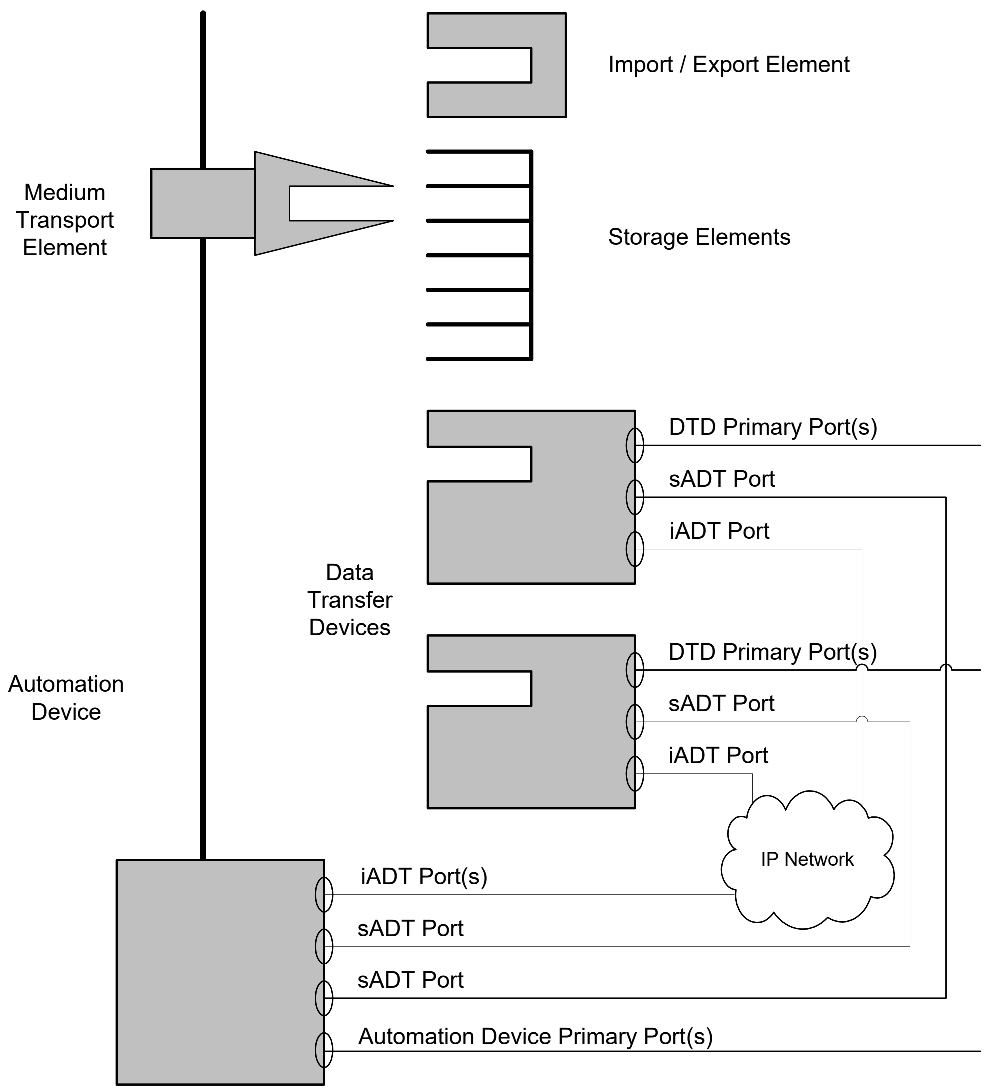


Figure 4 — Example Media Changer application of ADT

If ADI Bridging is enabled (see ADC-4), each ADT port in the DT device and automation device is a SCSI target port and a SCSI initiator port. If ADI Bridging is disabled, the ADT port in the DT device is a SCSI target port and the ADT port in the automation device is a SCSI initiator port.

4.1.2 ADT protocol layers

The ADT protocol defines communication between two ADT ports. The ADT protocol includes the SCSI Transport Protocol Layer (STPL) and the Interconnect Layer (see SAM-5). The STPL defined by ADT consists of the ADT Transport Layer. The Interconnect Layer defined by ADT consists of three layers, the Link Layer, the Connection Layer, and the Physical Layer.

Figure 5 shows the communication between ADT ports at the different layers of the protocol, from the physical layer to the SCSI transport protocol layer.

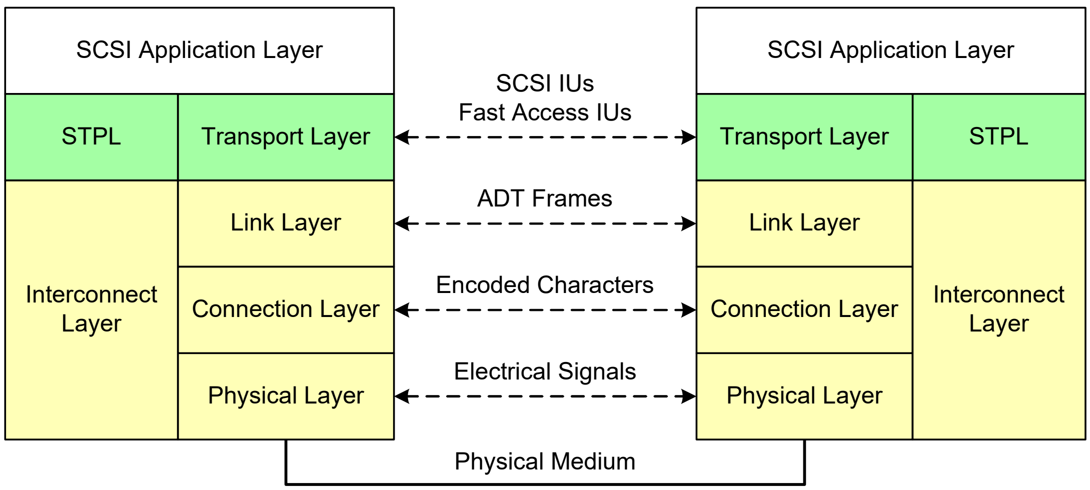


Figure 5 — ADT communication model

At the transport layer, information units (IUs) are passed between ADT ports. At the link layer, ADT frames are passed between ADT ports. At the connection layer, encoded characters are passed between ADT ports. At the physical layer, electrical signals are passed between ADT ports. The physical layers are connected by the physical medium.

The SCSI application layer (see clause 9) defines transport protocol services for processing SCSI commands and task management requests.

The ADT transport layer (see clause 8) provides transmission of two categories of information units (IUs), SCSI encapsulation IUs and fast access IUs, between ADT ports. The information units are represented as ADT frames.

The ADT link layer (see clause 7) provides establishment of sessions (see 3.1.50) between pairs of ADT ports and provides reliable transmission of ADT frames between the two ADT ports.

The ADT connection layer (see clause 6) provides transmission of encoded characters between ADT ports. Two transmission methods are defined: sADT and iADT. The sADT protocol supports transmission over an RS-422 physical layer. The iADT protocol supports transmission over a TCP connection (see RFC 793) and provides service discovery using UDP (see RFC 768). The TCP connection uses the Internet Protocol (IP) (see RFC 791 and RFC 2460) to provide transmission over a physical layer (e.g., Ethernet). The iADT protocol also supports transmission over a TCP connection secured using the Transport Layer Security (TLS) protocol (see RFC 5246).

The ADT physical layer (see clause 5) defines two physical connections for data, RS-422 and a physical layer capable of transporting IP packets (e.g., Ethernet), as well as sense, signal, and Ethernet LED connections.

The interface between the SCSI application layer and the SCSI transport protocol layer is called the protocol service interface. The interface between the SCSI transport protocol layer and the interconnect layer is called the interconnect service interface. The interface between the link layer and the connection layer is called the connection service interface.

Figure 6 shows the serial ADT (sADT) hierarchy of protocols which may be used to implement ADT on an RS-422 physical layer (see ANSI/EIA/TIA-422-B-1994 and 5.2.5.2).

<table>
<tr><th>ADT SCSI encapsulation</th><th>ADT fast access</th><th>Transport layer</th></tr>
<tr><td colspan="2">ADT link layer</td><td>Link layer</td></tr>
<tr><td colspan="2">sADT</td><td>Connection layer</td></tr>
<tr><td colspan="2">RS-422</td><td>Physical layer</td></tr>
</table>

Figure 6 — sADT protocol hierarchy

Figure 7 is an example of the Internet ADT (iADT) hierarchy of protocols using an Ethernet physical layer (see IEEE 802.3-2005 and 5.2.5.3). iADT may be used to implement ADT on a physical layer supporting the Internet Protocol (IP).

<table>
<tr><th>ADT SCSI encapsulation</th><th>ADT fast access</th><th>Transport layer</th></tr>
<tr><td colspan="2">ADT link layer</td><td>Link layer</td></tr>
<tr><td colspan="2">iADT</td><td rowspan="5">Connection layer</td></tr>
<tr><td colspan="2">TCP</td></tr>
<tr><td colspan="2">IP</td></tr>
<tr><td colspan="2">Ethernet LLC</td></tr>
<tr><td colspan="2">Ethernet MAC</td></tr>
<tr><td colspan="2">Ethernet PHY</td><td>Physical layer</td></tr>
</table>

Figure 7 — iADT protocol hierarchy

The term sADT port refers to an ADT port using the ADT serial transmit-receive connections (see 5.2.5.2) and the sADT connection layer (see 6.3). An sADT port may connect to one other sADT port in another device. Figure 8 shows connections corresponding to Figure 4.

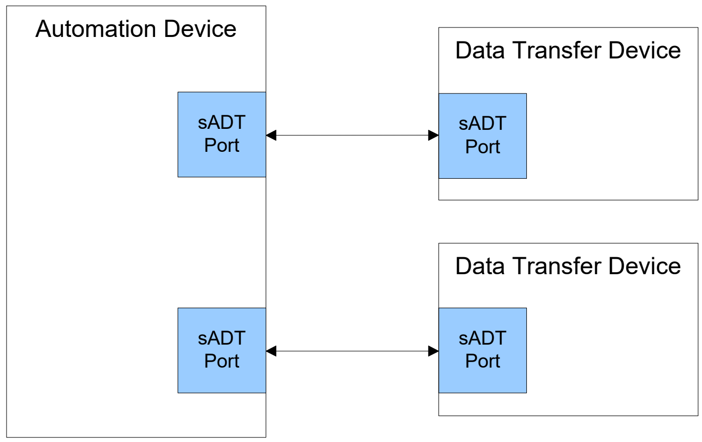


Figure 8 — sADT port example

The term iADT port refers to an ADT port using Internet Protocol (IP) transmit-receive connections, such as over Ethernet (see 5.2.5.3) and the iADT connection layer (see 6.4). An iADT port in one device may connect to multiple iADT ports in other devices. Figure 9 shows iADT ports connected via an IP network, corresponding to the connections shown in Figure 4.

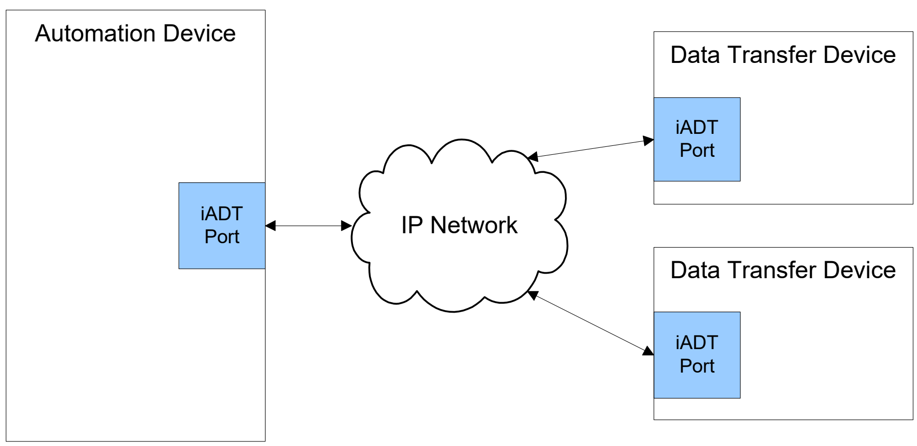


Figure 9 — iADT port example

4.2 Default operating parameters

The default operating parameters for an ADT port are as follows:

a) the baud rate shall be set to 9 600 by an sADT port;
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) ADT port;
b) link negotiation;
c) transmitter;
d) transmitter error recovery; and
e) receiver error recovery.

The ADT 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 ADT port state machine).

4.3.2 ADT Port state machine

4.3.2.1 ADT port state machine overview

The ADT 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 10 shows the ADT port state machine. The following subclauses describe the transitions and the actions taken in each state.

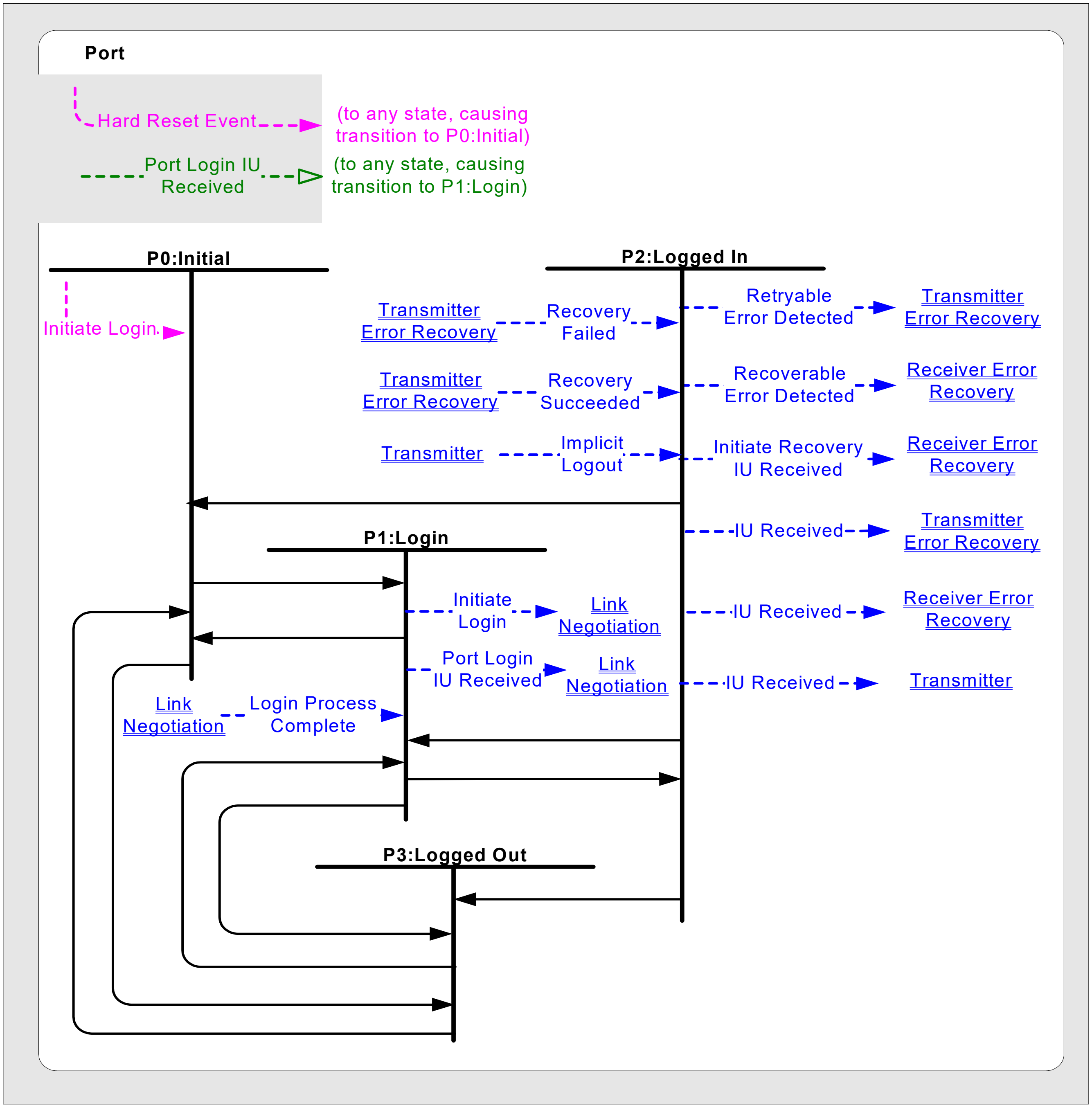


Figure 10 — ADT port state machine Diagram

4.3.2.2 P0:Initial state

4.3.2.2.1 State description

This is the initial state of the ADT port state machine.

An ADT port entering this state shall set its operating parameters to default values (see 4.2).

An ADT port in P0:Initial state shall send a NAK IU with status code of REJECTED, PORT IS LOGGED OUT (see table 51) in response to any frame other than Port Login IU, Port Logout IU, NOP IU, Device Reset IU, or acknowledgement IU. Upon entering this state, all other existing state machines shall be set to their initial states.

4.3.2.2.2 Transition P0:Initial to P1:Login

The ADT port shall transition to P1:Login after receiving a Port Login IU (see 7.5.4) or an Initiate Login request.

4.3.2.2.3 Transition P0:Initital to P3:Logged-Out

An ADT 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. 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 7.3)).

An ADT port in this state shall send a NAK IU with a status code of LOGIN IN PROGRESS (see table 51) in response to any frame other than Port Login IU, Port Logout IU, NOP IU, Device Reset IU, or acknowledgement IU.

While in this state, the ADT 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 ADT 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. If an ADT port enters this state as a result of a Recovery Failed message (see 4.3.2.4.3), the ADT port 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 7.5.4).

4.3.2.3.2 Transition P1:Login to P0:Initial

An ADT 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

An ADT port shall 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

An ADT 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

Upon entry to this state, an ADT port shall set its operating parameters to the negotiated values (see 4.3.2.3.1).

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

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

The ADT port shall send an Initiate Recovery IU Received message to the receiver error recovery state machine if the ADT 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 RE0:Idle, the ADT 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 ADT port shall send an IU Received message to the transmitter error recovery state machine.

If a Port Login IU, Port Logout IU or NOP IU frame is received, then that frame is processed.

If an ADT port receives a Recovery Failed message, the ADT port shall abort all open exchanges, set its operating parameters to default values (see 4.2), and set the AOE bit to one for the next Port Login IU it sends. If the ADT port receives a Recovery Succeeded message, the ADT port shall resume the transmission of frames.

4.3.2.4.2 Transition P2:Logged-In to P0:Initial

An ADT 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

An ADT port shall transition to P1:Login state after receiving a Recovery Failed message (see 4.3.5.4.2) or upon receiving a Port Login IU.

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

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

An ADT port shall transition to P3:Logged-Out state and set the logout duration time to a vendor-specific value which may be zero after it receives an Implicit Logout message.

When an ADT port in a DT device receives an Implicit Logout message it shall:

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

When an ADT port in an automation device receives an Implicit Logout message it shall:

a) abort all open exchanges;
b) disable initiating Port Login exchanges; and
c) set port operating parameters to the default operating parameters (see 4.2).

4.3.2.5 P3:Logged-Out state

4.3.2.5.1 State description

An ADT port entering this state shall set its operating parameters to the default operating parameters (see 4.2).

An ADT 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, Port Logout IU, NOP IU, or Device Reset IU, the ADT port shall send a NAK IU with a status code of REJECTED, PORT IS LOGGED OUT (see table 51).

4.3.2.5.2 Transition P3:Logged-out to P0:Initial

An ADT port shall transition to P0:Initial after the logout duration time has expired.

4.3.2.5.3 Transition P3:Logged-Out to P1:Login

An ADT port shall transition to P1:Login state upon receiving a Port Login IU.

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 ADT 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 ADT port enters the P1:Login state.

Figure 11 shows the link negotiation state machine. The following subclauses describe the transitions and the actions taken in each state.

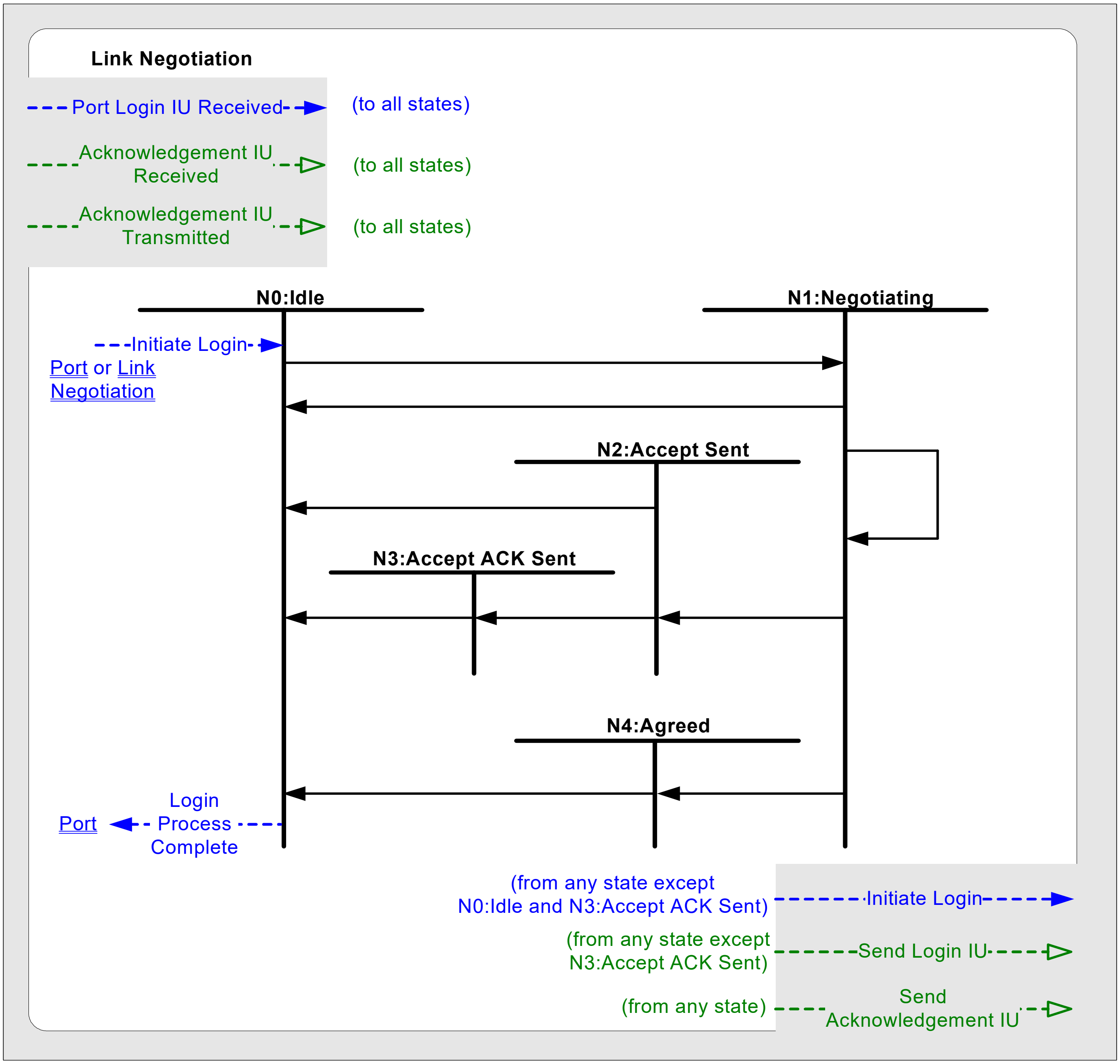


Figure 11 — Link Negotiation State Diagram

4.3.3.2 Precedence of Port Login exchanges

To avoid a deadlock condition if both ADT ports are attempting to initiate a Port Login exchange at the same time, the following rules shall apply. An ADT port in an automation device 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 ADT port if the automation device ADT 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 ADT port login process using the exchange originated by the other ADT 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 N1:Negotiating, and process the Port Login IU.

4.3.3.3 Common renegotiation conditions

The N1:Negotiating state and the N2:Accept Sent states behave the same under certain conditions. Both states set the ADT port operating parameters to the default operating parameters (see 4.2) and cause the link negotiation state machine to send an Initiate Login message to itself, if:

a) the link negotiation state machine receives a Port Login IU Received message and the ADT port detects a negotiation error (see 3.1.36);
b) the link negotiation state machine receives an Acknowledgement IU Received message for a NAK IU;
c) the ADT port detects an acknowledgement IU time-out; or
d) the following sequence occurs:
    1) the ADT port sends a Port Login IU in the current exchange;
    2) the ADT port receives an ACK IU for the Port Login IU;
    3) the ADT port sends an Acknowledgement IU Received message to the link negotiation state machine;
    4) The link negotiation state machine receives the Acknowledgement IU Received message sent by the ADT port;
    5) the ADT port transitions to a new state due to receiving the Acknowledgment IU Received message; and
    6) the link negotiation state machine fails to receive a Port Login IU Received message within 15 seconds after entering the state.

4.3.3.4 N0:Idle state

4.3.3.4.1 State description

The N0:Idle state waits for the link negotiation state machine to receive a Port Login IU Received message or an Initiate Login message.

While in the N0:Idle state, if the link negotiation state machine receives a Port Login IU Received message and the requested parameters (see 3.1.42) in the corresponding Port Login IU are not an element of the supported parameters (see 3.1.55), then for the next Port Login IU the ADT port sends it shall:

a) set the ACCEPT bit to zero; and
b) adjust link parameters, the MAJOR REVISION field, and the MINOR REVISION field as specified in 7.5.4.

While in the N0:Idle state, if the link negotiation state machine receives a Port Login IU Received message and the requested parameters in the corresponding Port Login IU are an element of the supported parameters, then for the next Port Login IU the ADT port sends it shall set:

a) the ACCEPT bit to one; and
b) all other parameters to the corresponding values from the received Port Login IU.

While in the N0:Idle state, if the link negotiation state machine receives a Port Login IU Received message and the ADT port detects a negotiation error, the ADT port shall set the ACCEPT bit to zero and set the other link parameters to the values from the starting parameter set (see 3.1.53) for the next Port Login IU it sends. See 7.5.4 for additional information about setting values for the MAJOR REVISION field and the MINOR REVISION field.

While in the N0:Idle state, if the link negotiation state machine receives an Initiate Login message, the ADT port shall set the ACCEPT bit to zero and set the other link parameters to the values from the starting parameter set (see 3.1.53) for the next Port Login IU it sends. See 7.5.4 for additional information about setting values for the MAJOR REVISION field and the MINOR REVISION field.

A link negotiation state machine in the N0:Idle state shall remain in the N0:Idle state when the ADT port sends a NAK IU for a Port Login IU Received message (see 4.6.2.6.1 and 4.6.2.7.1).

4.3.3.4.2 Transition N0:Idle to N1:Negotiating

The link negotiation state machine shall transition from N0:Idle to N1:Negotiating:

a) when the ADT port sends an ACK IU for a Port Login IU Received message; or
b) when the link negotiation state machine receives an Initiate Login message.

4.3.3.5 N1:Negotiating state

4.3.3.5.1 State description

When the link negotiation state machine enters the N1:Negotiating state the ADT port shall send a Port Login IU. The ADT port shall send the Port Login IU in a new exchange if the link negotiation state machine transitioned to the N1:Negotiating state as the result of an Initiate Login message (see 4.3.3.4.1 for the value of the link parameters and the ACCEPT bit). If no exchanges are available, then the ADT port shall abort all open exchanges and the Port Login IU shall contain an AOE bit set to one.

The ADT port shall send the Port Login IU in the existing exchange if the link negotiation state machine transitioned to the N1:Negotiating state as a result of a Port Login IU Received message and the ADT port does not detect a negotiation error (see 4.3.3.4.1 and this subclause for the value of the link parameters and the ACCEPT bit).

An ADT port with a link negotiation state machine in the N1:Negotiating state shall set the ADT port operating parameters to the default operating parameters (see 4.2) and the link negotiation state machine shall send an Initiate Login message to itself if detection of one of the common renegotiation conditions occurs (see 4.3.3.3).

