Working Draft American National Standard

X3T10 Project 856D

Revision 10
31-Mar-96

Information technology - SCSI-3 Interlocked Protocol (SIP)

ASC X3T10 Technical Editor: George O. Penokie
IBM
MS 2B7
3605 Highway 52 N.
Rochester, MN 55901
USA

Telephone: 507-253-5208
Facsimile: 507-253-2432
Email: gop@rchvmp3.vnet.ibm.com

Reference number
ISO/IEC ***** : 199x
ANSI X3.292 - 199x

Printed Sunday, July 21, 1996 12:09 am

POINTS OF CONTACT:

X3T10 Chair
John B. Lohmeyer
Symbios Logic
1635 Aeroplaza Drive
Colo Spgs, CO 80916
Tel: (719) 573-3362
Fax: (719) 593-3037
Email: john.lohmeyer@symbios.com

X3T10 Vice-Chair
Lawrence J. Lamers
Adaptec
691 South Milpitas Blvd
Milpitus, CA 95035
Tel: (408) 975-7817
Fax: (408) 957-7193
Email: ljlamers@aol.com

X3 Secretariat
Lynn Barra
Administrator Standards Processing
X3 Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005
Telephone: 202-626-5738
Facsimile: 202-638-4922
Email: lbarra@cbema.org

SCSI Reflector Internet address for subscription of the SCSI reflector: scsi-request@symbios.com
Internet address for distribution via SCSI reflector: scsi@symbios.com

SCSI Bulletin Board 719-574-0424

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

ABSTRACT

This standard defines the command and task management delivery protocol required to transfer commands and data between SCSI initiators and SCSI targets attached to an SCSI-3 Parallel Interface.

PATENT STATEMENT

CAUTION: The developers of this standard have requested that holder's 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 and following calls for the identification of patents that may be required for the implementation of this standard, no such claims have been made. 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 licenses are not required to avoid infringement in the use of this standard.

Contents

Tables

Figures

Foreword

This foreword is not part of ANSI X3.292 - 199x.

The SCSI-3 Interlocked Protocol standard is divided into the following 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 relationship between this standard, the SCSI-3 Architecture Model, and the SCSI-3 Parallel Interface;
- Clause 5 describes the protocol services between this standard and the SCSI-3 Architecture Model;
- Clause 6 describes the services between this standard and the SCSI-3 Parallel Interface;
- Clause 7 describes the SCSI pointers;
- Clause 8 defines the SCSI-3 Interlocked Protocol messaging system,
- Clause 9 describes the SCSI-3 Interlocked Protocol command processing definitions,

Annex A is informative and is not considered part of the SCSI-3 Interlocked Protocol Standard.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Computer and Business Equipment Manufacturers Association, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all committee members voted for approval.

At the time it approved this standard, Committee X3 had the following members:

Technical Committee X3T10 on Lower Level Interfaces, which approved this standard, had the following members:

John B. Lohmeyer, Chair
Lawrence J. Lamers, Vice-Chair
Ralph O. Weber, Secretary

Introduction

The SCSI protocol is designed to provide an efficient peer-to-peer I/O bus with up to 32 devices, including one or more hosts. Data may be transferred asynchronously or synchronously at rates that depend primarily on device implementation and cable length.

SCSI is an I/O interface that can be operated over a wide range of media and data rates. The objectives of the Parallel Interface and Interlocked Protocol in SCSI-3 are:

a) To provide host computers with device independence within a class of devices. Thus, different disk drives, tape drives, printers, optical media drives, and other devices can be added to the host computers without requiring modifications to generic system hardware. Vendor unique indications are accommodated. Reserved areas are provided for future standardization.
b) To provide interoperability with SCSI-2 devices. Devices meeting SCSI-2 and the SCSI-3 Parallel Interface standards can co-exist on the same bus. SCSI-3 devices should be permissive of the SCSI-2 or SCSI-3 compliant behavior of other devices including those not implementing optional extensions of the SCSI-3 Interlocked Protocol Standard.

The interface protocol includes provision for the connection of multiple initiators (SCSI devices capable of initiating a task) and multiple targets (SCSI devices capable of responding to a request to perform a task). Distributed arbitration (i.e., bus-contention logic) is built into the architecture of parallel SCSI. A priority system awards interface control to the highest priority SCSI device that is contending for use of the bus.

With any technical document there may arise questions of interpretation as new products are implemented, The X3 Committee has established procedures to issue technical opinions concerning the standards developed by the X3 organization. These procedures may result in SCSI Technical Information Bulletins being published by X3.

Any such 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 X3.292 - 199x, as approved though 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 any such Technical Information Bulletins.

Current X3 practice is to make Technical Information Bulletins available through:

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

1 Scope

This standard defines the SCS-3 Interlocked Protocol (SIP). The role of SIP is to supply a mapping of the SCSI-3 command sets to the SCSI parallel bus as defined in the SCSI-3 Parallel Interface (SPI) document. SIP is closely derived from the protocol portions of the SCSI-2 standard. SIP is intended to be a fully compliant expression of the SCSI-3 Architecture Model (SAM).

Figure 1 shows the relationship of this document to other standards and related projects as of the publishing of this standard. The figure is does not imply a relationship such as a hierarchy, protocol stack, or system architecture. It indicates the applicability of a standard or related project to the implementation of a given transport.

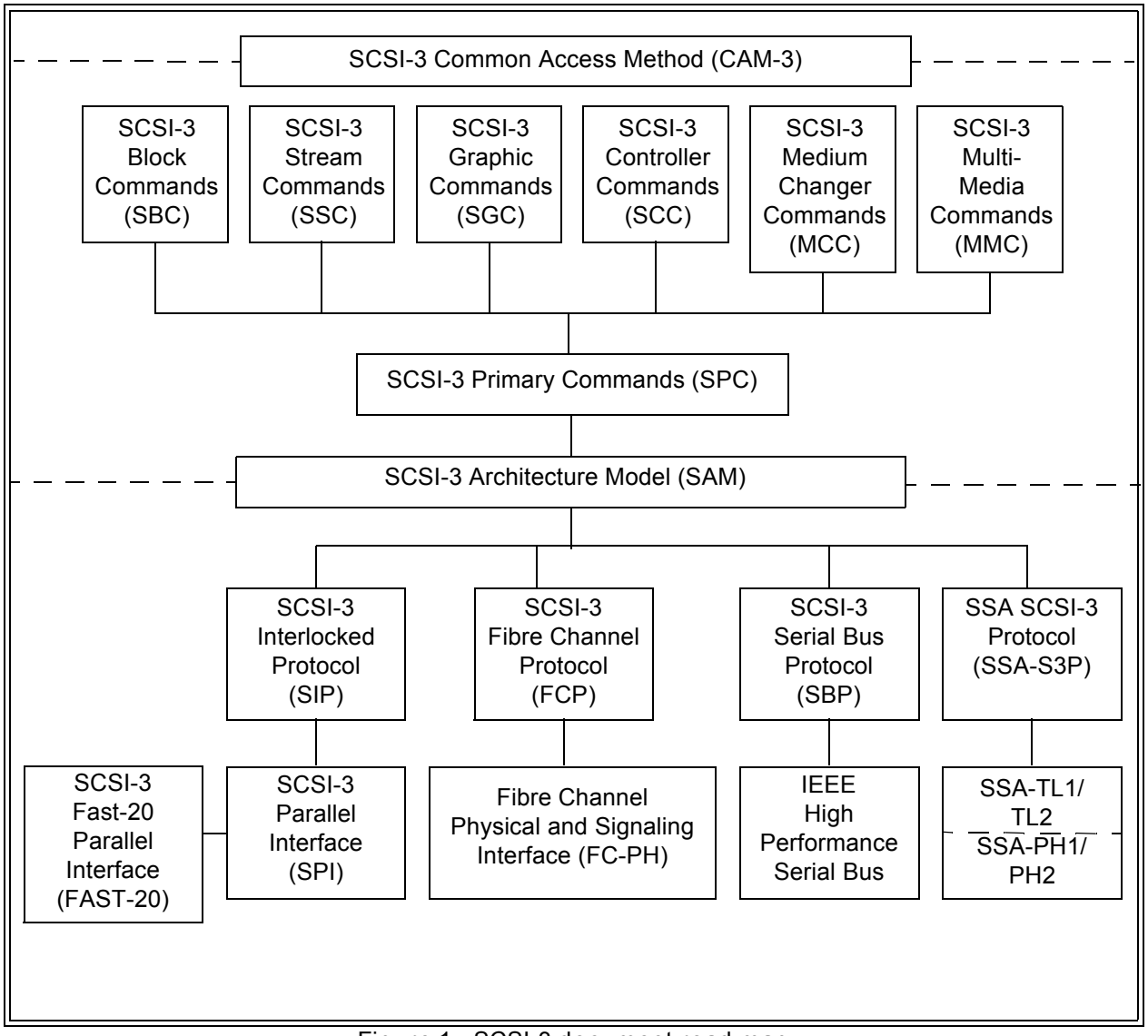


Figure 1 - SCSI-3 document road map

2 Normative references

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

encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

Members of IEC and ISO maintain registers of currently valid standards.

SCSI-3 Architecture Model Standard, ANSI X3.270-1996
SCSI-3 Parallel Interface Standard, ANSI X3.253-199x

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 agent: Carries out the actions of a requested service following the rules of the protocol.

3.1.2 application client: An object that is the source of SCSI commands. Further definition of an application client can be found in the SCSI-3 Architecture Model Standard.

3.1.3 asynchronous event notification: A procedure used by targets to notify initiators of events that occur when a pending task does not exist for that initiator.

3.1.4 auto-contingent allegiance: A condition of a task set following the return of a CHECK CONDITION or COMMAND TERMINATED status. See the SCSI-3 Architecture Model Standard for a detailed definition of auto-contingent allegiance.

3.1.5 byte: Indicates an 8-bit construct.

3.1.6 confirmation: A response returned to an object that signals the completion of a service request.

3.1.7 confirmed protocol service: A service available at the protocol service interface, that requires confirmation of completion.

3.1.8 connection: An initial connection or reconnection. A connection can only occur between one initiator and one target on the same bus. A connection begins with an initial connection or a reconnection and n ends with the next disconnect.

3.1.9 current task: A task that is in the process of sending status, transferring data, or transferring command data to or from the initiator.

3.1.10 disconnect: The action that occurs when a SCSI device releases control of the SCSI bus, allowing it to go to the BUS FREE phase.

3.1.11 dual port: SCSI devices that provide two SCSI connectors in a dual port configuration that allows any port to connect to the attached logical unit(s).

3.1.12 flag: An abstraction indicating that the condition will be communicated to the recipient of the flag.

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

3.1.14 indication: The second step of a four step confirmed service in reply to a request.

3.1.15 initial connection: An initial connection is the result of a connect. It exists from the assertion of the BSY signal (see SCSI-3 Parallel Interface Standard) in a SELECTION service until the next BUS FREE indication occurs.

3.1.16 initiator: An SCSI device containing application clients that originate device service and task management requests to be processed by a target SCSI device. See the SCSI-3 Architecture Model

Standard for a detailed definition of an initiator.

3.1.17 initiator role: The mode of operation of an SCSI device in which the SCSI device performs initiator functions.

3.1.18 initiator role agent: Carries out the actions of a requested service following the initiator role rules of the SCSI Interlocked Protocol.

3.1.19 invalid: An illegal or unsupported field or code value.

3.1.20 I_T nexus: A nexus which exists between an initiator and a target.

3.1.21 I_T_L nexus: A nexus which exists between an initiator, a target, and a logical unit. This relationship replaces the prior I_T nexus.

3.1.22 I_T_L_Q nexus: A nexus between an initiator, a target, a logical unit, and a queue tag following the successful receipt of one of the queue tag messages. This relationship replaces the prior I_T_L nexus.

3.1.23 logical unit: An externally addressable entity within a target that implements an SCSI device model. See the SCSI-3 Architecture Model Standard for a detailed definition of a logical unit.

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

3.1.25 lower level protocol: A protocol used to carry the information representing upper level transactions.

3.1.26 mandatory: The referenced item is required to claim compliance with this standard.

3.1.27 message: One or more bytes transferred between an initiator and a target to do link control, task management, and to attach task attributes to commands.

3.1.28 nexus: A relationship between an initiator and a target that begins with an initial connection and ends with the completion of the associated I/O process.

3.1.29 object: An architectural abstraction that encapsulates data types, services, or other objects that are related in some way.

3.1.30 one: A true signal value or a true condition of a variable.

3.1.31 optional: The referenced item is not required to claim compliance with this standard. Implementation of an optional item shall be as defined in this standard.

3.1.32 port: A single attachment to an SCSI bus from an SCSI device.

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

3.1.34 queue tag: The parameter associated with a task that uniquely identifies it from other tagged tasks for a logical unit from the same initiator.

3.1.35 parallel interface agent: Carries out the actions of a requested service following the rules of the SCSI parallel interface.

3.1.36 pending task: A task that is not the current task.

3.1.37 reconnect: The act of resuming a nexus to continue a task. A target reconnects when conditions are appropriate for the physical bus to transfer data associated with a nexus between an initiator and a

target.

3.1.38 reconnection: A reconnection is the result of a reconnect and it exists from the receipt of a selection confirmation with the selection won flag set to one or a reselection confirmation with the reselection won flag set to one until the next bus free indication occurs.

3.1.39 request: A transaction invoking a service.

3.1.40 reselection ID: The bit-significant representation of the target SCSI address in combination with the initiator SCSI address that is the result of a successful reselection service request.

3.1.41 reserved: Identifies bits, fields, and code values that are set aside for future standardization.

3.1.42 response: The third step of a four set confirmed service in reply to an indication.

3.1.43 SCSI device: An initiator or a target attached to the SCSI bus.

3.1.44 SCSI ID: The bit-significant representation of the SCSI address.

3.1.45 selection ID: The bit-significant representation of the target SCSI address in combination with the initiator SCSI address that is the result of a successful selection service request.

3.1.46 target: An SCSI device that receives SCSI commands and directs such commands to one or more logical units.

3.1.47 target role: The mode of operation of an SCSI device in which the SCSI device performs target functions.

3.1.48 target role agent: Carries out the actions of a requested service following the target role rules of the SCSI Interlocked Protocol.

3.1.49 task: An object within the logical unit representing the work associated with a command or group of linked commands. A task consists of one initial connection and zero or more reconnections, all pertaining to the task.

3.1.50 task set: A group of tasks within a target device, whose interaction is dependent on the queueing and auto-contingent allegiance rules. See the SCSI-3 Architecture Model Standard for a detailed definition of a task set.

3.1.51 upper level protocol: Any application specific protocol executed through services provided by a lower level protocol.

3.1.52 unconfirmed protocol service: A service available at the protocol service interface, that does not result in a completion confirmation.

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

3.1.54 word: In this standard this term indicates a 1-byte, 2-byte, or 4-byte construct.

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

3.2 Symbols and abbreviations

LSB	Least significant bit
LUN	Logical unit number

MSB	Most significant bit
RIRC	Request indication response confirmation
SCSI	Either SCSI-2 or SCSI-3
SCSI-2	The Small Computer System Interface - 2 (ANSI X3.131)
SCSI-3	The Small Computer System Interface - 3

3.3 Conventions

Certain words and terms used in this American National Standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in 3.1 or in the text where they first appear, Names of messages, commands, statuses, sense keys, additional sense codes, and additional sense code qualifiers are in all uppercase (e.g., REQUEST SENSE), names of fields are in small uppercase (e.g., STATE OF SPARE), lower case is used for words having the normal English meaning.

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

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

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

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

Some of the terms defined in the SCSI-3 Architecture Model Standard that are used throughout clause 5 are equivalent to terms defined in the SCSI-3 Parallel Interface Standard that are used throughout clause 6 even though the terms are labeled differently. Any term that has this characteristic is labeled as follows:

SCSI-3 Parallel Interface Standard term{SCSI-3 Architecture Model Standard term}

3.4 Notion for Procedures and Functions

Procedure Name ([input-1a|input-1b|inout-1c][,input-2a+input2b]...[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-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.

4 General

4.1 Overview

The SCSI-3 Interlocked Protocol Standard defines the protocol and services provided to a SCSI application client and an SCSI parallel interface. The SCSI-3 Interlocked Protocol Standard does not imply any particular implementation or any interface between the SCSI-3 Interlocked Protocol Standard and the other SCSI-3 standards except as defined within this standard.

This standard describes a device's behavior in terms of function levels, service interfaces between levels and peer-to-peer protocols. For a full description of the model used in this standard see the SCSI-3 Architecture Model Standard. Figure 2 shows the model as to appears from the point of view of the SCSI-3 Interlocked Protocol Standard.

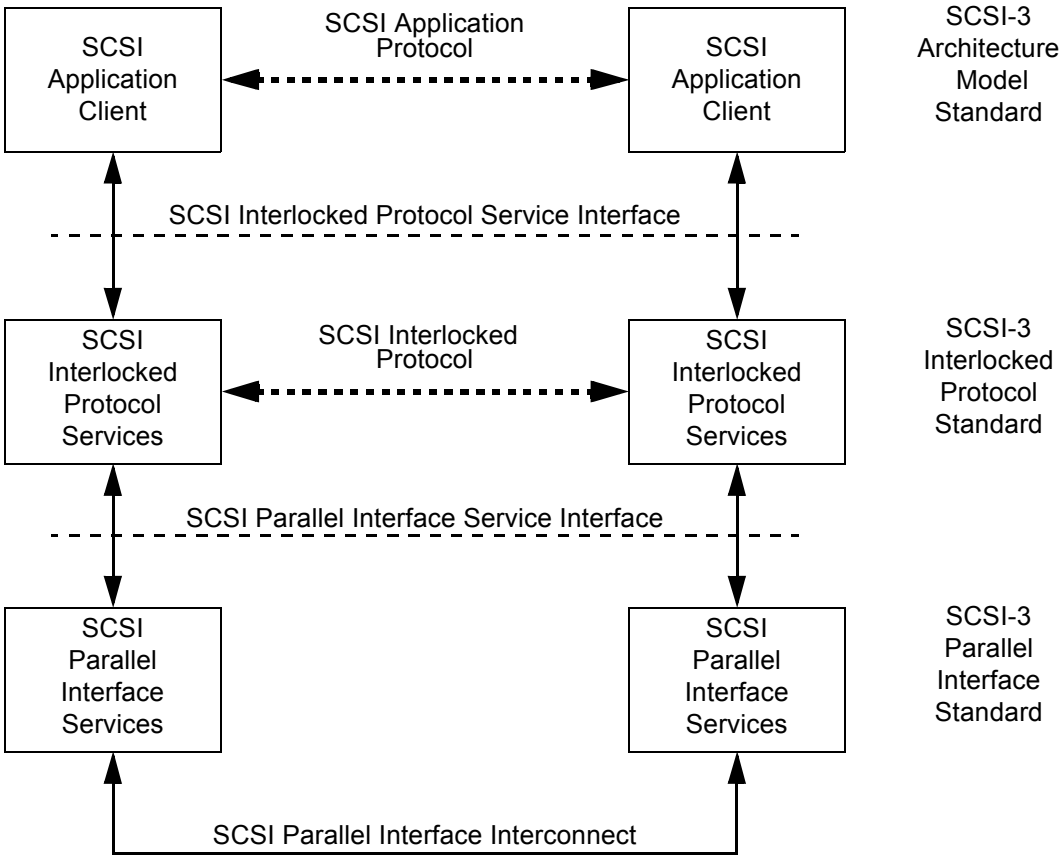


Figure 2 - SCSI-3 Interlocked Protocol service reference mode

Services between service levels are either four step confirmed services, two step confirmed services, or unconfirmed services. A four step confirmed service consists of a service request, indication, response, and confirmation. A two step confirmed service consists of a service request and confirmation. An unconfirmed service consists of only a service request and results in one or more service indications. Unconfirmed services only occur at the SCSI parallel interface service interface.

Figures 3 and 4 show the service and protocol interactions for a four step confirmed service.

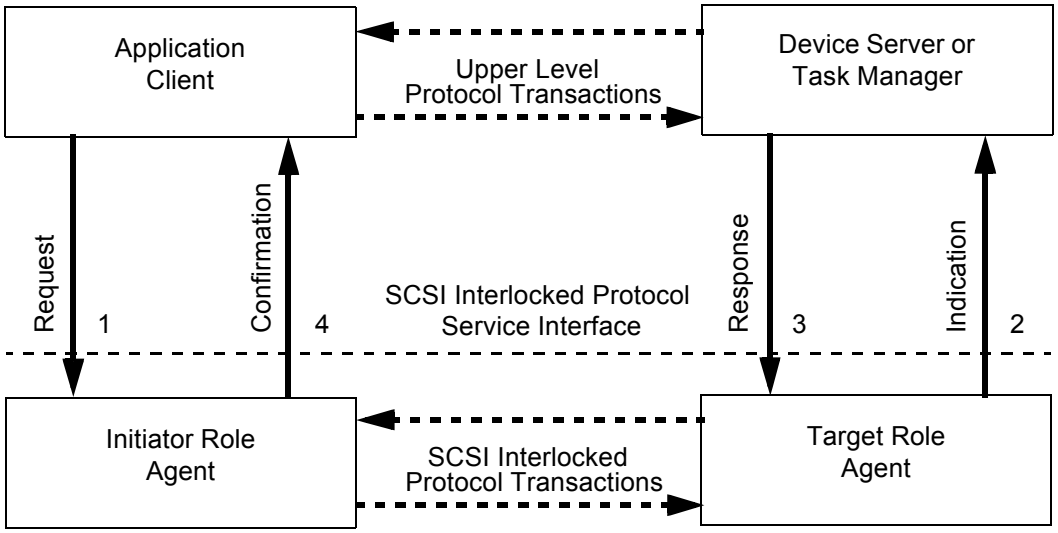


Figure 3 - Model for a four step confirmed services at the SCSI Interlocked Protocol service interface

The SCSI Interlocked Protocol service interface consists of the following interactions:

a) A request to the initiator role agent to invoke a service;
b) An indication from the target role agent notifying the device server or task manager of an event;
c) A response from the device server or task manager in reply to an indication;
d) A confirmation from the initiator role agent upon service completion.

At the SCSI Interlocked Protocol service interface, only application clients shall request a four step confirmed service be invoked.