If a link negotiation state machine in the N1:Negotiating state receives a Port Login IU Received message and the corresponding Port Login IU contains requested parameters that are not an element of the supported parameters, then for the next Port Login IU the ADT port sends it shall:

a) set the ACCEPT bit to zero; and
b) adjust link parameters, the MAJOR REVISION field, and the MINOR REVISION field as specified in 7.5.4.

If a link negotiation state machine in the N1:Negotiating state receives a Port Login IU Received message and the requested parameters in the corresponding Port Login IU are an element of the supported parameters, then for the next Port Login IU the ADT port sends it shall set:

a) the ACCEPT bit to one; and
b) all other parameters to the corresponding values from the received Port Login IU.

4.3.3.5.2 Transition N1:Negotiating to N0:Idle

A link negotiation state machine in the N1:Negotiating state shall transition to the N0:Idle state when it sends an Initiate Login message.

4.3.3.5.3 Transition N1:Negotiating to N1:Negotiating

A link negotiation state machine in the N1:Negotiating state shall transition to the N1:Negotiating state when the ADT port sends an ACK IU in response to the link negotiation state machine receiving a Port Login IU Received message.

NOTE 3 The transition from the N1:Negotiating state to the N1:Negotiating state implies entry into the N1:Negotiating state.

4.3.3.5.4 Transition N1:Negotiating to N2:Accept Sent

A link negotiation state machine in the N1:Negotiating state shall transition to the N2:Accept Sent state when the ADT port sends a Port Login IU with the ACCEPT bit set to one and the most recently received Port Login IU had the ACCEPT bit set to zero.

4.3.3.5.5 Transition N1:Negotiating to N4:Agreed

A link negotiation state machine in the N1:Negotiating state shall transition to the N4:Agreed state when the ADT port sends a Port Login IU with the ACCEPT bit set to one and the most recently received Port Login IU had the ACCEPT bit set to one.

4.3.3.6 N2:Accept Sent state

4.3.3.6.1 State description

The N2:Accept state waits for the link negotiation state machine to either receive an Acknowledgement IU Received message for an ACK IU or an acknowledgement IU time-out after sending a Port Login IU with the ACCEPT bit set to one and with parameters unchanged before it has received a Port Login IU with the ACCEPT bit set to one (i.e., it is the first ADT port to send a Port Login IU with the ACCEPT bit set to one).

When the link negotiation state machine enters the N2:Accept Sent state it shall wait until:

a) it receives an Acknowledgement IU Received message; or
b) the ADT port detects an acknowledgement IU time-out.

If a link negotiation state machine in the N2:Accept Sent state receives an Acknowledgement IU Received message for an ACK IU, it shall wait until it:

a) receives a Port Login IU Received message; or
b) has not received a Port Login IU Received message within 15 seconds after receiving the Acknowledgement IU Received message.

An ADT port with a link negotiation state machine in the N2:Accept Sent state shall set the ADT port operating parameters to the default operating parameters (see 4.2) and the link negotiation state machine shall send an Initiate Login message to itself if detection of one of the common renegotiation conditions occurs (see 4.3.3.3).

An ADT port with a link negotiation state machine in the N2:Accept Sent state shall send an ACK IU if the link negotiation state machine receives a Port Login IU Received message and:

a) each requested parameter in the received Port Login IU has the same value as the corresponding requested parameter in the most recently sent Port Login IU; and
b) the ACCEPT bit in the received Port Login IU is set to one.

An ADT port with a link negotiation state machine in the N2:Accept Sent state shall send a NAK IU if the link negotiation state machine receives a Port Login IU Received message and:

a) a requested parameter in the received Port Login IU has a different value than the corresponding requested parameter in the most recently sent Port Login IU; or
b) the ACCEPT bit in the received Port Login IU is set to zero.

4.3.3.6.2 Transition N2:Accept Sent to N0:Idle

A link negotiation state machine in the N1:Negotiating state shall transition to the N0:Idle state when it sends an Initiate Login message.

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

A link negotiation state machine in the N2:Accept Sent state shall transition to the N3:Accept ACK Sent state when the ADT port sends an ACK IU in response to receiving a Port Login IU Received message.

4.3.3.7 N3:Accept ACK Sent state

4.3.3.7.1 State description

A link negotiation state machine reaches the N3:Accept ACK Sent 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. When the link negotiation state machine enters the N3:Accept ACK Sent state:

1) the ADT port shall wait until completion of the transmission of the ACK IU sent at the transition from the N2:Accept Sent state to the N3:Accept ACK Sent state; and
2) the link negotiation state machine shall send a Login Process Complete message to the ADT port state machine.

4.3.3.7.2 Transition N3:Accept ACK Sent to N0:Idle

An ADT port in the N3:Accept ACK Sent state shall transition to the N0:Idle state when it sends a Login Process Complete message to the ADT port state machine.

4.3.3.8 N4:Agreed state

4.3.3.8.1 State description

A link negotiation state machine reaches the N4:Agreed state if the ADT port has sent a Port Login IU with the ACCEPT bit set to one and with parameters unchanged after it has received a Port Login IU with the ACCEPT bit set to one and with parameters unchanged (i.e., it is the second ADT port to send a Port Login IU with the ACCEPT bit set to one).

When the link negotiation state machine enters the N4:Agreed state, it shall wait until:

a) it receives an Acknowledgement IU Received message; or
b) the ADT port detects an acknowledgement IU time-out.

If the link negotiation state machine receives an Acknowledgement IU Received message for an ACK IU, it shall send a Login Process Complete message to the ADT port state machine.

If the link negotiation state machine receives an Acknowledgement IU Received message for a NAK IU or the ADT port detects an acknowledgement IU time-out, the ADT port shall set the ADT port operating parameters to the default operating parameters (see 4.2) and the link negotiation state machine shall send an Initiate Login message to itself.

4.3.3.8.2 Transition N4:Agreed to N0:Idle

An ADT port in the N4:Agreed state shall transition to the N0:Idle state when it receives an Acknowledgement IU Received message or the ADT port detects an acknowledgement IU time-out (see 4.6.1.2.2).

4.3.4 Transmitter state machine

4.3.4.1 Transmitter state machine overview

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

Figure 12 shows the transmitter state machine. The following subclauses describe the transitions and the actions taken in each state.

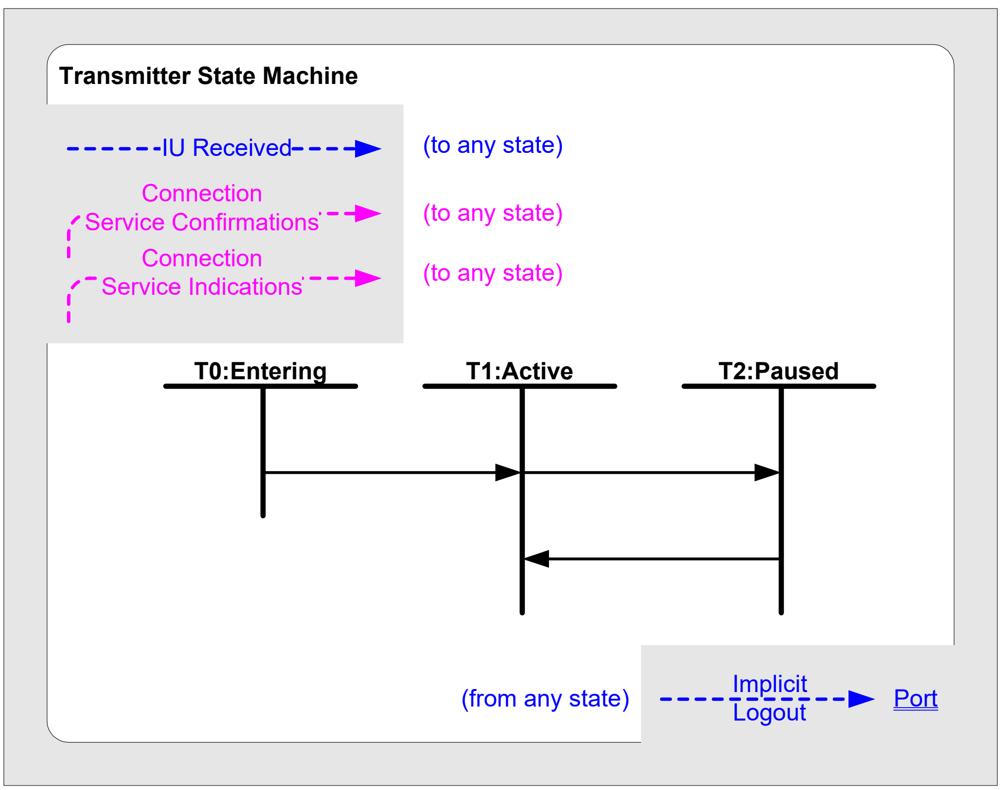


Figure 12 — Transmitter State Diagram

4.3.4.2 T0:Entering state

4.3.4.2.1 State description

On entry to this state the sADT port shall start a 100 millisecond timer. On entry to this state the iADT port may or may not start a 100 millisecond timer.

While in this state, an ADT port shall not transmit ADT information units.

If a **Disconnected** service confirmation or a **Disconnect Received** service indication is invoked by an iADT port while in this state, the ADT port shall send an Implicit Logout message to the ADT port state machine.

If a **Disconnect Immediate** service request is invoked by an iADT port while in this state, the ADT port shall send an Implicit Logout message to the ADT port state machine.

4.3.4.2.2 Transition T0:Entering to T1:Active

An ADT 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 or immediately if the iADT port is not using the 100 millisecond timer.

4.3.4.3 T1:Active state

4.3.4.3.1 State description

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

If a **Disconnected** service confirmation or a **Disconnect Received** service indication is invoked by an iADT port while in this state, the ADT port shall send an Implicit Logout message to the ADT port state machine.

If a **Disconnect Immediate** service request is invoked by an iADT port while in this state, the ADT port shall send an Implicit Logout message to the ADT port state machine.

4.3.4.3.2 Transition T1:Active to T2:Paused

An ADT 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

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

If a **Disconnect Immediate** service request is invoked by an iADT port while in this state, the ADT port shall send an Implicit Logout message to the ADT port state machine.

4.3.4.4.2 Transition T2:Paused to T1:Active

An ADT port shall transition to T1:Active state after receiving any frame other than a Port Login IU, Port Logout IU, Pause IU, Device Reset 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 ADT port. It is a sub-state machine of the ADT 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 ADT port enters P2:Logged-In state.

Figure 13 shows the transmitter error recovery state machine. The following subclauses describe the transitions and the actions taken in each state.

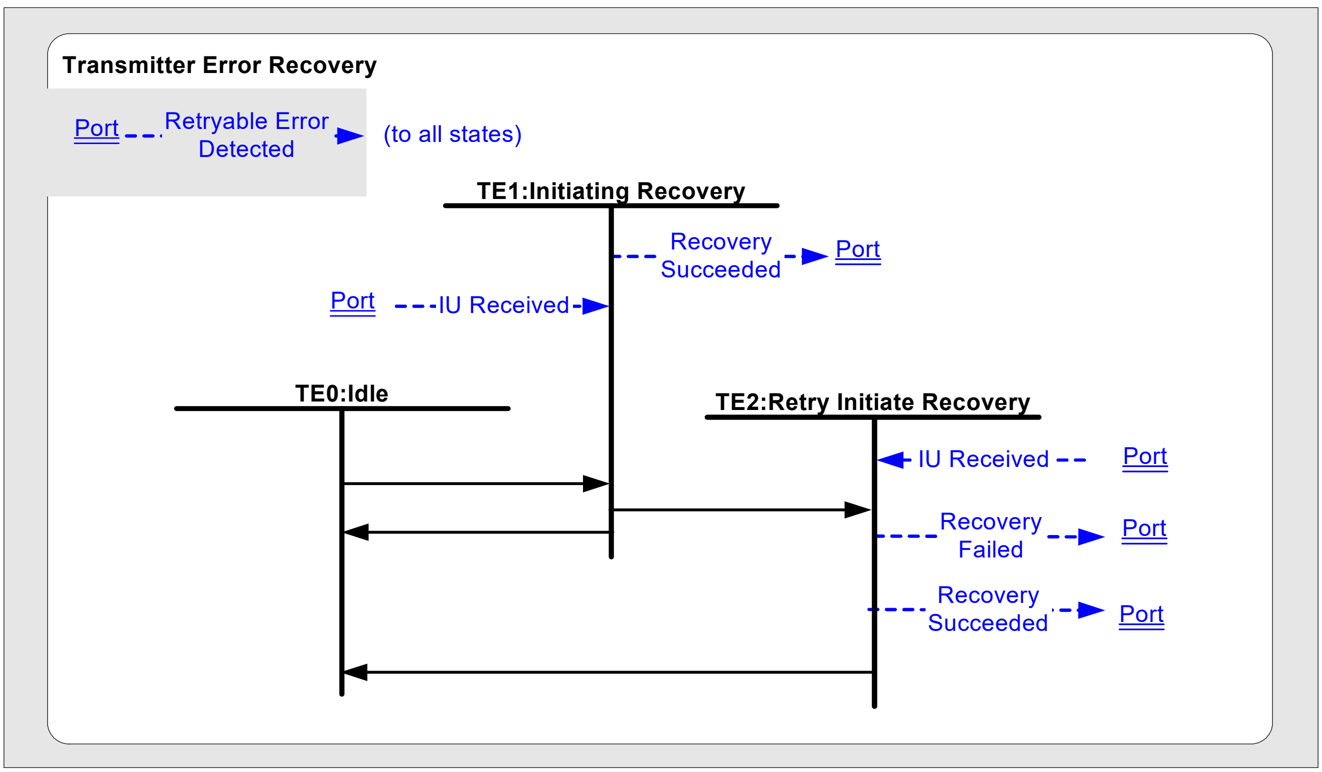


Figure 13 — Transmitter Error Recovery State Diagram

4.3.5.2 TE0:Idle state

4.3.5.2.1 State description

The TE0:Idle state waits for the ADT 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 ADT port shall transition to TE1:Initiating Recovery.

4.3.5.3 TE1:Initiating Recovery state

4.3.5.3.1 State description

An ADT port upon entering TE1:Initiating Recovery state shall send an Initiate Recovery IU. An ADT 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, Port Logout IUs, or Device Reset IUs.

An ADT port in TE1:Initiating Recovery state shall discard ACK IUs and NAK IUs for frames other than Port Login IUs, NOP IUs, Pause IUs, Port Logout IUs, and Device Reset 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 ADT port shall send a Recovery Succeeded message to the ADT port state machine and transition to TE0:Idle. This shall cause the retransmission of the frame that had failed.

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

An ADT port upon receiving a Retryable Error Detected message or an Initiate Recovery NAK IU shall transition to TE2:Retry Initiate Recovery.

4.3.5.4 TE2:Retry Initiate Recovery state

4.3.5.4.1 State description

An ADT port upon entering TE2:Retry Initiate Recovery state shall send an Initiate Recovery IU identical to the one sent in the TE1:Initiating Recovery state. An ADT port in TE2:Retry Initiate Recovery state shall not send any frames other than acknowledgement IUs, Port Login IUs, NOP IUs, Pause IUs, Port Logout IUs, Device Reset IUs.

An ADT 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, Port Logout IUs, and Device Reset 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 ADT port shall send a Recovery Succeeded message to the ADT port state machine and transition to TE0:Idle. This shall cause the retransmission 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 ADT port shall send a Recovery Failed message to the ADT 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 ADT port. It is a sub-state machine of the ADT port state P2:Logged-In. The state machine consists of the following states:

a) RE0:Idle;
b) RE1:Pending Recovery; and
c) RE2:Recovering.

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

Figure 14 shows the receiver error recovery state machine. The following subclauses describe the transitions and the actions taken in each state.

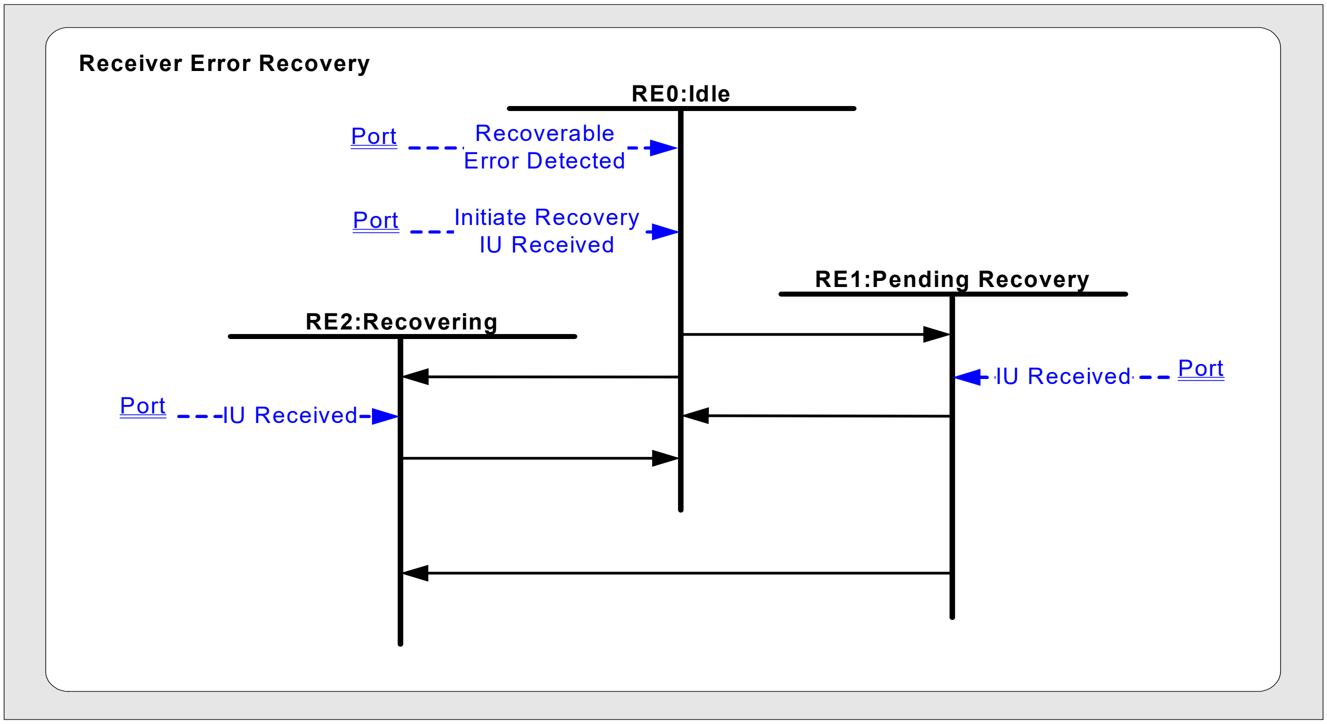


Figure 14 — Receiver Error Recovery State Diagram

4.3.6.2 RE0:Idle state

4.3.6.2.1 State description

The RE0:Idle state waits for the ADT 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 RE0:Idle to RE1:Pending Recovery

The ADT port shall transition to RE1:Pending Recovery upon receiving a Recoverable Error Detected message.

4.3.6.2.3 Transition RE0:Idle to RE2:Recovering

The ADT port shall transition to RE2: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 RE1:Pending Recovery state

4.3.6.3.1 State description

While an ADT port is in RE1:Pending Recovery state, receipt of a frame other than an Initiate Recovery IU, NOP IU, Port Login IU, Port Logout IU, Pause IU, Device Reset IU, or acknowledgment IU is an error and the ADT port shall send a NAK IU with a status code of AWAITING INITIATE RECOVERY IU (see table 51) and PR bit set to one.

4.3.6.3.2 Transition RE1:Pending Recovery to RE0:Idle

An ADT port shall transition to RE0: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 RE1:Pending Recovery to RE2:Recovering

An ADT port shall transition to RE2: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 RE2:Recovering state

4.3.6.4.1 State description

An ADT port in this state shall acknowledge frames with the same acknowledgement that was previously sent 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 RE2:Recovering to RE0: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 ADT port shall transition to the RE0:Idle state.

4.4 ACK Offset

The receiving ADT port explicitly acknowledges all frames that are transmitted. By default, an ADT 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 ADT 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 ADT ports to negotiate the number of frames that may be sent without waiting for acknowledgement, based on the resources available to the ADT ports. Regardless of the setting of this field, all frames shall be acknowledged in the order they are received.

Each ADT 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:

a) acknowledgement IU that is received; and
b) if an ACK timeout occurs.

Except for link service frames, an ADT port shall not transmit frames when the Unacknowledged Frame Counter is equal to the maximum ACK offset value. If the ADT 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, an ADT port may detect that the maximum ACK offset value has been exceeded. An ADT 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 51); 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 ADT port to detect missing frames; and
c) it allows an ADT port to identify a frame in order to retry transmission of frames that fail.

To accomplish these, each ADT 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 ADT 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 ADT port is operating in RE2:Recovering state; and
6) if the ADT port is operating in P2:Logged-In state, the ADT 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 ADT port shall send a NAK IU in response to the frame with a status code of UNEXPECTED FRAME NUMBER (see table 51) 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.

An ADT port that receives a Port Login IU, Port Logout IU, Pause IU, NOP IU, or Device Reset 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 ADT port shall respond with a NAK IU with a status code of INVALID OR ILLEGAL IU RECEIVED (see table 51).

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 ADT 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, an ADT 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.

4.6.1.2.3 NAK acknowledgement

It shall be considered an error condition if an ADT port receives a NAK IU (see 7.5.3.3).

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

NOTE 4 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 4.6.2.5.4.

4.6.2.2 Error recovery for symbol framing errors

After detecting four or more symbol framing errors without the receipt of a frame, an ADT port in P1:Login state or P2:Logged-In state shall abort all exchanges, set its operating parameters to the default operating parameters (see 4.2), transition to P1:Login state, and initiate a Port Login exchange with the AOE bit set to one. An ADT port in P0:Initial state or P3:Logged-Out state shall ignore symbol framing errors.

4.6.2.3 Recoverable error

If an ADT port detects a recoverable error (see 4.6.2.5) with a frame it receives it shall send a NAK IU to the other ADT port with the appropriate status code (see table 51) and the Pending Recovery (PR) bit set to one so that the ADT 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 ADT port shall send a Recoverable Error Detected message to the Receiver Error Recovery state machine causing it to transition to the RE1:Pending Recovery state.

4.6.2.4 Initiating recovery

If an ADT port receives an Initiate Recovery IU it is an indication that the other ADT port is attempting to recover from a retryable error. The following steps shall be taken by the receiving ADT 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 RE0:Idle state, the ADT port shall remain in its current state. If the frame numbers match and the Receiver Error Recovery state machine is in RE1:Pending Recovery state, the Receiver Error Recovery state machine shall transition to RE0:Idle; and
c) if the frame number does not match, this is an indication that an acknowledgement IU (i.e., either an ACK IU or a NAK IU) IU was lost in transmission. The Receiver Error Recovery state machine shall transition into RE2:Recovering state. While in this state, frames that are received by the ADT 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 RE0:Idle state.

4.6.2.5 Retryable error

4.6.2.5.1 Port login IUs

An ADT port with an ADT port state machine in P1:Login state that receives a NAK IU or detects an acknowledgment IU time-out shall restart the negotiation by sending an Initiate Login message, transitioning the link negotiation state machine to N0:Idle state, setting operating parameters to the default operating parameters (see 4.2), and initiating a new login exchange using starting parameters.

4.6.2.5.2 Port Logout, NOP, Initiate Recovery, Pause IUs, and Device Reset IUs

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

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

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

4.6.2.5.3 Initiate Recovery IU

For error recovery on Initiate Recovery IUs, see 4.6.2.5.4.

4.6.2.5.4 Non link service IUs

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

1) the ADT 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 ADT port waits for an Initiate Recovery ACK IU for that frame. No other frames shall be sent by that ADT 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, then the error ADT 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, then the ADT 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, then the ADT port shall send a Recovery Failed message to the ADT port state machine (see 4.3.5.4.2).

4.6.2.6 Protocol error

4.6.2.6.1 Port Login IU

If a protocol error is detected on a Port Login IU, then the ADT port shall send a NAK IU with the PR bit set to zero and the status code of NEGOTIATION ERROR (see table 51). The ADT port shall set the operating parameters to the default operating parameters (see 4.2), the ADT 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 IU, NOP IU, Pause IU, and Device Reset IU

If a protocol error is detected on a Port Logout IU, NOP IU, Pause IU, or Device Reset IU, then the ADT port shall send a NAK IU with PR bit set to zero and the appropriate status code (see table 51) then discard the frame.

4.6.2.6.3 Initiate Recovery IU

If a protocol error is detected on an Initiate Recovery IU, then the ADT port shall send an Initiate Recovery NAK IU with PR bit set to zero and the appropriate status code (see table 51) then discard the frame.

4.6.2.6.4 Non link service IUs

If an ADT 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 51) 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.

4.6.2.7 Resource limitation error

4.6.2.7.1 Port Login IU

If a resource limitation error is detected on a Port Login IU:

a) the ADT port shall send a NAK IU with a status code of OUT OF RESOURCES (see table 51) and set the operating parameters to default (see 4.2);
b) the ADT port state machine shall transition to P1:Login; and
c) the Link Negotiation state machine shall transition to N0:Idle.

4.6.2.7.2 Port Logout IU, NOP IU, Pause IU, and Device Reset IU

If a resource limitation error is detected on a Port Logout IU, NOP IU, or Pause IU, or Device Reset IU, then the ADT port shall send a NAK IU with the PR bit set to zero and the appropriate status code (see table 51) and discard the frame.

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

4.6.2.7.3 Initiate Recovery IU

If a resource limitation error is detected on an Initiate Recovery IU, then the ADT 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 51) and discard the Initiate Recovery IU.

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

4.6.2.7.4 Non link service IUs

If an ADT port detects a resource limitation error on a frame it receives, then it should send a NAK IU with the PR bit set to one and a status code of OUT OF RESOURCES (see table 51). An ADT port that receives a NAK IU indicating a resource limitation error shall consider the error retryable and perform the actions described in 4.6.2.5.4.

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

4.7 Hard reset

A hard reset is a response to an event that causes a hard reset condition (e.g., a power on condition, a Reset_a event, or a Device Reset IU). The target ADT 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-5.

The effect of the hard reset on tasks that have not completed, SCSI device reservations, and SCSI device operating modes is defined in SAM-5.

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) receives an ACK IU in response to a Device Reset IU;
d) is an sADT port and detects the change of state of the Sense line from presence to absence (i.e., Sense_a for DT device port and Sense_d for automation device ADT port) (see figure 17);
e) receives a **Reset Received** service indication (see 6.2.13); or
f) receives an Implicit Logout message while the ADT port state machine is in the P2:Logged-In state.

If an ADT port detects an I_T nexus loss event it shall send a Nexus Loss event notification indication (see SAM-5) to the SCSI Application layer.

If an I_T Nexus loss occurs, then the ADT port shall:

a) abort all open exchanges; and
b) set the operating parameters to default (see 4.2).

4.9 Transport protocol variations from SAM-5

The ADT transport protocol provides all of the services mandated by SAM-5. In addition to the mandatory protocol services, ADT provides several extensions for the initiator ADT 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 (see ADC-4)) from the perspective of SAM-5. 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-5 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-5, 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, for a command identified by an I_T_L nexus and Command Identifier, shall:

1) if there is data to be sent to the device server for that command:
   A) invoke the Send Data-Out transport protocol service after performing the Execute Command remote procedure call if the Data-Out Buffer Size argument in the Execute Command remote procedure call is set to zero; or
   B) not invoke the Send Data-Out transport protocol service after performing the Execute Command remote procedure call until it receives a Data-Out Delivered confirmation for that command if the Data-Out Buffer Size argument in the Execute Command remote procedure call is not set to zero;
2) while there is more data to be sent to the device server,
   A) invoke the Send Data-Out transport protocol service for that command after receiving the Data-Out Delivered confirmation; and
   B) not invoke the Send Data-Out transport protocol service for that command until it receives a Data-Out Delivered confirmation for that command.