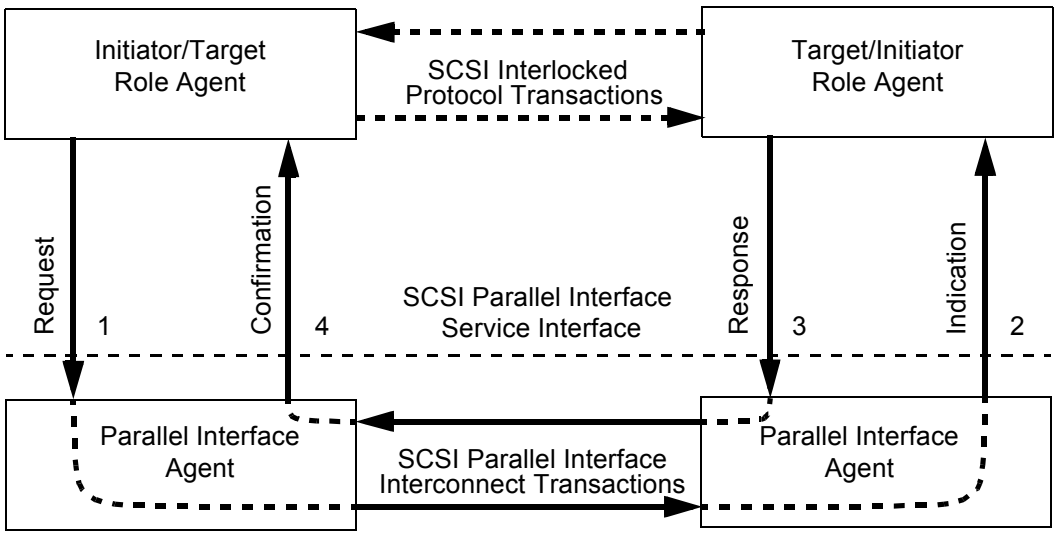


Figure 4 - Model for a four step confirmed services at the SCSI parallel interface service interface

The SCSI parallel interface service interface consists of the following interactions:

a) A request to the parallel interface agent, to invoke a service;
b) An indication from the parallel interface agent notifying the target/initiator role agent of an event;
c) A response from the target/initiator role agent in reply to an indication;
d) A confirmation from the parallel interface agent upon service completion.

At the SCSI parallel interface service interface, either the initiator role agent or the target role agent may request a service be invoked.

Figure 5 shows the service and protocol interactions for a two step confirmed service.

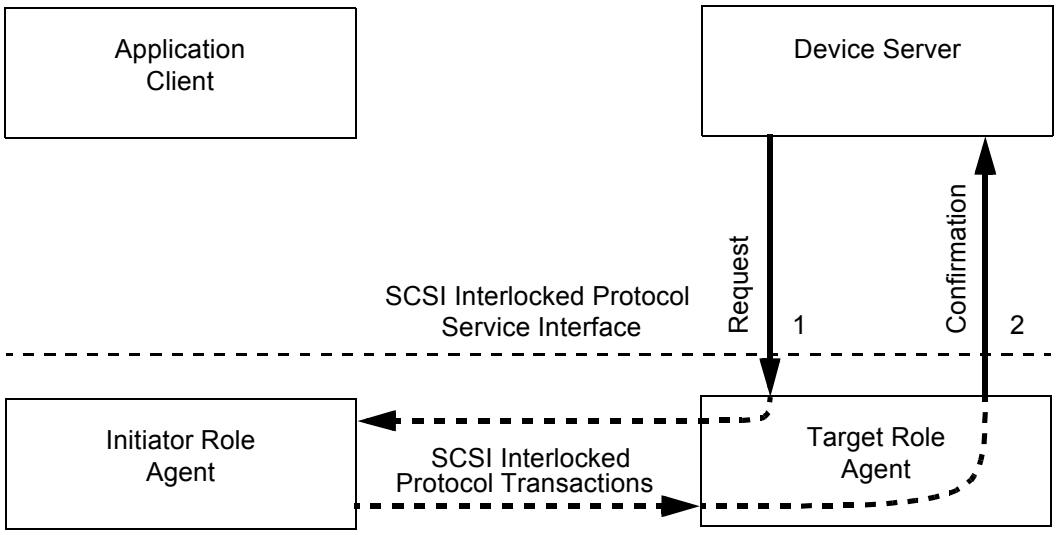


Figure 5 - Model for a two step confirmed services at the SCSI Interlocked Protocol service interface

The SCSI Interlocked Protocol service interface consists of the following interactions:

a) A request to the target role agent to invoke a service;
b) A confirmation from the target role agent upon service completion.

At the SCSI Interlocked Protocol service interface, only device servers shall request a two step confirmed service be invoked.

Figure 6 shows the service and protocol interactions for an unconfirmed service.

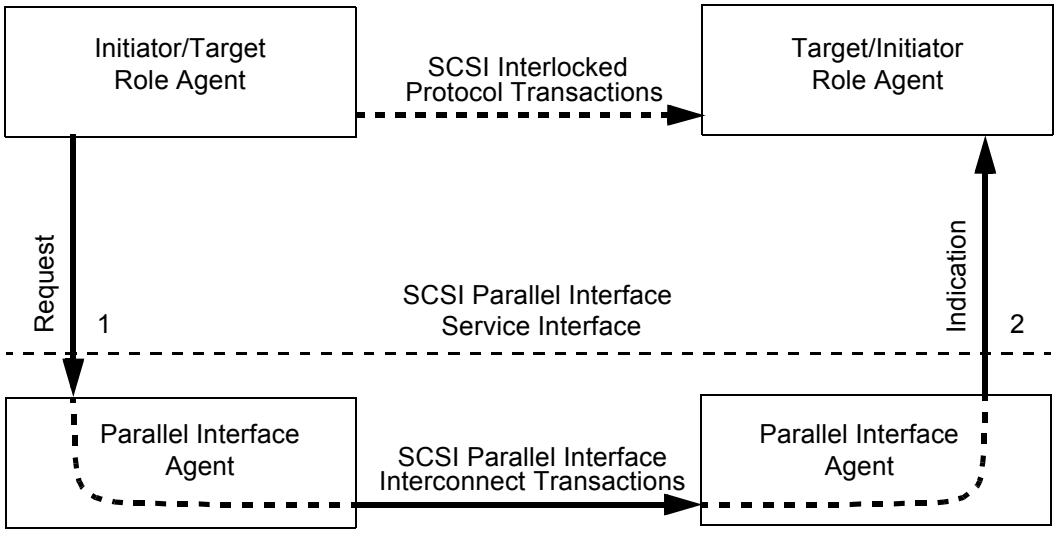


Figure 6 - Model for unconfirmed services at the SCSI parallel interface service interface

An unconfirmed service at the SCSI parallel interface service interface consists of the following interactions:

a) A request to the parallel interface agent, invoking an SCSI parallel interface service;
b) An indication from the parallel interface agent notifying all target role agents and initiator role agents of an event.

This standard uses a RIRC diagram to show the relationships in time between the various service requests. The RIRC diagram shows the sequence of service requests generated by the target or initiator role agents to a parallel interface agent in response to a service request received from a SCSI application. Any activities carried out by the initiator role agent or the target role agent relating to a service request from an application client or device server are enclosed by a — · — · — · — or a — · · — · · — · · —. The enclosed area representing the service is pointed to by a ⟶ within the application client or device server columns. All service requests generated by the initiator role agent or the target role agent as a result of requests from the application client or device server are shown within the parallel interface agent column and enclosed by a — — — — — — — —. The enclosed area representing the service is pointed to by a ⟶ within the initiator role agent or target role agent columns. Figures 7, 8, and 9 show the generic form of the RIRC diagram used by this standard.

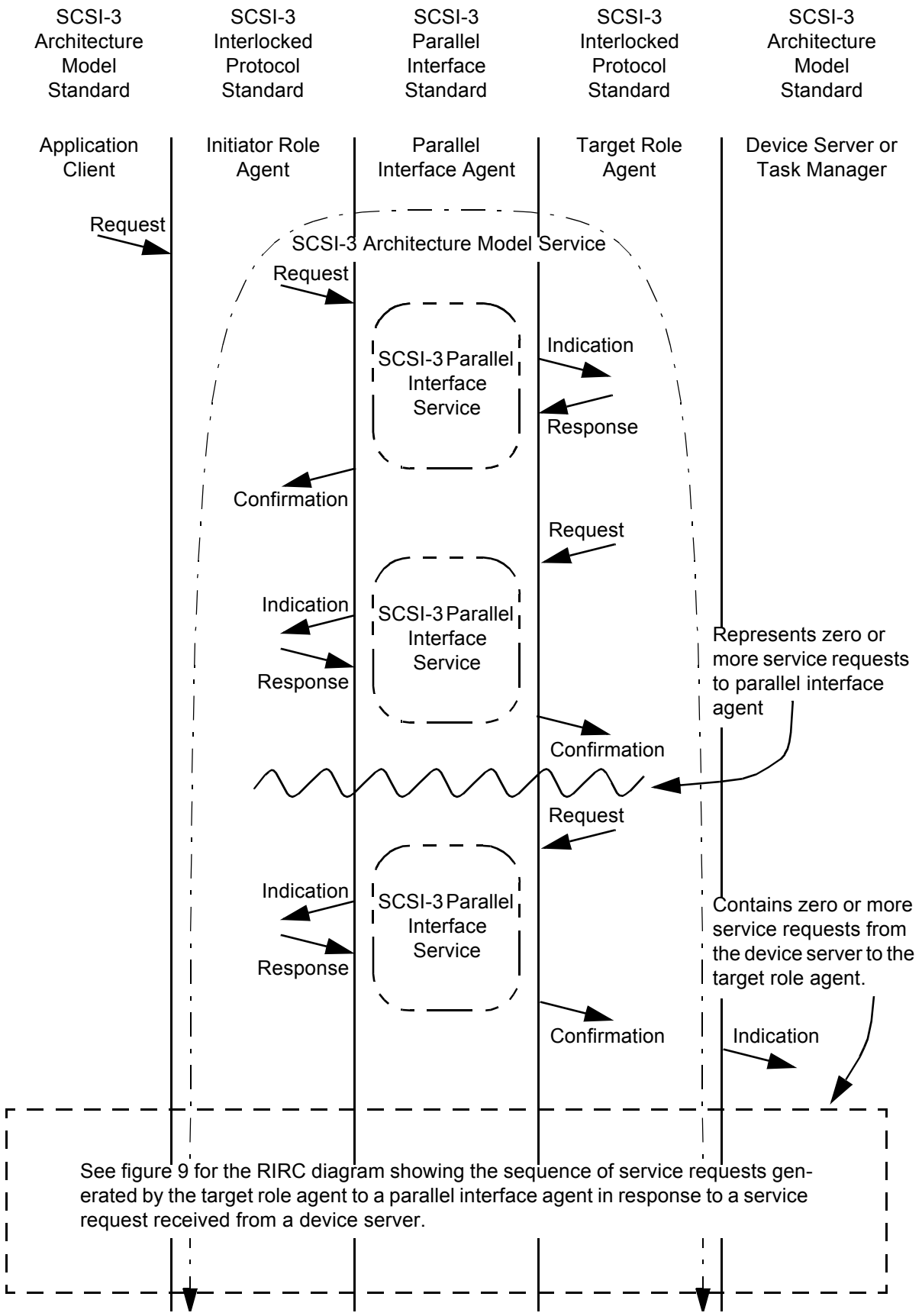


Figure 7 - SCSI Interlocked Protocol service request(s) sequencing (four step confirmed) (Part 1)

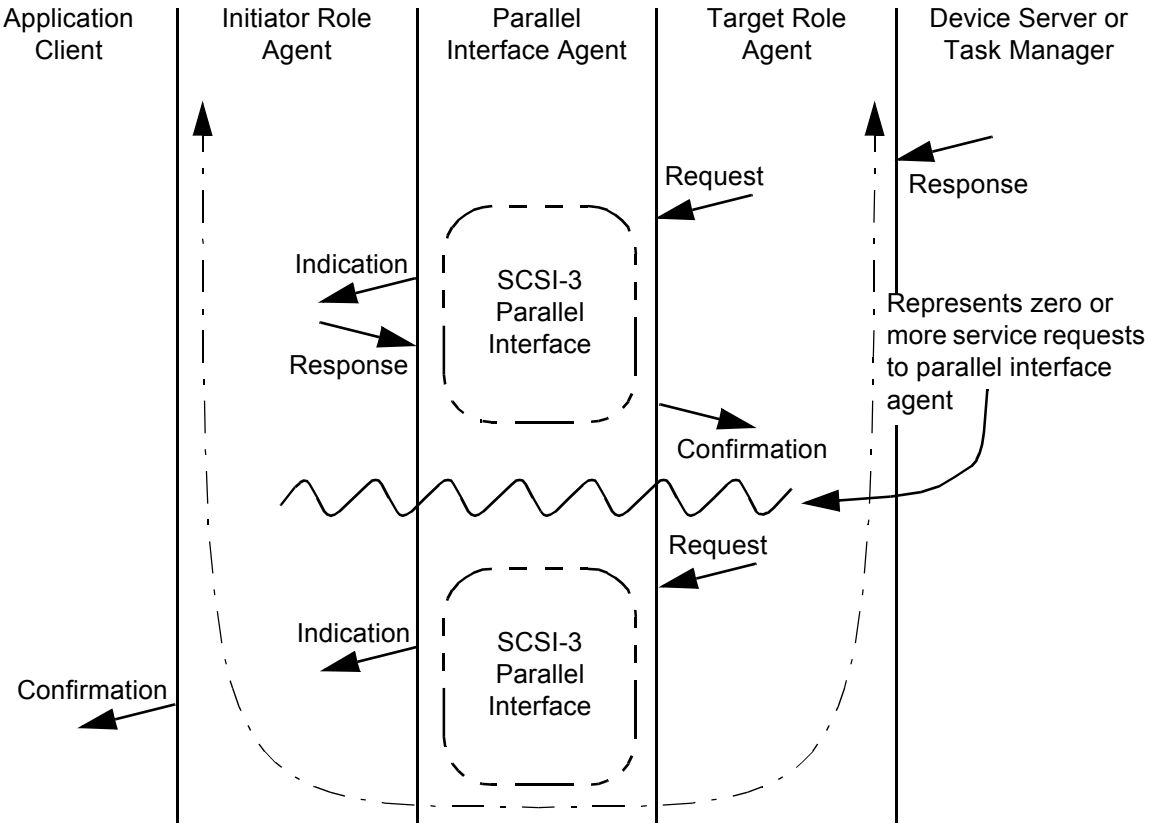


Figure 8 - SCSI Interlocked Protocol service request(s) sequencing (four step confirmed) (Part 1 continued)

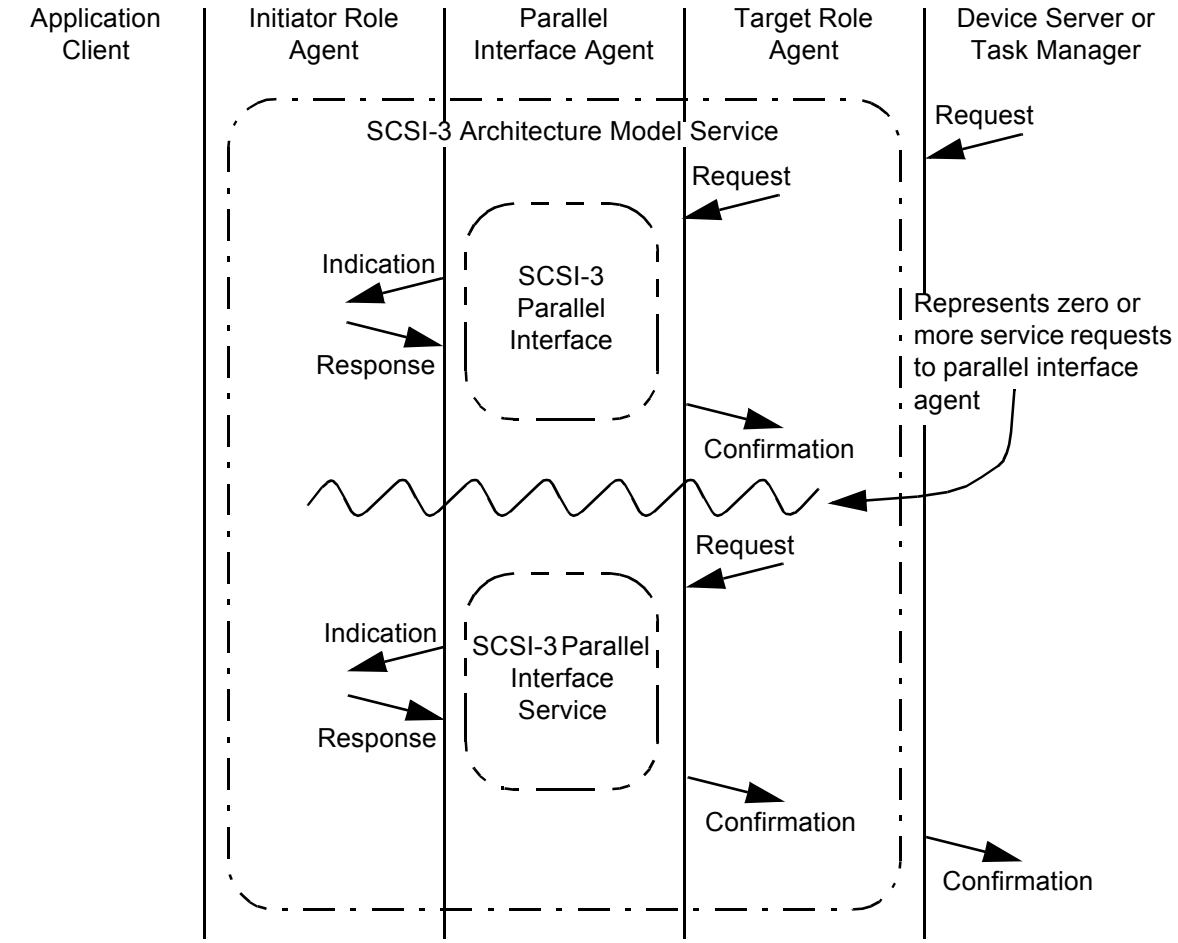


Figure 9 - SCSI Interlocked Protocol service requests sequencing (two step confirmed) (Part 2)

Subclause 5.1 describes the services provided by the SCSI Interlocked Protocol to an application client.

Subclause 5.2 describes the services provided by the SCSI Interlocked Protocol to a device server.

Clause 6 describes the services provided by the SCSI Interlocked Protocol to the parallel interface agent.

5 SCSI Interlocked Protocol services

SCSI Interlocked Protocol services are provided by the initiator role agent enabling the application client to accomplish tasks and task management functions (see SCSI-3 Architecture Model Standard) and by the target role agent enabling the device server to receive commands and move data to/from an application client. The SCSI Interlocked Protocol services are described in terms of the services the initiator role agent and target role agent provide. Each SCSI Interlocked Protocol service causes a sequence of SCSI parallel interface service requests to be sent to a parallel interface agent.

The services requested from the target role agent to the parallel interface agent may be issued in more sequences than shown in the following subclauses. For example those clauses do not consider retry sequences or error scenarios. Figure 35 shows all the valid service request sequences.

5.0.1 Procedure terms

See table 1 for the mapping of the procedure terms used in the SCSI-3 Interlocked Protocol Standard to the equivalent procedure terms used in the SCSI-3 Architecture Model Standard and the SCSI-3 Parallel Interface Standard.

Table 1 - SCSI-3 Interlocked Protocol Standard terms mapped to terms from other SCSI-3 standards

SCSI-3 Interlocked Protocol Standard terms	Equivalent SCSI-3 Architecture Model Standard terms	Equivalent SCSI-3 Parallel Interface Standard terms
initiator identifier	initiator identifier	SCSI ID
target identifier	target identifier	SCSI ID
initiator identifier+target identifier+logical unit number[+tag]	task identifier	none
target identifier+logical unit number[+tag]	task address	none
target identifier | target identifier+logical unit number | initiator identifier+target identifier+logical unit number+tag	object identifier	none
target identifier | target identifier+logical unit number | target identifier+logical unit number+tag	object address	none
target identifier+initiator identifier	none	selection IDs | reselection IDs

See table 2 for a list of the procedure terms used when passing services across the SCSI Interlocked Protocol service interface. See table 2 for the definitions of the SCSI-3 Interlocked Protocol Standard names and the equivalent SCSI-3 Architecture Model Standard names of the procedure terms, the name of the standard where the terms are defined, the standard where the binary contents of the terms are defined, and the routing of the terms

.

Table 2 - Procedure terms

SCSI-3 Interlocked Protocol Standard terms	Standard where term defined	Standard where binary contents of term defined	Term routing
application client buffer offset	SAM	SAM	DS ► TRA► IRA
command byte count	SAM	SAM	AC ► IRA
command descriptor block	SAM	SAM/cmd (Note 2)	AC ► IRA ► TRA ► DS
data-in buffer	SAM	cmd (Note 3)	DS ► TRA► IRA ► AC
data-out buffer	SAM	cmd (Note 3)	AC ► IRA ► TRA ► DS
device server buffer	SAM	cmd (Note 3)	DS ► TRA► IRA
initiator identifier	SAM	SPI	DS ► TRA | TM ► TRA
link control function	SIP	SIP	AC ► IRA ► TRA
logical unit number	SAM	SIP	AC ► IRA ► TRA ► DS | AC ► IRA ► TRA ► TM | DS ► TRA► IRA
request byte count	SAM	SAM	DS ► TRA
service response	SAM	SIP (Note 4)	DS ► TRA► IRA ► AC | TRA ► DS
service response (Note 1)	SAM	SIP (Note 4)	IRA ► AC
status	SAM	SAM	DS ► TRA► IRA ► AC
tag	SAM	SIP	AC ► IRA ► TRA ► DS | AC ► IRA ► TRA ► TM | DS ► TRA► IRA
target identifier	SAM	SPI	AC ► IRA ► TRA ► DS | AC ► IRA ► TRA ► TM | DS ► TRA
target identifier+initiator identifier	SIP	SPI	TRA ► DS | TRA ► TM
task attribute	SAM	SIP	AC ► IRA ► TRA ► DS

Key: AC=application client, cmd=SCSI command standards, DS=device server, IRA=initiator role agent, SAM=SCSI-3 Architecture Model Standard, SIP=SCSI-3 Interlocked Protocol Standard, SPI=SCSI-3 Parallel Interface Standard, TM=task manager, TRA=target role agent

Notes

1 Only occurs when unexpected bus free (see 9.5) is detected by the initiator role agent.

2 The portions not defined in the SCSI-3 Architecture Model Standard are defined in the SCSI command standards (e.g.,SCSI-3 Block Commands Standard, SCSI-3 Primary Commands Standard).

3 Parameter lists are defined within one of the SCSI command standards (e.g.,SCSI-3 Block Commands Standard, SCSI-3 Primary Commands Standard). SCSI standards do not define non-parameter list information.

4 The SERVICE DELIVERY OR TARGET FAILURE value of the service response is not defined in SCSI-3.

5.1 Application client SCSI command services

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

> Service response = execute command (target identifier+logical unit number[+tag], command descriptor block, [task attribute], [link control function], [data-out buffer], [command byte count] || [data-in buffer], status, service response).