An application client, for a command identified by an I_T_L nexus and Command Identifier, shall:

1) if there is data expected from the device server for that command:
   A) invoke the Receive Data-In transport protocol service after performing the Execute Command remote procedure call if the Data-In Buffer Size argument in the Execute Command remote procedure call is set to zero; or
   B) not invoke the Receive Data-In transport protocol service after performing the Execute Command remote procedure call until it receives a Data-In Received confirmation for that command if the Data-In Buffer Size argument in the Execute Command remote procedure call is not set to zero;
2) while there is more data expected from the device server:
   A) invoke the Receive Data-In transport protocol service for that command after receiving the Data-In Received confirmation; and
   B) not invoke the Receive Data-In transport protocol service for that command until it receives a Data-In Received confirmation for that command.

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

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

4.10 Acknowledgement time-out period

4.10.1 Acknowledgement time-out period calculation

When changing operating parameters (see 3.1.39), an sADT port shall calculate a new acknowledgement IU time-out period using the formula in figure 15. The sADT port shall apply the new acknowledgement IU time-out period to every frame transmitted after changing operating parameters

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

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 7.1).

$Offset_{Max}$ 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 7.5.3.3).

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 15 — sADT port acknowledgement time-out period

An iADT port shall use an initial acknowledgement time-out period of 2 500 milliseconds.

4.10.2 . Acknowledgement time-out period negotiation

4.10.2.1 Acknowledgement time-out period negotiation introduction

This standard allows for an ADT port to have an acknowledgement time-out period that differs from the acknowledgement time-out period of the other ADT port forming the link. The Time-out IU (see 7.5.12) provides the mechanism by which an ADT port:

a) reports its own acknowledgement time-out period parameters;
b) discovers the acknowledgement time-out period parameters of the other ADT port; and
c) requests that the other ADT port alter its acknowledgement time-out period value. This standard does not mandate the usage model described in 4.10.2.2 and 4.10.2.3.

4.10.2.2 Discovering another ADT port's acknowledgement time-out period parameters

An ADT port interrogates the acknowledgement IU time-out parameters of the other ADT port forming the link through the use of a time-out discovery exchange (see 3.1.59). To discover the acknowledgement IU time-out parameters of the other ADT port forming the link, an ADT port initiates the exchange by sending a Time-out IU (see 7.5.12) with the ACTION CODE field set to DISCOVER. If the other ADT port forming the link receives the Time-out IU without error, it responds with:

1) an ACK IU; and
2) a Time-out IU with the ACTION CODE field set to REPORT and the CURRENT ACKNOWLEDGEMENT TIME-OUT field, MAXIMUM ACKNOWLEDGEMENT IU TIME-OUT field, MINIMUM ACKNOWLEDGEMENT IU TIME-OUT field, and TIME-OUT RESOLUTION field set as specified by table 56.

Provided it received this Time-out IU without error, the ADT port that initiated the time-out discovery exchange concludes it by sending an ACK IU.

4.10.2.3 Altering another ADT port's acknowledgement time-out period

An ADT port requests a change to the acknowledgement IU time-out period of the other ADT port forming the link by sending a Time-out IU (see 7.5.12) in a simple link service exchange (see 3.1.52) with the ACTION CODE field set to REQUEST CHANGE and the CURRENT ACKNOWLEDGEMENT IU TIME-OUT field set to the desired time-out value.

4.10.2.4 Exception conditions

If an ADT port sends a Time-out IU with the ACTION CODE field set to DISCOVER in a new exchange and receives a corresponding ACK IU but fails to receive a Time-out IU with the ACTION CODE field set to REPORT in the same exchange, then the ADT port may:

a) send another Time-out IU with the ACTION CODE field set to DISCOVER in the same exchange; or
b) send a Time-out IU with the ACTION CODE field set to REQUEST CHANGE and the CURRENT ACKNOWLEDGEMENT IU TIME-OUT field set to the desired time-out value in a simple link exchange (see 4.10.2.3).

If an ADT port receives a Time-out IU with the ACTION CODE field set to REPORT in a nonexistent exchange, then it shall transmit a NAK IU with a status code of INVALID EXCHANGE ID (see table 51) and discard the Time-out IU.

If an ADT port receives a Time-out IU with an exchange ID that differs from an existing time-out discovery exchange, then the other ADT port forming the link has started a new time-out discovery exchange or a new simple link exchange. The ADT port shall abort the existing time-out discovery exchange and process the new exchange as specified in 4.10.2.2 and 4.10.2.3.

NOTE 5 Exception conditions may arise that this clause does not describe.

5 Physical layer

5.1 Physical layer introduction

The ADT physical layer defines a number of connection types. Some of these connections are used by all ADT ports, some are used only by sADT ports, and some are used only by iADT ports. A connector is defined which may be used by sADT ports.

5.2 Electrical Characteristics

5.2.1 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. An ADT compliance point may be an output compliance point or an input compliance point.

5.2.2 Cabling

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

NOTE 6 The sADT connection specifications in sub clauses 5.2.3 through 5.2.5 assume cable with a R < 400 ohms/km, Z_0 = 100 ohms (nominal), and C = 50 pF/m (nominal).

5.2.3 Sense connection

A Sense connection is a complete uni-directional signal path from one device supporting the ADT protocol to a second device supporting the ADT protocol. 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.		

5.2.4 sADT signal connection

An sADT signal connection is a complete uni-directional signal path from one sADT port to a second sADT port. An sADT signal connection includes:

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

An sADT signal connection shall use single-ended signalling. An sADT port shall include termination for sADT 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 an sADT signal connection at the output compliance point.

Table 3 — sADT 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	Not specified	$V_{OH} <= 3.6\ V$
[a] V_{dd} is the positive supply voltage at the receiving end.		

NOTE 7 There is no need to specify the current for the negated state.

Table 4 describes the electrical characteristics of an sADT signal connection at the input compliance point.

Table 4 — sADT 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	Not specfied	$0.7\ V_{dd}$[a] $< V_{IH} <= 3.6\ V$; $400\ mV < V_{hysteresis}$
[a] V_{dd} is the positive supply voltage at the receiving end.		

NOTE 8 There is no need to specify the current for the negated state.

Table 5 describes the timing characteristics of an sADT signal connection.

Table 5 — sADT signal connection timing characteristics

Characteristic	Timing	Timing
Duration	$1.5\ \mu s < t_L$	$1.5\ \mu s < t_H$
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 16 defines $V_{hysteresis}$.

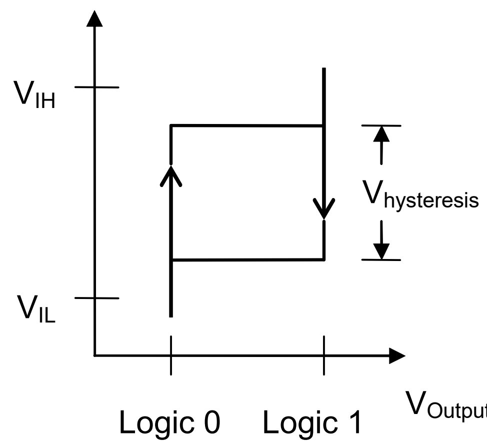


Figure 16 — $V_{hysteresis}$ definition

5.2.5 Transmit-receive connections

5.2.5.1 Transmit-receive connections introduction

This standard defines two sets of transmit-receive connections. The serial transmit-receive connection applies to implementations using the transmit-receive connections defined in 5.2.5.2. The Ethernet transmit-receive connection applies to implementations using the transmit-receive connections defined in 5.2.5.3.

5.2.5.2 Serial transmit-receive connections

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

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

A Tx-Rx connection shall conform to ANSI/EIA/TIA-422-B-1994 as measured at the associated compliance points.

A Tx-Rx connection shall support 9 600 baud and may support additional modulation rates. Example optional Tx-Rx modulation rates are shown 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
230 400
460 800
921 600

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 17 .

start	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	stop

Figure 17 — Asynchronous Transmission Format

5.2.5.3 Ethernet transmit-receive connections

The electrical characteristics of Ethernet transmit-receive connections are defined in IEEE 802.3-2005.

5.2.6 Ethernet LED connections

Ethernet LED connections may be used by a DT device to drive light-emitting diodes (LEDs) to indicate the status of the Ethernet connections. Table 7 describes the electrical characteristics of an Ethernet LED connection at the output compliance point. The description assumes that:

a) the output is an open-collector type; and
b) an LED and a resistor are connected in series between the output and the positive supply voltage.

Table 7 — Ethernet LED connection output characteristics

Signal State	Current	Voltage
Asserted	$-25\ mA < I_{OL}$	$0\ V < V_{OL} < 0.4\ V$
Negated	$I_{OL} < 20\ \mu A$	$V_{OH} < 5.5\ V$

5.3 Connection Instances

5.3.1 Sense connection instances

Table 8 defines the sense connections used by ADT ports.

Table 8 — Sense connections

Connection name	O/M[a]	Connection type	Driven by	Connection definition
Sense_a	O/M[b]	Sense	automation device port	A DT device may use this connection to sense the presence or absence of an automation device.
Sense_{aux}	O	Sense		This standard does not define the use of this connection.
Sense_d	O	Sense	DT device port	An automation device shall use this connection to sense the presence or absence of a DT device.

[a] O indicates support is optional; M indicates support is mandatory.
[b] Mandatory for sADT ports. Optional for iADT ports.

5.3.2 sADT signal connection instances

Table 9 defines the signal connections used by sADT ports.

Table 9 — Signal connections

Connection name	O/M[a]	Connection type	Driven by	Connection definition
Reset_a	O	Signal	automation device ADT port	An automation device may use this connection to signal a reset request to a DT device by invoking the **Reset** service request. A DT device shall treat the receipt of a signal on this connection as an invocation of the **Reset Received** service indication (see 6.2.13).
Signal_{aux}	O	Signal		This standard does not define the use of this connection.

[a] O indicates support is optional; M indicates support is mandatory.

5.3.3 Serial transmit-receive connection instances

Table 10 defines the transmit-receive connections for sADT ports.

Table 10 — ADT serial transmit-receive connections

Connection name	O/M[a]	Connection type	Driven by	Connection definition
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 for sADT ports.				

5.3.4 Ethernet transmit-receive connection instances

Table 11 defines the transmit-receive connections for iADT ports using Ethernet.

Table 11 — Ethernet transmit-receive connections for iADT ports using Ethernet

Connection name	O/M[a]	Connection type	Driven by	Connection definition
TX_D1+	M	MDI[b]	c	See IEEE 802.3-2005
TX_D1-	M	MDI[b]	c	See IEEE 802.3-2005
RX_D2+	M	MDI[b]	c	See IEEE 802.3-2005
RX_D2-	M	MDI[b]	c	See IEEE 802.3-2005
BI_D3+	O	MDI[b]	d	See IEEE 802.3-2005
BI_D3-	O	MDI[b]	d	See IEEE 802.3-2005
BI_D4+	O	MDI[b]	d	See IEEE 802.3-2005
BI_D4-	O	MDI[b]	d	See IEEE 802.3-2005
[a] O indicates support is optional, M indicates support is mandatory for iADT ports. [b] Medium Dependent Interface (MDI) and alternate MDI (MDI-X) are defined in IEEE 802.3-2005. An MDI connection shall support autonegotiation of link speed. [c] In the MDI configuration, the iADT port drives the TX_D1 pair. In the MDI-X configuration, the iADT port drives the RX_D2 pair. [d] The BI_D3 and BI_D4 pairs are driven as indicated by IEEE 802.3-2005.				

5.3.5 Ethernet LED connection instances

Table 12 defines the Ethernet LED connections used by the DT device supporting Ethernet transmit-receive connections.

Table 12 — Ethernet LED connections

Connection name	O/M[a]	Connection type	Asserted by
LED_{active}	O	LED	DT device port
LED_{signal}	O	LED	DT device port

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

A DT device supporting both the LED_{signal} and LED_{active} connections may signal in the following manner:

a) if carrier is detected (see IEEE 802.3-2005), the LED_{signal} connection is asserted. If no carrier is detected, the LED_{signal} connection is deasserted; and
b) if data is being transmitted or received on the TX_D1 or RX_D2 connections (see IEEE 802.3-2005), the LED_{active} connection is alternately asserted and deasserted. If no data is being received on the TX_D1 or RX_D2 connections, the LED_{active} connection is deasserted.

A DT device supporting only the LED_{signal} connection may signal in the following manner:

a) if no carrier is detected, the LED_{signal} connection is deasserted;
b) if carrier is detected and no data is being received on the TX_D1 and RX_D2 connections, the LED_{signal} connection is asserted; and
c) if data is being received on the TX_D1 or RX_D2 connections, the LED_{signal} connection is alternately asserted and deasserted.

5.4 Connector pin-out

sADT ports may use the plug connector defined in SFF-8054. Table 13 defines the pinout for the ADT port connector on the DT device.

Table 13 — DT device sADT port connector pinout

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

No connector pin-out is defined for the use of iADT ports.

6 Connection layer

6.1 Connection layer introduction

An ADT port shall establish a connection with another ADT port before transmitting or receiving encoded characters. Each connection is associated with one and only one session. A connection shall exist implicitly between two sADT ports which are physically connected. A connection between two iADT ports shall be established explicitly.

The ADT connection layer provides connection services for transmitting and receiving sequences of encoded characters between ADT ports. Table 14 lists the ADT connection services.

Table 14 — ADT connection services

Connection service	Connection service type	Invoked by device type	Supported by port type
Connect	Request	Either	iADT
Connected	Confirmation	Either	iADT
Connect Received	Indication	Either	iADT
Send	Request	Either	sADT and iADT
Sent	Confirmation	Either	sADT and iADT
Receive	Request	Either	sADT and iADT
Received	Confirmation	Either	sADT and iADT
Disconnect	Request	Either	iADT
Disconnected	Confirmation	Either	iADT
Disconnect Immediate	Request	Either	iADT
Disconnect Received	Indication	Either	iADT
Reset	Request	Automation device	sADT and iADT
Reset Received	Indication	DT device	sADT and iADT

An iADT port in a DT device shall listen for a connection from any iADT port and an iADT port in an automation device may listen for a connection from any iADT port. An iADT port may initiate a connection. An iADT port awaits a connection by performing a TCP passive OPEN specifying the iADT port number (4169/tcp) or the iADT over TLS (iADT-TLS) port number (9614/tcp) for the local port. An iADT port initiates a connection by invoking the **Connect** service request. When the connection is established, the active iADT port receives a **Connected** service confirmation and the passive iADT port receives a **Connect Received** service indication. The iADT ports may exchange TCP control segments in order to establish the connection.

All ADT IUs using the iADT-TLS port number shall use the Transport Layer Security (TLS) protocol (see RFC 5246).

Figure 18 shows an example of the relationships among the connection services used to establish a connection between two iADT ports. The communication between the two devices is defined in RFC 793 and may constitute more than the two communications shown.

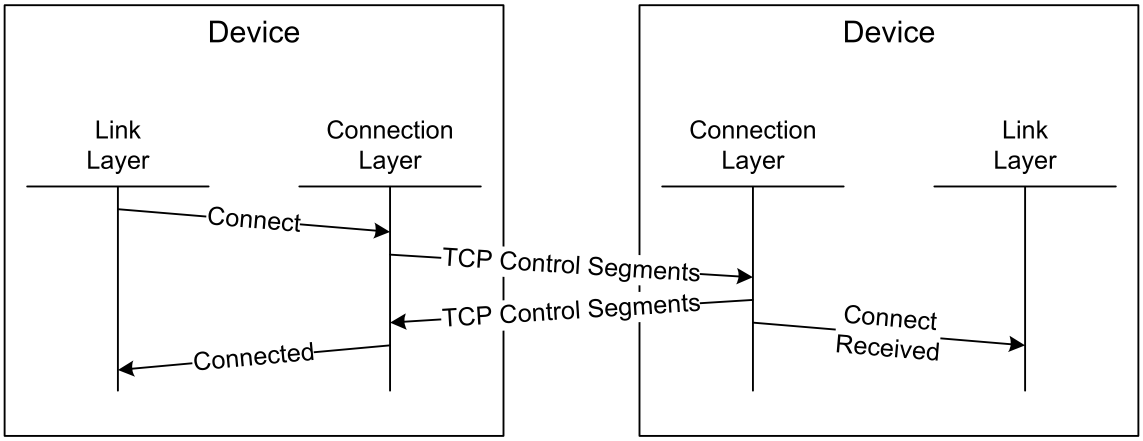


Figure 18 — Connection services for establishing a connection between iADT ports

Figure 19 shows the relationships among the connection services used to transfer data.

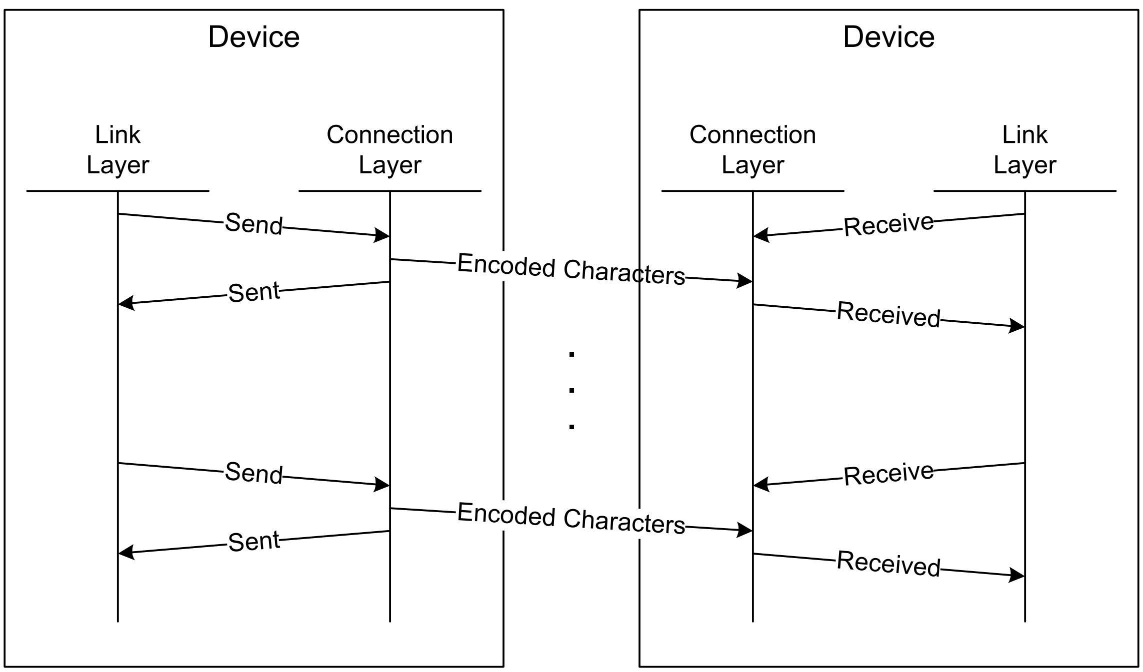


Figure 19 — Connection services for transferring data

An ADT port link layer sends encoded characters on a connection within a session by invoking the **Send** service request. When the **Sent** service confirmation is invoked, the characters have been accepted by the connection layer for delivery, and may have been transmitted.

An ADT port link layer receives encoded characters on a connection by invoking the **Receive** service request and then processing the **Received** service confirmation. When characters have been received by the connection layer, the **Received** service confirmation is invoked. The **Received** service confirmation indicates the number of characters that have been received. To receive more characters on the connection, the ADT port link layer must invoke the **Receive** service request again.

sADT ports transmit encoded characters on the RS-422 physical layer. iADT ports transmit encoded characters in TCP segments over the physical layer used by the iADT port.

Figure 20 shows the relationships among the connection services used to close a connection between two iADT ports. The communication between the two devices is defined in RFC 793 and may constitute more than the two communications shown.

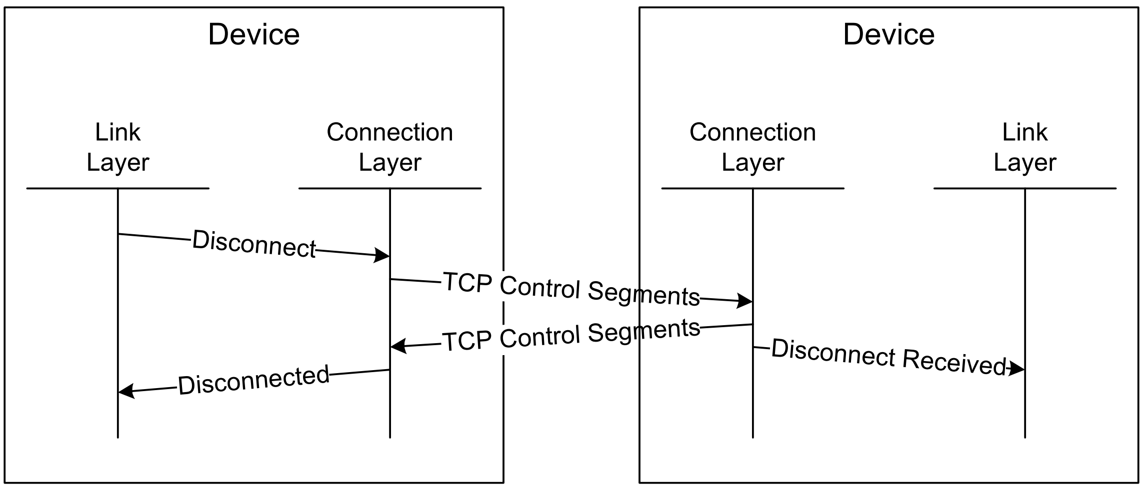


Figure 20 — Connection services for closing a connection between iADT ports

An iADT port link layer closes a connection by invoking the **Disconnect** service request. An iADT port shall attempt to transmit any characters that have been successfully submitted for delivery by earlier **Send** service requests before the connection is closed. When an iADT port receives a **Disconnected** service confirmation or a **Disconnect Received** service indication, the connection is closed and no more characters shall be received. iADT ports exchange TCP control segments in order to close the connection.

Figure 21 shows the relationships among the connection services used to perform a reset.

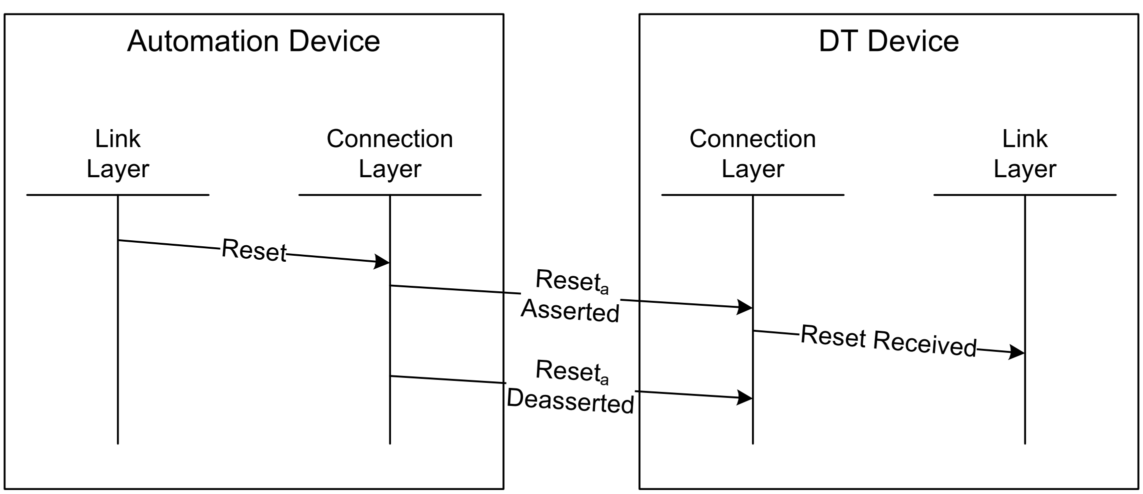


Figure 21 — Connection services for performing a reset

An ADT port link layer in an automation device resets an ADT port in a DT device by invoking the **Reset** service request. The ADT port then asserts the $Reset_a$ connection. Assertion of the $Reset_a$ connection causes the connection layer in the ADT port in the DT device to invoke a **Reset Received** service indication.

6.2 Connection layer connection service definitions

6.2.1 Connect request

A local iADT port link layer invokes the **Connect** connection service request to initiate a connection with a specific remote iADT port.

Service Response = **Connect (IN (Session ID, Local IP Address, Remote IP Address, Method))**

Input arguments:

Session ID: A link layer identifier for the session with which the connection will be associated.

Local IP Address: The IP address of the local iADT port.

Remote IP Address: The IP address of the remote iADT port.

Method: See table 15.

Table 15 — Method

Value	Description
Unsecured	The connection is made via unsecured TCP.
TLS	The connection is made via Transport Layer Security (TLS).

NOTE 9 The TCP port number of the remote port is defined in 6.1.

Service Response may be one of the values specified in table 16.

Table 16 — Connect service request error processing

Service response	Cause	Possible error recovery procedure
GOOD	The request completed successfully.	n/a
INVALID LOCAL IP ADDRESS	The Local IP Address argument did not specify a valid local IP address.	Retry request with valid local IP address.
INVALID REMOTE IP ADDRESS	The Remote IP Address argument did not specify a valid remote IP address.	Retry request with valid remote IP address.
NO PHYSICAL CONNECTION	The Sense connection was not asserted or because the Ethernet iADT port did not detect a signal.	Not specified by this standard.

6.2.2 Connected service confirmation

A local iADT port connection layer invokes the **Connected** service confirmation to notify the iADT port link layer that the connection initiated by a **Connect** service request has been established.

Connected (IN (Session ID, Connection ID))

Input arguments:

Session ID: A link layer identifier for the session with which the connection is associated.

Connection ID: A connection layer identifier for the connection.

6.2.3 Connect Received service indication

A local iADT port connection layer invokes the **Connect Received** service indication to notify the iADT port link layer that a connection initiated by a remote iADT port has been established.

Connect Received (IN (Connection ID, Remote IP Address))

Input arguments:

Connection ID: A connection layer identifier for the connection.

Remote IP Address: The IP address of the remote iADT port.

6.2.4 Send service request

An ADT port link layer invokes the **Send** service request to send data on a connection. An ADT port link layer shall not invoke the **Send** service request for a session for which:

a) there is no established connection; or
b) a connection close has been initiated (i.e., the **Disconnect** service request has been invoked but the **Disconnected** service confirmation has not been invoked).

If a subsequent **Send** service request is invoked before all of the data in the buffer specified by a previous **Send** service request has been transmitted, then the ADT port shall transmit all of the data in the buffer for the previous invocation before transmitting any data in the buffer of the subsequent invocation.

If the **Send** service request returns a service response of GOOD, then the ADT port link layer may modify the contents of the buffer without affecting the data to be transmitted by the connection layer.

When the **Send** service request returns a service response of GOOD, then the characters may or may not have been transmitted on the physical connection.

Service Response = Send (IN (Session ID, Connection ID, Buffer, Buffer Size))

Input arguments:

Session ID:	A link layer identifier for the session with which the connection is associated.
Connection ID:	A connection layer identifier for the connection.
Buffer:	A buffer containing data to be transmitted. The data in the buffer shall be encoded (see 7.2).
Buffer Size:	The number of characters of encoded data to be transmitted on the connection.

Service Response may be one of the values specified in table 17.

Table 17 — Send service request error processing

Service response	Cause	Possible error recovery procedure
GOOD	The request completed successfully.	n/a
INVALID CONNECTION STATE	The Connection argument did not specify an established connection.	Create new connection and retry operation.
INVALID BUFFER	The Buffer argument did not specify a valid buffer.	Retry request with valid buffer.
OUT OF RESOURCES	The ADT port lacked resources to accept more characters for transmission.	Retry send after a delay.

6.2.5 Sent service confirmation

An ADT port connection layer invokes the **Sent** service confirmation to notify the ADT port link layer that the characters specified by the **Send** service request have been accepted for transmission. In an sADT port, the **Sent** service confirmation also indicates that the characters have been transmitted by the physical layer.

Sent (IN (Session ID, Connection ID))

Input arguments:

Session ID:	A link layer identifier for the session with which the connection is associated.
Connection ID:	A connection layer identifier for the connection.

6.2.6 Receive service request

An ADT port link layer invokes the **Receive** service request to receive data from a connection. The data received shall be processed as specified in clause 7. An ADT port link layer shall not invoke the **Receive** service request for a session for which there is no established connection.