5.1.1 Send SCSI command service

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

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

Table 3 - Processing of send SCSI command service procedure

Step	Protocol service name	SCSI Interlocked Protocol Service Interface procedure call
request	send SCSI command request	send SCSI command (target identifier+logical unit number[+tag], command descriptor block, [task attribute], [link control function], [data-out buffer], [command byte count] ||).
indication	send SCSI command indication	SCSI command received (target identifier+initiator identifier+logical unit number[+tag], [command descriptor block], [task attribute], ||).
response	send SCSI command response	send command complete (target identifier+initiator identifier+logical unit number[+tag], [status], [service response], ||).
confirmation	send SCSI command confirmation	command complete received (target identifier+logical unit number[+tag], [data-in buffer], [status], service response ||).

5.1.1.1 Send SCSI command request

The application client shall issue a send SCSI command request as follows:

> send SCSI command (target identifier+logical unit number[+tag], command descriptor block, [task attribute], [link control function], [data-out buffer], [command byte count] ||).

The link control function (see 8.2) shall be sent to the target role agent as part of the message out service shown in figure 11.

When a send SCSI command request is received and a bus free is indicated the initiator role agent shall issue a selection service request (see figure 10) to the parallel interface agent. The selection service request shall contain the target identifier from the send SCSI command service request.

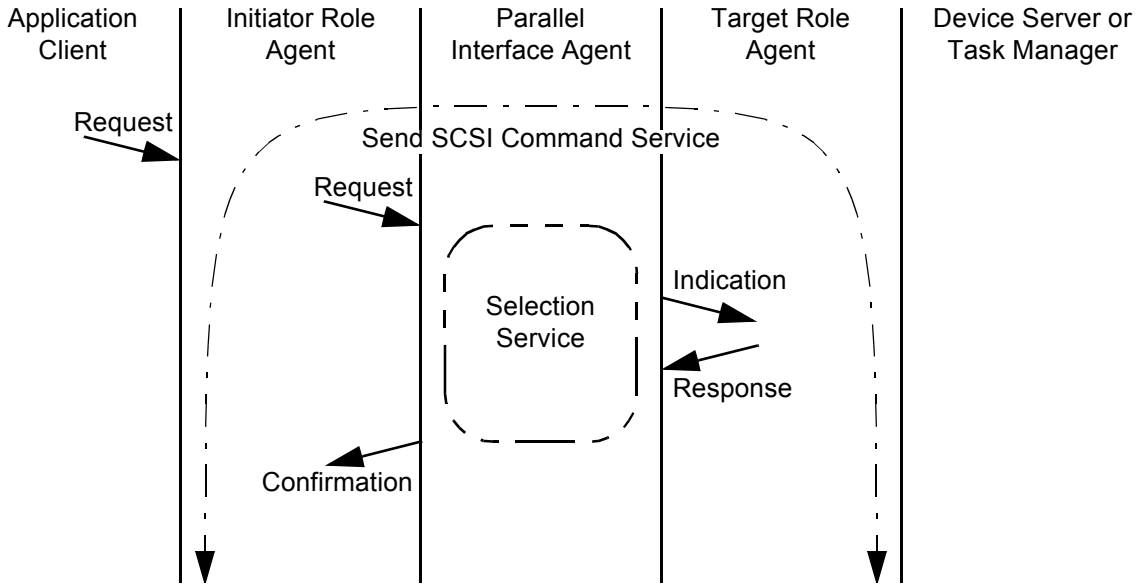


Figure 10 - Send SCSI command service request sequencing

The data-out delivery service (see 5.2.2) moves data from the data-out buffer to a device server buffer.

5.1.1.2 Send SCSI command indication

After confirmation from the parallel interface agent to the target role agent that a selection service was successful the target role agent shall issue the sequence of service requests shown in figure 35.

If the selection service was not successful the initiator role agent shall issue a confirmation to the application client with a SERVICE DELIVERY OR TARGET FAILURE value in the service response parameter of the command complete received procedure call. The initiator role agent shall not place any values in the status parameter.

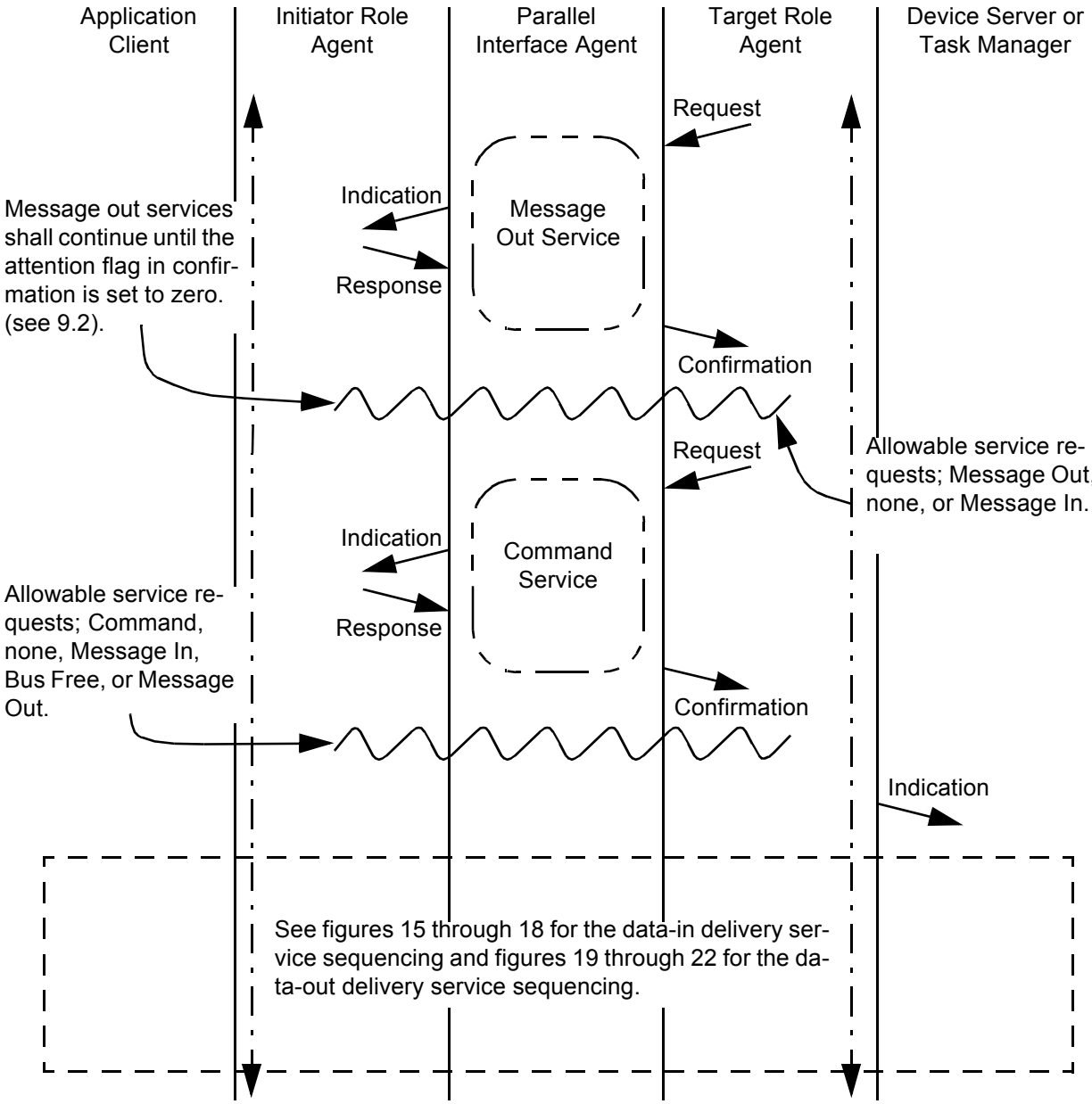


Figure 11 - Send SCSI command service indication sequencing

After the completion of the above sequence of services a send SCSI command indication is generated to the device server. The send SCSI command indication follows:

> SCSI command received (target identifier+initiator identifier+logical unit number[+tag], [command descriptor block], [task attribute], ||).

If there is a service delivery failure and the target role agent cannot recover or the target role agent receives a reset indication the target role agent shall issue a bus free service. An initiator role agent shall consider this an unexpected bus free (see 9.5) unless it issued the bus device reset or a DISCONNECT message. See 5.1.1.4 for information on how the initiator role agent handles unexpected bus free indications.

If the device server requires data or parameter list information to be transferred from the application client it shall issue a data-out delivery service request (see 5.2.2) to the target role agent. If the device server

requires data or parameter list information to be transferred to the application client it shall issue a data-in delivery service request (see 5.2.1) to the target role agent.

5.1.1.3 Send SCSI command response

After completion of any data-out delivery service or data-in delivery service requests a send SCSI command response is generated by the device server to the target role agent. The send SCSI command response follows:

> send command complete (target identifier+initiator identifier+logical unit number[+tag], [status], [service response], ||).

A service response of SERVICE DELIVERY OR TARGET FAILURE indicates there is no status parameter. Any other service response indicates there is a valid value in the status parameter.

After receiving a send SCSI command response with a valid status from the device server, if the target role agent determines a reselection is not required, then it shall issue the sequence of service requests shown in figure 12. After receiving a send SCSI command response with a service response of SERVICE DELIVERY OR TARGET FAILURE from the device server, if the target role agent determines a reselection is required, then it shall issue the sequence of service requests shown in figure 13.

> NOTE 1 - Several factors control whether a target role agent can or must disconnect. These factors include information in the IDENTIFY message parameter data passed in some commands and other vendor specific factors.

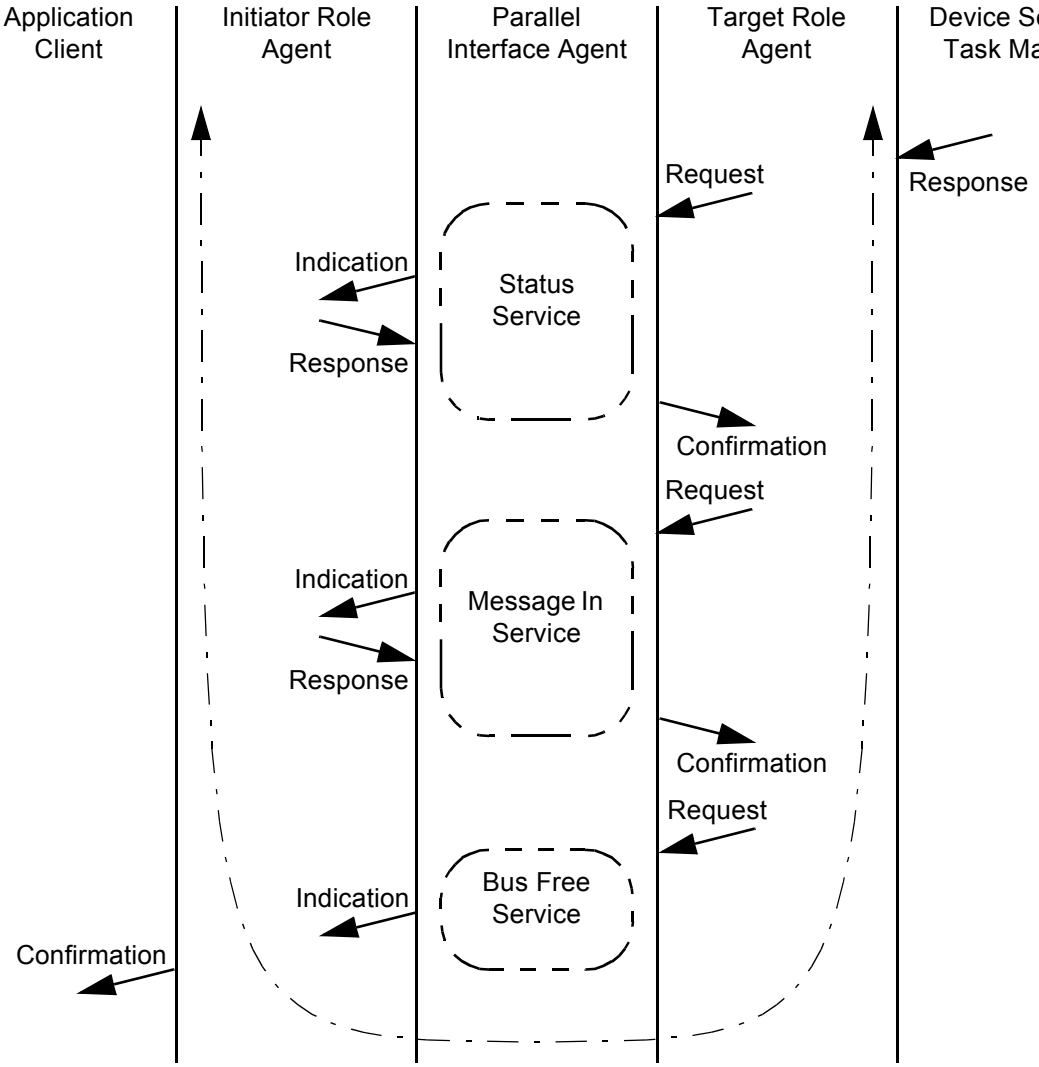


Figure 12 - Send SCSI command service response sequencing (without reselection/with status)

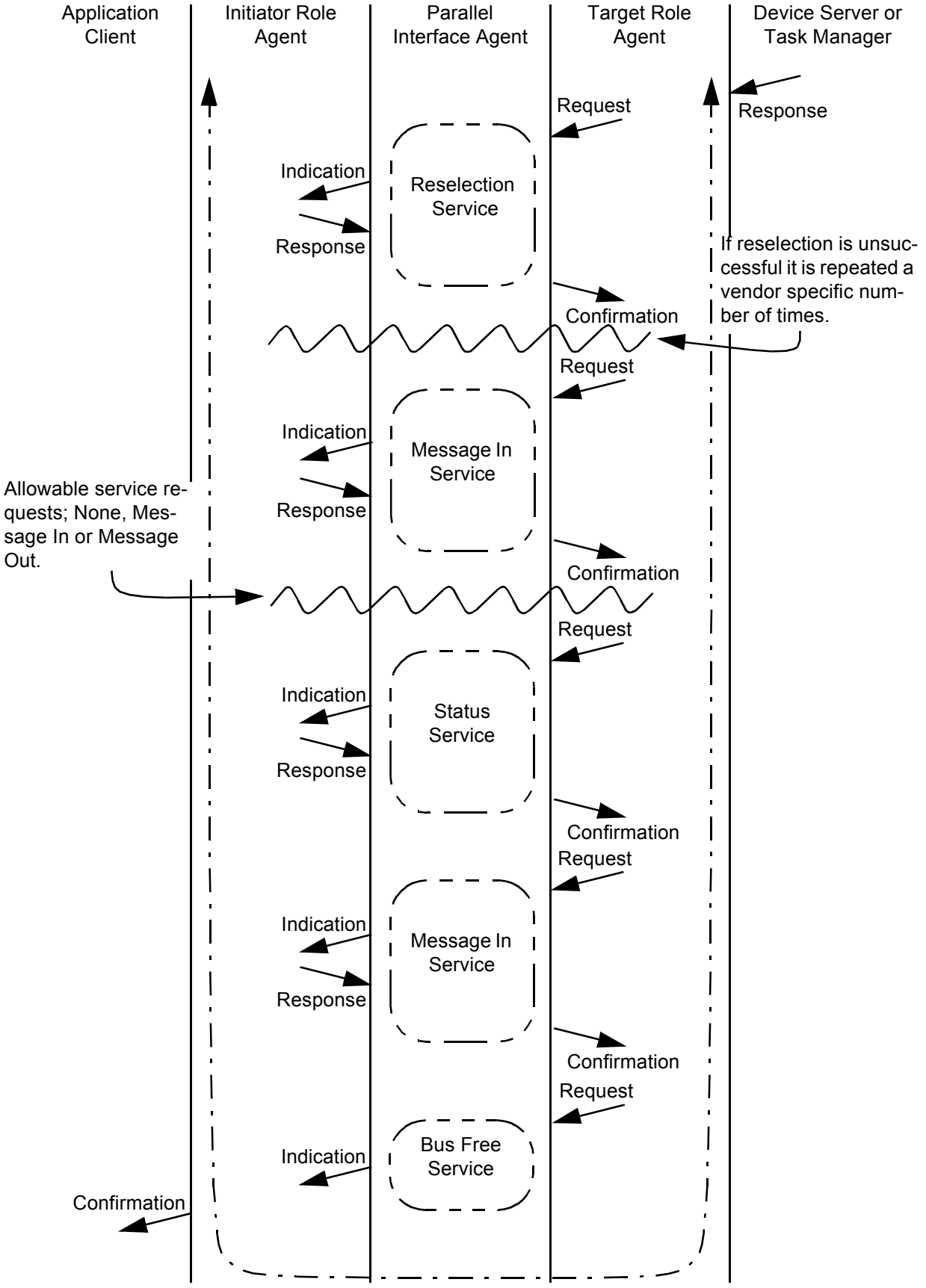


Figure 13 - Send SCSI command service response sequencing (with reselection/with status)

After receiving a send SCSI command response with the bus free service request flag set to one from the device server the target role agent shall issue the sequence of service requests shown in figure 14.

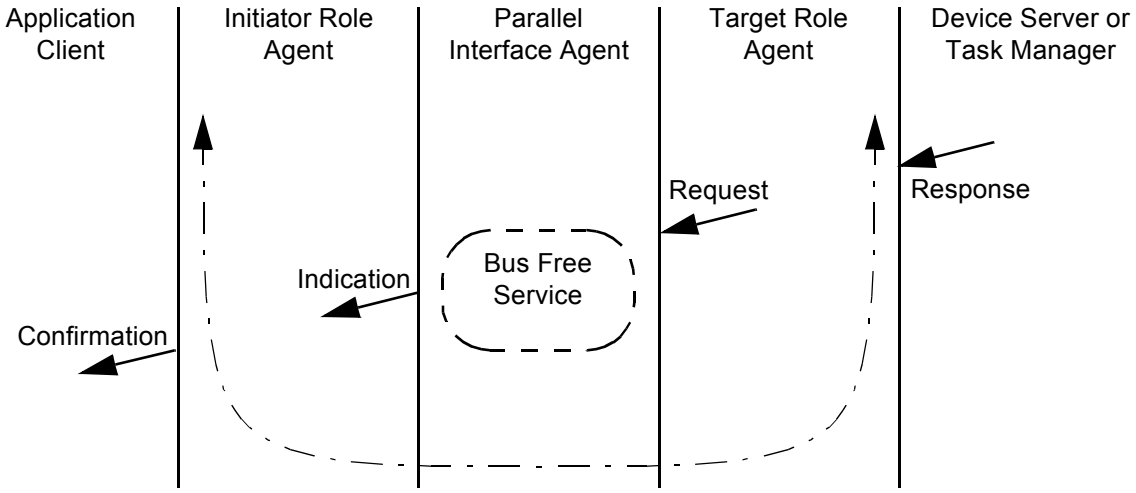


Figure 14 - Send SCSI command service response sequencing (without status)

5.1.1.4 Send SCSI command confirmation

After receipt of a bus free indication to the initiator role agent from the parallel interface agent a send SCSI command confirmation is generated by the initiator role agent to the application client. If the bus free indication is preceded by a message in service confirmation then the initiator role agent shall place the contents of the message in message into the service response parameter.

If the bus free indication is an unexpected bus free the initiator role agent shall place the SERVICE DELIVERY OR TARGET FAILURE value in the service response parameter. In this case, the initiator role agent shall not place any values in the status parameter.

> NOTE 2 - When an application client receives a service response of SERVICE DELIVERY OR TARGET FAILURE it is recommended that a REQUEST SENSE command be attempted to obtain any valid sense data.

The send SCSI command confirmation follows:

> command complete received (target identifier+logical unit number[+tag], [data-in buffer], [status], service response ||).

The initiator role agent shall not issue any service requests to the parallel interface agent to carry out the send SCSI command confirmation. The data-in delivery service (see 5.2.1) moves data from device server buffer to data-in buffer.

5.2 Device server SCSI command services

The SCSI data buffer movement services shall be requested from the device server using a procedure call defined as:

> Service response = move data buffer (target identifier+initiator identifier+logical unit number [+tag], device server buffer, application client buffer offset, request byte count ||).

Only one type of data buffer movement procedure call shall be used while processing one command, either data-in delivery or data-out delivery.

5.2.1 Data-in delivery service

The data-in delivery service is a two step confirmed service that provides the means to transfer a parameter list or data from a device server to an initiator role agent.

Processing the execute command procedure call for a data-in delivery service shall be composed of the 2 step confirmed service shown in table 4

Table 4 - Processing of data-in delivery service procedure

Step	Protocol service name	SCSI Interlocked Protocol Service Interface procedure call
request	data-in delivery request	send data-in (target identifier+initiator identifier+logical unit number[+tag], device server buffer, application client buffer offset, request byte count ||).
confirmation	data-in delivery confirmation	data delivered (target identifier+initiator identifier+logical unit number[+tag], service response ||).

5.2.1.1 Data-in delivery request

The device server shall issue a data-in delivery request as follows:

> send data-in (target identifier+initiator identifier+logical unit number[+tag], device server buffer, application client buffer offset, request byte count ||).

After receiving a data-in delivery request the target role agent determines if a reselection service is required. If a reselection service is not required, and no disconnect is required, the target role agent shall issue the sequence of service requests shown in figure 15. If a reselection service is required and no disconnect is required the target role agent shall issue the sequence of service requests shown in figure 16. If a reselection service is not required and a disconnect is required the target role agent shall issue the sequence of service requests shown in figure 17. If a reselection service is required and a disconnect is required the target role agent shall issue the sequence of service requests shown in figure 18.

.