An ADT port link layer may invoke the **Receive** service request for a connection for which a connection close has been initiated (i.e., the **Disconnect** service request has been invoked but the **Disconnected** service confirmation has not been processed).

If the link layer invokes the **Receive** service request a second time before the **Received** service confirmation has been invoked, then the connection layer shall reject the second **Receive** service request with a service response of RECEIVE PENDING.

Service Response = **Receive (IN (Session ID, Connection ID, Buffer, Buffer Size))**

Input arguments:

Session ID:	A link layer identifier for the session with which the connection is associated.
Connection ID:	A connection layer identifier for the connection.
Buffer:	A buffer to contain received data.
Buffer Size:	The maximum number of characters of encoded data to be placed in the buffer.

Service Response may be one of the values specified in table 18.

Table 18 — Receive service request error processing

Service response	Cause	Possible error recovery procedure
GOOD	The request completed successfully.	n/a
INVALID CONNECTION STATE	The request failed because the Connection ID argument did not specify a valid connection.	Retry operation with a valid connection.
RECEIVE PENDING	The request failed because the ADT port has invoked the **Receive** service request and the ADT port has not yet invoked the **Received** service confirmation.	Retry **Receive** service request after processing **Received** service confirmation.

6.2.7 Received service confirmation

A local ADT port connection layer invokes the **Received** service confirmation to notify the ADT port link layer that a number of characters have been received.

There is not a one-to-one correspondence between invocations of **Send** in the local ADT port and invocations of **Received** in the remote ADT port (i.e., the characters delivered in one invocation of **Received** on the remote ADT port may have been sent by one or more invocations of **Send** on the local ADT port). Similarly, the characters sent in one invocation of **Send** may be delivered in one or more invocations of **Received**.

An ADT frame may be transferred to the link layer in multiple invocations of the **Received** service indication. The beginning of the frame may not align with the start of the buffer and the end of the frame may not align with the end of the buffer (i.e., the first character in the buffer may not be an SOF and the last character in the buffer may not be an EOF).

If a connection is closed then the **Received** service confirmation may be invoked to deliver the characters which have been received. The received character count may be zero.

Received (IN (Session ID, Connection ID, Buffer, Received Character Count))

Input arguments:

Session ID:	A link layer identifier for the session with which the connection is associated.
Connection ID:	A connection layer identifier for the connection.
Buffer:	A buffer containing data received. The data in the buffer shall be encoded (see 7.2).
Received Character Count:	The number of characters received and placed in the buffer. The number of characters may be less than the buffer size specified in the corresponding invocation of the **Receive** service request (e.g., in a non-blocking implementation).

6.2.8 Disconnect service request

A local iADT port connection layer receives the **Disconnect** service request to close a connection to a remote iADT port. An iADT port link layer shall not invoke the **Disconnect** service request for a session for which:

a) there is no established connection; or
b) a connection close has been initiated (i.e., the **Disconnect** service request has been invoked but the **Disconnected** service confirmation has not been invoked).

Service Response = Disconnect (IN (Session ID, Connection ID))

Input arguments:

Session ID:	A link layer identifier for the session with which the connection is associated.
Connection ID:	A connection layer identifier for the connection.

Service Response may be one of the values specified in table 19.

Table 19 — Disconnect service request error processing

Service response	Cause	Possible error recovery procedure
GOOD	The request completed successfully.	n/a
INVALID CONNECTION STATE	The request failed because the Connection ID argument did not specify a session with an established connection.	Not specified by this standard.

6.2.9 Disconnect Immediate service request

An iADT port link layer invokes the **Disconnect Immediate** service request to close a connection without receiving any further data on the connection and without invoking a service confirmation.

Disconnect Immediate (IN (Connection ID))

Input arguments:

Connection ID: A connection layer identifier for the connection.

6.2.10 Disconnected service confirmation

A local iADT port connection layer invokes the **Disconnected** service confirmation to notify the iADT port link layer that either the connection has been closed or a **Connect** service request has failed. The iADT port connection layer shall not invoke the **Disconnected** service confirmation or the **Disconnect Received** service indication until all received characters have been transferred to the iADT port link layer by use of the **Received** service confirmation.

Disconnected (IN (Session ID, Reason))

Input arguments:

Session ID: A link layer identifier for the session with which the connection was associated.

Reason: The reason that the connection was closed.

Reason may be one of the values specified in table 20.

Table 20 — Disconnected service confirmation error processing

Reason	Cause	Possible error recovery procedure
GOOD	The request completed successfully.	n/a
SENSE DEASSERTED	The ADT port detected transition of the $Sense_a$ connection from asserted to deasserted.	n/a
LOSS OF SIGNAL	The iADT port detected loss of signal.	n/a
CONNECTION REFUSED	The remote iADT port rejected the connection attempt.	Retry connection later.

6.2.11 Disconnect Received service indication

A local iADT port connection layer invokes the **Disconnect Received** service indication to notify the iADT port link layer that the connection has been closed by the remote ADT port. The iADT port connection layer shall not invoke the **Disconnect Received** service indication until all received characters have been transferred to the link layer.

The value in the Connection ID argument shall not be used in any service requests until it is reported in the **Connection ID** argument of a subsequent **Connected** service confirmation or in the **Connection ID** argument of a subsequent **Connect Received** service indication.

If an iADT port detects loss of signal, then the connection layer may invoke the **Disconnect Received** service indication.

Disconnect Received (IN (Connection ID, Reason))

Input arguments:

Connection ID: A connection layer identifier for the connection.

Reason: The reason that the connection was closed.

Reason may be one of the values specified in table 21.

Table 21 — Disconnect Received service indication error processing

Reason	Cause	Possible error recovery procedure
GOOD	The request completed successfully.	n/a
CLOSED STATE	The iADT port detected loss of the TCP connection (see RFC 793) but not loss of signal.	Open a new connection and retry.
SENSE DEASSERTED	The ADT port detected transition of the $Sense_a$ connection from asserted to deasserted.	Try again later when sense line is asserted.
LOSS OF SIGNAL	The iADT port detected loss of signal.	Try again later when signal is restored.
CONNECTION LOST	The connection is no longer valid.	Open a new connection and retry.

6.2.12 Reset service request

An ADT port link layer in an automation device invokes the **Reset** service request to request the reset of the remote ADT port by asserting the $Reset_a$ connection (see table 9).

Reset (IN (DTD Identifier))

Input arguments:

DTD Identifier: The identifier for the DT device to be reset.

6.2.13 Reset Received service indication

An ADT port connection layer invokes the **Reset Received** service indication in a DT device to indicate to the link layer that the ADT port has been reset (e.g., by assertion of the $Reset_a$ connection (see table 9)).

Reset Received (IN (Reset Signal Connection Identifier))

Input arguments:

Reset Signal Connection Identifier: An identifier for the Reset connection that was asserted.

6.3 sADT port support of connection services

6.3.1 Data transmission

Table 22 shows how the arguments to the **Send** service request are used by the sADT port.

Table 22 — Send service request usage by sADT port

Argument	sADT port implementation
Session ID	The link layer identifier for the session with which the connection is associated.
Connection ID	The connection layer identifier for the connection.
Buffer	The buffer containing data to be transmitted.
Buffer Size	The number of characters in the buffer to be sent. The characters are encoded (i.e., the number includes Escape characters).

When the **Send** service request is successfully invoked by the link layer, the sADT port connection layer shall invoke the **Sent** service confirmation after the encoded characters have been transmitted by the physical port.

Table 23 shows how the argument to the Sent service confirmation is used by the sADT port.

Table 23 — Sent service confirmation usage by sADT port

Argument	sADT port implementation
Session ID	The value of the **Session ID** argument of the **Send** service request
Connection ID	The connection layer identifier for the connection.

6.3.2 Data reception

Table 24 shows how the arguments to the **Receive** service request are used by the sADT port.

Table 24 — Receive service request usage by sADT port

Argument	sADT port implementation
Session ID	The link layer identifier for the session with which the connection is associated.
Connection ID	The connection layer identifier for the connection.
Buffer	The buffer to contain received data.
Buffer Size	The maximum number of characters to be placed in the buffer.

Table 25 shows how the arguments to the **Received** service confirmation are set by the sADT port.

Table 25 — Received service confirmation usage by sADT port

Argument	sADT port implementation
Session ID	The link layer identifier for the session with which the connection is associated.
Connection ID	The connection layer identifier for the connection.
Buffer	The buffer containing the received data. The buffer shall be the same buffer specified in the previous successful invocation of the **Receive** service request for this connection.
Received Character Count	The number of characters placed in the buffer.

6.3.3 Performing a reset

An automation device link layer invokes the **Reset** service request to request a reset of the remote ADT port in a DT device. Table 26 shows how the argument to the **Reset** service request is used by the sADT port.

Table 26 — Reset service request usage by sADT port

Argument	sADT port implementation
DTD Identifier	The identifier for the DT device to be reset.

The link layer in a DT device shall treat the processing of the **Reset Received** service indication either:

a) as an ADT port logout (see 7.5.5); or
b) as a hard reset (see 4.7).

Table 27 shows how the argument to the **Reset Received** service indication is set by the sADT port.

Table 27 — Reset Received service indication usage by sADT port

Argument	sADT port implementation
Reset Connection Identifier	An identifier for the Reset connection that was asserted.

6.4 iADT port support of connection services

6.4.1 Connection establishment

When an iADT port link layer invokes the **Connect** service request, the iADT port shall perform an active **OPEN** call (see RFC 793) with the foreign socket specified by the iADT port number and the remote IP address associated with the session. The local iADT port may learn the IP address of the remote iADT port by service discovery using UDP (see 6.5).

The iADT port may support more than one session. The local iADT port shall have at most one session for any given pair of IP addresses.

Table 28 shows how the arguments to the **Connect** service request are used by the iADT port.

Table 28 — Connect service request usage by iADT port

Argument	iADT port implementation
Session ID	The link layer identifier for the session with which the connection will be associated.
Local IP Address	The IP address component of the local port argument of the **OPEN** call. The connection layer shall assign the port number component of the local port argument of the **OPEN** call.
Remote IP Address	The IP address component of the foreign socket argument of the **OPEN** call. The port number component of the foreign socket argument of the **OPEN** call shall be the port number based on the **Method** argument
Method	Shall be one of the arguments specified in table 15.

Table 29 shows how the arguments to the **Connected** service confirmation are set by the iADT port.

Table 29 — Connected service confirmation usage by iADT port

Argument	iADT port implementation
Session ID	The value of the Session ID argument of the **Connect** service request.
Connection ID	The connection layer identifier for the connection.
Remote IP Address	IP address of the remote iADT port.

Table 30 shows how the arguments to the **Connect Received** service indication are set by the iADT port.

Table 30 — Connect Received service indication usage by iADT port

Argument	iADT port implementation
Connection ID	The connection layer identifier for the connection.
Remote IP Address	IP address of the remote iADT port.

6.4.2 Data transmission

When the **Send** service request is invoked, the iADT port shall invoke the **SEND** call (see RFC 793) with an indication that the data is to be transmitted immediately. Table 31 shows how the arguments to the **Send** service request are used by the iADT port.

Table 31 — Send service request usage by iADT port

Argument	iADT port implementation
Session ID	The link layer identifier for the session with which the connection is associated.
Connection ID	The connection layer identifier for the connection.
Buffer	**buffer address** argument of the **SEND** call.
Buffer Size	**byte count** argument of the **SEND** call.

When the **Send** service request is successfully processed, the iADT port connection layer shall invoke the **Sent** service confirmation. Invocation of the **Sent** service confirmation by the iADT port does not indicate that the characters have been transmitted by the physical port.

Table 32 shows how the arguments to the **Sent** service confirmation is used by the iADT port.

Table 32 — Sent service confirmation usage by iADT port

Argument	iADT port implementation
Session ID	The value of the Session ID argument of the **Send** service request.
Connection ID	The connection layer identifier for the connection

6.4.3 Data reception

Table 33 shows how the arguments to the **Receive** service request are used by the iADT port.

Table 33 — Receive service request usage by iADT port

Argument	iADT port implementation
Session ID	The link layer identifier for the session with which the connection is associated.
Connection ID	The connection layer identifier for the connection.
Buffer	**buffer address** argument of the **RECEIVE** call (see RFC 793).
Buffer Size	**byte count** argument of the **RECEIVE** call.

Table 34 shows how the arguments to the Received service confirmation are used by the iADT port.

Table 34 — Received service confirmation usage by iADT port

Argument	iADT port implementation
Session ID	The link layer identifier for the session with which the connection is associated.
Connection ID	The connection layer identifier for the connection.
Buffer	**buffer address** argument of the **RECEIVE** call.
Received Character Count	The number of characters placed in the buffer.

6.4.4 Closing a connection

When the link layer in an iADT port successfully invokes the **Disconnect** service request, then the iADT port shall invoke the **CLOSE** call (see RFC 793).

NOTE 10 TCP guarantees that characters previously transferred with the **SEND** call shall be sent prior to closing the connection.

Table 35 shows how the arguments to the **Disconnect** service request is used by the iADT port.

Table 35 — Disconnect service request usage by iADT port

Argument	iADT port implementation
Session ID	The link layer identifier for the session with which the connection is associated.
Connection ID	The connection layer identifier for the connection.

Table 36 shows how the argument to the **Disconnect Immediate** service request is used by the iADT port.

Table 36 — Disconnect Immediate service request usage by iADT port

Argument	iADT port implementation
Connection ID	The connection layer identifier for the connection.

When an iADT port that had invoked the **Disconnect** service request enters the **TCP CLOSED** state (see RFC 793), the connection layer shall invoke the **Disconnected** service confirmation. Table 37 shows how the argument to the **Disconnected** service confirmation is set by the iADT port. When an iADT port has invoked the **Connect** service request and the connection attempt fails, the connection layer shall invoke the **Disconnected** service confirmation.

Table 37 — Disconnected service confirmation usage by iADT port

Argument	iADT port implementation
Session ID	The link layer identifier for the session with which the connection was associated.
Reason	One of the Reasons listed in table 20.

When an iADT port that had not invoked the **Disconnect** service request enters the TCP CLOSED state (see RFC 793), the connection layer shall invoke the **Disconnect Received** service indication. Table 38 shows how the argument to the **Disconnect Received** service indication is set by the iADT port.

Table 38 — Disconnect Received service indication usage by iADT port

Argument	iADT port implementation
Connection ID	The connection layer identifier for the connection.
Reason	One of the Reasons listed in table 21.

6.4.5 Performing a reset

An automation device link layer shall invoke the **Reset** service request to request the reset of the remote ADT port in a DT device. Table 39 shows how the argument to the **Reset** service request is used by the iADT port.

Table 39 — Reset service request usage by iADT port

Argument	iADT implementation
DTD Identifier	The identifier for the DT device to be reset.

The link layer in a DT device shall treat the reception of the **Reset Received** service indication either:

a) as a **Disconnect Received** service indication (see 6.2.11); or
b) as a hard reset (see 4.7).

Table 40 shows how the argument to the **Reset Received** service indication is set by the iADT port.

Table 40 — Reset Received service indication usage by iADT port

Argument	iADT implementation
Reset Connection Identifier	An identifier for the Reset connection that was asserted.

6.5 Service discovery for iADT ports using the iADT Discovery protocol

6.5.1 Service discovery introduction

Service discovery using the iADT Discovery (iADT-DISC) protocol is optional and provides for the automated discovery of IP addresses used by remote iADT ports.

6.5.2 Service discovery message

When the physical port inside an iADT port supporting the iADT-DISC protocol has been enabled the iADT port shall:

a) begin the service discovery transmission processing (see 6.5.3);
b) continually receive packets on the iADT-DISC port (4169/udp); and
c) process service discovery messages received on the iADT-DISC port (see 6.5.4).

The data octets of the user datagram shall contain the service discovery message defined in table 41.

Table 41 — Service discovery message

Byte \ Bit	7	6	5	4	3	2	1	0
0 to 3	SIGNATURE ("iADT")							
4	INFORMATION TYPE							
5	DEVICE TYPE							
6	(MSB) ADDITIONAL LENGTH							
7								(LSB)
8	ADT MAJOR REVISION				ADT MINOR REVISION			
9	UNSEC	TLS	Reserved					
10	Reserved							
11	Reserved							
	Service discovery descriptor(s)							
12	Service discovery descriptor [first] (see table 44)							
	. . .							
n	Service discovery descriptor [last]							

The SIGNATURE field shall be set to the ASCII string "iADT".

The INFORMATION TYPE field is defined in table 42.

Table 42 — INFORMATION TYPE field

Value	Description
00h	Announcement
01h	Response
02h - EFh	Reserved
F0h - FFh	Vendor specific

The DEVICE TYPE field is defined in Table 43.

Table 43 — DEVICE TYPE field

Value	Description
00h	DT device
01h	Automation device
02h	Monitoring application
03h - FFh	Reserved

The ADDITIONAL LENGTH field is the length of the data that follows. The value in the ADDITIONAL LENGTH field shall be a multiple of four and shall not be greater than 500.

The ADT MAJOR REVISION field shall be set to the value that is used in the Port Login Information Unit (see 7.5.4) when initiating a Port Login exchange.

The ADT MINOR REVISION field shall be set to the value that is used in the Port Login Information Unit when initiating a Port Login exchange .

The UNSEC bit set to one indicates that the iADT port accepts iADT connections on port 4169/tcp.

The TLS bit set to one indicates that the iADT port accepts iADT-TLS connections on port 9614/tcp.

The service discovery descriptor is defined in table 44 and shall be a multiple of four bytes in length.

Table 44 — Service discovery descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE							
1	ADDITIONAL LENGTH (n-1)							
2	Reserved							
3	DESCRIPTOR LENGTH (m-3)							
4 … m	DESCRIPTOR							
m+1 … n	Pad bytes (optional)							

The DESCRIPTOR TYPE field (see table 45) identifies the service discovery descriptor.

The ADDITIONAL LENGTH field indicates the length of the data that follows.

The DESCRIPTOR LENGTH field indicates the length of the DESCRIPTOR field.

See table 45 for the definition of the DESCRIPTOR field.

Table 45 — Descriptor types (part 1 of 3)

DESCRIPTOR TYPE	Description	Support: Device type	Support: Requirements	DESCRIPTOR field
00h	Reserved			
01h	T10 vendor ID	DT device	O	The same data as the T10 VENDOR IDENTIFICATION field of the standard INQUIRY data (see SPC-4) reported by the RMC device server (see ADC-4) in the DT device.
		Automation device	O	The same data as the T10 VENDOR IDENTIFICATION field of the standard INQUIRY data reported by the SMC device server (see SMC-3) in the automation device.
		Monitoring application	I	
Support Key:	**M = mandatory** **O = optional** **I = invalid**			

Table 45 — Descriptor types (part 2 of 3)

DESCRIPTOR TYPE	Description	Support: Device type	Support: Requirements	DESCRIPTOR field
02h	T10 product ID	DT device	O	The same data as the T10 PRODUCT IDENTIFICATION field of the standard INQUIRY data (see SPC-4) reported by the RMC device server (see ADC-4) in the DT device.
		Automation device	O	The same data as the T10 PRODUCT IDENTIFICATION field of the standard INQUIRY data reported by the SMC device server (see SMC-3) in the automation device.
		Monitoring application	I	
03h	T10 product revision level	DT device	O	The same data as the T10 PRODUCT REVISION LEVEL field of the standard INQUIRY data (see SPC-4) reported by the RMC device server (see ADC-4) in the DT device.
		Automation device	O	The same data as the T10 PRODUCT REVISION LEVEL field of the standard INQUIRY data reported by the SMC device server (see SMC-3) in the automation device.
		Monitoring application	I	
04h	T10 serial number	DT device	O	The same data as the T10 PRODUCT SERIAL NUMBER field of the Manufacturer-assigned Serial Number VPD page reported by the ADC device server (see ADC-4) in the DT device.
		Automation device	O	The same data as the T10 PRODUCT SERIAL NUMBER field of the Unit Serial Number VPD page (see SPC-4) reported by the SMC device server (see SMC-3) in the automation device.
		Monitoring application	I	
05 - 3Fh	Reserved		O	
Support Key:	**M = mandatory** **O = optional** **I = invalid**			

Table 45 — Descriptor types (part 3 of 3)

DESCRIPTOR TYPE	Description	Support		DESCRIPTOR field
		Device type	Requirements	
40h	Monitoring application vendor ID	Monitoring application	O	Left-aligned ASCII data containing a vendor-specific vendor identifier. If the vendor has an assigned T10 vendor ID, that value should be used.
		DT or automation device	I	
41h	Monitoring application product ID	Monitoring application	O	Left-aligned ASCII data containing a vendor-specific product identifier.
		DT or automation device	I	
42h - EFh	Reserved			
F0h - FFh	Vendor specific		O	
Support Key:	**M = mandatory** **O = optional** **I = invalid**			

The pad bytes shall be used to pad the descriptor to end on a four byte boundary. Each pad byte shall be set to 00h.

6.5.3 Service discovery transmission processing

If the INFORMATION TYPE field in the Service discovery message is set to Announcement, then the Service Discovery Message shall be sent to the iADT-DISC port on the iADT-DISC multicast address (i.e., 224.0.23.63 for IPv4 or FF02::6F for IPv6).

The iADT port shall delay a random time in the range zero to three seconds and then transmit the Service discovery message (see table 41) once every three seconds until it receives a Service discovery message addressed to this iADT port or until a vendor specific number of messages have been sent. In the service discovery message the INFORMATION TYPE field shall be set to Announcement (i.e., 00h).

The purpose of the random delay is to avoid collisions between UDP packets from multiple devices.

6.5.4 Service discovery message reception processing

When processing a service discovery message the iADT port shall check the fields to determine how to process the message.

The iADT port shall save the address information for the device that transmitted the announcement message, delay a random time in the range zero to three seconds, and then transmit a service discovery message with the INFORMATION TYPE field set to Response (i.e., 01h) to the IP Address and port number that sent the announcement message if the INFORMATION TYPE field is set to Announcement (i.e., 00h) and:

a) this device is a DT device and the DEVICE TYPE field is not set to DT device (i.e., 00h); or
b) this device is an Automation device and
    A) the DEVICE TYPE field is set to Monitoring Application (i.e., 02h); or
    B) the DEVICE TYPE field is set to DT device (i.e., 00h) and:

a) it is known that this DT device is inside the automation device; or
b) it is unknown whether this DT device is inside or outside the automation device.

The iADT port should save the address information for the other device and may stop the transmission processing if in the received service discovery message:

a) the INFORMATION TYPE field is set to Response (i.e., 01h);
b) the DEVICE TYPE field is not set to Monitoring Application (i.e., 02h); and
c) the DEVICE TYPE field is not the same as this device.

NOTE 11 RFC 1112 and RFC 3170 provide general guidance for implementing multicasting. Various sockets API's provide specific guidance for implementing multicasting.

7 Link layer

7.1 Basic frame format

The general layout of an ADT frame is shown in figure 22. 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 22 — Basic ADT frame format

7.2 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 46 — 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.

7.3 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 47 defines the ADT frame header.

Table 47 — 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 48 defines the values for the PROTOCOL field. If an ADT 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 51).

Table 48 — PROTOCOL field values

Value	Description	Reference
0	Link service	7.5
1	Encapsulated SCSI	8.1
2	Fast access	8.2
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 48 for a description of the values in this field.

The X_ORIGIN bit shall be set to zero if the device originating the exchange is acting as an automation device within that session. The X_ORIGIN bit shall be set to one if the device originating the exchange is acting as a DT device within that session. This bit shall remain constant for all frames associated with a given exchange. If the X_ORIGIN bit is not set to the correct value, then the port shall send a NAK IU with status code of INVALID EXCHANGE ID (see table 51).

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 ADT 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. An ADT port shall check for re-use of exchange ID values for exchanges initiated by the other ADT port. If re-use is detected, the ADT port shall send a NAK IU with status code of INVALID EXCHANGE ID

The FRAME NUMBER field is assigned by the transmitting ADT 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, NOP IU, or Device Reset IU shall be set to zero. A transmitting ADT 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 an ADT port receives a frame where the payload length does not match the value in the PAYLOAD SIZE field, the ADT port shall return a NAK IU with the appropriate status code from table 51. If an ADT port receives a frame with a value in the PAYLOAD SIZE field that exceeds the maximum payload size, the ADT port shall return an NAK IU with a status code of MAXIMUM PAYLOAD SIZE EXCEEDED (see table 51).

A receiving ADT 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 51).

Except for a Port Login IU, Port Logout IU, Pause IU, NOP IU, or Device Reset IU, a receiving ADT 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 51).

7.4 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 7.2.

NOTE 12 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.

7.5 Link service information units

7.5.1 Link service frames overview

Either ADT port may initiate link service frames. Link service frames are used to manage the transport layer. Table 49 defines the values for the FRAME TYPE field in the ADT frame header for link service protocol frames.

Table 49 — Link service information units

Frame Type	Automation device support	DT device support	Description
0h	Mandatory	Mandatory	ACK (acknowledge)
1h	Mandatory	Mandatory	NAK (negative acknowledge)
2h	Mandatory	Mandatory	Port login
3h	Mandatory	Mandatory	Port logout
4h	Not allowed	Mandatory	Pause
5h	Mandatory	Mandatory	NOP (no operation)
6h	Mandatory	Mandatory	Initiate recovery
7h	Mandatory	Mandatory	Initiate recovery ACK (acknowledgement)
8h	Mandatory	Mandatory	Initiate recovery NAK (negative acknowledgement)
9h	Not allowed	Mandatory	Device Reset
Ah	Mandatory	Mandatory	Time-out
Bh - Fh			Reserved

7.5.2 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 defined size of the link service information unit for those link service information units that have a defined size. If the size of the payload exceeds the defined size of the link service information unit, then 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 defined size of the link service information unit, then the frame receiver shall not change the current setting(s) of the parameter(s) controlled by any missing field(s) or incomplete field(s). All other fields in frame shall be treated normally.

7.5.3 Acknowledgement information units

7.5.3.1 Acknowledgement information units introduction

An acknowledgement IU is an ACK IU, NAK IU, Initiate Recovery ACK IU, or Initiate Recovery NAK IU. An ADT 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.

7.5.3.2 ACK information unit

An ACK IU shall be sent by an ADT port that has received a frame without error. Except for acknowledgement IUs, an ADT port shall send an ACK IU for every frame that it receives without error. A sending ADT port shall set the ACK IU payload to zero bytes.

7.5.3.3 NAK information unit

A NAK IU is sent by the transport layer to indicate that the ADT port has detected an error during the reception of a frame. Except for acknowledgement IUs and corrupted frames (see 4.6.1.3), an ADT port shall send a NAK IU for every frame that it receives with a detected error. 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 50.

Table 50 — 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 ADT port is in RE1: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 51.

Table 51 — NAK frame STATUS CODE value (part 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 ADT port supports.
06h	INVALID EXCHANGE ID
07h - 2Fh	Reserved
30h - 3Fh	Vendor specific
40h	UNSUPPORTED PROTOCOL
41h	OUT OF RESOURCES (e.g., the receiving ADT port has run out of buffers to store the frame)
[a] This error is recoverable (see 4.6.2.3) and the PR bit may be set to one	

Table 51 — NAK frame STATUS CODE value (part 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) and the PR bit may be set to one

7.5.3.4 Interleaving acknowledgement and other frame types

An ADT 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. An ADT port that receives a Port Login IU or Port Logout IU may terminate transmission of a frame in progress.

An ADT port shall acknowledge all frames that it has received before starting transmission of any other frame type, except Port Logout IUs, Pause IUs, or a Port Login IU that is initiating a port login exchange.

7.5.4 Port login information unit

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

An ADT 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 51).

Table 52 defines the payload of the Port Login IU.

Table 52 — Port Login IU payload contents

Byte \ Bit	7	6	5	4	3	2	1	0
0	ACCEPT	Reserved					Vendor Specific	
1	MAJOR REVISION			MINOR REVISION				
2	Reserved							
3	AOE	Reserved					MAXIMUM ACK OFFSET	
4	(MSB)	MAXIMUM PAYLOAD SIZE						
5								(LSB)
6	(MSB)	BAUD RATE						
7								(LSB)

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 an ADT 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 an ADT port matches the parameters of the last Port Login IU it received, the ACCEPT bit shall be set to one. If an ADT port receives a Port Login IU with the ACCEPT bit set to one in a nonexistent exchange and the link negotiation state machine is not in the N0:Idle state, then it shall transmit a NAK IU with a status code of INVALID EXCHANGE ID (see table 51) and discard the Port Login IU.

An ADT port that receives a Port Login IU with the ACCEPT bit set to one shall respond with a NAK IU with a status code of NEGOTIATION ERROR (see table 51) if:

a) the link negotiation state machine is in the N0:Idle state (see 4.3.3.4); or
b) the parameter values, other than the ACCEPT bit, of the Port Login IU received differ from the parameter values of the most recent Port Login IU sent by the ADT port.

An ADT port that receives a Port Login IU with the ACCEPT bit set to zero shall respond with a NAK IU with a status code of NEGOTIATION ERROR (see table 51) if:

a) the link negotiation state machine is in N2:Accept Sent state (see 4.3.3.6); or
b) the Port Login IU parameters received by an ADT port match the parameters of the last Port Login IU it sent.

Ports claiming compliance with a draft revision of this standard shall set the MAJOR REVISION field to 001b and the MINOR REVISION field to the revision number of the supported draft. Ports claiming compliance with the INCITS approved version of this standard shall set the MAJOR REVISION field to 010b and the MINOR REVISION field to 00000b.

When initiating a Port Login exchange: a port shall set the MAJOR REVISION field and the MINOR REVISION field to the values corresponding to the latest supported revision of this standard. During the negotiation process, an ADT 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 ADT port supports the major revision level in the received Port Login IU, but not the minor revision level in the IU, then 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, then it shall respond as if it does not support the major revision level; or

b) if the ADT port does not support the major revision in the Port Login IU it receives, then 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 an ADT port shall be validated according to the revision level specified by the MAJOR REVISION and MINOR REVISION fields. An ADT port that receives a Port Login IU with supported MAJOR REVISION and MINOR REVISION field values and a payload that does not validate with the indicated revision shall send a NAK IU with a status code of NEGOTIATION ERROR (see table 51). Once the Port Login process has completed, both ADT 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 an ADT port under the following conditions:

a) the ADT port has experienced a hard reset condition;
b) the ADT port has experienced an error condition that may have led to loss of data or state on one or more exchanges;
c) the ADT port has received a Port Login IU with the AOE bit set to one; or
d) the ADT port terminated the previous session due to an implicit logout (see 4.3.2.4.4).

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

An ADT 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 (i.e., the port has transitioned into the P2:Logged-In state).

The MAXIMUM ACK OFFSET field indicates the number of frames that may be sent to the ADT port without receiving an acknowledgement IU in response (see 4.4). A value of zero indicates the ADT 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 ADT 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 8.1.4). If an ADT 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 51).

The BAUD RATE field indicates the speed that the physical interface in an sADT port shall run after completion of negotiation. The BAUD RATE field contains the desired nominal Baud rate divided by 100. All sADT 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 an sADT 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 51).

In an iADT port, the BAUD RATE field shall contain a value of 0000h.

If an ADT port receives a Port Login IU containing requested parameters (see 3.1.42) that are not an element of the supported parameters (see 3.1.55), then the ADT port shall set the value of the link parameters for the next Port Login IU it sends in the current exchange such that:

a) the value for each link parameter in the supported parameter set (see 3.1.54) is less than or equal to the value of that parameter in the requested parameters; and
b) the value of at least one link parameter in the supported parameter set is strictly less than the value of the parameter in the requested parameters.

7.5.5 Port logout information unit

After sending a Port Logout IU and before receiving the corresponding acknowledgement IU, an ADT 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;
d) set ADT port operating parameters to default following transmission of the ACK IU for the Port Logout IU (see 4.2); and
e) set the logout duration time to the value in the LOGOUT DURATION field. If the value in the field is zero, then set the logout duration time to infinite.

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

a) abort all open exchanges;
b) disable initiating Port Login exchanges;
c) set ADT port operating parameters to default following transmission of the ACK IU for the Port Logout IU (see 4.2); and
d) set the logout duration time to the value in the LOGOUT DURATION field. If the value in the field is zero, then set the logout duration time to infinite.

If a DT device port sends a Port Logout IU to an automation device port, then it may send a Port Login IU to the automation device port within the logout duration time.

Knowledge of the logged out state may be volatile (e.g., a hard reset condition in an ADT port may cause the port to become active again and initiate a login process wit the attached ADT port prior to the logout duration completing).

The payload of the Port Logout IU is shown in table 53.

Table 53 — Port Logout IU payload contents

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)	LOGOUT DURATION						
1								(LSB)
2	ESR	REASON CODE						
3	Reserved							

The LOGOUT DURATION field contains the value to which the ADT port that receives the Port Logout IU shall set the logout duration time (i.e., the length in seconds that the port shall remain in P3:Logged-out state). A value of zero indicates that the ADT port that receives the Port Logout IU shall set the logout duration time to infinite (i.e., the port shall remain in the P3:Logged-out state until it receives a Port Login IU).

The external stimulus required (ESR) bit set to one indicates that the ADT port requires an external stimulus before initiating a negotiation exchange (see 7.5.13.1).

The REASON CODE field contains the reason the ADT port logged out. Table 54 defines the values for the REASON CODE field values.

Table 54 — REASON CODE field

Code	Description
00h	Reason not stated
01h	Microcode update in progress
02h	Fatal microcode error (e.g., bugcheck or assert)
03h	Microcode reboot in progress
04h	Going offline for diagnostics
05h	Out of resources
06h	Error recovery in progress
07h - 5Fh	Reserved
60h - 7Fh	Vendor Specific

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

7.5.6 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, then it shall acknowledge the frame and, if the acknowledgement is an ACK IU, 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 ADT port back into T1:Active state. The T2:Paused state is volatile, a power cycle or other hard reset condition in the paused device may cause the ADT 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 51).

The Pause IU shall contain zero bytes of payload.

7.5.7 NOP information unit

A NOP IU may be sent by an ADT 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). An ADT 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.

7.5.8 Initiate Recovery information unit

An Initiate Recovery IU shall be sent by an ADT 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.

7.5.9 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.

7.5.10 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.

7.5.11 Device Reset information unit

A Device Reset IU may be sent by an automation device port to cause the DT device to perform a hard reset. DT device ports shall not initiate a Device Reset IU exchange. If a DT device port receives a Device Reset IU, it shall send an acknowledge IU. If the acknowledgement is an ACK IU, then the DT Device port shall generate a Hard Reset Event (see 4.7) for the ADT port state machine.

Following reception of an ACK IU for a Device Reset IU an automation device port shall:

a) abort all open exchanges; and
b) set the operating parameters to default (see 4.2).

An automation device port that receives a Device Reset IU shall respond with a NAK IU with a status code of INVALID OR ILLEGAL IU RECEIVED (see table 51).

The Device Reset IU shall contain zero bytes of payload.

7.5.12 Time-out information unit

The Time-out IU provides a mechanism for the ADT ports forming a link to negotiate changes to one another's acknowledgement IU time-out period (see 4.10.2). Table 55 defines the payload of the Time-out IU.

Table 55 — Time-out IU payload contents

Bit Byte	7	6	5	4	3	2	1	0
0	ACTION CODE		Reserved				Vendor Specific	
1 3	Reserved							
4 5	(MSB) CURRENT ACKNOWLEDGEMENT TIME-OUT (LSB)							
6 7	(MSB) MAXIMUM ACKNOWLEDGEMENT TIME-OUT (LSB)							
8 9	(MSB) MINIMUM ACKNOWLEDGEMENT TIME-OUT (LSB)							
10 11	(MSB) TIME-OUT RESOLUTION (LSB)							

The ACTION CODE field specifies the meaning of the CURRENT ACKNOWLEDGEMENT TIME-OUT field, MAXIMUM ACKNOWLEDGEMENT TIME-OUT field, MINIMUM ACKNOWLEDGEMENT TIME-OUT field, and TIME-OUT RESOLUTION field. Table 56 defines the CURRENT ACKNOWLEDGEMENT TIME-OUT field, MAXIMUM ACKNOWLEDGEMENT TIME-OUT field, MINIMUM ACKNOWLEDGEMENT TIME-OUT field, and TIME-OUT RESOLUTION field for the various ACTION CODE field values.

Table 56 — Time-out IU Action Codes

Code	Code name	Received IU field	Description
00b	REPORT	current acknowledgement time-out	Acknowledgement IU time-out period, in milliseconds, currently in effect in the sending ADT port.
		maximum acknowledgement time-out	Maximum acknowledgement IU time-out period, in milliseconds, supported by the sending ADT port.
		minimum acknowledgement time-out	Minimum acknowledgement IU time-out period, in milliseconds, supported by the sending ADT port.
		time-out resolution	Minimum difference between acknowledgement IU time-out periods, in milliseconds, supported by the sending ADT port.
01b	DISCOVER	current acknowledgement time-out	Ignored
		maximum acknowledgement time-out	Ignored
		minimum acknowledgement time-out	Ignored
		time-out resolution	Ignored
10b	REQUEST CHANGE	current acknowledgement time-out	Requested value of the receiving ADT port's acknowledgement IU time-out, in milliseconds.
		maximum acknowledgement time-out	Ignored
		minimum acknowledgement time-out	Ignored
		time-out resolution	Ignored
11b	Reserved		

An ADT port that processes a Time-out IU with the ACTION CODE field set to REQUEST CHANGE and a supported value in the CURRENT ACKNOWLEDGEMENT TIME-OUT field shall set its acknowledgement IU time-out to the value contained in the CURRENT ACKNOWLEDGEMENT TIME-OUT field. An ADT port that processes a Time-out IU with the ACTION CODE field set to REQUEST CHANGE and an unsupported value in the CURRENT ACKNOWLEDGEMENT TIME-OUT field shall:

a) set its acknowledgement IU time-out period to the closest supported value that is greater than the value contained in the CURRENT ACKNOWLEDGEMENT TIME-OUT field and send an ACK IU; or
b) not alter the acknowledgement IU time-out period and send a NAK IU with a status code of INVALID OR ILLEGAL IU RECEIVED (see table 51).

An ADT port that processes a Time-out IU with the ACTION CODE field set to a value other than REQUEST CHANGE shall not alter its acknowledgement IU time-out period.

An ADT port that receives a Time-out IU with the ACTION CODE field set to 11b shall send a NAK IU with a status code of INVALID OR ILLEGAL IU RECEIVED (see table 51).

7.5.13 Link service exchange lifetime

7.5.13.1 Link service exchange types

Link service exchanges may be negotiation exchanges, port logout exchanges, pause exchanges, NOP exchanges, device reset exchanges, or time-out discovery exchanges.

7.5.13.2 Simple link service exchange lifetime

Port logout IUs, Pause IUs, NOP IUs, Device Reset IUs, and Time-out IUs with the ACTION CODE field set to REQUEST CHANGE are sent in simple exchanges. A simple exchange begins in the sending ADT 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 ADT 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.

7.5.13.3 Negotiation exchange lifetime

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

a) the ADT 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 ADT 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;
c) the ADT port has received a Login IU with a different exchange ID, indicating that negotiation has restarted; or
d) the ADT port sends a Login IU with a different exchange ID, indicating that negotiation has been restarted.

7.5.13.4 Time-out discovery exchange lifetime

In an ADT port initiating a time-out discovery exchange, the exchange begins when the ADT port transmits a Time-out IU with the ACTION CODE field set to DISCOVER in a nonexistent exchange. In an ADT port not initiating a time-out discovery exchange, the exchange begins when the ADT port receives a Time-out IU with the ACTION CODE field set to DISCOVER in a nonexistent exchange. A time-out discovery exchange ends in an ADT port when:

a) the ADT port has sent a Time-out IU with the ACTION CODE field set to REPORT in an existing exchange and received an ACK IU in response to it;
b) the ADT port has received a Time-out IU with the ACTION CODE field set to REPORT in an existing exchange and sent an ACK IU in response to it;
c) the ADT port has sent a Time-out IU and received a NAK IU in response to it;
d) the ADT port has received a Time-out IU and sent a NAK IU in response to it; or
e) the ADT port has received a Time-out IU with a different exchange ID, indicating that a new time-out exchange has started.

7.6 Sessions

A session is a relationship between two ADT ports. The link layer manages sessions and associates connections with sessions. A session between two sADT ports is identified by two physically-connected sADT ports. A session between two iADT ports is identified by a unique pair of IP addresses.

A session between a local sADT port and a remote sADT port begins in the local sADT port when no session exists between the two sADT ports and the local sADT port transmits or receives a Login IU.

A session between a local iADT port and a remote iADT port begins in the local iADT port when no session exists between the two iADT ports and:

a) the link layer invokes the **Connect** service request specifying the remote iADT port; or
b) the connection layer invokes the **Connect Received** service indication specifying the remote iADT port.

A session between two ADT ports ends in the local ADT port after the local ADT port:

a) processes a Logout IU (see 7.5.5) that was received on the local ADT port;
b) receives an ACK for a Logout IU the local ADT port sent; or
c) receives an implicit logout message (see 4.3.2.4.4).

In an iADT port, if:

1) a session exists;
2) a Login IU has been received; and
3) a new connection is established in the same session

then the previously-established session shall be terminated (i.e., perform an implicit logout and invoke the **Disconnect Immediate** service request) and a new session shall be created using the new connection.

In an iADT port, if:

1) a session exists;
2) a Login IU has not been received; and
3) a new connection is established in the same session

then the connection initiated by the DT device shall be closed (i.e., invoke the **Disconnect Immediate** service request) and the connection initiated by the automation device shall be used for this session.

Annex E contains examples of resolving redundant connections.

8 Transport layer

8.1 SCSI Encapsulation

8.1.1 SCSI encapsulation overview

SCSI information units contain information required to implement the SCSI protocol. Information units are exchanged between an automation device transport layer and a DT device transport layer (see figure 5). 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 command identifier from SAM-5. The LUN is included in the SCSI Command IU, SCSI Command - Expanded IU, and SCSI Task Management IU payload contents. See 4.9 for transport protocol variations from SAM-5. See clause 9 for the mapping of the IUs described in this clause to the SCSI transport protocol services.

Table 57 defines the values for the FRAME TYPE field in the ADT frame header for encapsulated SCSI protocol information units.

Table 57 — 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	SCSI Command - Expanded
6h - Fh	Reserved

8.1.2 SCSI Command information unit

The SCSI Command IU payload shall contain the information described in Table 58.

Table 58 — 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-5.

The TASK ATTRIBUTE field is defined in table 59.

Table 59 — 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-5).
1h	HEAD OF QUEUE	Requests that the task be managed according to the rules for a head of queue task attribute (see SAM-5).
2h	ORDERED	Requests that the task be managed according to the rules for an ordered task attribute (see SAM-5).
3h	Reserved	
4h	ACA	Requests that the task be managed according to the rules for an automatic contingent allegiance task attribute (see SAM-5).
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 ADT initiator port. A non-zero value in the FIRST DATA-IN BURST LENGTH field requests that the ADT target port transfer one or more SCSI Data 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 DATA 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 ADT initiator port and no data shall be sent by the ADT target port until it receives a SCSI Transfer Ready IU. An ADT initiator port may put a non-zero value in the FIRST DATA-IN BURST LENGTH field of any SCSI Command IU. The ADT target port shall ignore the value in the FIRST DATA-IN BURST LENGTH field for non-data or data-out commands.

NOTE 13 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, then 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-5.

8.1.3 SCSI Task Management information unit

Table 60 defines the SCSI Task Management IU. The SCSI Task Management IU is sent by an ADT initiator port to request that a task management function be processed by a task manager in a logical unit.

Table 60 — 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-5. 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 61 defines the values for the TASK MANAGEMENT FUNCTION field. See SAM-5 for a definition of the task management functions provided.

Table 61 — TASK MANAGEMENT FUNCTION values

Value	Description
00h	Reserved
01h	ABORT TASK
02h	ABORT TASK SET

Table 61 — TASK MANAGEMENT FUNCTION values

Value	Description
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	QUERY TASK SET
82h - FFh	Reserved

If TASK MANAGEMENT FUNCTION is set to ABORT TASK or QUERY TASK, then the TAG OF TASK TO BE MANAGED field specifies the tag value (see 8.1.1) from the SCSI Command IU or SCSI Command - Expanded 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-5 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, then 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, then it may return a NAK IU with a status code of INVALID EXCHANGE ID (see table 51).

8.1.4 SCSI Response information unit

A SCSI Response IU shall be returned to the exchange Initiator for every SCSI Command IU, SCSI Command - Expanded IU, and SCSI Task Management IU that is received. Table 62 defines the payload of a SCSI Response IU.

Table 62 — SCSI Response IU payload contents

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

The RESPONSE CODE field indicates the results of the operation as an extension to the SCSI Status. Table 63 defines the values for this field.

Table 63 — 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 (see SMC-3) 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. The SCSI STATUS field shall contain GOOD and the SCSI AUTOSENSE DATA field shall contain the sense data to be reported to the other initiator ports. Additionally, any data cached by the local SMC device server (see SMC-3) shall be invalidated (see ADC-4).

The SCSI STATUS field contains SCSI status as defined in SAM-5. 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 or SCSI Command - Expanded 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 CHEDCK CONDITION, or the response code is set to COMMAND COMPLETE WITH UNIT ATTENTION, then autosense data shall be included in the IU as defined in SPC-4 and the SENSE LENGTH field shall be set to indicate how much sense data is included.

8.1.5 SCSI Transfer Ready information unit

A SCSI Transfer Ready IU shall be sent by one ADT port to inform another ADT 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 64.

Table 64 — SCSI Transfer Ready IU payload contents

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

The DATA OFFSET field contains the offset from first data byte transferred in the exchange to the first byte of data in the requested transfer. Data shall not be requested out of order. The DATA OFFSET field shall be set to a value that is a multiple of four.

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 transmitted or until all available data has been transmitted.

If an ADT initiator port receives a SCSI Transfer Ready IU that is not 8 bytes long, then it shall send an ACK IU and discard the frame. The ADT initiator port shall use the **Command Complete Received** transport protocol service (see 9.2.4) to return a service response value of SERVICE DELIVERY OR TARGET FAILURE to the application client that generated the command. The application client should abort the command (see 9.4).

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

If an ADT initiator port receives a SCSI Transfer Ready IU requesting zero bytes, then it shall send an ACK IU and discard the frame. The ADT initiator port shall use the **Command Complete Received** transport protocol service (see 9.2.4) to return a service response value of SERVICE DELIVERY OR TARGET FAILURE to the application client that generated the command. The application client should abort the command (see 9.4).

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

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

8.1.6 SCSI Data information unit

The SCSI Data IU is used to send data associated with SCSI Data In and Data Out operations. Table 65 describes the contents of a SCSI Data IU.

Table 65 — SCSI Data IU payload contents

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

The DATA OFFSET field contains the offset from the first data byte in the exchange to the first byte in the DATA field. 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, then it shall send an ACK IU, discard that frame and any subsequent SCSI Data IUs received for that command, and terminate the command with CHECK CONDITION status, with a sense key set to ABORTED COMMAND, and the additional sense code set to 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), then it shall send an ACK IU, discard the frame, and terminate the command with CHECK CONDITION status, with the sense key set to ABORTED COMMAND, and the additional sense code set to TOO MUCH WRITE DATA.

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

If an ADT initiator port receives a SCSI Data IU with more read data than expected, then it shall send an ACK IU and discard the frame. The ADT initiator port shall use the **Command Complete Received** transport protocol service (see 9.2.4) to return a service response value of SERVICE DELIVERY OR TARGET FAILURE to the application client that generated the command. The application client should abort the command (see 9.4).

If an ADT initiator port receives a SCSI Data IU with zero bytes, then it shall send an ACK IU and discard the frame. The ADT initiator port shall use the **Command Complete Received** transport protocol service (see 9.2.4) to return a service response value of SERVICE DELIVERY OR TARGET FAILURE to the application client that generated the command. The application client should abort the command (see 9.4).

If an ADT initiator port receives a SCSI Data IU with a data offset that was not expected, then it shall send an ACK IU and discard that frame and any subsequent SCSI Data IUs received for that command. The ADT initiator port shall use the **Command Complete Received** transport protocol service (see 9.2.4) to return a service response value of SERVICE DELIVERY OR TARGET FAILURE to the application client that generated the command. The application client should abort the command (see 9.4)

8.1.7 SCSI Command - Expanded information unit

The SCSI Command - Expanded IU payload shall contain the information described in table 66.

Table 66 — SCSI Command - Expanded IU payload contents

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)	LUN						
1								(LSB)
2	Reserved				TASK ATTRIBUTE			
3	Reserved							
4	(MSB)	FIRST DATA-IN BURST LENGTH						
7								(LSB)
8	Reserved							
9								
10	(MSB)	CDB LENGTH (n-11)						
11								(LSB)
12	CDB							
n								

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-5.

The TASK ATTRIBUTE field is defined in table 67.

Table 67 — 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-5).
1h	HEAD OF QUEUE	Requests that the task be managed according to the rules for a head of queue task attribute (see SAM-5).
2h	ORDERED	Requests that the task be managed according to the rules for an ordered task attribute (see SAM-5).
3h	Reserved	
4h	ACA	Requests that the task be managed according to the rules for an automatic contingent allegiance task attribute (see SAM-5).
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 ADT initiator port. A non-zero value in the FIRST DATA-IN BURST LENGTH field requests that the ADT target port transfer one or more SCSI Data 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 DATA 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 ADT initiator port and no data shall be sent by the ADT target port until it receives a SCSI Transfer Ready IU. An ADT initiator port may put a non-zero value in the FIRST DATA-IN BURST LENGTH field of any SCSI Command IU. The ADT target port shall ignore the value in the FIRST DATA-IN BURST LENGTH field for non-data or data-out commands.

NOTE 14 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.

The CDB LENGTH field specifies the length of the CDB. If an ADT target port receives a SCSI Command - Expanded IU and the length of the payload of the frame does not match the value from the CDB LENGTH field plus 12, then 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-5.

8.1.8 SCSI encapsulation exchange lifetime

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

A SCSI encapsulation exchange ends in an ADT initiator port after:

a) the ADT 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 ADT 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 an ADT target port after:

a) the ADT 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 ADT 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 (see 4.8).

8.1.9 Reception of Encapsulated SCSI Information Units in exceptional circumstances

If an ADT port receives a SCSI Response IU, SCSI Transfer Ready IU, or SCSI Data IU in a nonexistent exchange, then it shall transmit a NAK IU with a status code of INVALID EXCHANGE ID (see table 51) and discard the SCSI IU.

Within a valid exchange, if an ADT port receives a SCSI Transfer Ready IU for which it has no data to send, then it shall transmit an ACK IU and ignore the SCSI Transfer Ready IU.

If an ADT port that does not support SCSI target functions receives a SCSI Command IU, SCSI Command - Expanded IU, or SCSI Task Management IU, then it shall transmit a NAK IU with the PR bit set to zero and a status code of UNSUPPORTED FRAME TYPE FOR SELECTED PROTOCOL (see table 14) and discard the IU.

NOTE 15 The exchange originator should not interpret this response as an indication that the ADT port does not have SCSI target port capabilities, only that it does not have SCSI target port capabilities enabled at this time (see 4.1).

8.2 Fast Access

8.2.1 Fast Access overview

This protocol is intended to provide a feature set beyond what is provided by SAM-5 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-4;
b) an asynchronous event report, a method for a DT device to report asynchronous activity; and
c) a method to control these asynchronous reports.

Fast Access protocol IUs shall be routed to the remote ADT port associated with the session.

Table 68 defines the values for the FRAME TYPE field in the ADT frame header for Fast Access protocol IUs.

Table 68 — Fast Access protocol IUs

Frame Type	Description
0h	Request for VHF Data
1h	VHF Data
2h	AER
3h	AER Control
4h	AER - Expanded
5h	AER - Expanded Control
6h - Fh	Reserved

If an ADT device receives an unsupported frame type, then it shall send a NAK IU in response to that frame with the PR bit set to zero and a status code of UNSUPPORTED FRAME TYPE FOR SELECTED PROTOCOL (see table 51) and discard the IU.

If an automation device port receives a VHF Data IU, an AER Control IU or an AER - Expanded Control IU in a nonexistent exchange, then it shall transmit a NAK IU with a status code of INVALID EXCHANGE ID (see table 51) and discard the IU.

8.2.2 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, then 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, then the IU receiver shall not change the current setting(s) of the parameter(s) controlled by any missing field(s).

8.2.3 Request for VHF Data information unit

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

8.2.4 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 descriptor as defined in ADC-4.

8.2.5 AER information unit

Asynchronous Event Report (AER) IUs may be used to report that an event has occurred. Only a DT device port is allowed to initiate AER IUs. The payload of an AER IU shall contain the VHF data descriptor as defined in ADC-4.

8.2.6 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 descriptor, with the bits set to one for each field that the device shall report a change. The values in VHF data descriptor shall be maintained separately for each I_T nexus. 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 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 for the I_T nexus 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 for the I_T nexus 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 ADT 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.

8.2.7 AER - Expanded information unit

Asynchronous Event Report (AER) – Expanded IUs may be used to report that an event has occurred. Only a DT device port is allowed to initiate AER – Expanded IUs. The AER – Expanded information unit payload shall contain the information described in table 69. Each AER – Expanded IU shall report a single log parameter. If there are multiple log parameters to be reported, then they shall be reported in separate IUs.

Table 69 — AER - Expanded IU payload contents

Bit Byte	7	6	5	4	3	2	1	0
0	DT DEVICE STATUS LOG PAGE PARAMETER CODE							
1								
2	DT DEVICE STATUS LOG PAGE PARAMETER VALUE							
n								

The DT DEVICE STATUS LOG PAGE PARAMETER CODE field shall contain a DT Device Status log page parameter code that is supported for reporting in AER – Expanded IUs (see ADC-4).

The DT DEVICE STATUS LOG PAGE PARAMETER VALUE field shall contain the parameter value from the DT Device Status log page parameter defined in ADC-4.

8.2.8 AER - Expanded Control information unit

An AER - Expanded Control IU may be sent by an automation device port to a DT device port to enable or disable AER - Expanded reporting. The payload of an AER - Expanded Control IU shall contain an AER - Expanded payload (see table 69), with the bits set to one for each field that the device shall report a change. Each AER – Expanded Control IU shall contain information for a single log parameter. To enable AER – Expanded IUs for multiple log parameters, each log parameter shall be configured in a separate AER – Expanded Control IU.

If the DT DEVICE STATUS LOG PAGE PARAMETER CODE field indicates a log parameter that is supported but does not exist at this time (e.g., the log parameter is a supported potential conflict list log parameter but has not yet been created, see ADC-4), then the DT device shall process the AER–Expanded Control IU as if the log parameter exists. If the log parameter is created, each field in that log parameter is considered to have changed. If that log parameter exists and is destroyed (see ADC-4) each field in that log parameter is considered to have changed. If a log parameter enabled for AER is destroyed, that log parameter is returned in an AER – Expanded IU which contains a DT DEVICE STATUS LOG PAGE PARAMETER CODE field set to the parameter code of the destroyed log parameter and no DT DEVICE STATUS LOG PAGE PARAMETER VALUE field (i.e., a 2-byte length response).

The values in log parameters reported in AER - Expanded IUs shall be maintained separately for each I_T nexus. Multiple-bit fields shall have either all of the bits in 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 support AER - Expanded shall respond to the receipt of an AER - Expanded Control IU by sending an AER - Expanded Control IU back to the automation device with the same X_ORIGIN and EXCHANGE ID values. The payload of the IU shall contain an AER - Expanded payload for the maximum length of the log parameter regardless of the payload size of the AER – Expanded Control information unit received. Each field that has been enabled for AER - Expanded notification for the I_T nexus 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 - Expanded notification for the I_T nexus or is not supported for AER - Expanded notification by the device shall be set to zero. The default setting for all AER - Expanded events in a DT device shall be zero.

All AER - Expanded 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 - Expanded Control IU in response to receiving an AER - Expanded Control IU from an automation device port.