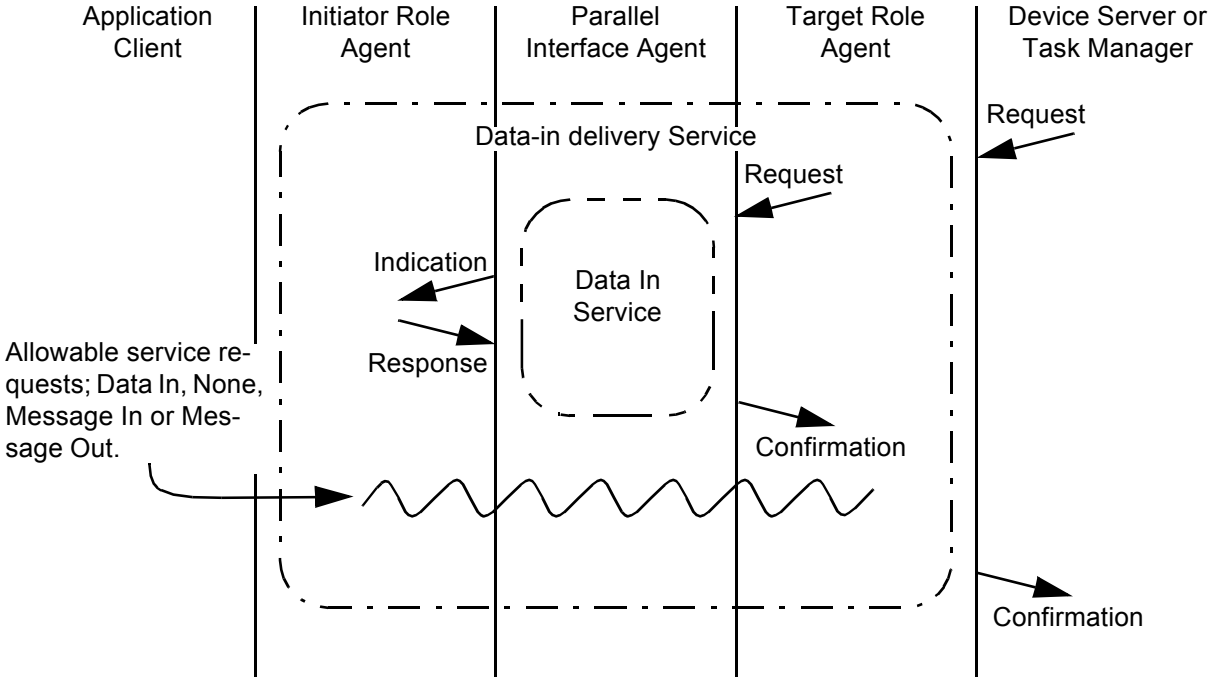


Figure 15 - Data-in delivery service request sequencing (without reselection/without disconnection)

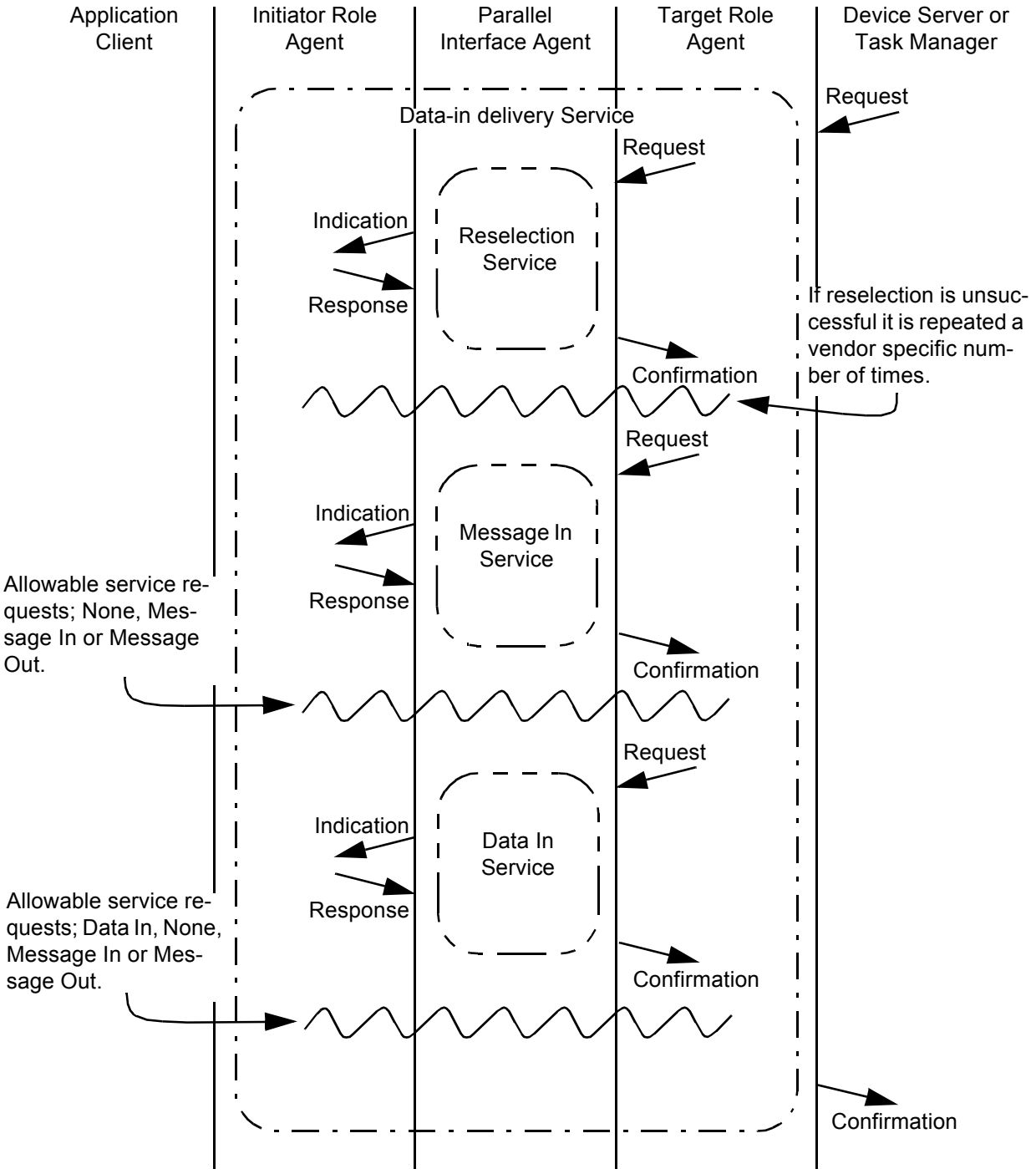


Figure 16 - Data-in delivery service request sequencing (with reselection/without disconnection)

.

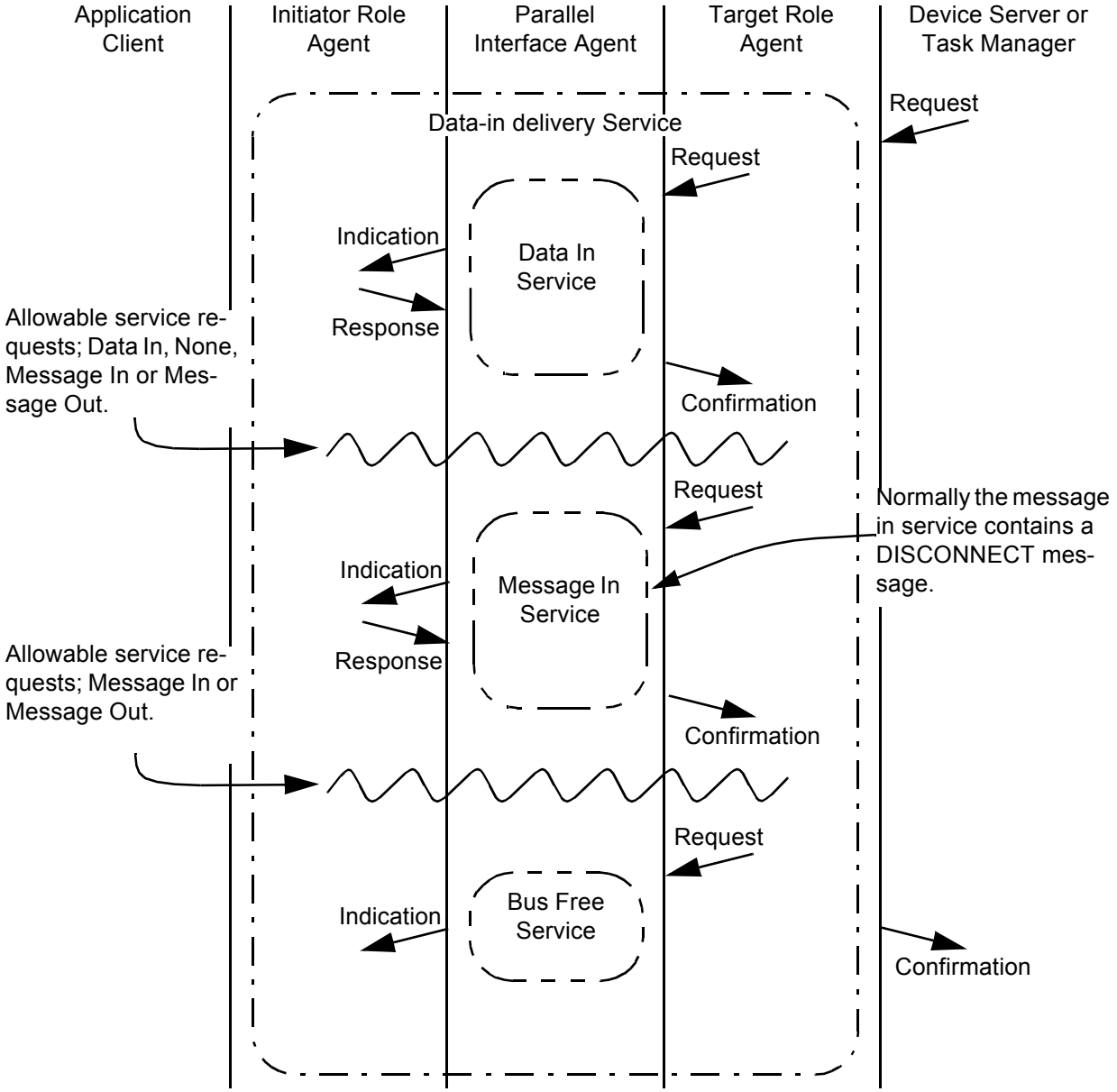


Figure 17 - Data-in delivery service request sequencing (without reselection/with disconnection)

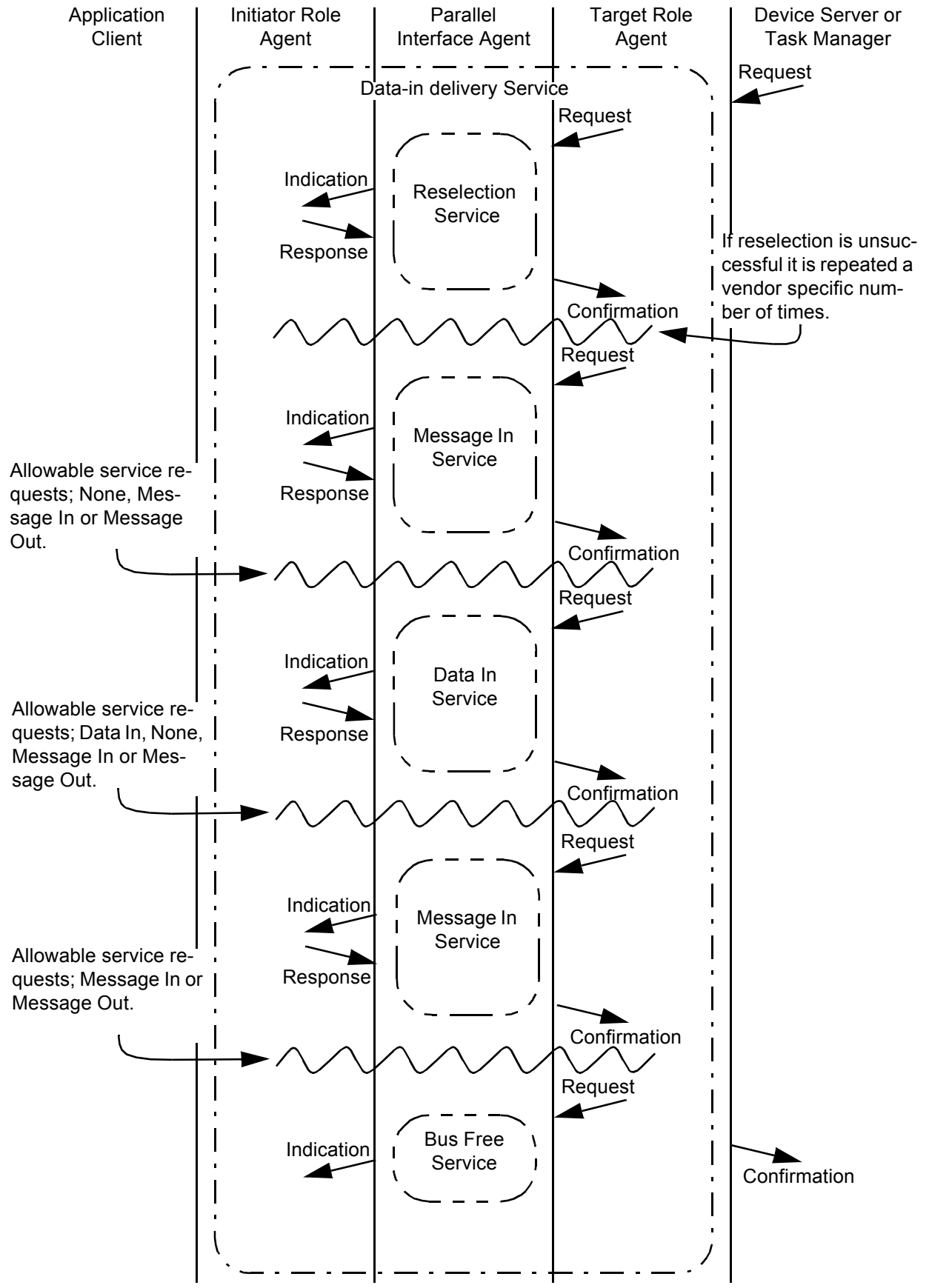


Figure 18 - Data-in delivery service request sequencing (with reselection/with disconnection)

5.2.1.2 Data-in delivery confirmation

After confirmation from the parallel interface agent that the last byte of information for the data-in delivery service request has been transferred the target role agent shall notify the device server. The data-in delivery confirmation follows:

data delivered (target identifier+initiator identifier+logical unit number[+tag], service response ||).

The service response parameter is set to DATA TRANSFER COMPLETE by the target role agent to indicate to the device server the successful completion of the data-in delivery service request sequence. The service response parameter is set to SERVICE DELIVERY OR TARGET FAILURE by the target role agent to indicate to the device server the data-in delivery service request sequence did not successfully complete (i.e., reset indication, repeating parity error, reselection failure, etc.).

The target role agent shall respond to a service delivery failure by issuing a bus free service.

The target role agent shall not issue any service requests to the parallel interface agent to carry out the data-in delivery confirmation.

5.2.2 Data-out delivery service

The data-out delivery service is a two step confirmed service that provides the means to transfer a parameter list or data from an initiator role agent to a device server.

Processing the execute command procedure call for a data-out delivery service shall be composed of the 2 step confirmed service shown in table 5..

Table 5 - Processing of data-out delivery service procedure

Step	Protocol service name	SCSI Interlocked Protocol Service Interface procedure call
request	data-out delivery request	receive data-out (target identifier+initiator identifier+logical unit number[+tag], application client buffer offset, request byte count, device server buffer ||).
confirmation	data-out delivery confirmation	data-out received (target identifier+initiator identifier+logical unit number [+tag] service response ||)

5.2.2.1 Data-out delivery request

The device server shall issue a data-out delivery request as follows:

receive data-out (target identifier+initiator identifier+logical unit number[+tag], application client buffer offset, request byte count, device server buffer ||).

After receiving a data-out delivery request the target role agent determines if a reselection service is required. If a reselection service is not required and no disconnect is required the target role agent shall issue the sequence of service requests shown in figure 19. If a reselection service is required and no disconnect required the target role agent shall issue the sequence of service requests shown in figure 20. If a reselection service is not required and a disconnect is required the target role agent shall issue the sequence of service requests shown in figure 21. If a reselection service is required and a disconnect is required the target role agent shall issue the sequence of service requests shown in figure 22.

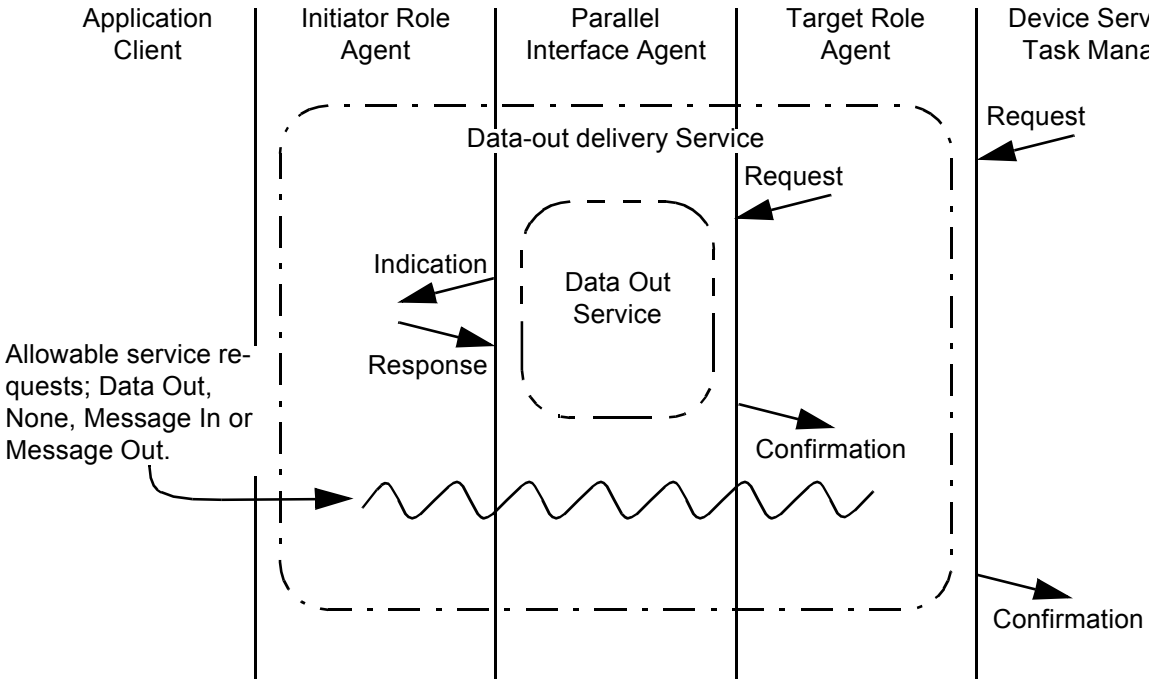


Figure 19 - Data-out delivery service request sequencing (without reselection/without disconnection)

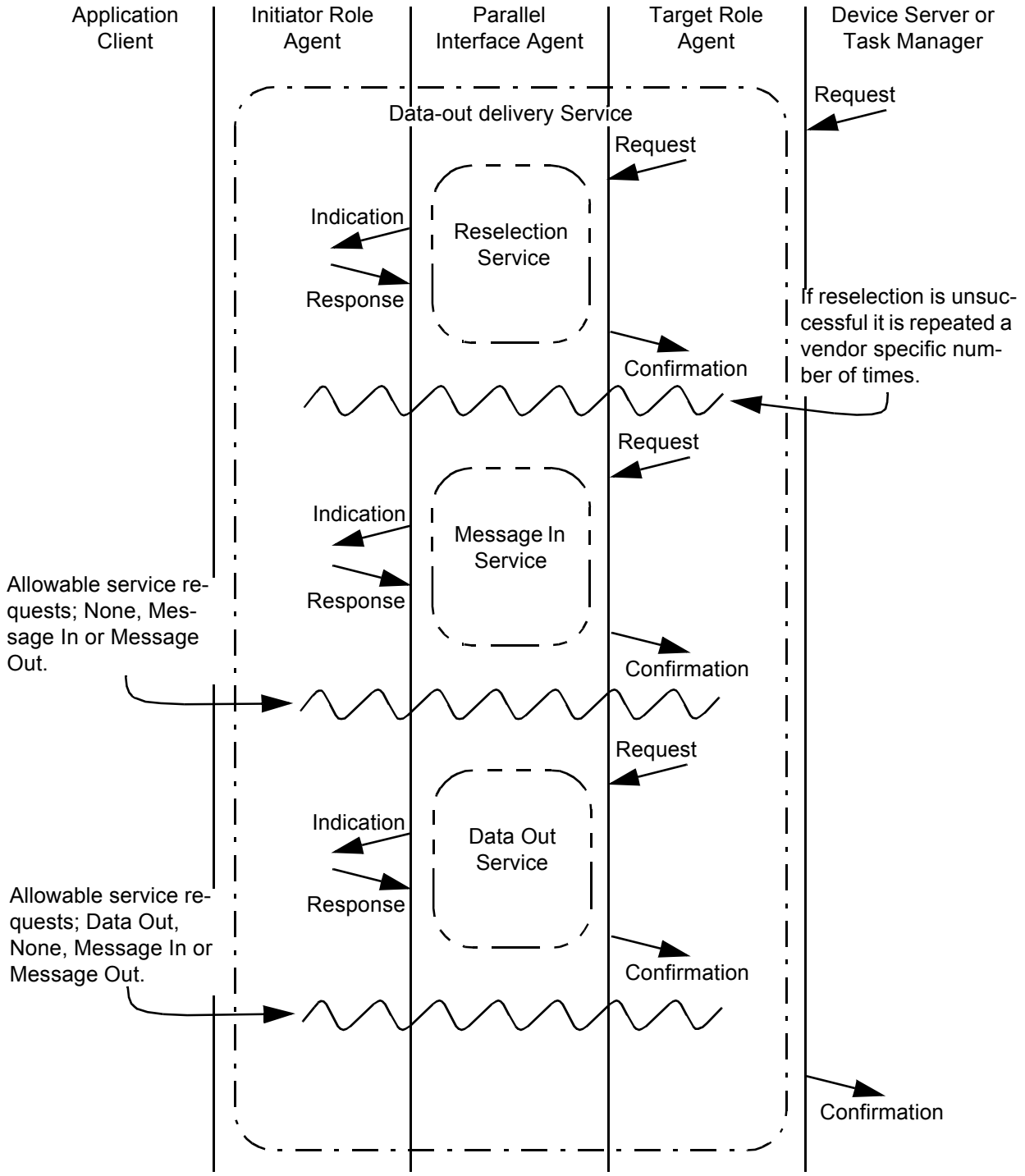


Figure 20 - Data-out delivery service request sequencing (with reselection/without disconnection)