8.2.9 Fast Access exchange lifetime

8.2.9.1 Fast Access exchange types

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

8.2.9.2 VHF Data exchange lifetime

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

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

8.2.9.3 AER Control exchange lifetime

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

An AER control exchange ends in an automation device port after the ADT 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 ADT 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.

8.2.9.4 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.

8.2.9.5 AER - Expanded Control exchange lifetime

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

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

8.2.9.6 AER - Expanded exchange lifetime

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

9 SCSI Application layer

9.1 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-5):

Service response = Execute Command (IN (I_T_L Nexus, Command Identifier, CDB, Task Attribute, [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [CRN], [Command Priority]), OUT ([Data-In Buffer], [Sense Data], [Sense Data Length], Status, [Status Qualifier]))

This standard defines the transport protocol services required by SAM-5 in support of this remote procedure call (see 9.2) and transport protocol services that are extensions of the services required by SAM-5 (see 4.9). Table 70 describes the mapping of the **Execute Command** procedure call to transport protocol services and the ADT-3 implementation of each transport protocol service.

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 70 — Execute Command procedure call transport protocol services

Transport protocol service	I/T[a]	ADT-3 implementation	Reference
Command and status			
Send SCSI Command request	I	SCSI Command IU or SCSI Command - Expanded IU	9.2.1
SCSI Command Received indication	T	Receipt of SCSI Command IU or SCSI Command - Expanded IU	9.2.2
Send Command Complete response	T	SCSI Response IU	9.2.3
Command Complete Received confirmation	I	Receipt of SCSI Response IU	9.2.4
Data-In delivery [b]			
Send Data-In request	T	SCSI Data IU	9.2.5
Data-In Delivered confirmation	T	Positive acknowledgement of the last SCSI Data IU	9.2.6
Receive Data-In request [c]	I	SCSI Transfer Ready IU	9.2.11
Data-In Received confirmation [c]	I	Positive acknowledgement of the last SCSI Data IU containing data to satisfy the SCSI Transfer Ready IU	9.2.12

[a] I/T indicates whether the initiator port (I) or the target port (T) implements the transport protocol service.
[b] Data transfer transport protocol services for SCSI initiator ports are not specified by SAM-5.
[c] Extensions of the services required by SAM-5. See 4.9 for details of the use of these transport protocol services.

Table 70 — Execute Command procedure call transport protocol services

Transport protocol service	I/T[a]	ADT-3 implementation	Reference
Data-Out delivery [b]			
Receive Data-Out request	T	SCSI Transfer Ready IU	9.2.7
Data-Out Received confirmation	T	Receipt of last SCSI Data IU	9.2.8
Send Data-Out request [c]	I	SCSI Data IU	9.2.9
Data-Out Delivered confirmation [c]	I	Positive acknowledgement of the last SCSI Data IU	9.2.10
Terminate Data Transfer [b]			
Terminate Data Transfer request	T		
Data Transfer Terminated confirmation	T		

[a] I/T indicates whether the initiator port (I) or the target port (T) implements the transport protocol service.
[b] Data transfer transport protocol services for SCSI initiator ports are not specified by SAM-5.
[c] Extensions of the services required by SAM-5. See 4.9 for details of the use of these transport protocol services.

Editors Note 1 - CCB: Editor to research how ADT should handle the Terminate Data Transfer transport protocol services to determine if ADT should handle these and if so how.

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-5):

a) Service Response = **ABORT TASK (IN (Nexus, Command Identifier))**;
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 (I_T_L Nexus, Command Identifier), OUT ([Additional Response Information]))**; and
g) Service Response = **QUERY TASK SET (IN (I_T_L Nexus))**.

This standard defines the transport protocol services required by SAM-5 in support of these procedure calls. Table 71 describes the mapping of these procedure calls to transport protocol services and the ADT-3 implementation of each transport protocol service

Table 71 — Task management function procedure call transport protocol services

Transport protocol service	I/T[a]	ADT-3 implementation	Reference
Task management			
Send Task Management request	I	SCSI Task Management IU	9.3.1
Task Management Request Received indication	T	Receipt of SCSI Task Management IU	9.3.2
Task Management Function Executed response	T	SCSI Response IU	9.3.3
Receive Task Management Function-Executed confirmation	I	Receipt of SCSI Response IU	9.3.4
[a] I/T indicates whether the initiator port (I) or the target port (T) implements the transport protocol service.			

9.2 Transport layer protocol services to support Execute Command

9.2.1 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 or SCSI Command - Expanded IU containing a SCSI command.

Send SCSI Command (IN (I_T_L Nexus, Command Identifier, CDB, Task Attribute, [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [CRN], [Command Priority], [First Burst Enabled]))

Table 72 shows how the arguments to the **Send SCSI Command** transport protocol service are used in the generation of a SCSI Command IU or SCSI Command - Expanded IU.

Table 72 — Send SCSI Command transport layer protocol service arguments (part 1 of 2)

Argument	ADT Implementation
I_T_L nexus	I_T_L nexus, where: a) I_T is used to select the session and to set the X_ORIGIN bit in the ADT frame header; and; b) L is used to set the LUN field in the command IU (i.e., the SCSI Command IU or the SCSI Command - Expanded IU).
Command Identifier	Used to set the EXCHANGE ID in the ADT frame header.
CDB	Used to set the CDB field in the command IU.
Task Attribute	Used to set the TASK ATTRIBUTE field in the command IU.
[Data-In Buffer Size]	Used to set the FIRST DATA-IN BURST LENGTH field in the command IU.

Table 72 — Send SCSI Command transport layer protocol service arguments (part 2 of 2)

Argument	ADT Implementation
[Data-Out Buffer]	Buffer of data to send.
[Data-Out Buffer Size]	Maximum of 2^{32}-1
[CRN]	Not used
[Command Priority]	Not used
[First Burst Enabled]	Not used

9.2.2 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 or SCSI Command - Expanded IU.

SCSI Command Received (IN (I_T_L Nexus, Command Identifier, CDB, Task Attribute, [CRN], [Command Priority], [First Burst Enabled]))

Table 73 shows how the arguments to the **SCSI Command Received** transport protocol service are used.

Table 73 — SCSI Command Received transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	I_T_L nexus, where: a) I_T is indicated by the session and the X_ORIGIN bit in the ADT frame header; and b) L is indicated by the LUN field in the command IU (i.e., the SCSI Command IU or the SCSI Command - Expanded IU).
Command Identifier	From the EXCHANGE ID field in the ADT frame header.
CDB	From the CDB field in the command IU (i.e., the SCSI Command IU or the SCSI Command - Expanded IU).
Task Attribute	From the TASK ATTRIBUTE field in the command IU.
[CRN]	Not used
[Command Priority]	Not used
[First Burst Enabled]	Not used

9.2.3 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 Nexus, Command Identifier, [Sense Data], [Sense Data Length], Status, Service Response, [Status Qualifier]))

Table 74 shows how the arguments to the **Send Command Complete** transport protocol service are used in the generation of a SCSI Response IU.

Table 74 — Send Command Complete transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	From the **SCSI Command Received** transport protocol service call that established the command.
Command Identifier	From the **SCSI Command Received** transport protocol service call that established the command.
[Sense Data]	Used to set the SCSI AUTOSENSE DATA field in the SCSI Response IU.
[Sense Data Length]	Used to set the SENSE LENGTH field in the SCSI Response IU.
Status	Used to set the SCSI STATUS field in the SCSI Response IU.
Service Response	Used to set the RESPONSE CODE and SCSI STATUS fields in the SCSI Response 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 SCSI STATUS field is set to a Status Code value (see SAM-5); or b) SERVICE DELIVERY OR TARGET FAILURE: The RESPONSE CODE field is set to SERVICE DELIVERY FAILURE.
Status Qualifier	Not used

9.2.4 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 or SCSI Command - Expanded 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 Nexus, Command Identifier, [Data-In Buffer], [Sense Data], [Sense Data Length], Status, Service Response, [Status Qualifier]))

Table 75 shows how the arguments to the **Command Complete Received** transport protocol service are determined.

Table 75 — Command Complete Received transport layer protocol service arguments (part 1 of 2)

Argument	ADT Implementation
I_T_L nexus	I_T_L nexus, where: a) I_T is indicated by the session and the X_ORIGIN bit in the ADT frame header; and; b) L is indicated by the LUN field in the command IU (i.e., in the SCSI Command IU or the SCSI Command - Expanded IU).

Table 75 — Command Complete Received transport layer protocol service arguments (part 2 of 2)

Argument	ADT Implementation
Command Identifier	From 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.
[Sense Data Length]	From the SCSI Response IU SENSE LENGTH field.
Status	From the SCSI Response IU SCSI STATUS field.
Service Response	From the SCSI Response IU RESPONSE CODE and SCSI STATUS field, or from a NAK on the SCSI Command IU or SCSI Command - Expanded 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 SCSI STATUS field is set to a Status Code value (see SAM-5); or b) SERVICE DELIVERY OR TARGET FAILURE: RESPONSE CODE field is set to Service delivery failure.
Status Qualifier	Not used

9.2.5 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 ADT initiator port using one or more SCSI Data IUs. An ADT 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 8.1.5); or
b) the SCSI Command IU or SCSI Command - Expanded IU contained a non-zero value in the FIRST DATA-IN BURST LENGTH field (see 8.1.2).

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

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

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

Table 76 shows how the arguments to the **Send Data-In** transport protocol service are used.

Table 76 — Send Data-In transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	From the **SCSI Command Received** transport protocol service call that established the command.
Command Identifier	From the **SCSI Command Received** transport protocol service call that established the command.
Device Server Buffer	Pointer to a buffer where the data is located.
Application Client Buffer Offset	Used to set the DATA 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 DATA 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.

9.2.6 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 Nexus, Command Identifier, Delivery Result))

Table 77 shows how the arguments to the **Data-In Delivered** transport protocol service are determined.

Table 77 — Data-In Delivered transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	I_T_L nexus value passed to the **Send Data-In** transport layer protocol service request that initiated the transfer.
Command Identifier	Command Identifier value passed to the **Send Data-In** transport layer protocol service request that initiated the transfer.
Delivery Result	Result of the data delivery operation.

9.2.7 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 Nexus, Command Identifier, Application Client Buffer Offset, Request Byte Count, Device Server Buffer))

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

A device server shall not call **Receive Data-Out** for a given command until **Data-Out Received** has completed without error for the previous **Receive Data-Out** call for that command (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 command).

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

Table 78 shows how the arguments to the **Receive Data-Out** transport protocol service are used.

Table 78 — Receive Data-Out transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	From the **SCSI Command Received** transport protocol service call that established the command.
Command Identifier	From the **SCSI Command Received** transport protocol service call that established the command.
Application Client Buffer Offset	Used to set the DATA 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.

9.2.8 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 Nexus, Command Identifier, Delivery Result))

Table 79 shows how the arguments to the **Data-Out Received** transport protocol service are determined.

Table 79 — Data-Out Received transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	I_T_L nexus value passed to the **Receive Data-Out** transport layer protocol service request that initiated the transfer.
Command Identifier	Command Identifier value passed to the **Receive Data-Out** transport layer protocol service request that initiated the transfer.
Delivery Result	Result of the data delivery operation.

9.2.9 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 an ADT target port using one or more SCSI Data IUs. An ADT 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 Nexus, Command Identifier, Application Client Buffer, Device Server Buffer Offset, Request Byte Count))

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

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

Table 80 shows how the arguments to the **Send Data-Out** transport protocol service are used.

Table 80 — Send Data-Out transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	Used to select the session and to set the X_ORIGIN bit in the ADT frame(s) header.
Command Identifier	Used to set the EXCHANGE ID field 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 DATA 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 DATA 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.

9.2.10 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 Nexus, Command Identifier, Delivery Result))

Table 81 shows how the arguments to the **Data-Out Delivered** transport protocol service are determined.

Table 81 — Data-Out Delivered transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	I_T_L nexus value passed to the **Send Data-Out** transport layer protocol service request that initiated the transfer.
Command Identifier	Command Identifier value passed to the **Send Data-Out** transport layer protocol service request that initiated the transfer.
Delivery Result	Result of the data delivery operation.

9.2.11 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 Nexus, Command Identifier, Device Server Buffer Offset, Request Byte Count, Application Client Buffer))

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

An application client shall not call **Receive Data-In** for a given command 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 command).

Table 82 shows how the arguments to the **Receive Data-In** transport protocol service are used.

Table 82 — Receive Data-In transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	Used to select the session and to set the X_ORIGIN bit in the ADT frame(s) header.
Command Identifier	Used to set the EXCHANGE ID field in the ADT frame(s) header.
Device Server Buffer Offset	Used to set the DATA 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.

9.2.12 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 Nexus, Command Identifier, Delivery Result))

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 command is received. If a SCSI Response IU for the command is received after **Receive Data-In** has been called but before **Data-In Received** has been called, the ADT initiator port shall call **Data-In Received** before it calls **Command Complete Received**.

Table 83 shows how the arguments to the **Data-In Received** transport protocol service are determined.

Table 83 — Data-In Received transport layer protocol service arguments

Argument	ADT Implementation
I_T_L nexus	I_T_L nexus value passed to the **Receive Data-I**n transport protocol service that initiated the transfer.
Command Identifier	Command Identifier value passed to the **Receive Data-I**n transport protocol service that initiated the transfer.
Delivery Result	Result of the data delivery operation.

9.3 Task management protocol services

9.3.1 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, [Command Identifier], Function Identifier))

Table 84 shows how the arguments to the **Send Task Management Request** transport protocol service are used.

Table 84 — Send Task Management Request transport layer protocol service arguments

Argument	ADT Implementation
Nexus	I_T or I_T_L nexus identifier. If the identifier is an I_T_L nexus identifier, then the L is used to set the LUN field in the SCSI Task Management IU.
Command Identifier	Used to set the TAG OF THE TASK TO BE MANAGED field 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 and Command Identifier specifies the command to be managed); 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); and f) QUERY TASK (Nexus and Command Identifier specifies the command to be managed); and g) QUERY TASK SET (Nexus argument specifies an I_T_L Nexus).

9.3.2 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, [Command Identifier], Function Identifier))

Table 85 shows how the arguments to the **Task Management Request Received** transport protocol service are determined.

Table 85 — Task Management Request Received transport layer protocol service arguments

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

9.3.3 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, Command Identifier, Service Response, [Additional Response Information]))

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

Table 86 shows how the arguments to the **Task Management Function Executed** transport protocol service are used.

Table 86 — Task Management Function Executed transport layer protocol service arguments

Argument	ADT Implementation
Nexus	I_T or I_T_L nexus from the Task Management Request Received argument list.
Command Identifier	Command Identifier from the Task management Request Received argument list.
Service Response	Used to set the RESPONSE CODE field in the SCSI Response IU: a) FUNCTION COMPLETE: The RESPONSE CODE field is set to COMMAND OR TASK MANAGEMENT FUNCTION COMPLETE; b) FUNCTION SUCCEEDED: The RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION SUCCEEDED; c) FUNCTION REJECTED: The RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION NOT SUPPORTED or INVALID FIELD IN ENCAPSULATED SCSI IU (EXCLUDES CDB); d) INCORRECT LOGICAL UNIT NUMBER: The RESPONSE CODE field is set to INVALID LOGICAL UNIT NUMBER IN SCSI TASK MANAGEMENT IU (see SAM-5); or e) SERVICE DELIVERY OR SUBSYSTEM FAILURE: The RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION FAILED.
Additional Response Information	Not used

9.3.4 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, [Command Identifier], Service Response, [Additional Response Information]))

Table 87 shows how the arguments to the **Received Task Management Function-Executed** transport protocol service are determined.

Table 87 — Received Task Management Function-Executed transport layer protocol service arguments

Argument	ADT Implementation
Nexus	I_T or I_T_L nexus from the Send Task Management Request argument list.
Command Identifier	Command Identifier 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; c) SERVICE DELIVERY OR SUBSYSTEM FAILURE: The RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION FAILED; or d) INCORRECT LOGICAL UNIT NUMBER: The RESPONSE CODE field is set to INVALID LOGICAL UNIT NUMBER IN SCSI TASK MANAGEMENT IU (see SAM-5).
Additional Response Information	Not Used

9.4 Application client error handling

If an application client receives a service response of SERVICE DELIVERY OR TARGET FAILURE from an ADT initiator port via the **Command Complete Received** transport protocol service (see 9.2.4), then the application client should abort the command by sending an ABORT TASK task management function (i.e., call **Send Task Management Request** specifying the ABORT TASK task management function (see 9.3.1)).

9.5 SCSI mode parameters

9.5.1 Disconnect-Reconnect mode page

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

9.5.2 Protocol-Specific Port mode page

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

9.5.3 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.

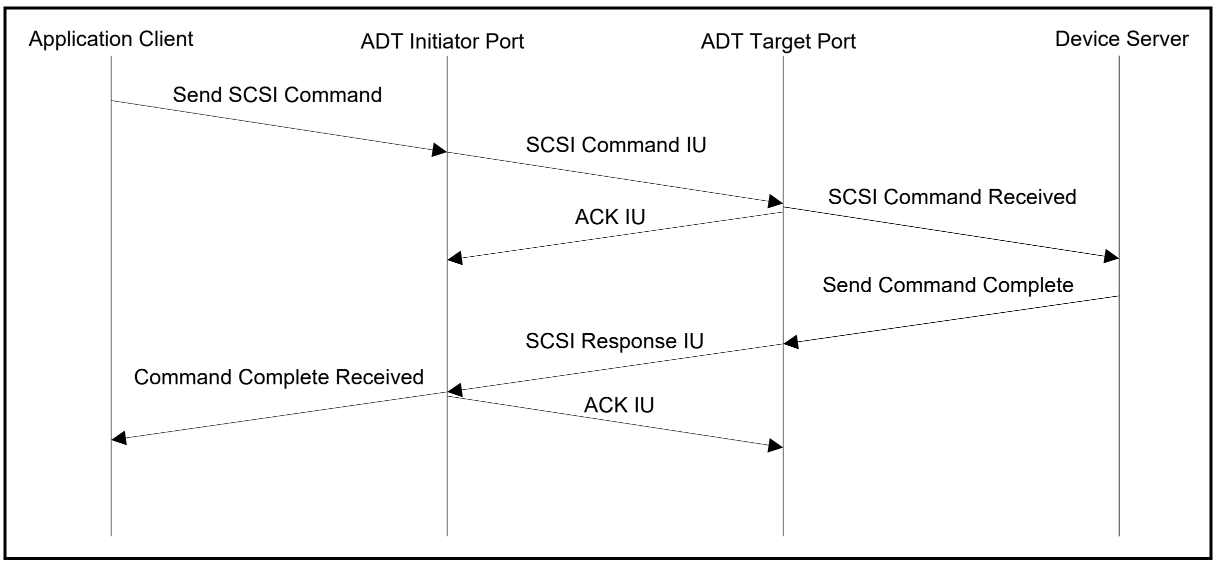


Figure A.1 — SCSI command with no data phase

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 8.1.2).

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 8.1.5) 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 8.1.5) 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.

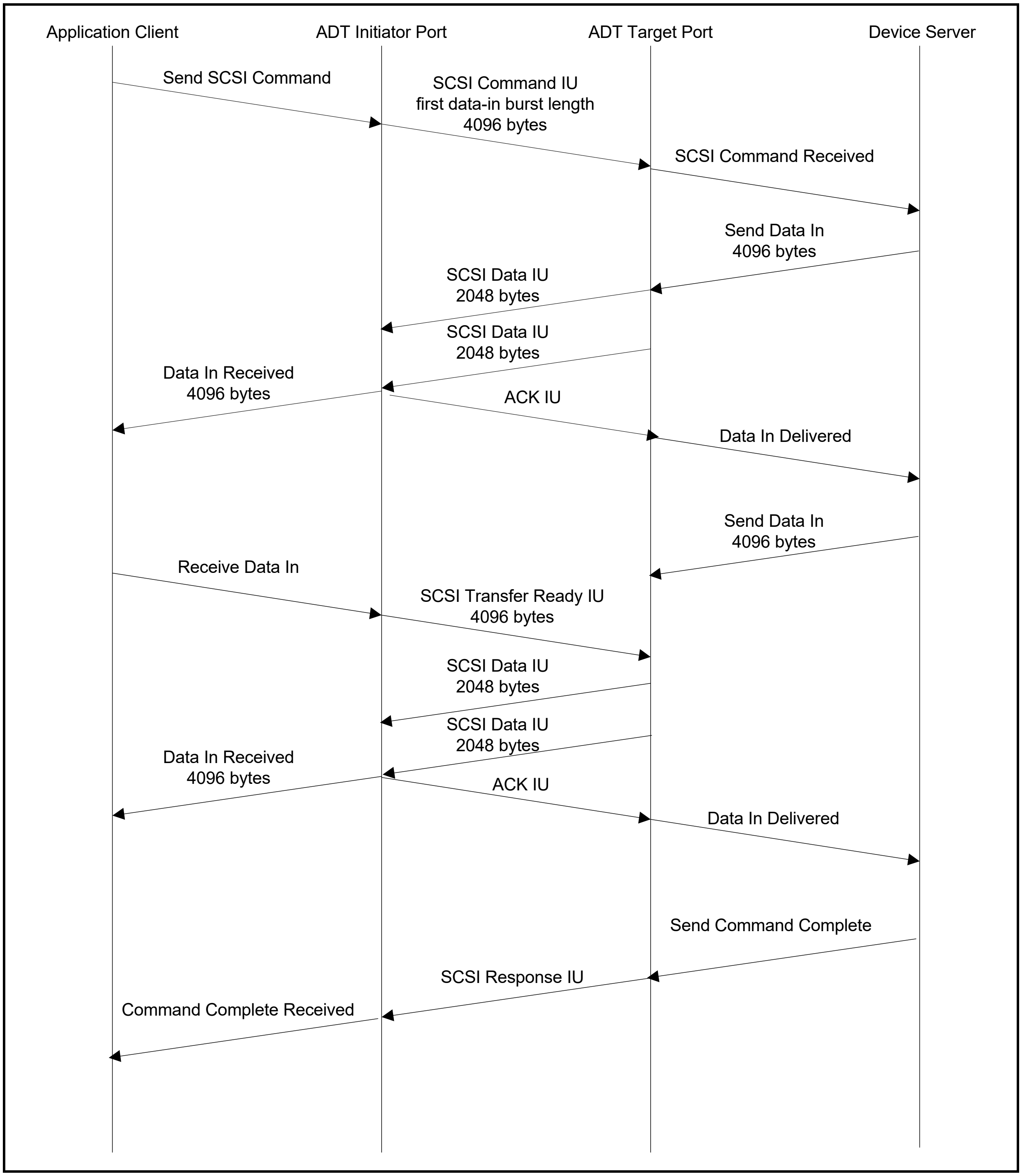


Figure A.2 — SCSI command with data-in

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 8192 bytes, 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 4096 bytes 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 8.1.5). 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 4096 bytes 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.

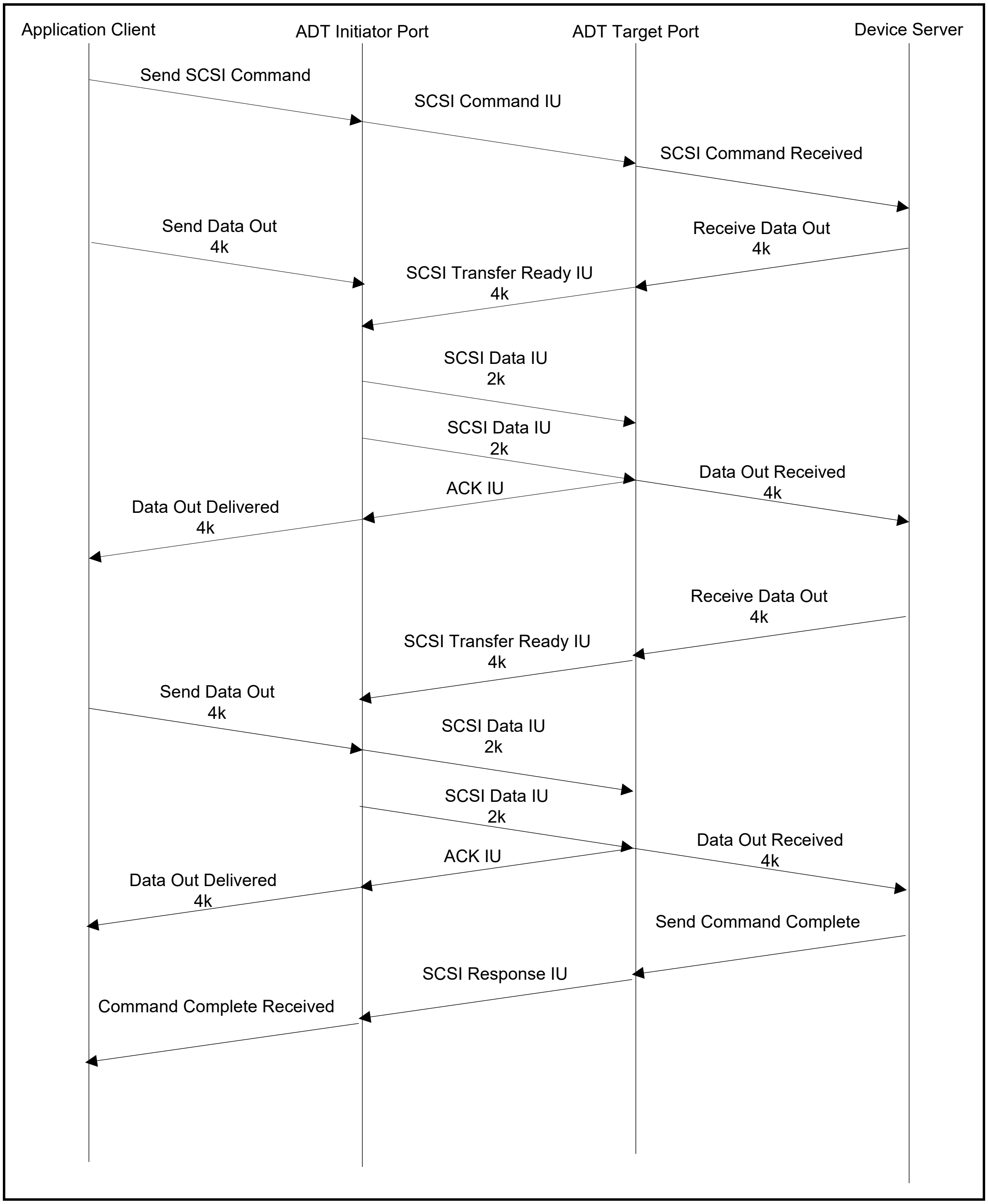


Figure A.3 — SCSI command with data-out

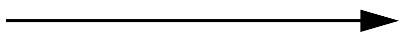
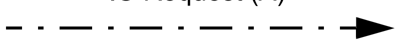
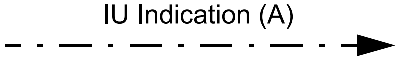
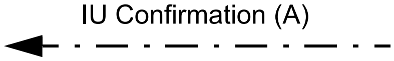
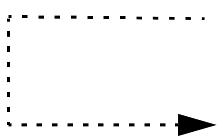
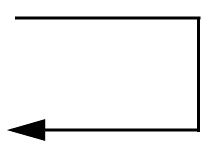
Annex B
(informative)

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.

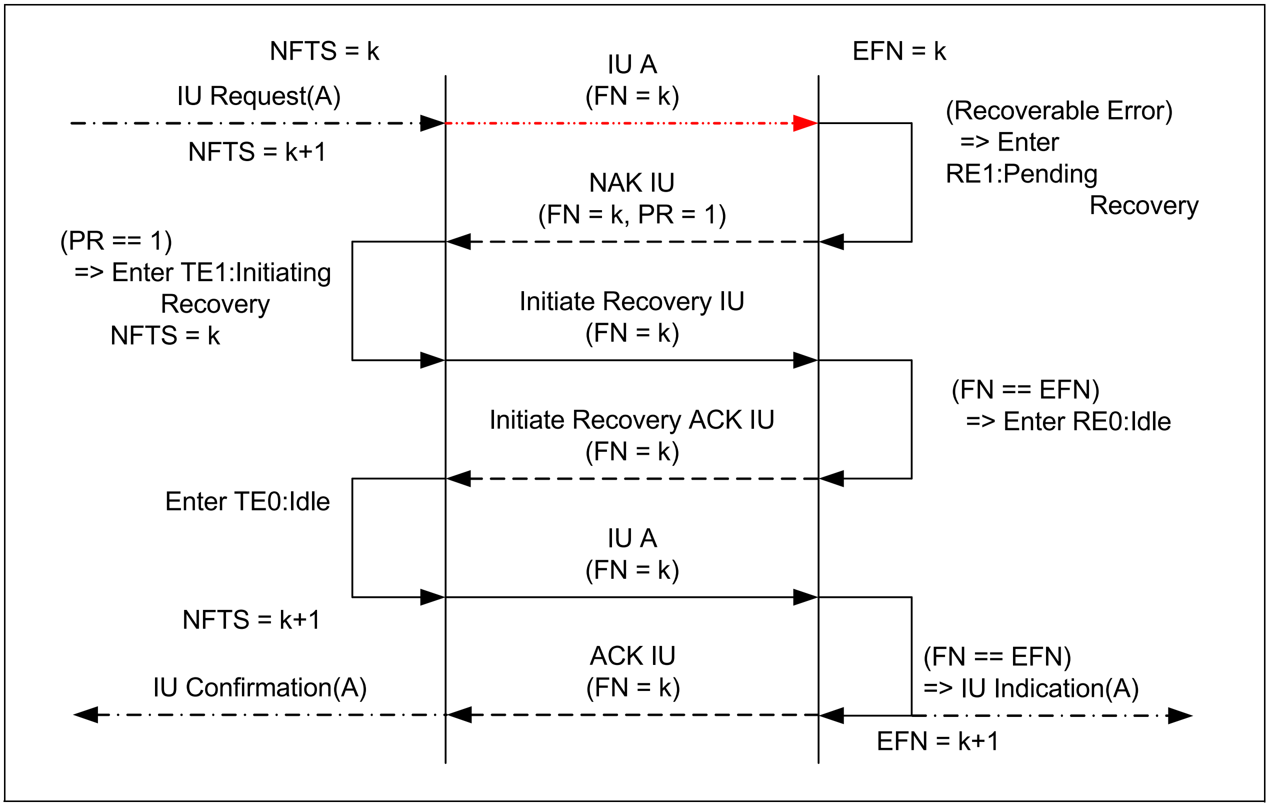


Figure B.1 — Receiver-detected retryable error

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.

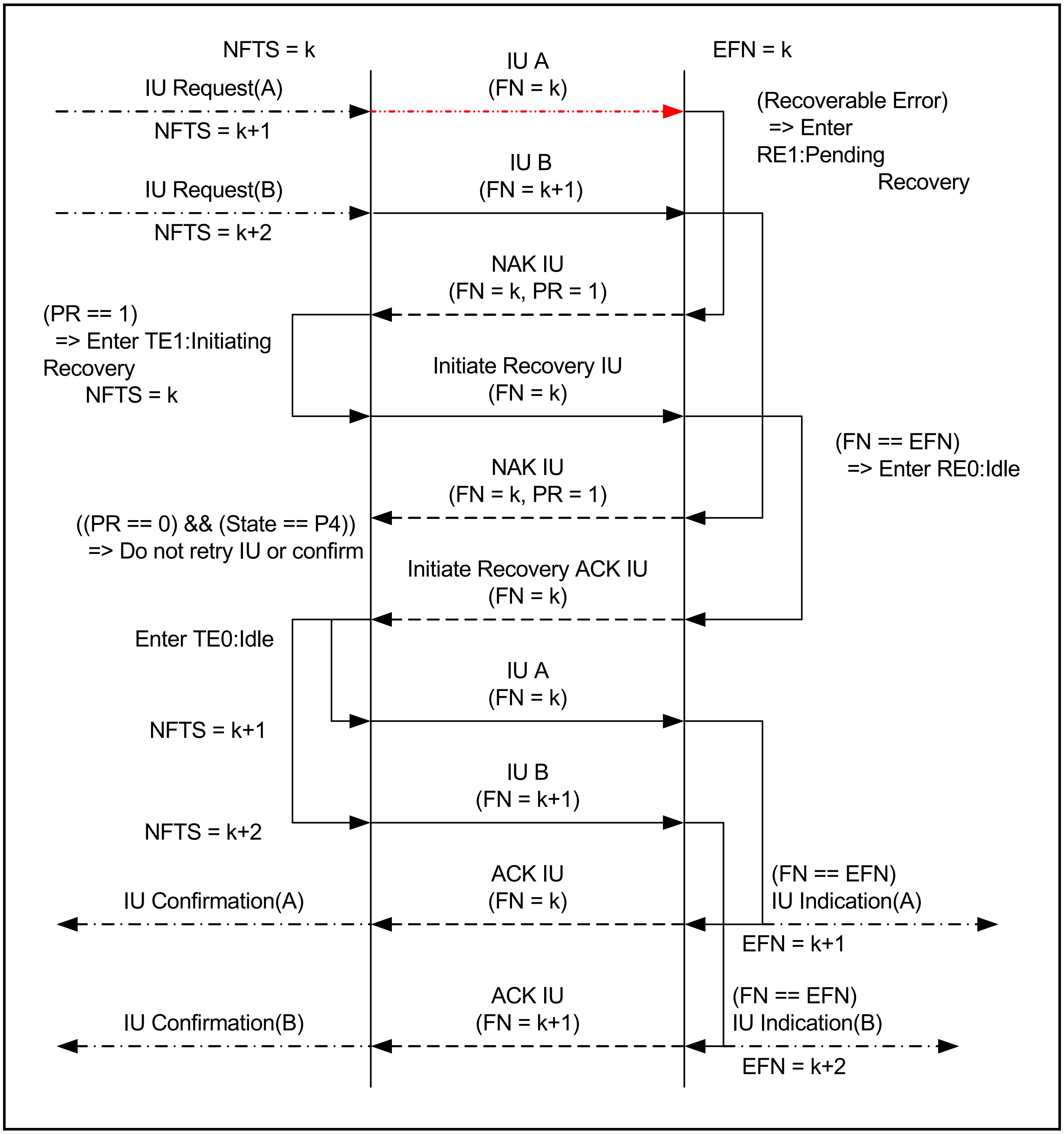


Figure B.2 — Receiver-detected retryable error with multiple active IUs

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.

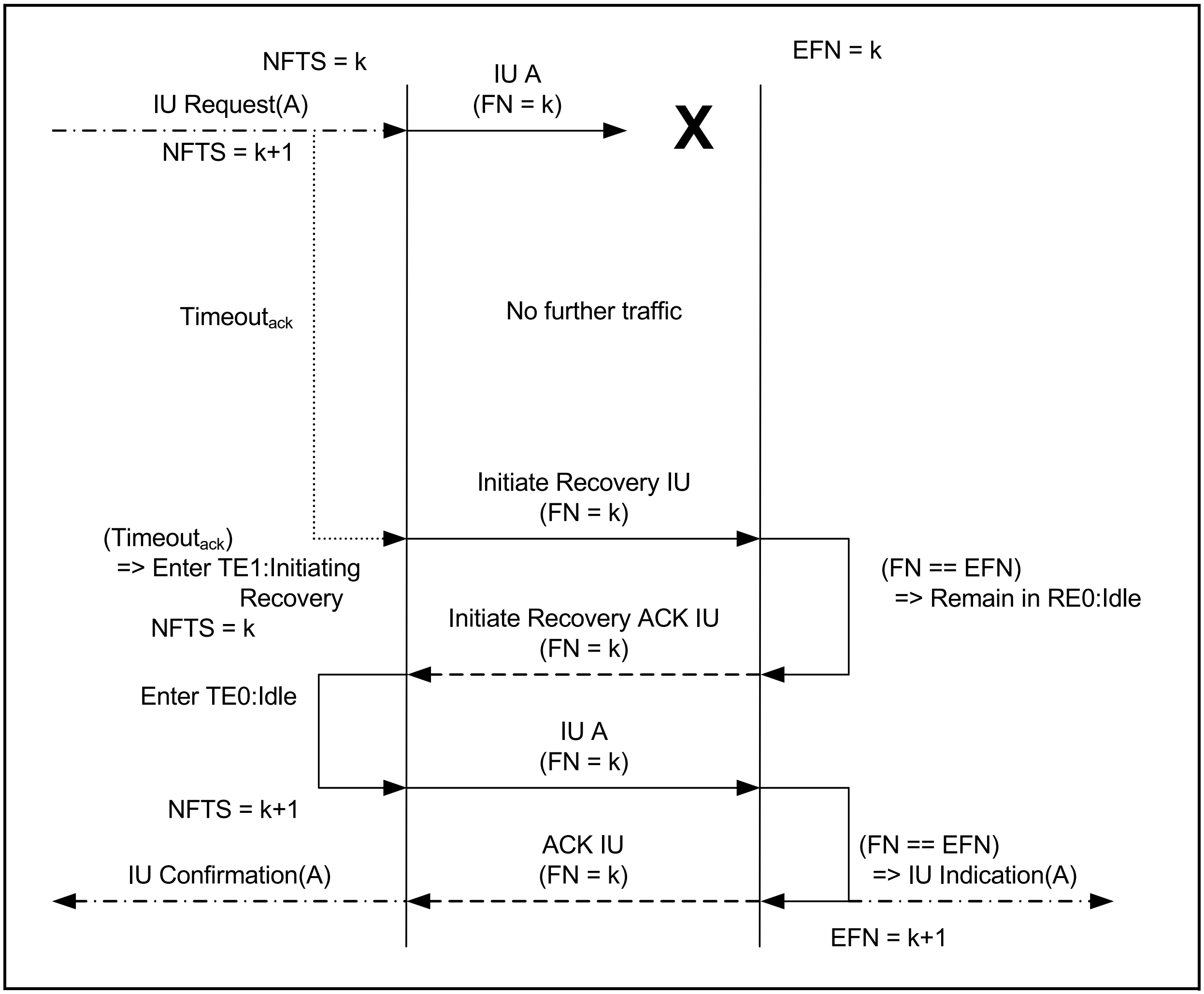


Figure B.3 — Lost IU with no further traffic

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.

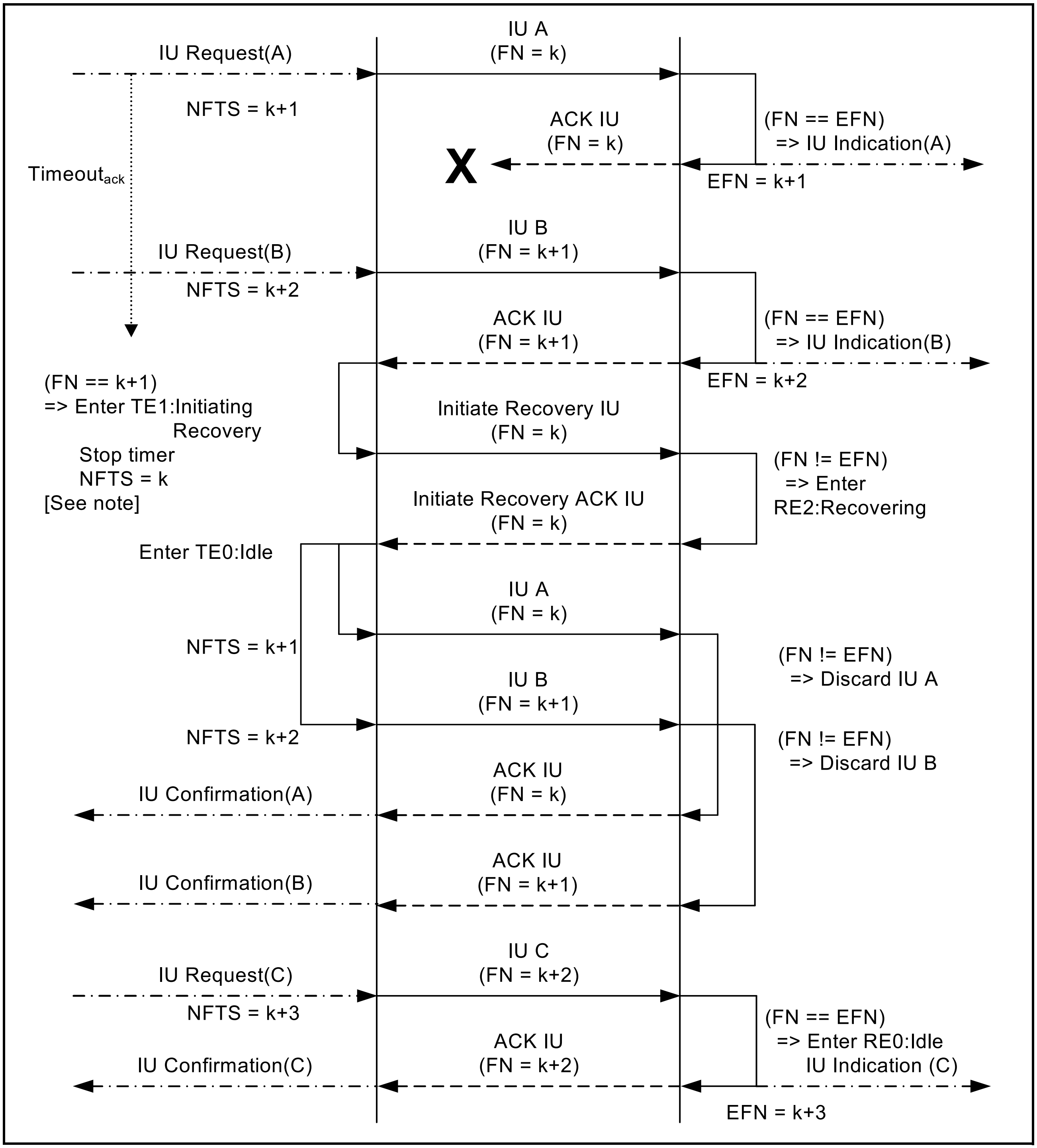


Figure B.4 — Lost ACK with recovery driven by out-of-order ACK

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.

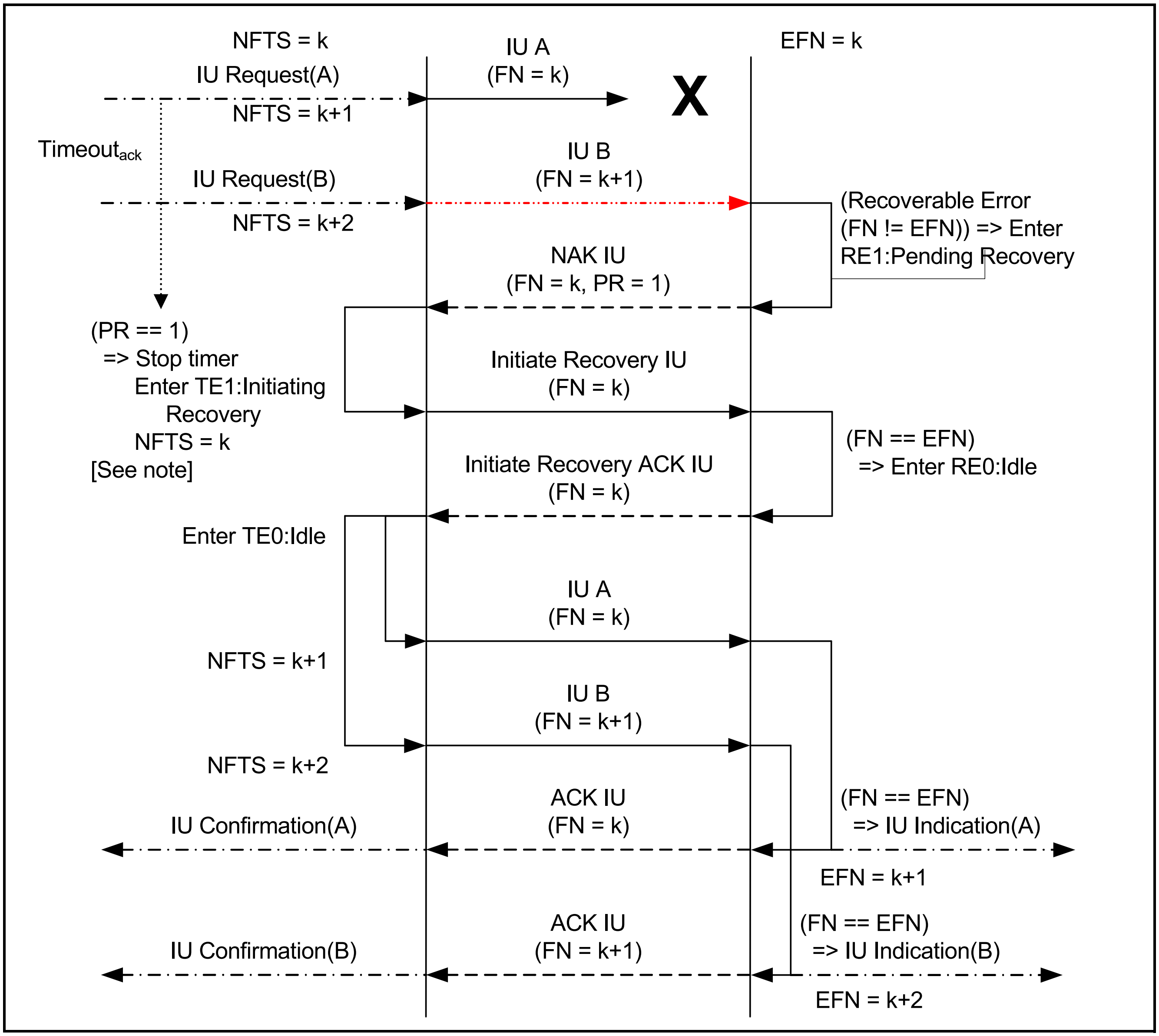


Figure B.5 — Lost IU with recovery driven by out-of-order NAK

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

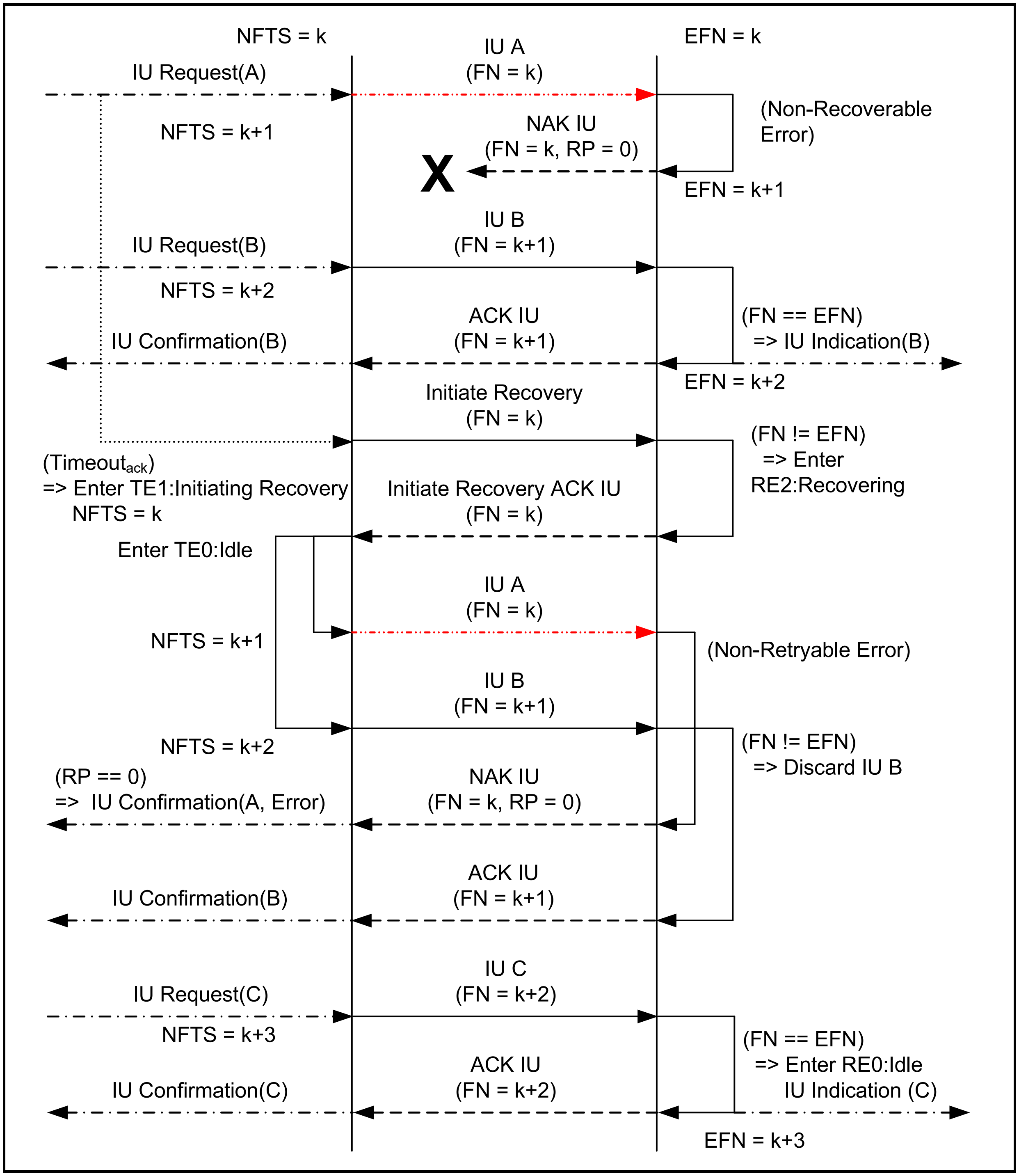


Figure B.6 — Lost NAK with recovery driven by timeout

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.

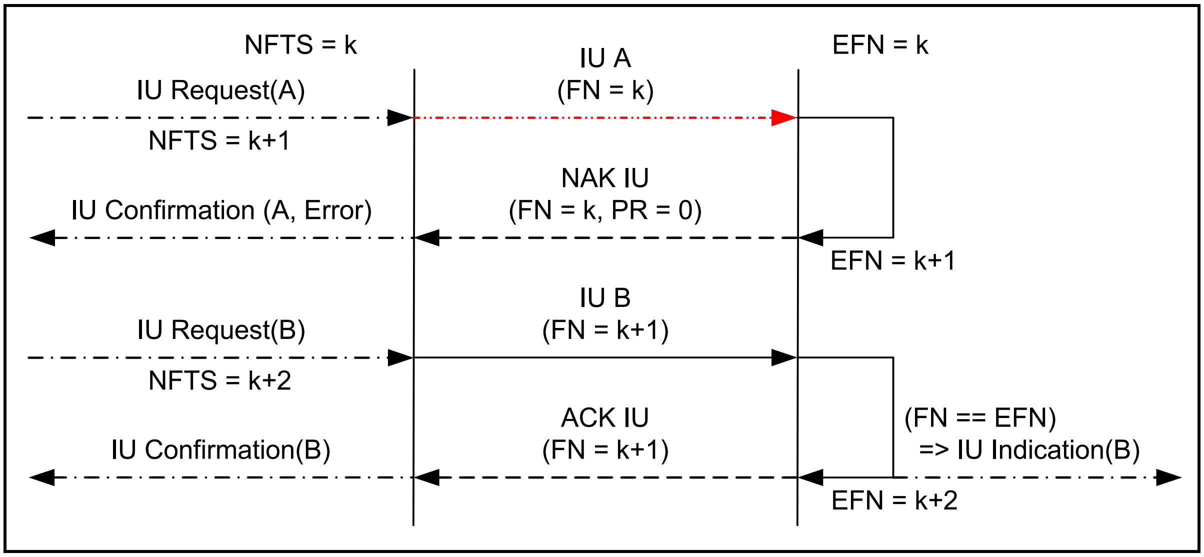


Figure B.7 — Non-retryable error

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.

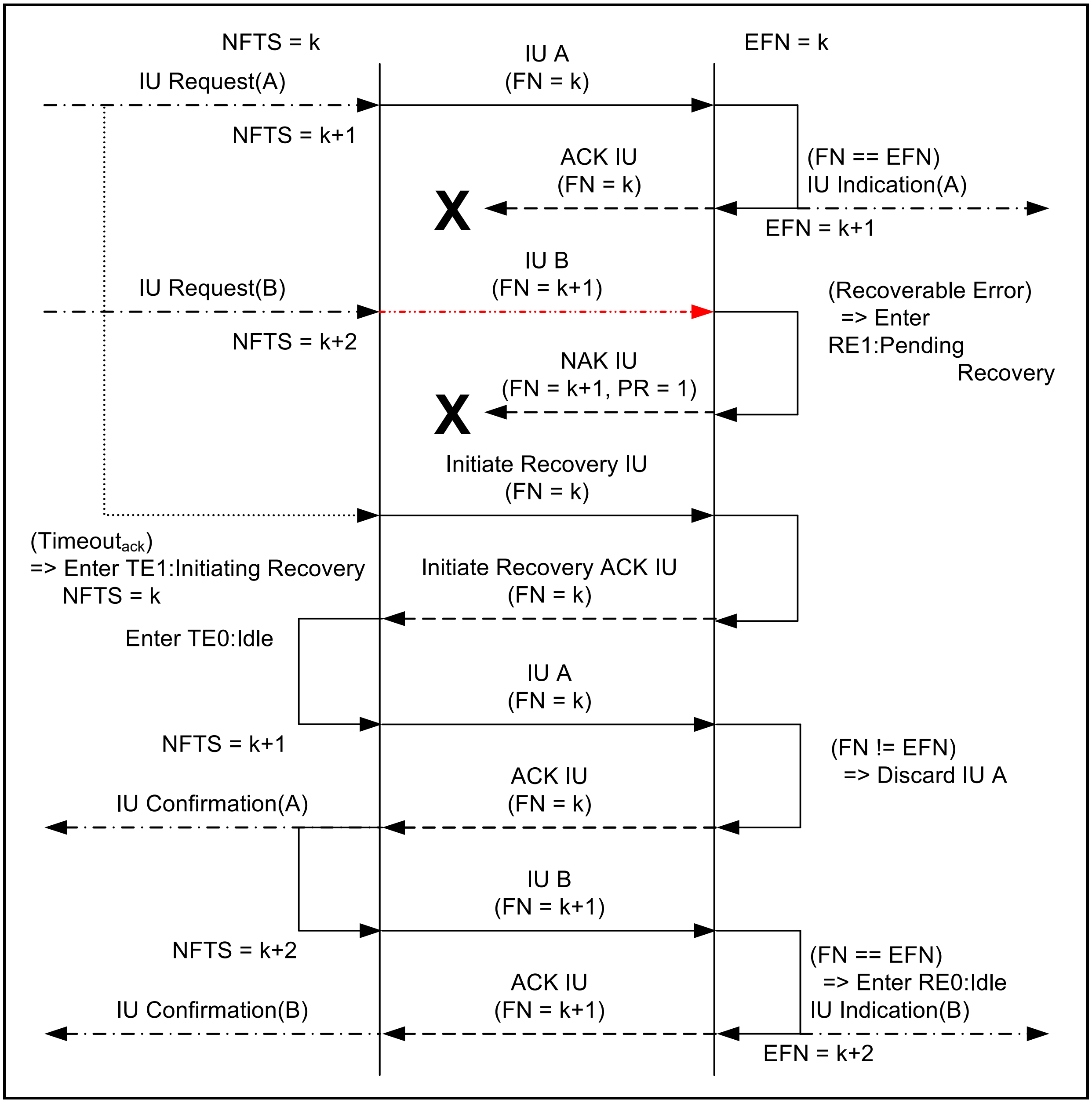


Figure B.8 — Lost ACK with errors on next IU

B.10 Delayed response with recovery driven by timeout

Figure B.9 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.