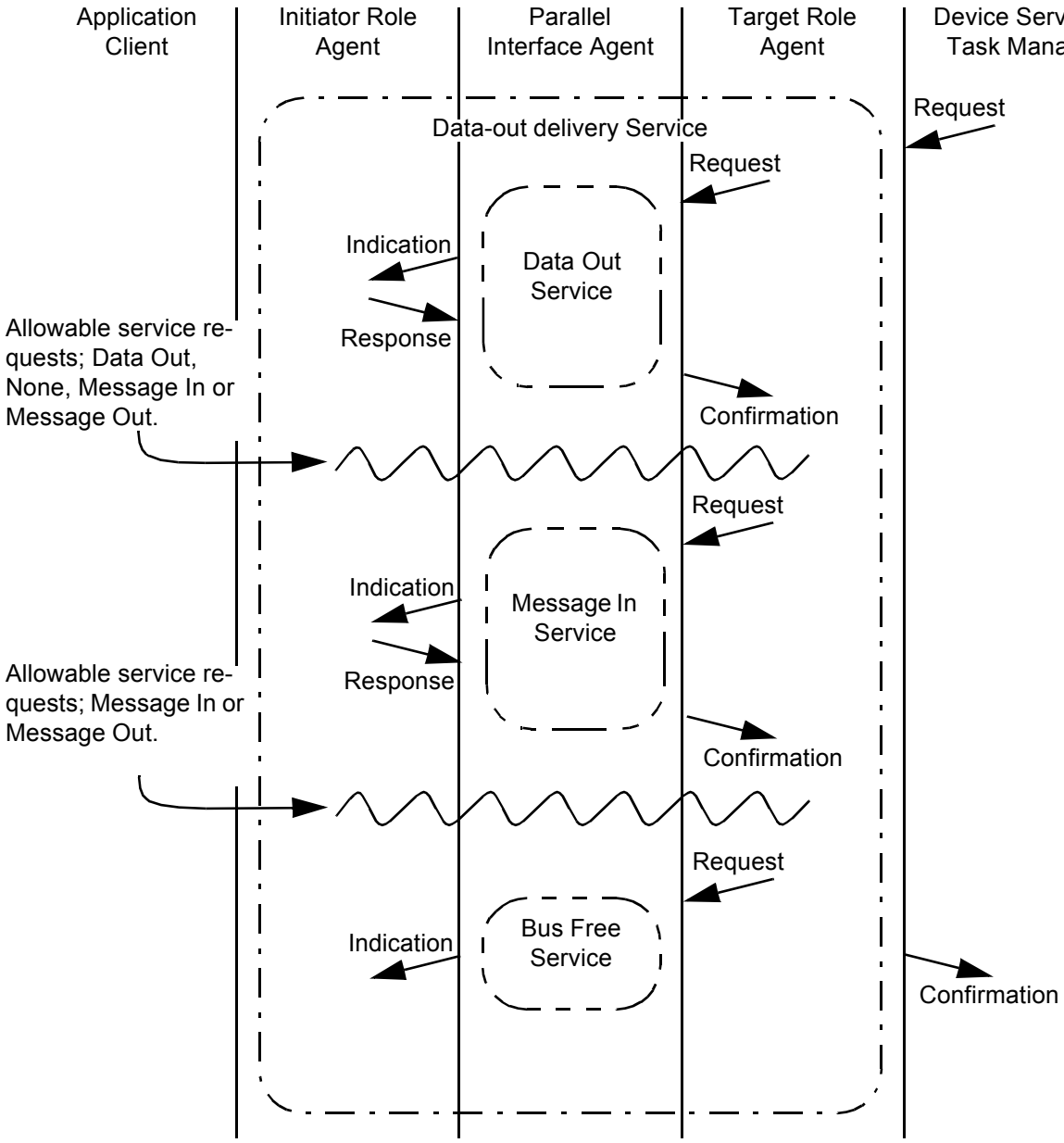


Figure 21 - Data-out delivery service request sequencing (without reselection/with disconnection)

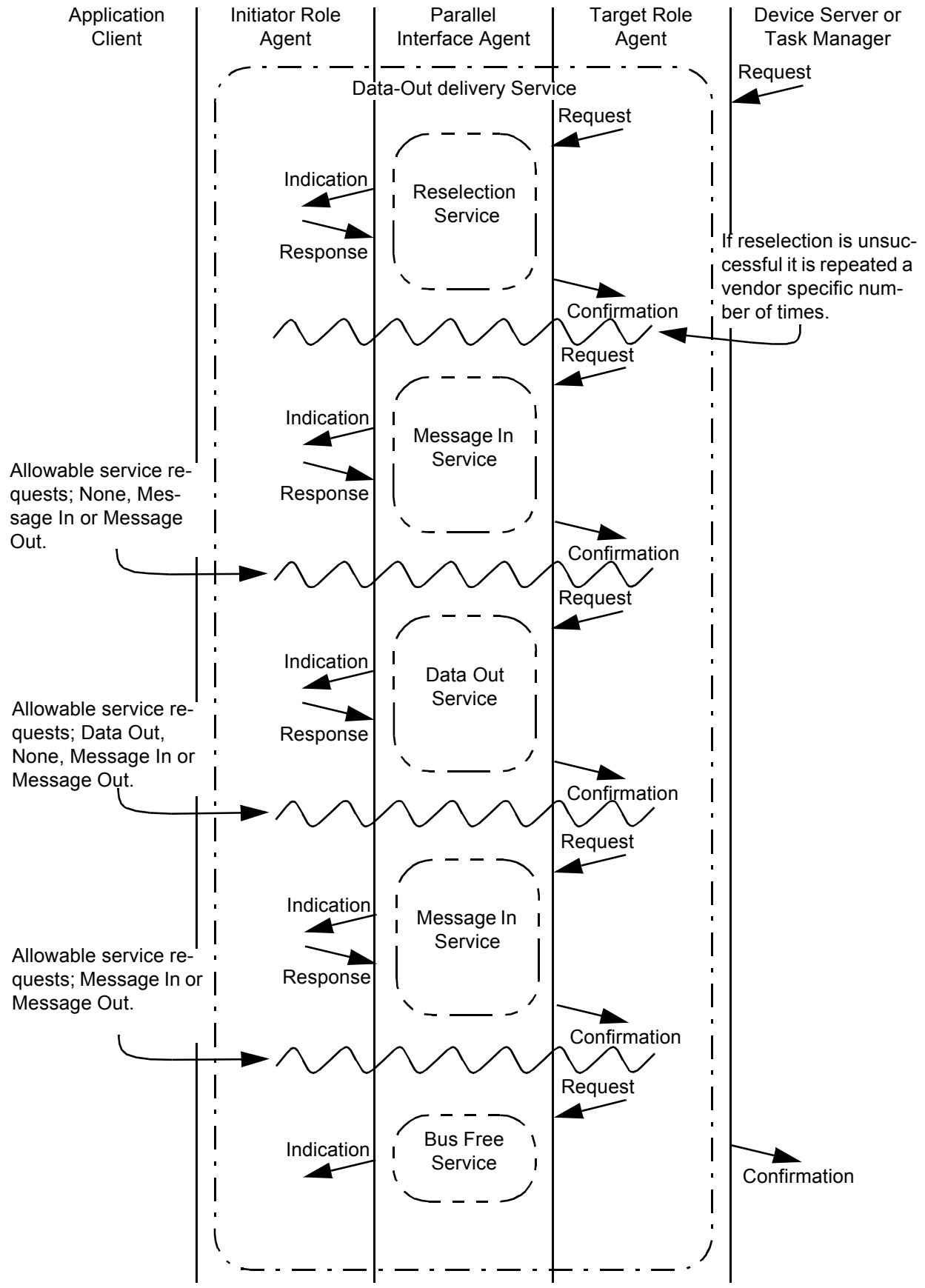


Figure 22 - Data-out delivery service request sequencing (with reselection/with disconnection)

5.2.2.2 Data-out delivery confirmation

After confirmation from the parallel interface agent that the last byte of information for the data-out delivery service request has been transferred the target role agent shall notify the device server. The data-out delivery confirmation follows:

data-out received (target identifier+initiator identifier+logical unit number [+tag] service response ||).

The service response parameter is set to DATA TRANSFER COMPLETE by the target role agent to indicate to the device server the successful completion of the data-out delivery service request sequence. The service response parameter is set to SERVICE DELIVERY OR TARGET FAILURE by the target role agent to indicate to the device server the data-out delivery service request sequence did not successfully complete (i.e., reset indication, repeating parity error, reselection failure, etc.).

The target role agent shall respond to a service delivery failure by issuing a bus free service.

The target role agent shall not issue any service requests to the parallel interface agent to carry out the data-out delivery confirmation.

5.3 Task management services

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

Service response = task management function (target identifier|target identifier+logical unit number|target identifier+logical unit number+tag, service delivery failure flag || service response).

5.3.1 Task management function service

The SCSI-3 Interlocked Protocol Standard handles task management functions in one of two methods depending on the specific function. Both methods handle the task management function service as a four step confirmed service that provides the means to transfer task management functions to a task manager however the sequences of the service requests to the parallel interface agent are different.

Message task management functions are task management functions that require messages to be delivered to a task manager. Non-message task management function is a task management function that causes the initiator role agent to issue a reset service request to the parallel interface agent.

5.3.1.1 Message task management function service sequencing

5.3.1.1.1 Task management function request

The application client shall issue a task management function request as follows:

send task management request (target identifier|target identifier+logical unit number|target identifier+logical unit number+tag ||).

When a task management function request is received the initiator role agent either issues a selection service request (see figure 23) to the parallel interface agent or sets the attention flag in the next response to an indication from a parallel interface agent service (see figure 24). The selection service request shall contain the target identifier.

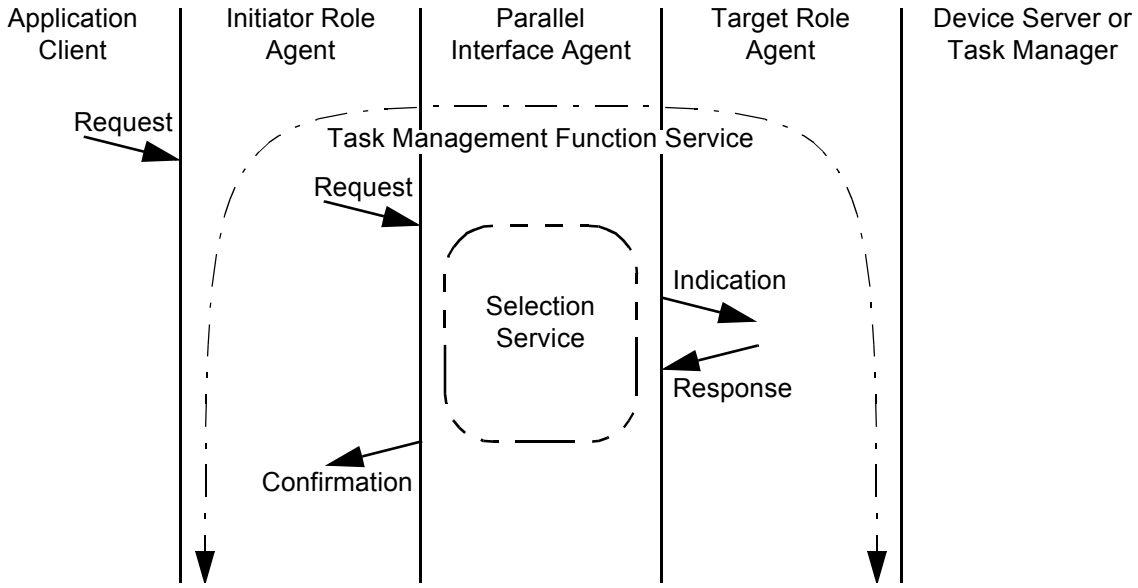


Figure 23 - Task management function service request sequencing (selection option)

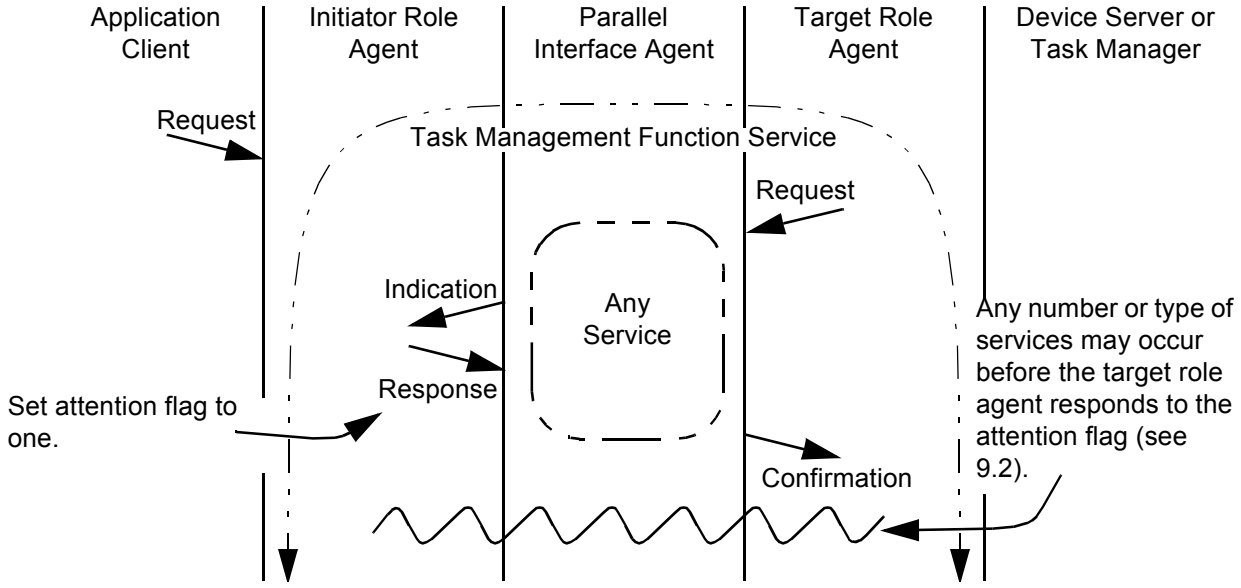


Figure 24 - Task management function service request sequencing (attention option)

5.3.1.1.2 Task management function indication

If the selection option is used then on confirmation from the parallel interface agent to the target role agent that a selection service was successful the target role agent shall issue the service requests shown in figure 25. If the attention option is used the target agent shall issue the service requests shown in figure 26.

If the selection service was not successful the initiator role agent shall issue a confirmation to the application client with a SERVICE DELIVERY OR TARGET FAILURE value in the service response parameter.

After the completion of the sequence of services in figure 25 or figure 26 a task management function indication is generated to the task manager. The task management function indication follows:

task management request received (target identifier|target identifier+logical unit number|target identifier+initiator identifier+logical unit number+tag], ||).

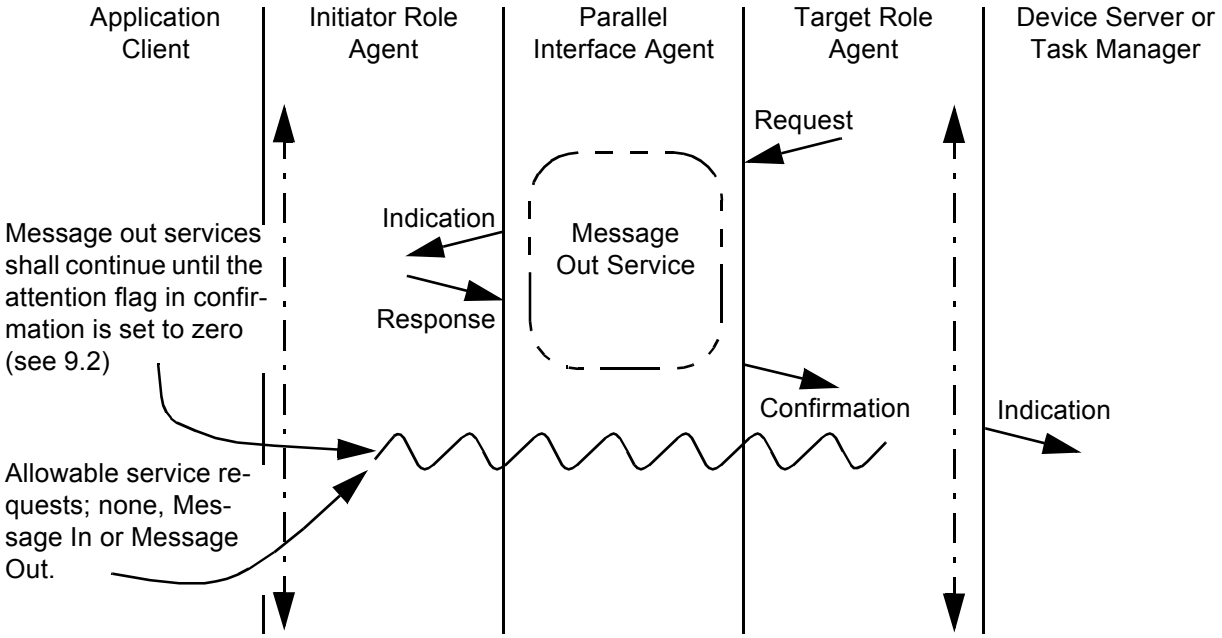


Figure 25 - Task management function service indication sequencing (selection option)

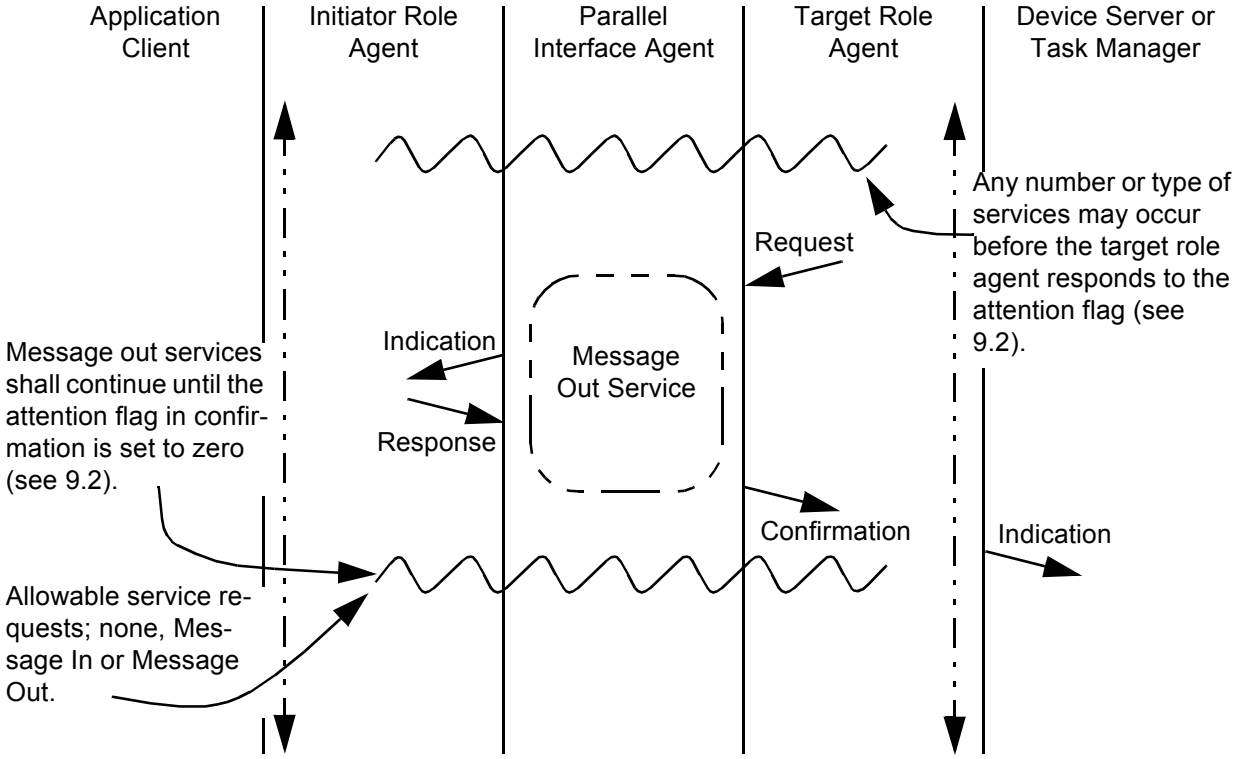


Figure 26 - Task management function service indication sequencing (attention option)

If there is a service delivery failure and the target role agent cannot recover the target role agent shall issue the service requests shown in figure 27. If there is a service delivery failure the target role agent shall not generate an indication to the task manager. After receipt of the indication to the initiator role agent from the parallel interface agent a task management function is generated by the initiator role agent to the application client (see 5.3.1.1.4). The service response shall be set to FUNCTION COMPLETE.

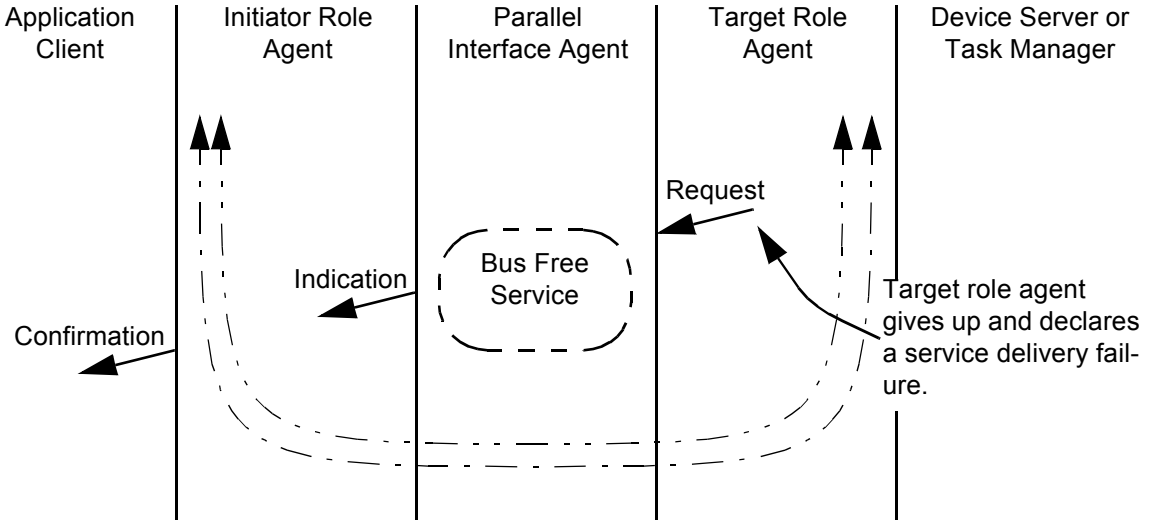


Figure 27 - Task management function service indication sequencing (service delivery failure)

5.3.1.1.3 Task management function response

After the task manager completes the task management function it shall send the following response to the target role agent:

task management function executed (target identifier|target identifier+logical unit number|target identifier+initiator identifier+logical unit number+tag, service response ||).

If the selection option is used then on receiving a task management function response with a service response of function rejected from the task manger the target role agent shall issue the sequence of service requests shown in figure 28. After receiving a task management function response with a service response of function complete from the task manager the target role agent shall issue the sequence of service requests shown in figure 29.

If the attention option is used then on receiving a task management function response with a service response of function rejected from the task manger the target role agent shall issue the sequence of service requests shown in figure 30. After receiving a task management function response with a service response of function complete from the task manager the target role agent shall issue the sequence of service requests shown in figure 31.

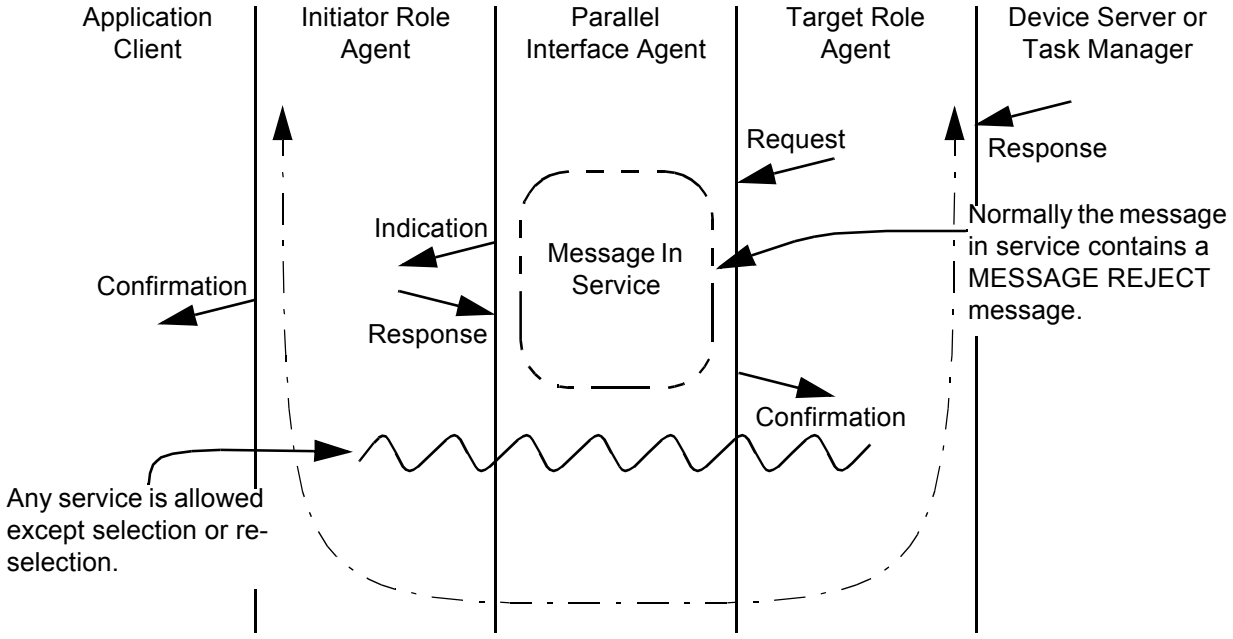


Figure 28 - Task management function service response sequencing (with message in) (selection option)

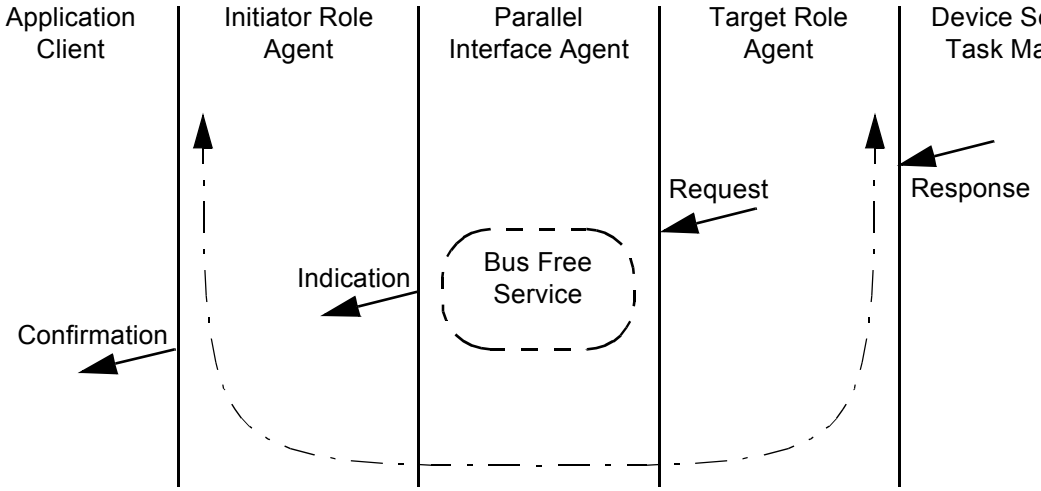


Figure 29 - Task management function service response sequencing (without message in) (selection option)

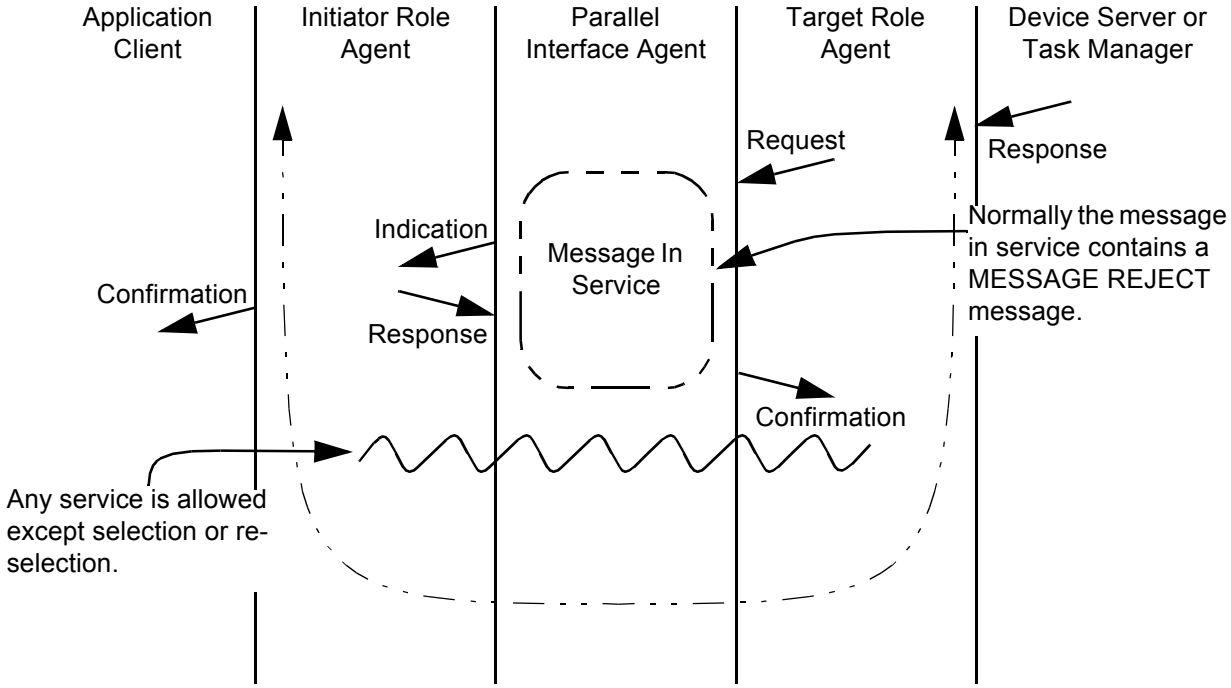


Figure 30 - Task management function service response sequencing (with message in) (attention option)

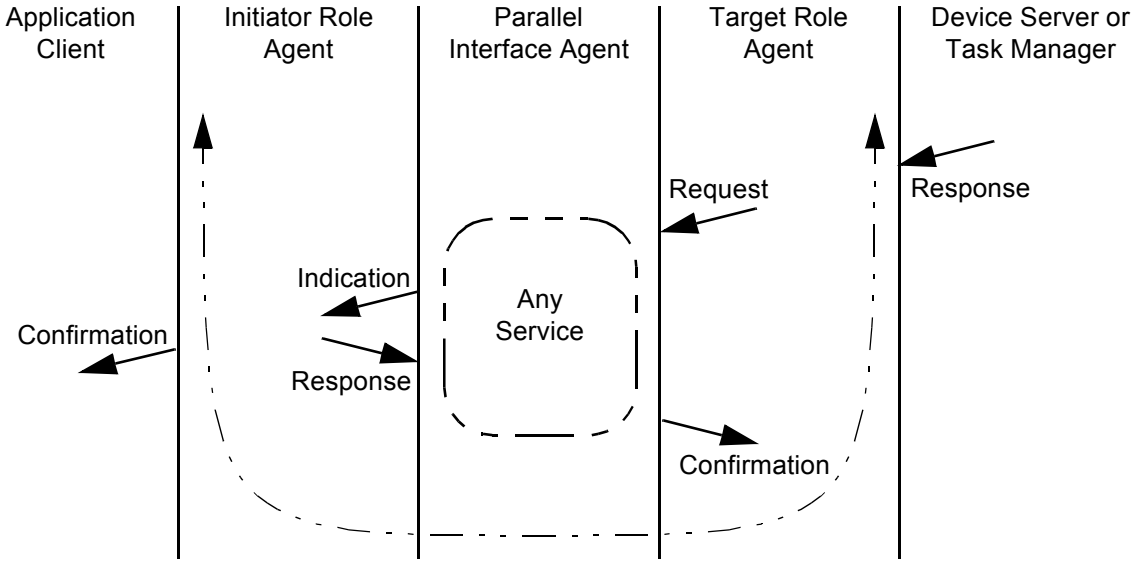


Figure 31 - Task management function service response sequencing (without message in) (attention option)

5.3.1.1.4 Task management function confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent a task management function confirmation is generated by the initiator role agent to the application client. The task management function follows:

received function-executed (target identifier|target identifier+logical unit number|target identifier+logical unit number+tag, service response ||).

The initiator role agent shall not issue any service requests to the parallel interface agent to carry out the task management function confirmation.

The service response shall be set to FUNCTION REJECTED if the initiator role agent received a MESSAGE REJECT message from a parallel interface agent. If a MESSAGE REJECT message was not received the service response shall be set to FUNCTION COMPLETE.

5.3.1.2 Message task management functions

The task management functions are defined in the SCSI-3 Architecture Model Standard. This standard defines the services used by the SCS-3I Interlocked Protocol to move the task management functions from the application client to the task manager. This standard, also, defines the binary values (see table 21) of the task management functions.

5.3.1.2.1 ABORT TASK request

The application client shall issue an ABORT TASK service request as follows:

ABORT TASK (target identifier+logical unit number[+tag] ||).

See 5.3.1.1.1 for more information.

5.3.1.2.2 ABORT TASK indication

After the completion of the sequence of services described in 5.3.1.1.2 an ABORT TASK indication is

generated to the task manager. The ABORT TASK indication follows:

ABORT TASK (target identifier+initiator identifier+logical unit number[+ tag] ||).

See 5.3.1.1.2 for more information.

5.3.1.2.3 ABORT TASK response

After the task manager completes the ABORT TASK function it shall send the following response to the target role agent:

ABORT TASK (target identifier+initiator identifier+logical unit number[+tag], service response ||).

See 5.3.1.1.3 for more information.

5.3.1.2.4 ABORT TASK confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent an ABORT TASK confirmation is generated by the initiator role agent to the application client. The ABORT TASK follows:

ABORT TASK (target identifier+logical unit number[+tag], service response ||).

See 5.3.1.1.4 for more information.

5.3.1.2.5 ABORT TASK SET request

The application client shall issue an ABORT TASK SET service request as follows:

ABORT TASK SET (target identifier+logical unit number ||).

See 5.3.1.1.1 for more information.

5.3.1.2.6 ABORT TASK SET indication

After the completion of the sequence of services described in 5.3.1.1.2 an ABORT TASK SET indication is generated to the task manager. The ABORT TASK SET indication follows:

ABORT TASK SET (target identifier+initiator identifier+ logical unit number, ||).

See 5.3.1.1.2 for more information.

5.3.1.2.7 ABORT TASK SET response

After the task manager completes the ABORT TASK SET function it shall send the following response to the target role agent:

ABORT TASK SET (target identifier+initiator identifier+logical unit number, service response ||).

See 5.3.1.1.3 for more information.

5.3.1.2.8 ABORT TASK SET confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent an ABORT TASK SET confirmation is generated by the initiator role agent to the application client. The ABORT TASK SET follows:

ABORT TASK SET (target identifier+logical unit number, service response ||).

See 5.3.1.1.4 for more information.

5.3.1.2.9 CLEAR ACA request

The application client shall issue a CLEAR ACA service request as follows:

CLEAR ACA (target identifier+logical unit number ||).

See 5.3.1.1.1 for more information.

5.3.1.2.10 CLEAR ACA indication

After the completion of the sequence of services described in 5.3.1.1.2 a CLEAR ACA indication is generated to the task manager. The CLEAR ACA indication follows:

CLEAR ACA (target identifier+initiator identifier+ logical unit number, ||).

See 5.3.1.1.2 for more information.

5.3.1.2.11 CLEAR ACA response

After the task manager completes the CLEAR ACA function it shall send the following response to the target role agent:

CLEAR ACA (target identifier+initiator identifier+logical unit number, service response ||).

See 5.3.1.1.3 for more information.

5.3.1.2.12 CLEAR ACA confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent a CLEAR ACA confirmation is generated by the initiator role agent to the application client. The CLEAR ACA follows:

CLEAR ACA (target identifier+logical unit number, service response ||).

See 5.3.1.1.4 for more information.

5.3.1.2.13 CLEAR TASK SET request

The application client shall issue a CLEAR TASK SET service request as follows:

CLEAR TASK SET (target identifier+logical unit number ||).

See 5.3.1.1.1 for more information.

5.3.1.2.14 CLEAR TASK SET indication

After the completion of the sequence of services described in 5.3.1.1.2 a CLEAR TASK SET indication is generated to the task manager. The CLEAR TASK SET indication follows:

CLEAR TASK SET (target identifier+initiator identifier+logical unit number, ||).

See 5.3.1.1.2 for more information.

5.3.1.2.15 CLEAR TASK SET response

After the task manager completes the CLEAR TASK SET function it shall send the following response to

the target role agent:

CLEAR TASK SET (target identifier+initiator identifier+logical unit number, service response ||).

See 5.3.1.1.3 for more information.

5.3.1.2.16 CLEAR TASK SET confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent a CLEAR TASK SET confirmation is generated by the initiator role agent to the application client. The CLEAR TASK SET follows:

CLEAR TASK SET (target identifier+logical unit number, service response ||).

See 5.3.1.1.4 for more information.

5.3.1.2.17 LOGICAL UNIT RESET request

The application client shall issue a LOGICAL UNIT RESET service request as follows:

LOGICAL UNIT RESET (target identifier+logical unit number||).

See 5.3.1.1.1 for more information.

5.3.1.2.18 LOGICAL UNIT RESET indication

After the completion of the sequence of services described in 5.3.1.1.2 a LOGICAL UNIT RESET indication is generated to the task manager. The LOGICAL UNIT RESET indication follows:

LOGICAL UNIT RESET (target identifier+initiator identifier+logical unit number, ||).

See 5.3.1.1.2 for more information.

5.3.1.2.19 LOGICAL UNIT RESET response

After the task manager completes the LOGICAL UNIT RESET function it shall send the following response to the target role agent:

LOGICAL UNIT RESET (target identifier+initiator identifier+logical unit number, service response ||).

See 5.3.1.1.3 for more information.

5.3.1.2.20 LOGICAL UNIT RESET confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent a LOGICAL UNIT RESET confirmation is generated by the initiator role agent to the application client. The LOGICAL UNIT RESET follows:

LOGICAL UNIT RESET (target identifier+logical unit number ||).

See 5.3.1.1.4 for more information.

5.3.1.2.21 TARGET RESET request

The application client shall issue a TARGET RESET service request as follows:

TARGET RESET (target identifier ||).

See 5.3.1.1.1 for more information.

5.3.1.2.22 TARGET RESET indication

After the completion of the sequence of services described in 5.3.1.1.2 a TARGET RESET indication is generated to the task manager. The TARGET RESET indication follows:

TARGET RESET (target identifier+initiator identifier, ||).

See 5.3.1.1.2 for more information.

5.3.1.2.23 TARGET RESET response

After the task manager completes the TARGET RESET function it shall send the following response to the target role agent:

TARGET RESET (target identifier+initiator identifier, service response ||).

See 5.3.1.1.3 for more information.

5.3.1.2.24 TARGET RESET confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent a TARGET RESET confirmation is generated by the initiator role agent to the application client. The TARGET RESET follows:

TARGET RESET (target identifier, service response ||).

See 5.3.1.1.4 for more information.

5.3.1.2.25 TERMINATE TASK request

The application client shall issue a TERMINATE TASK service request as follows:

TERMINATE TASK (target identifier+logical unit number[+tag] ||).

See 5.3.1.1.1 for more information.

5.3.1.2.26 TERMINATE TASK indication

After the completion of the sequence of services described in 5.3.1.1.2 a TERMINATE TASK indication is generated to the task manager. The TERMINATE TASK indication follows:

TERMINATE TASK (target identifier+initiator identifier+logical unit number[+tag] ||).

See 5.3.1.1.2 for more information.

5.3.1.2.27 TERMINATE TASK response

After the task manager completes the TERMINATE TASK function it shall send the following response to the target role agent:

TERMINATE TASK (target identifier+initiator identifier+logical unit number[+tag], service response ||).

See 5.3.1.1.3 for more information.

5.3.1.2.28 TERMINATE TASK confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent a TERMINATE TASK confirmation is generated by the initiator role agent to the application client. The TERMINATE TASK follows:

TERMINATE TASK (target identifier+logical unit number[+tag], service response ||).

See 5.3.1.1.4 for more information.

5.3.1.3 Non-message task management function service sequencing

5.3.1.3.1 Task management function request

The application client shall issue a task management function request as follows:

send task management request ([target identifier] ||).

NOTE 3 - The SCSI-3 Interlocked Protocol Standard ignores the target identifier for non-message task management function services.

When a task management function request is received the initiator role agent issues a reset service request (see figure 32) to the parallel interface agent.

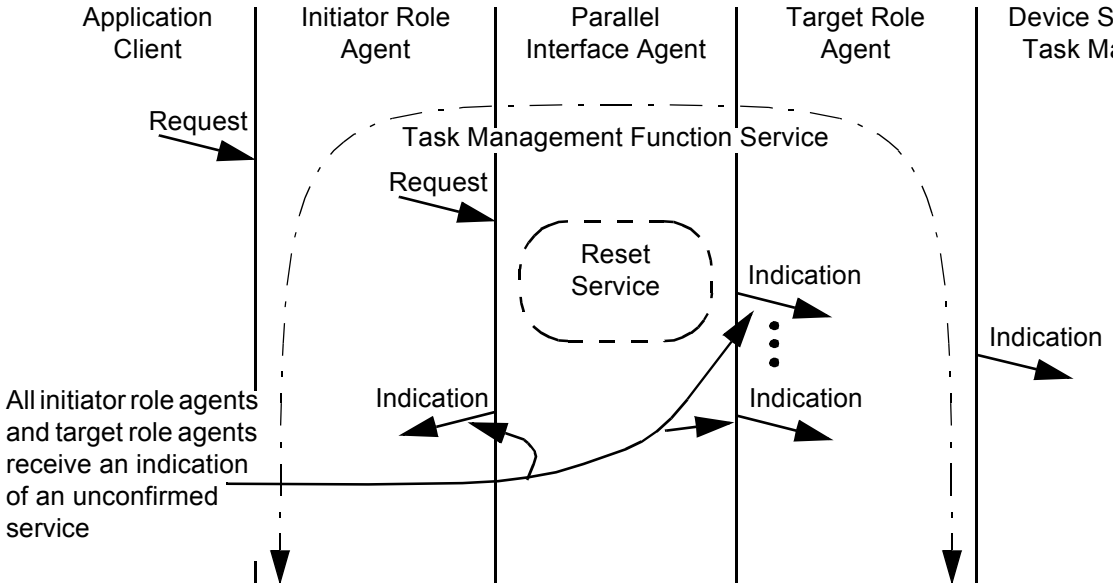


Figure 32 - Task management function service request sequencing

5.3.1.3.2 Task management function indication

Upon receipt of a reset indication a task management function indication is generated to the task manager. The task management function follows:

task management request received (target identifier ||).

5.3.1.3.3 Task management function response

After the task manager completes the task management function it shall send the following response to the

target role agent:

task management function executed (target identifier, service response ||).

After receiving a task management function response with a service response of function complete from the task manager the target role agent shall issue the service requests shown in figure 33.

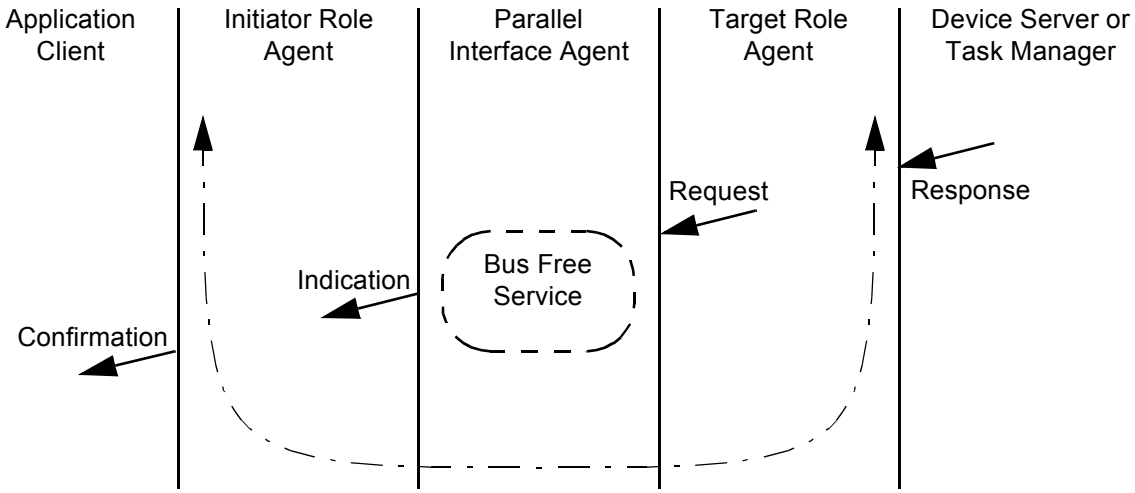


Figure 33 - Task management function service response sequencing

5.3.1.3.4 Task management function confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent a task management function confirmation is generated by the initiator role agent to the application client. The task management function follows:

received function-executed ([target identifier], service response ||).

The initiator role agent shall not issue any service requests to the parallel interface agent to carry out the task management function confirmation.