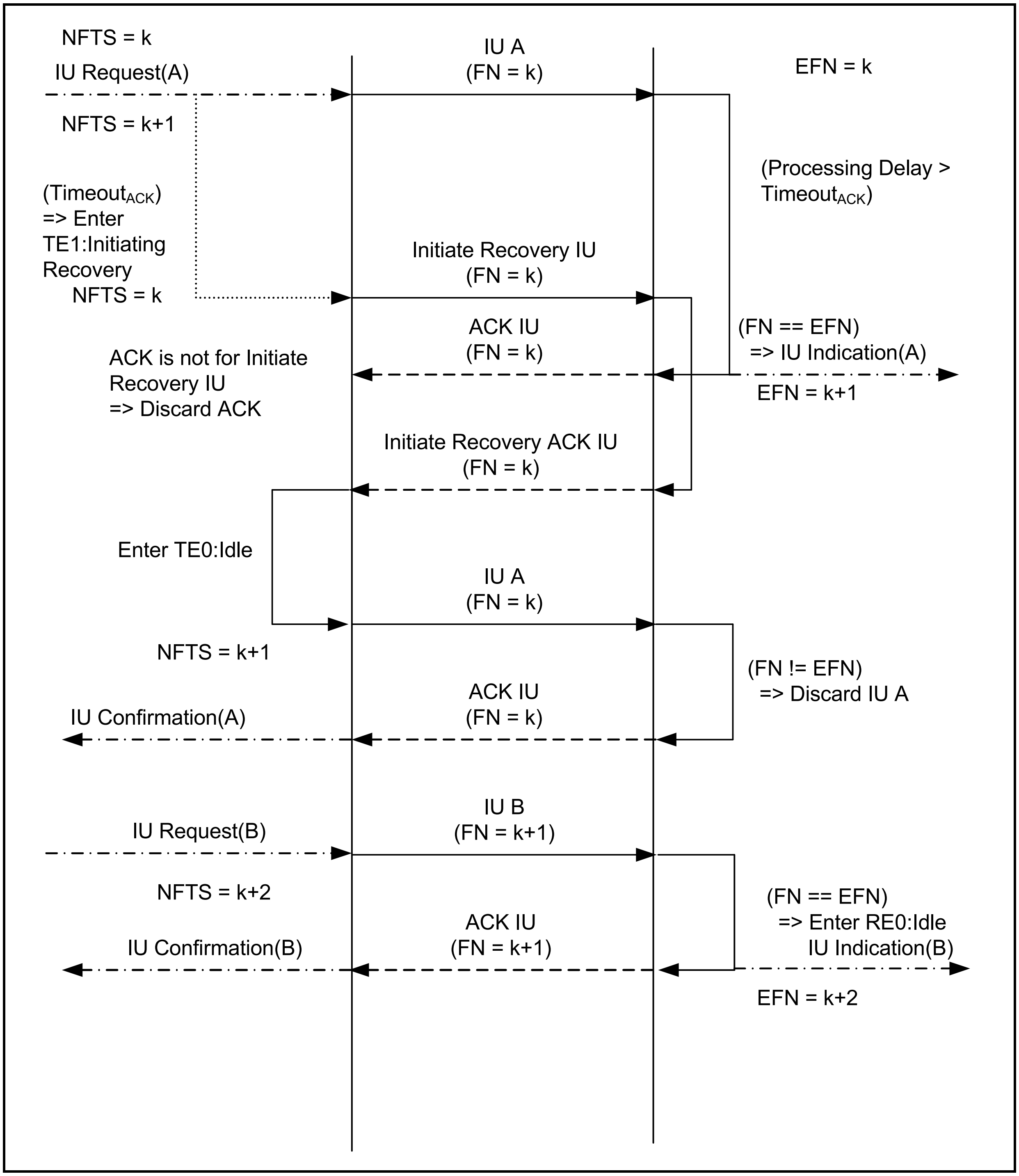


Figure B.9 — Lost ACK with errors on next IU

Annex C
(informative)

Link Negotiation Examples

C.1 Introduction

This annex provides an example link negotiation sequence as described in 4.3.3 and 7.5.4. 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;
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 the FRAME NUMBER field that matches 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 the FRAME NUMBER field that matches 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. 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 the FRAME NUMBER field that matches 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 the FRAME NUMBER field that matches 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. 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 the FRAME NUMBER field that matches 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 the FRAME NUMBER field that matches 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.

Annex D
(informative)

iADT Connection Services Relationship to Sockets API

In TCP/IP implementations, the TCP calls referenced in clause 6 are typically invoked via a Sockets application programming interface (API). The details of the Sockets API varies between implementations. This annex describes the typical semantics of the Sockets API function calls and how the connection services may be mapped to those function calls.

Table D.1 describes the function calls in a typical Sockets API.

Table D.1 — Sockets API function calls (part 1 of 2)

Function	Description
socket()	Creates a socket descriptor that represents a communication endpoint. The arguments to the **socket()** function tell the system which protocol to use, and what format address structure will be used in subsequent functions.
bind()	Assigns a name to an unnamed socket that represents the address of the local communications endpoint (i.e., IP address and port number). When a socket is created with **socket()**, it exists in a name space (address family), but has no name assigned. **bind()** requests that a name be assigned to the socket.
connect()	Assigns the name of the remote communications endpoint and a connection is established between the endpoints. Performs a TCP active OPEN and causes entry to the SYN-SENT state. Return from **connect()** indicates transition to the ESTABLISHED state.
listen()	Enables the socket to accept a specified number of connection requests from remote sockets. Up to that number of requests may be queued on the socket; if additional requests are received before a queued request is removed, then the additional requests are rejected. **listen()** performs a TCP passive OPEN and causes entry to the LISTEN state. When an **accept()** is invoked on a socket with queued requests, then one request is removed and an additional request may be queued.
accept()	Accepts a connection request on a socket that is listening for connections. When **accept()** is processed: 1) a queued request is removed from the socket; 2) a new socket is created for the connection. The connection is defined by a remote IP address and port number and the local IP address and port number; 3) further packets on that connection are routed to the new socket; and 4) the original socket may be used to accept additional connection requests. If no connection request is queued on the original socket, then the **accept()** may block until one arrives or until a **close()** is invoked on the socket.

Table D.1 — Sockets API function calls (part 2 of 2)

Function	Description
send()	Sends outgoing data on a connected socket. An error status on return from **send()** may indicate entry to the CLOSE-WAIT or FIN-WAIT-1 state.
recv()	Receives incoming data that has been received by a connected socket. An error status on return from **recv()** may indicate entry to the CLOSE-WAIT or FIN-WAIT-1 state.
shutdown()	Closes a connection, optionally preventing further sends and/or receives. Causes a transition from the ESTABLISHED state to the FIN-WAIT-1 state and eventually causes a transition to the CLOSED state.
close()	Deletes a socket descriptor created by the **socket()** function. If the socket was connected, the connection is terminated. Data that has yet to be delivered to the remote endpoint is discarded. To ensure transmission and reception of all pending packets, **close()** should be invoked after **shutdown()** has returned. If the deleted socket was the original one upon which the **listen()** was invoked, then no new connections shall be accepted. Existing connections are unaffected.

Table D.2 shows how connection services may be mapped to Sockets API function calls. The **Reset** service request and **Reset Received** service indication are not listed because they are not relevant to the Sockets API.

Table D.2 — Connection service mapping to Sockets API functions (part 1 of 2)

Connection service	Socket function	Notes
Connect	**socket()**	
	bind()	As part of processing a **Connect** service request, **bind()** specifies a dynamic local port number.
	connect()	**connect()** specifies the remote socket address. This socket may not be reused for additional connections. Creating another connection requires invoking **socket()** to allocate a new socket resource and then invoking **bind()** and **connect()** on that new socket.
Connected		**connect()** may block until a connection is established with a remote socket.
Connect Received	**accept()**	**accept()** may block until a remote socket connects to the local socket. If so, then it returns the address of the remote socket. This return causes invocation of the **Connected** service confirmation, which returns the address of the remote socket in the **Connection ID** argument.
Send	**send()**	Invocation of the **Send** service request causes invocation of **send()**.
Sent		The **Sent** service confirmation is invoked after **send()** returns.
Receive	**recv()**	Invocation of the **Receive** service request causes the invocation of **recv()**, which blocks until a message is received.

Table D.2 — Connection service mapping to Sockets API functions (part 2 of 2)

Connection service	Socket function	Notes
Received		**recv()** returns when a message is received. This return causes the invocation of the **Received** service confirmation.
Disconnect	**shutdown()**	The **Disconnect** service request causes invocation of **shutdown()**
	close()	**close()** deallocates the socket used for the TCP connection.
Disconnected	**shutdown() return**	Successful completion of closing a TCP connection which was initiated by the local port causes a return from shutdown(). This causes invocation of the **Disconnected** service confirmation. Other events (e.g., physical port failure) may also cause invocation of **Disconnected**.
Disconnect Received	**send() return** **recv() return**	Closing of a TCP connection by the remote port causes invocation of the **Disconnect Received** service indication.
	close()	**close()** deallocates the socket used for the TCP connection.
Disconnect Immediate	**shutdown()**	The **Disconnect Immediate** service request causes invocation of **shutdown()**

Annex E
(informative)

iADT Redundant Connection Resolution Examples

E.1 Introduction

If two iADT ports initiate TCP connections simultaneously, then one of the connections is closed as specified in 7.6. The order in which each connection is established may appear differently on each iADT port depending upon transmission delays. This annex provides examples to illustrate the application of the rules in 7.6.

Figure E.1 shows the drawing conventions used in this annex. The communications associated with each connection are distinguished by use of color and solid or dashed lines. A vertical line with two horizontal lines adjacent to the Link Layer symbol indicates the beginning and end of a session. A letter on the line identifies the session. A shaded box connecting the link layers in each device and labeled "Login Processing" indicates when login processing is performed. Connection services that have a **Connection ID** argument show that argument in parentheses.

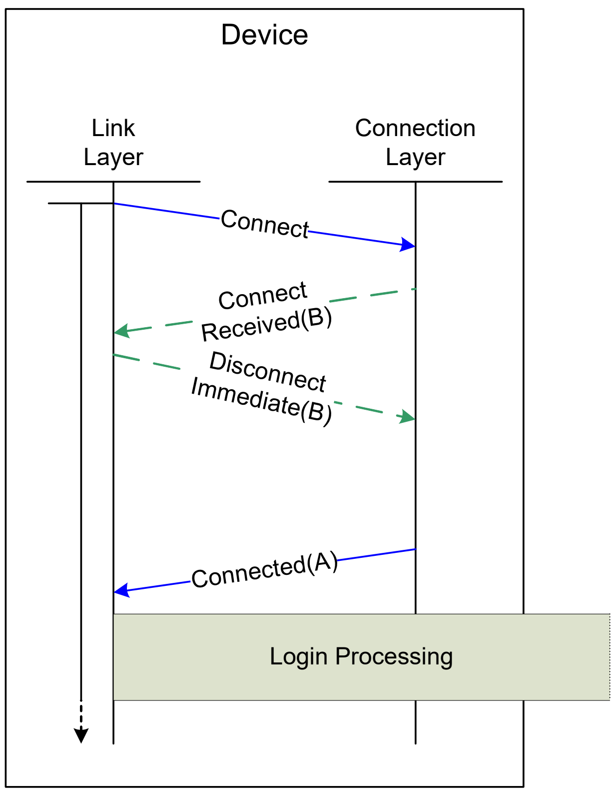


Figure E.1 — Examples of drawing conventions

E.2 Ambiguous connection establishment order

Figure E.2 shows the case in which each device sees the connections established in a different order (i.e., in the automation device the connection initiated by the DT device is established first, and in the DT device the connection initiated by the automation device is established first). No login is processed during the actions shown in this figure. Following the rules in 7.6, the automation device and the DT device close the connection initiated by the DT device.

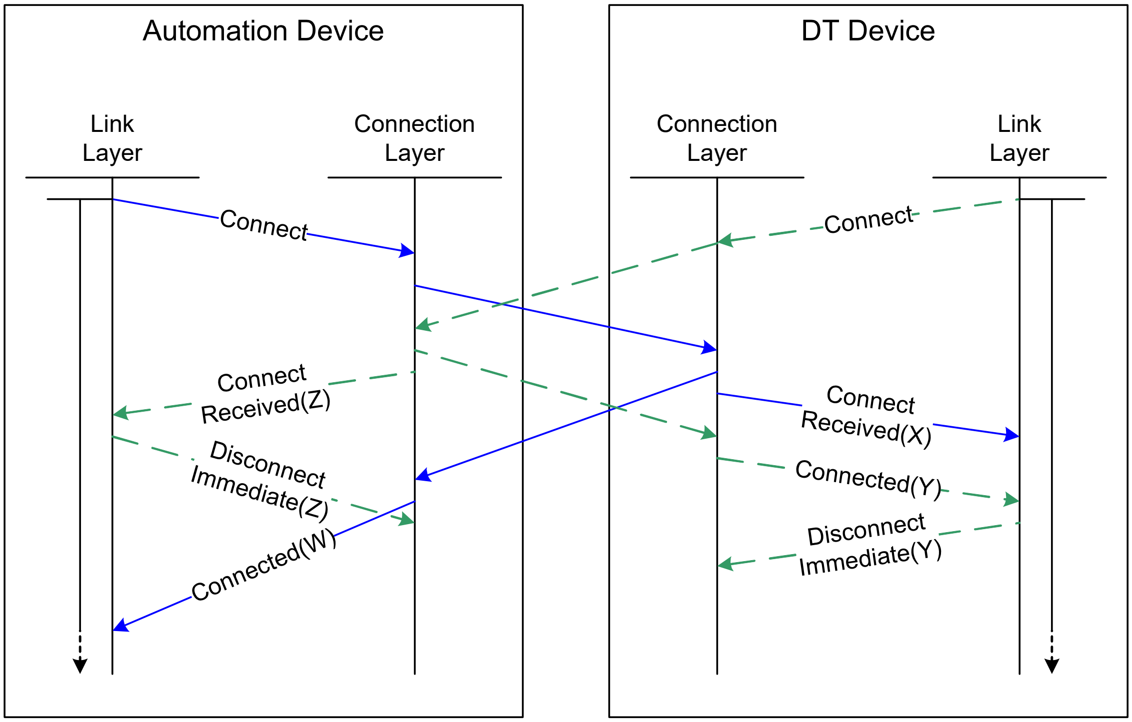


Figure E.2 — Ambiguous connection establishment order

E.3 Automation device-initiated connection established first

Figure E.3 shows the case in which both devices see the connection initiated by the automation device being established first. No login is processed during the actions shown in this figure. Following the rules in 7.6, the automation device and the DT device close the connection initiated by the DT device.

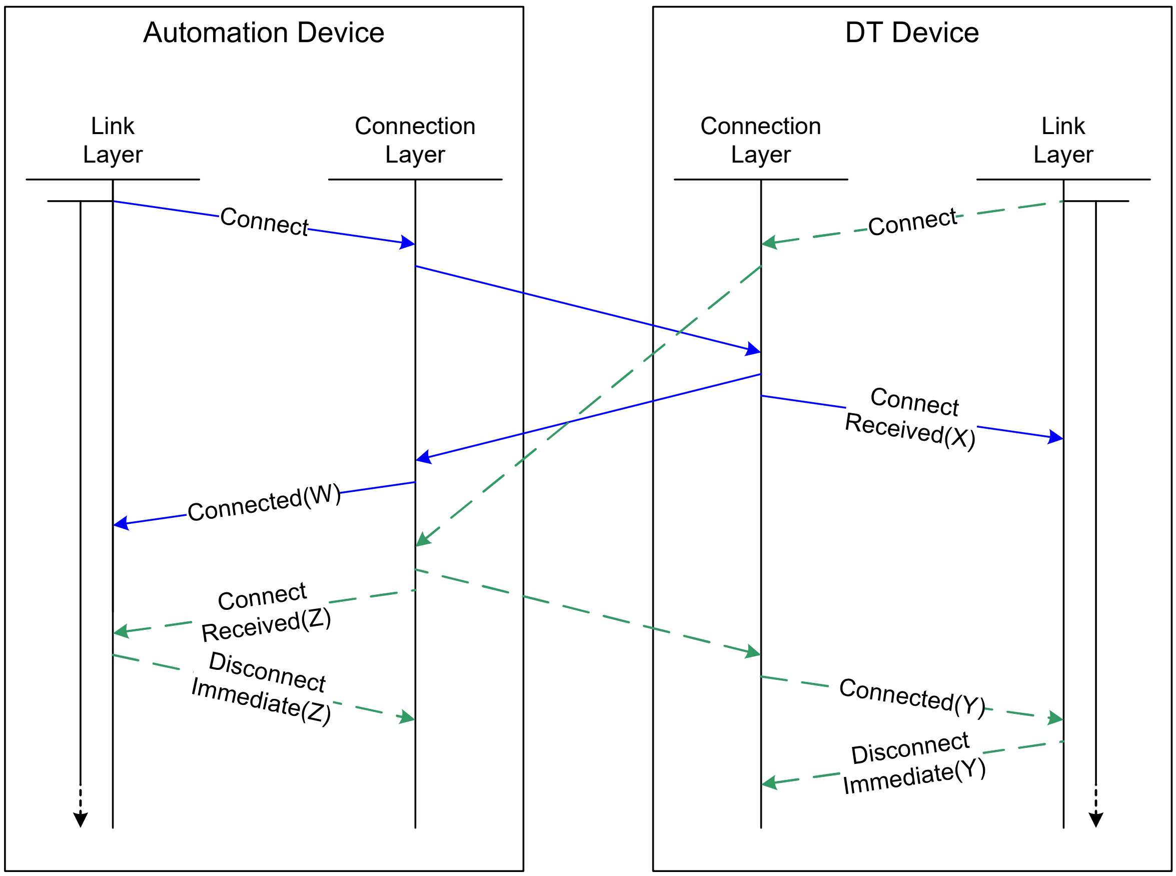


Figure E.3 — Automation device-initiated connection established first

E.4 DT device-initiated connection established first

Figure E.4 shows the case in which both devices see the connection initiated by the DT device being established first. No login is processed during the actions shown in this figure. Following the rules in 7.6, the automation and the DT device close the connection initiated by the DT device.

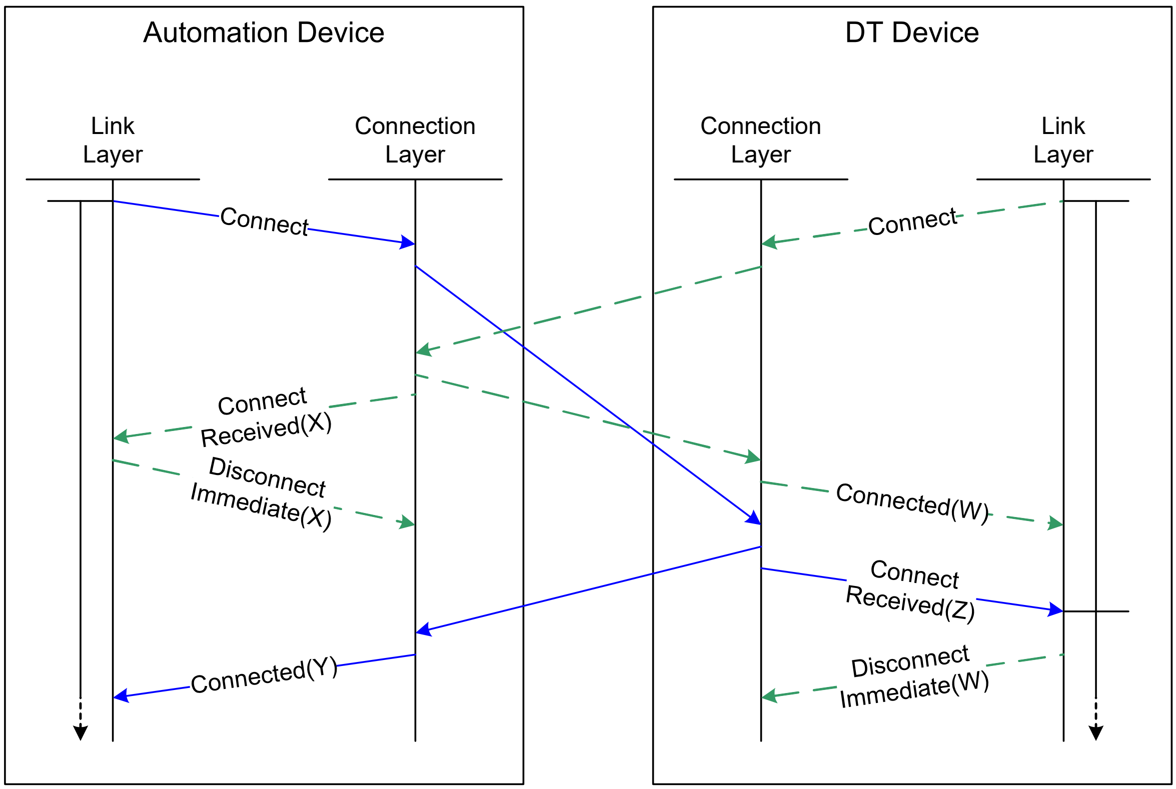


Figure E.4 — DT device-initiated connection established first

E.5 Login sent before second connection established

Figure E.5 shows the case in which the DT device establishes a connection and initiates login before the simultaneous connection initiated by the automation device has been established. Following the rules in 7.6, the connection initiated by the DT device is closed.

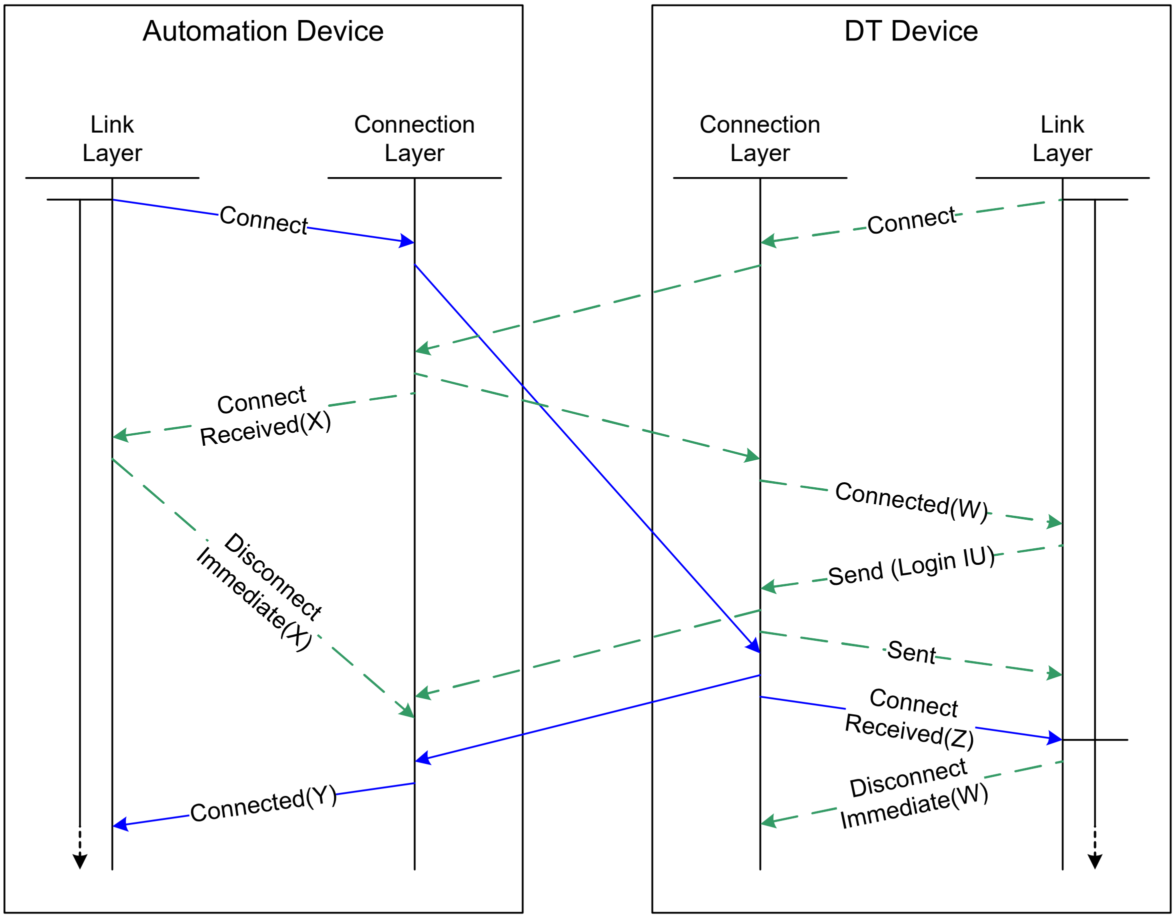


Figure E.5 — Login sent before second connection established

E.6 Connection received in session with established connection

Figure E.6 shows the case in which session A exists in both ports and then unexpectedly ends in one port, requiring establishment of a new session B and a new connection. Following the rules in 7.6, in the port processing the **Connect Received** service indication, session A ends, session B begins, connection Y is established, and the

Disconnect Immediate service request is invoked to close the previous connection X. In the port initiating the connection, the new connection identifier is W. A new login is then processed.

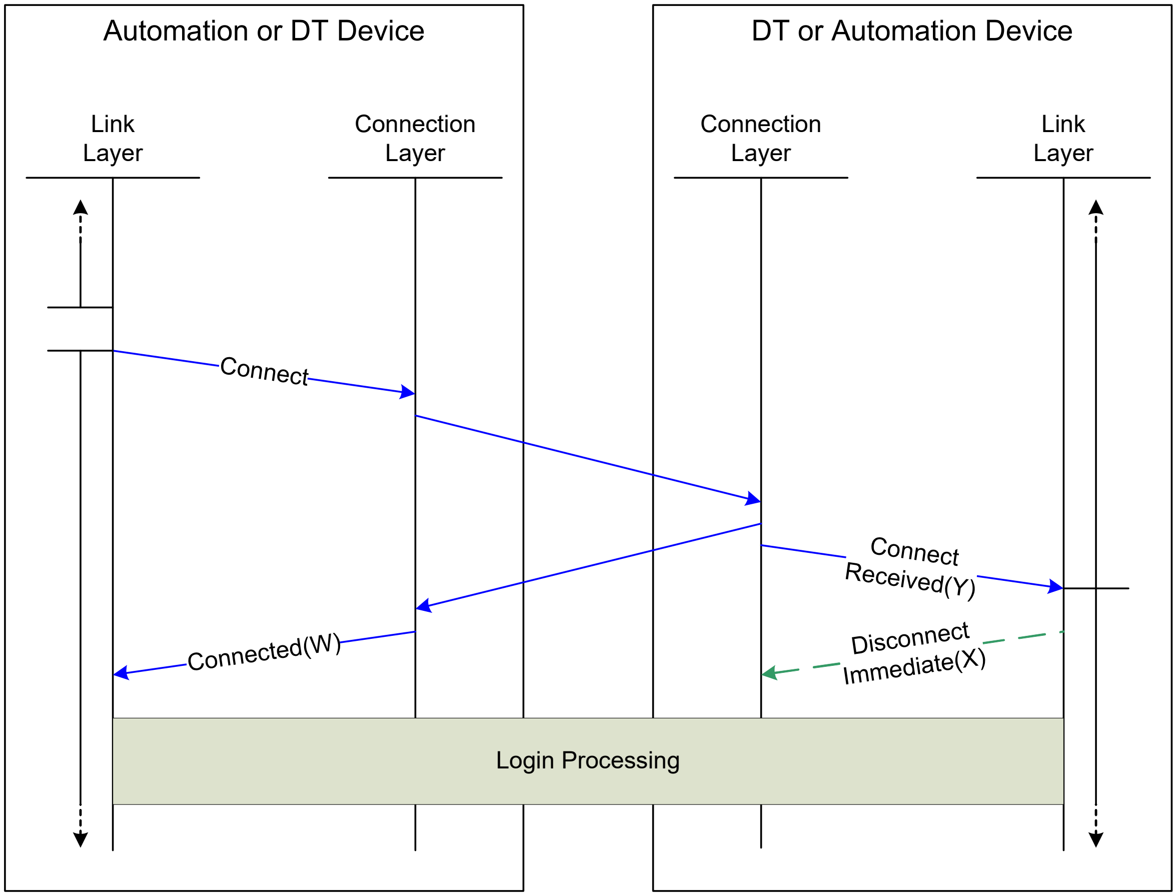


Figure E.6 — Connection received in session with established connection