The service response shall be set to FUNCTION COMPLETE.

5.3.1.4 Non-message task management functions

The task management functions are defined in the SCSI-3 Architecture Model Standard. This standard defines the services used by the SCS-3I Interlocked Protocol to move the task management functions from the application client to the task manager. This standard does not define the binary values of the non-message task management functions.

5.3.1.4.1 RESET SERVICE DELIVERY SUBSYSTEM request

The application client shall issue an RESET SERVICE DELIVERY SUBSYSTEM service request as follows:

RESET SERVICE DELIVERY SUBSYSTEM (||).

See 5.3.1.3.1 for more information.

5.3.1.4.2 RESET SERVICE DELIVERY SUBSYSTEM indication

After the completion of the sequence of services described in 5.3.1.3.2 an RESET SERVICE DELIVERY SUBSYSTEM indication is generated to the task manager. The RESET SERVICE DELIVERY SUBSYSTEM indication follows:

RESET SERVICE DELIVERY SUBSYSTEM (target identifier, ||).

See 5.3.1.3.2 for more information.

5.3.1.4.3 RESET SERVICE DELIVERY SUBSYSTEM response

After the task manager completes the RESET SERVICE DELIVERY SUBSYSTEM function it shall send the following response to the target role agent:

RESET SERVICE DELIVERY SUBSYSTEM (target identifier, service response ||).

See 5.3.1.3.3 for more information.

5.3.1.4.4 RESET SERVICE DELIVERY SUBSYSTEM confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent an RESET SERVICE DELIVERY SUBSYSTEM confirmation is generated by the initiator role agent to the application client. The RESET SERVICE DELIVERY SUBSYSTEM follows:

RESET SERVICE DELIVERY SUBSYSTEM (service response ||).

See 5.3.1.3.4 for more information.

5.3.1.4.5 WAKEUP request

The application client shall issue a WAKEUP service request as follows:

WAKEUP (target identifier ||).

See 5.3.1.3.1 for more information.

5.3.1.4.6 WAKEUP indication

After the completion of the sequence of services described in 5.3.1.3.2 a WAKEUP indication is generated to the task manager. The WAKEUP indication follows:

WAKEUP (target identifier ||).

See 5.3.1.3.2 for more information.

5.3.1.4.7 WAKEUP response

After the task manager completes the WAKEUP function it shall send the following response to the target role agent:

WAKEUP (target identifier, service response ||).

See 5.3.1.3.3 for more information.

5.3.1.4.8 WAKEUP confirmation

After receipt of an indication to the initiator role agent from the parallel interface agent a WAKEUP confirmation is generated by the initiator role agent to the application client. The WAKEUP follows:

WAKEUP (target identifier, service response ||).

See 5.3.1.3.4 for more information.

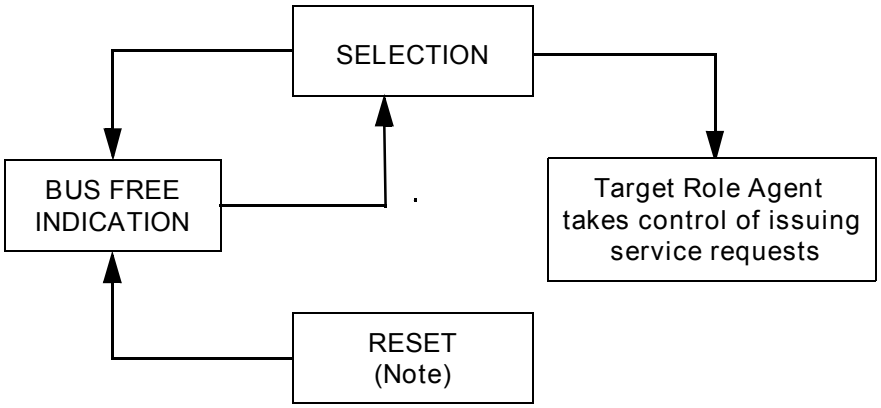
6 SCSI parallel interface services

The SCSI parallel interface services (see SCSI-3 Parallel Interface Standard) are used by the target role agent and initiator role agent to request the parallel interface agent to carry out specific actions on the SCSI parallel interface interconnect.

Except where stated, this standard does not define retry sequences to recover from errors that are reported by the parallel interface agent or detected by the target role agent and initiator role agent. However retry sequences are allowed.

6.1 Valid service sequences

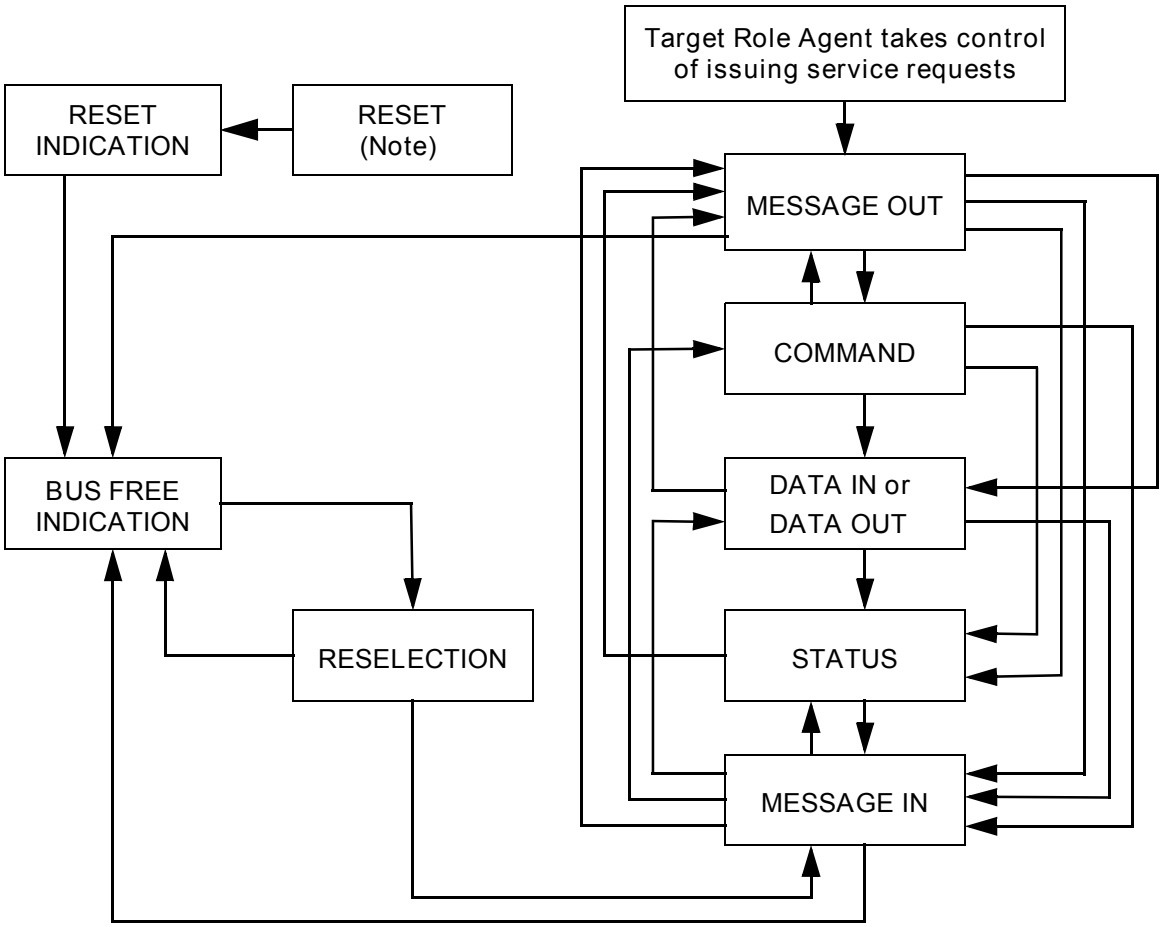
Figure 34 shows all the valid service request sequences allowed by the SCSI-3 Interlocked Protocol Standard for an initiator role agent. Any attempt to request a sequence not defined in figure 34 shall result in a protocol error. When an initiator role agent detects a protocol error it shall request a reset service.



Note-The initiator role agent may generate a reset service at any time.

Figure 34 - Valid service request sequences for initiator role agents

Figure 35 shows all the valid service request sequences allowed by the SCSI-3 Interlocked Protocol Standard for a target role agent. Any attempt to request a sequence not defined in figure 35 shall result in a protocol error. When a target role agent detects a protocol error it shall request a bus free service.



Note-The target role agent may generate a reset service at any time.

Figure 35 - Valid service request sequences for target role agents

6.2 Initiator role SCSI parallel interface services

6.2.1 Reset service

A reset service is an unconfirmed service that is generated by the initiator role agent to request the parallel interface agent to generate an SCSI bus reset. The initiator role agent may request a reset service at any time.

The reset service may be made available to application clients in a vendor specific manner.

6.2.1.1 Reset request

A reset request contains no parameters.

6.2.1.2 Reset indication

A reset indication contains no parameters.

Upon receipt of a reset indication the target role agent shall carry out a hard reset as defined in the SCSI-3 Architecture Model Standard.

Upon receipt of a reset indication a target role agent shall generate a bus free service request to the parallel interface agent as its next action (see figure 36).

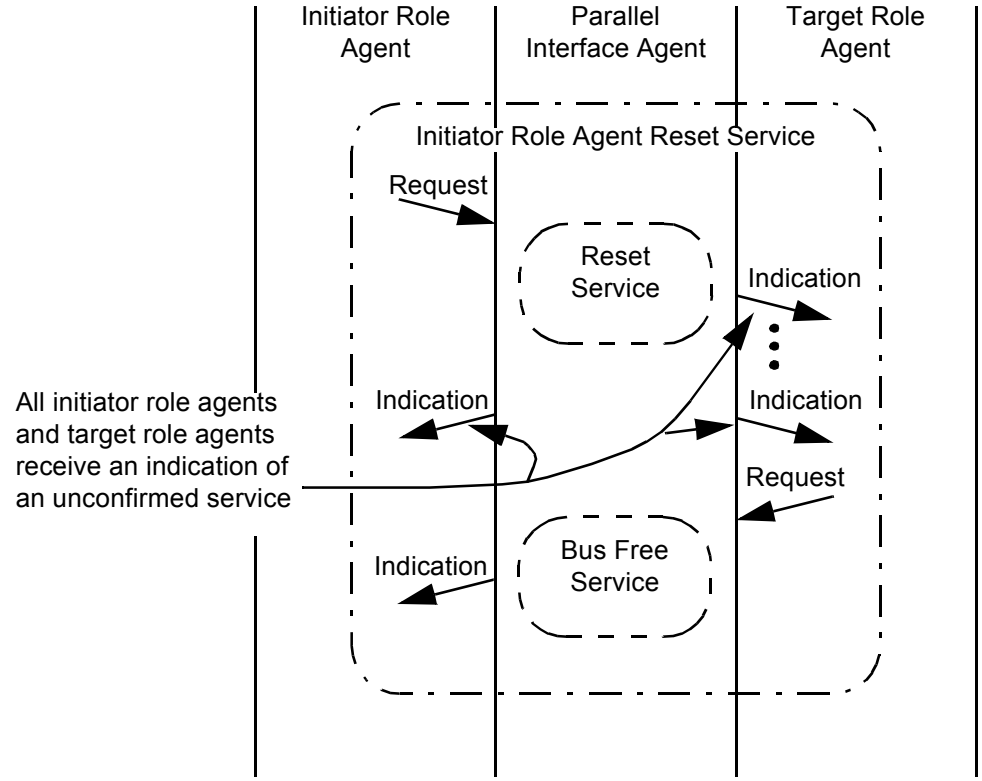


Figure 36 - Initiator role agent reset service request sequencing

6.2.2 Selection service

The selection service is a four step confirmed service that establishes an initial connection. At the successful completion of a selection service an I_T nexus has been established between an initiator and a target.

6.2.2.1 Selection request

The selection request contains the SCSI ID{target identifier} and the attention flag. The SCSI ID{target identifier} shall contain the address of the target role agent to be selected. The attention flag shall be set to one to indicate to the target role agent that a message out service shall be requested after a successful selection service.

6.2.2.2 Selection indication

The selection indication contains the selection ID{target identifier+initiator identifier}, the attention flag, and the parity flag. The selection ID{target identifier+initiator identifier} contains the address of the selected target role agent and the selecting initiator role agent. The attention flag set to one indicates a message out service shall be requested after a successful selection service. A parity flag set to zero indicates there was a successful selection. A parity flag set to one indicates the selection failed due to a parity error.

6.2.2.3 Selection response

The selection response contains the selection accepted flag. The selection accepted flag shall be set to zero to indicate the selection failed. A failed selection occurs if the parity flag in the selection indication was

set to one. The selection accepted flag shall be set to one to indicate the selection was successful. A successful selection occurs if the parity flag in the selection indication was set to zero.

6.2.2.4 Selection confirmation

The selection confirmation contains the arbitration lost flag, selection won flag, and selection time-out flag. An arbitration lost flag of zero indicates a successful arbitration occurred. An arbitration lost flag of one indicates the arbitration was lost. When arbitration is lost no selection indication nor selection response shall occur (see figure 38). A selection won flag set to zero indicates the selection failed. A selection won flag set to one indicates the selection was successful (see figure 37). A selection time-out flag set to zero indicates the parallel interface agent did not detect a selection time-out. A selection time-out flag set to one indicates the parallel interface agent detected a selection time-out.

If arbitration is lost (see figure 38) or there is a selection time-out (see figure 39) the initiator role agent shall not reissue the selection service. The initiator role agent shall notify the application client of the selection failure (see 5.1.1.2 and 5.3.1.1.2).

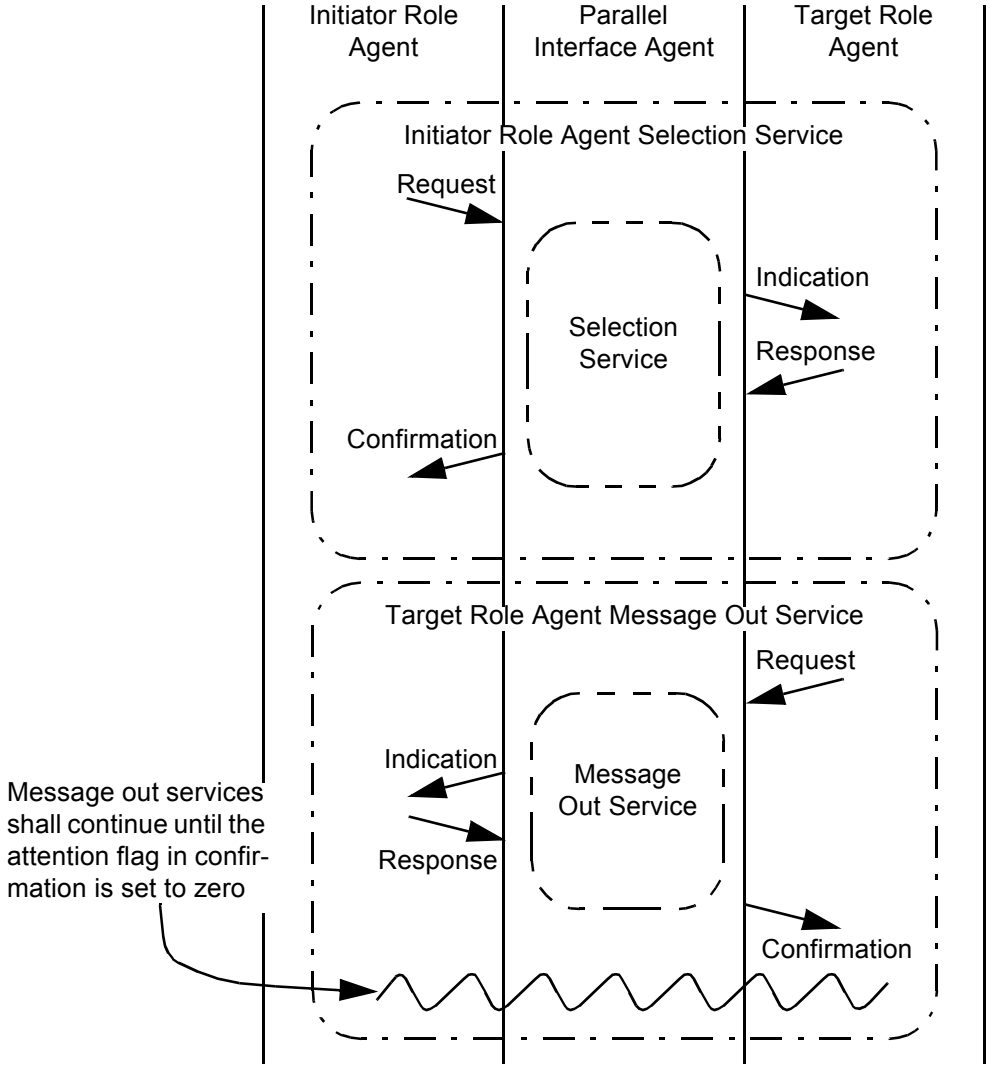


Figure 37 - Initiator role agent selection service request sequencing (attention flag=1/selection won flag=1)

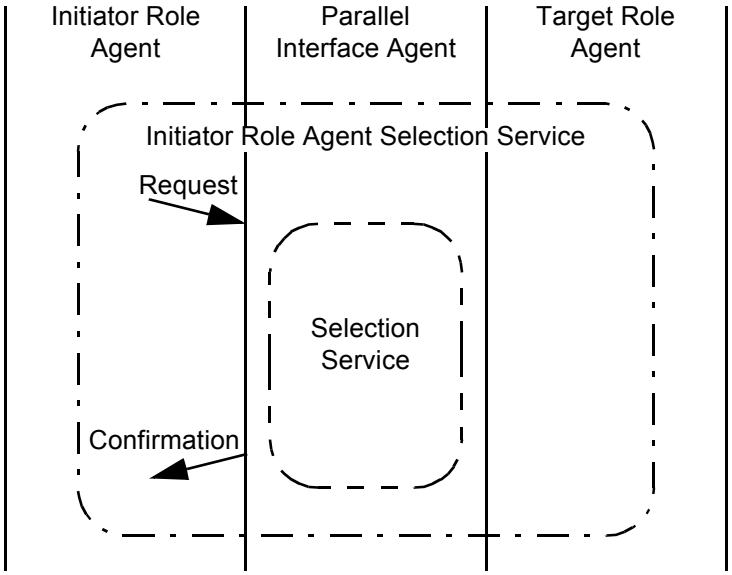


Figure 38 - Initiator role agent selection service request sequencing (arbitration lost flag = 1)

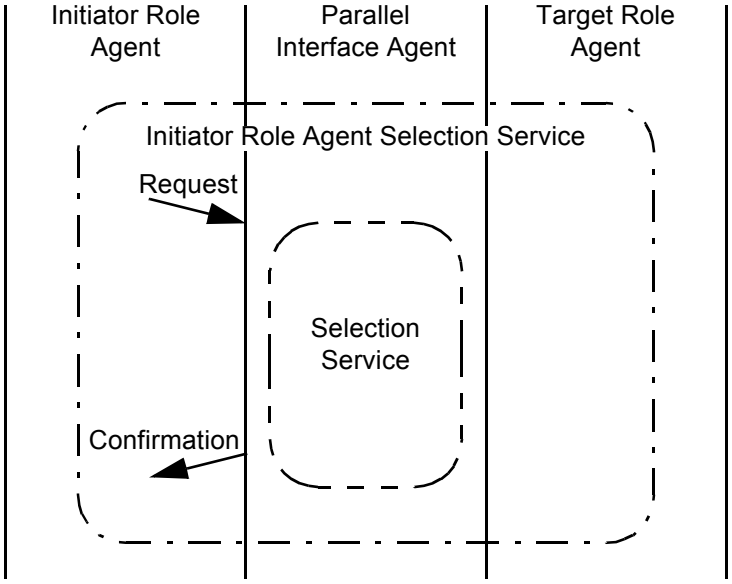


Figure 39 - Initiator role agent selection service request sequencing (selection time-out flag = 1)

6.3 Target role SCSI parallel interface services

6.3.1 Reset service

A reset service is an unconfirmed service that is generated by the target role agent to request the parallel interface agent to generate an SCSI bus reset. The target role agent may request a reset service at any time.

The reset service may be made available to device servers in a vendor specific manner.

6.3.1.1 Reset request

A reset request contains no parameters.

Upon generating a reset request a target role agent shall generate a bus free service request to the parallel interface agent as its next action (see figure 40).

6.3.1.2 Reset indication

A reset indication contains no parameters.

Upon receipt of a reset indication the initiator role agent shall carry out a hard reset as defined in the SCSI-3 Architecture Model Standard.

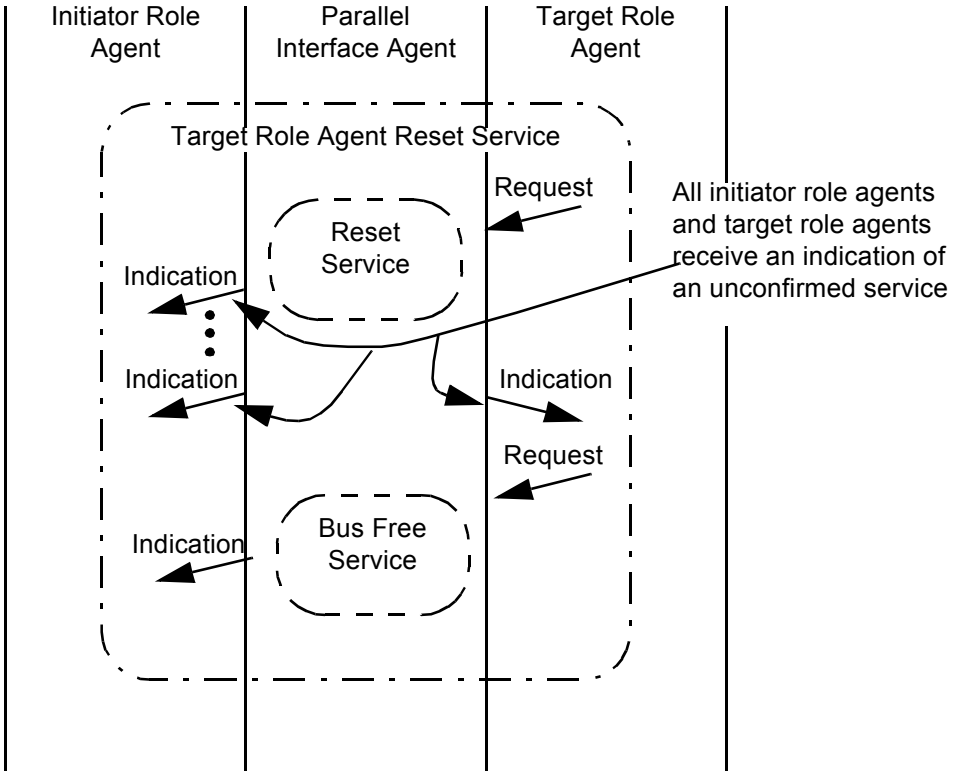


Figure 40 - Target role agent reset service request sequencing

6.3.2 Bus free service

A bus free service is an unconfirmed service that is generated by the target role agent to request the parallel interface agent to generate a bus free. The target role agent may request a bus free service at any time.

6.3.2.1 Bus free request

A bus free request contains no parameters.

After generating a bus free request a target role agent may request a reselection service.

6.3.2.2 Bus free indication

A bus free indication contains no parameters.

After receipt of a bus free indication the initiator role agent may request a selection service.

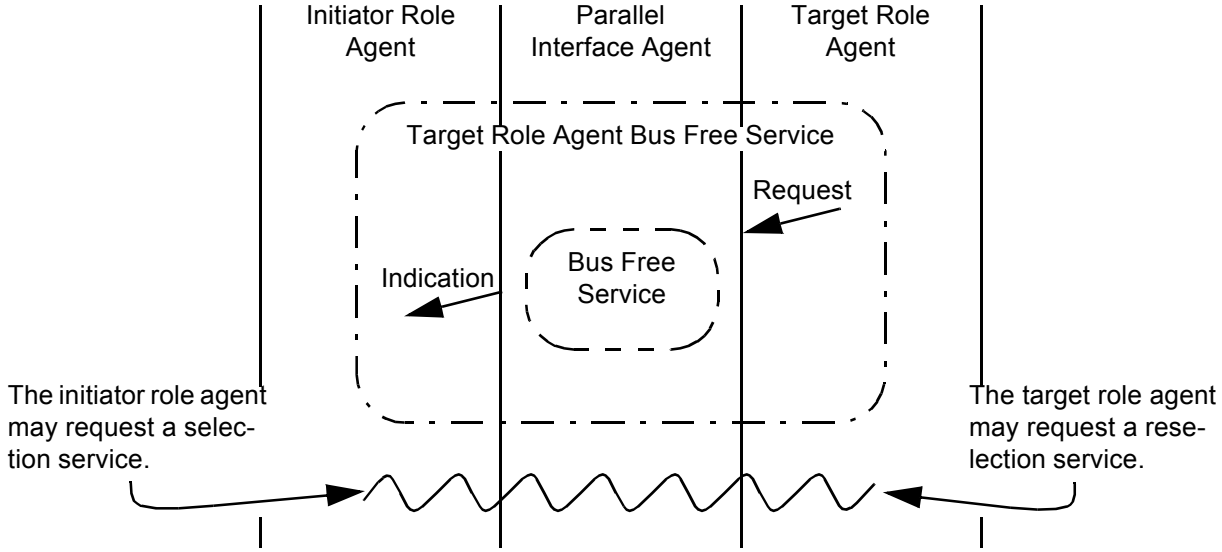


Figure 41 - Target role agent reset service request sequencing

6.3.3 Reselection service

The reselection service is a four step confirmed service that reestablishes an I_T nexus. At the successful completion of a resection service an I_T nexus has been established between a target and an initiator.

6.3.3.1 Reselection request

The reselection request contains the SCSI ID{initiator identifier}. The SCSI ID{initiator identifier} shall contain the address of the initiator role agent to be reselected.

6.3.3.2 Reselection indication

The reselection indication contains the reselection ID{target identifier+initiator identifier} and the parity flag. The reselection ID{target identifier+initiator identifier} contains the address of the reselected initiator role agent and the selecting target role agent. A parity flag set to zero indicates there was a successful reselection. A parity flag set to one indicates the reselection failed due to a parity error.

6.3.3.3 Reselection response

The reselection response contains the reselection accepted flag. The reselection accepted flag shall be set to zero to indicate the reselection failed. A failed reselection occurs if the parity flag in the reselection indication was set to one. The reselection accepted flag shall be set to one to indicate the reselection was successful. A successful reselection occurs if the parity flag in the reselection indication was set to zero.

6.3.3.4 Reselection confirmation

The reselection confirmation contains the arbitration lost flag, reselection won flag, and reselection time-out flag. An arbitration lost flag of zero indicates a successful arbitration occurred. An arbitration lost flag of one indicates the arbitration was lost. When arbitration is lost no reselection indication nor

reselection response shall occur (see figure 43). A reselection won flag set to zero indicates the reselection failed. A reselection won flag set to one indicates the reselection was successful (see figure 42). A reselection time-out flag set to zero indicates the parallel interface agent did not detect a reselection time-out. A reselection time-out flag set to one indicates the parallel interface agent detected a reselection time-out.

If arbitration is lost (see figure 43) or there is a reselection time-out (see figure 44) the target role agent may reissue the reselection service a vendor specific number of times. The target role agent may terminate the reselections at any time.

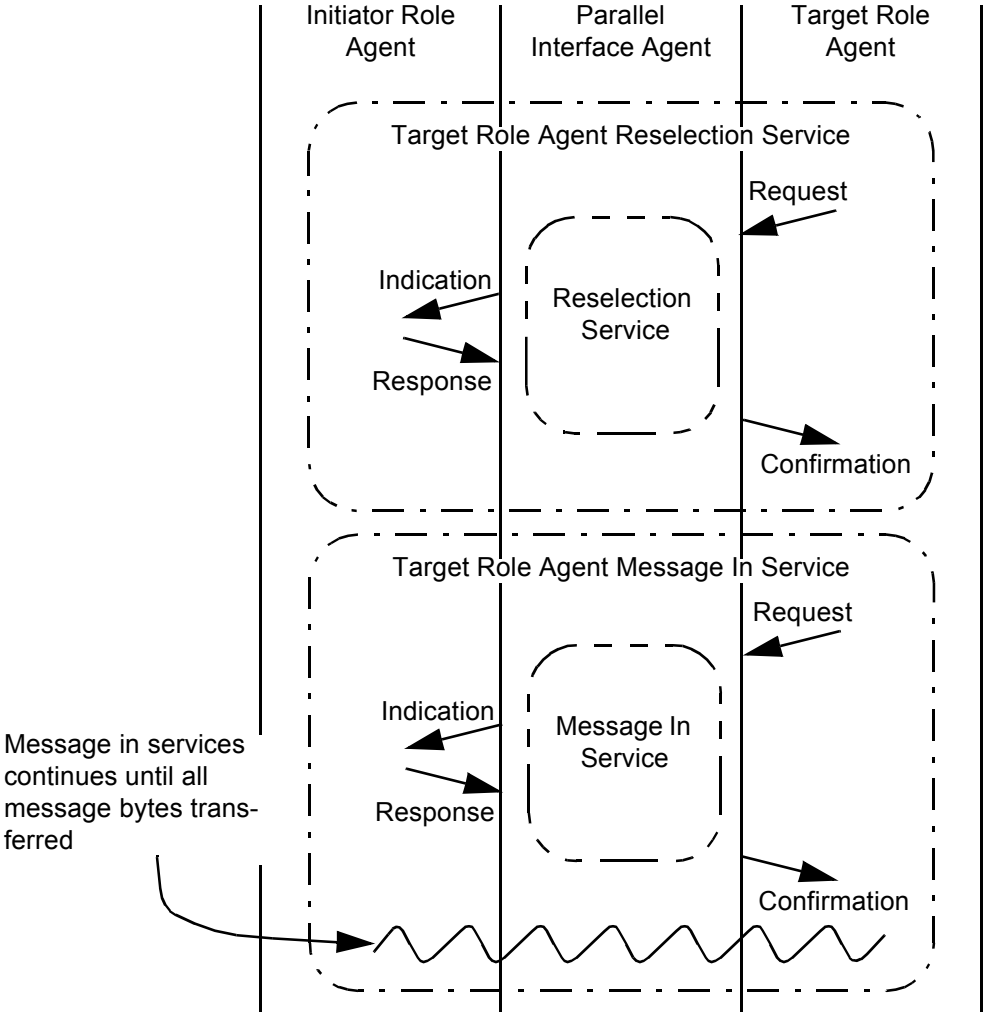


Figure 42 - Target role agent reselection service request sequencing (reselection won flag = 1)

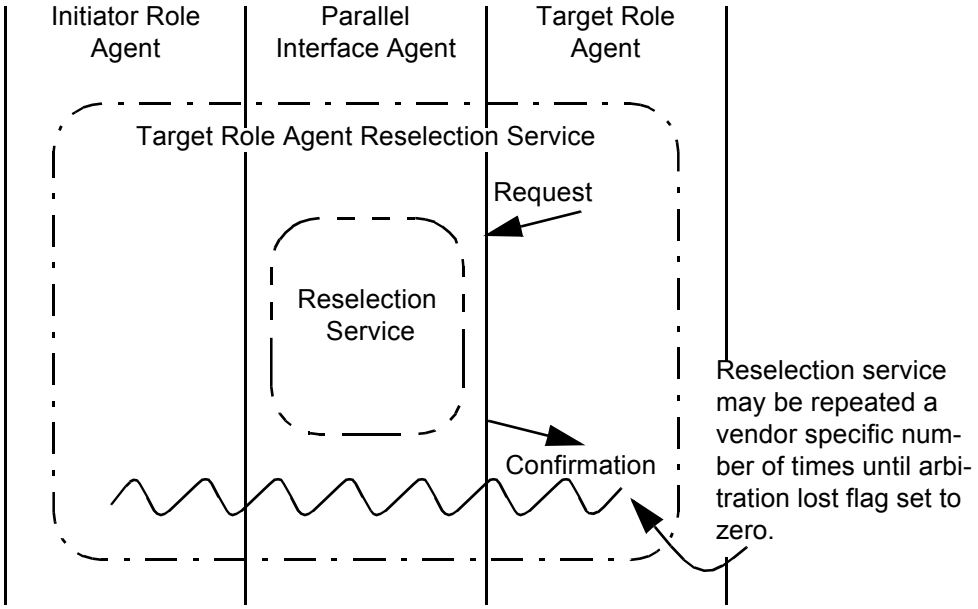


Figure 43 - Target role agent reselection service request sequencing (arbitration lost flag = 1)

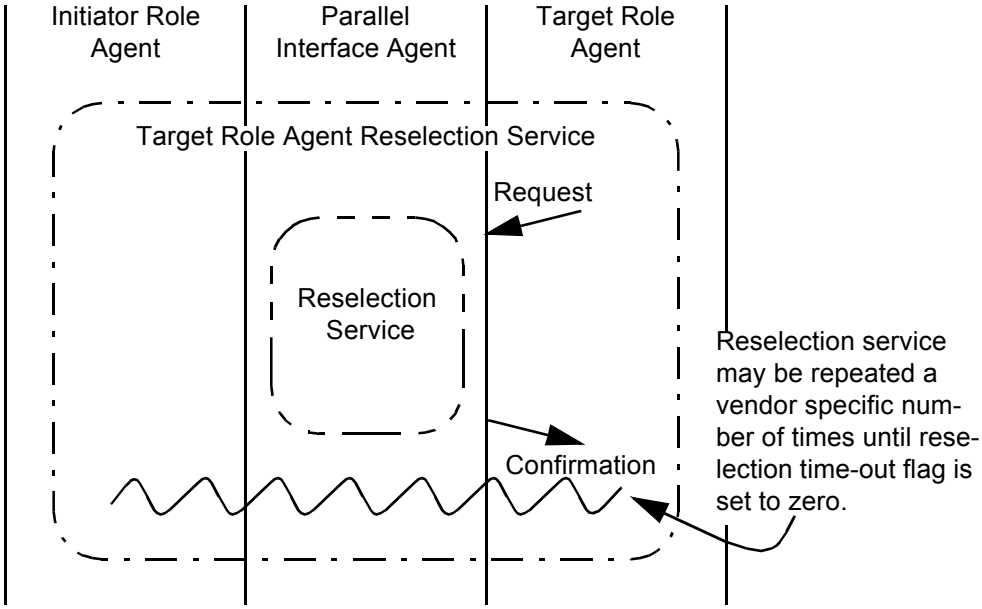


Figure 44 - Target role agent reselection service request sequencing (reselection time-out flag = 1)

6.3.4 Command service

The command service is a four step confirmed service that provide the means to transfer a single command byte from the initiator role agent to the target role agent (see figure 45).

6.3.4.1 Command request

The command request contains no parameters.

6.3.4.2 Command indication

The command indication contains no parameters.

6.3.4.3 Command response

The command response contains a command byte and the attention flag. The attention flag set to zero indicates to the initiator role agent is not requesting a message out service. The attention flag set to one indicates to the initiator role agent is requesting that a message out service be generated by the target role agent, at its discretion.

6.3.4.4 Command confirmation

The command confirmation contains the command byte, the attention flag, and the parity flag. An attention flag set to zero indicates to the target role agent that no message out service is being requested. An attention flag set to one indicates to the target role agent that a message out service is being requested by the initiator role agent. A parity flag set to zero indicates the command byte is valid. A parity flag set to one to indicate the command byte is not valid.

The device server shall be notified that one of the command bytes is invalid when the parity flag is set to one. The target role agent may, however, continue to generate command services until all the command bytes have been received before it notifies the device server. The target role agent makes no attempt to retry the command service.

When the attention flag is set to one the next service request, after the command service, shall be either a command service or a message out service (see figure 46). The target role agent may, however, continue to generate command services until all the command bytes have been received.

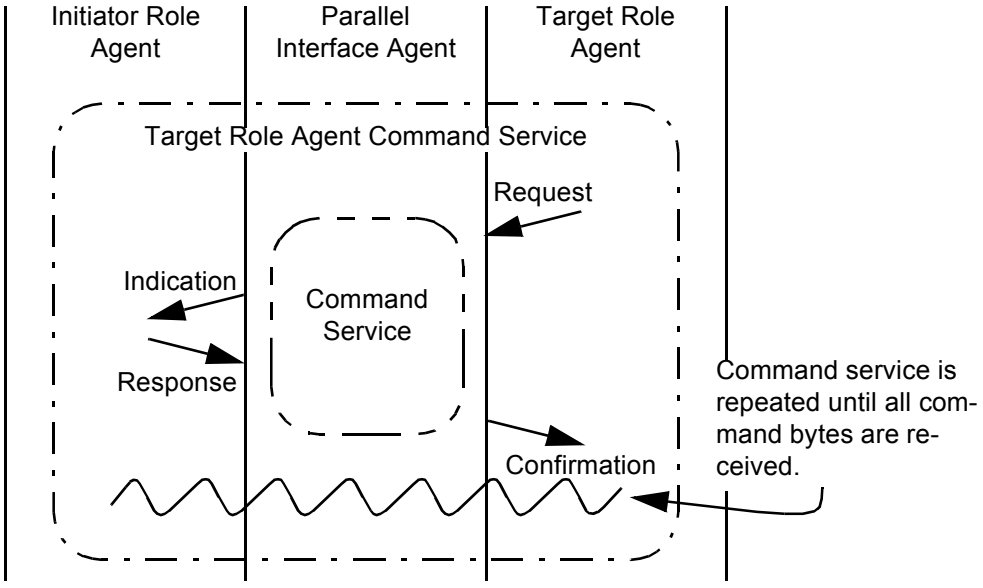


Figure 45 - Target role agent command service request sequencing

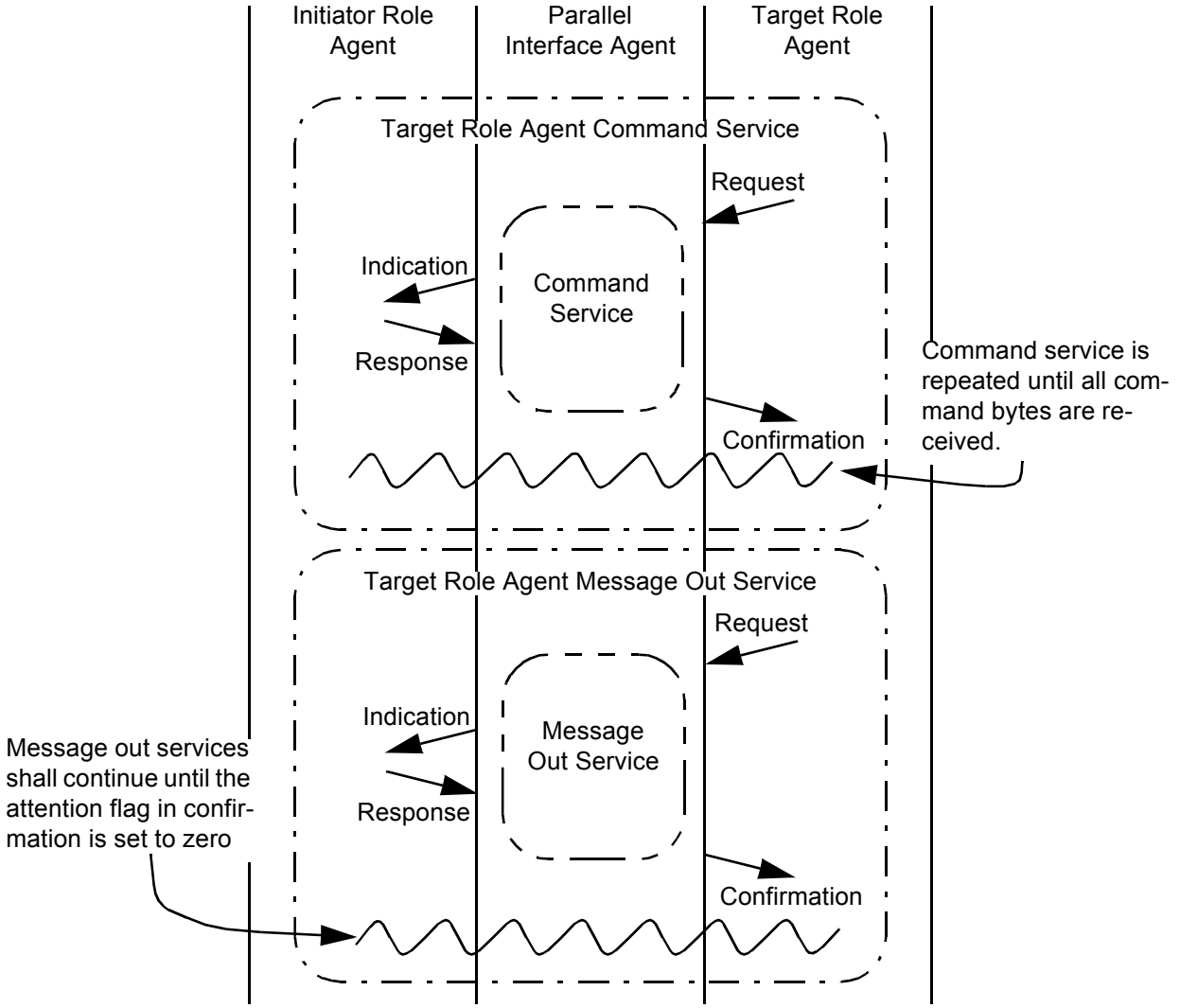


Figure 46 - Target role agent command service request sequencing (attention flag = 1)

6.3.5 Data out service

The data out service is a four step confirmed service that provide the means to transfer a single data word from the initiator role agent to the target role agent (see figure 47).

6.3.5.1 Data out request

The data out request contains the data bus width, the transfer period, and the REQ/ACK offset. See the SCSI-3 Parallel Interface Standard for the definitions of the data bus width, transfer period, and REQ/ACK offset. The values of the transfer period and the REQ/ACK offset are determined by a negotiation of a synchronous data transfer agreement (see 8.2.12). The value of the data bus width is determined by a negotiation of a wide data transfer agreement (see 8.2.15).

6.3.5.2 Data out indication

The data out indication contains no parameters.

6.3.5.3 Data out response

The data out response contains a data word and the attention flag. The attention flag is set to zero to indicate that the initiator role agent is not requesting a message out service. The attention flag is set to one to indicate to the initiator role agent is requesting that a message out service be generated by the target role initiator, at its discretion.

6.3.5.4 Data out confirmation

The data out confirmation contains the data word, the attention flag, and the parity flag. An attention flag set to zero indicates to the target role agent that no message out service is being requested. An attention flag set to one indicates to the target role agent that a message out service is being requested by the initiator role agent. A parity flag set to zero indicates the data word is valid. A parity flag set to one indicates the data word is not valid.

The device server shall be notified that one of the data words is invalid when the parity flag is set to one. The target role agent may, however, continue to generate data out services until a vendor specific number of data word(s) have been received before it notifies the device server. The target role agent shall not retry the data out service.

NOTE 4 - Any data retry attempts are controlled by the device server.

When the attention flag is set to one the next service request, after the data out service, shall be either a data-out service or a message out service (see figure 48). The target role agent may, however, continue to generate data out services until a vendor specific number of data words have been received.

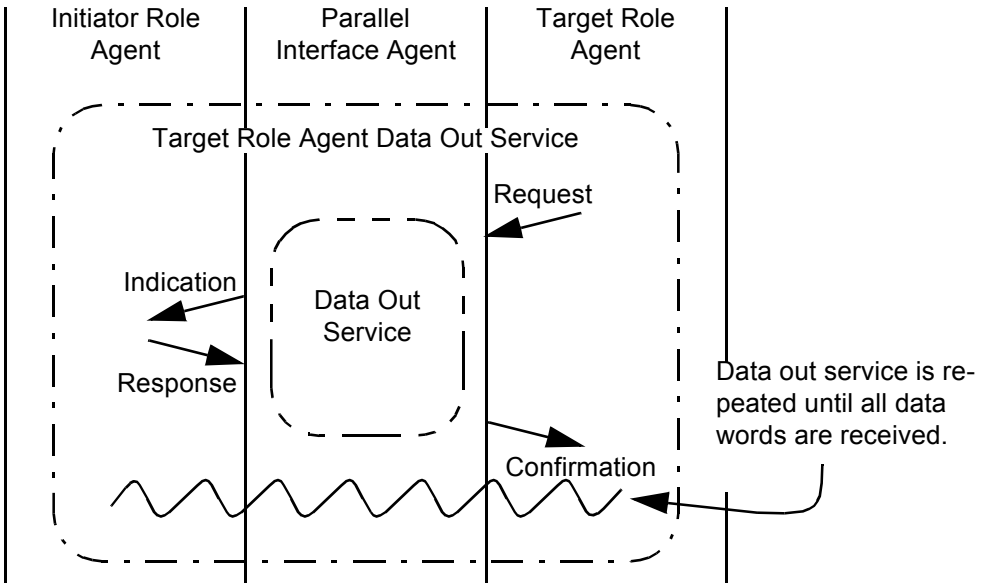


Figure 47 - Target role agent data out service request sequencing

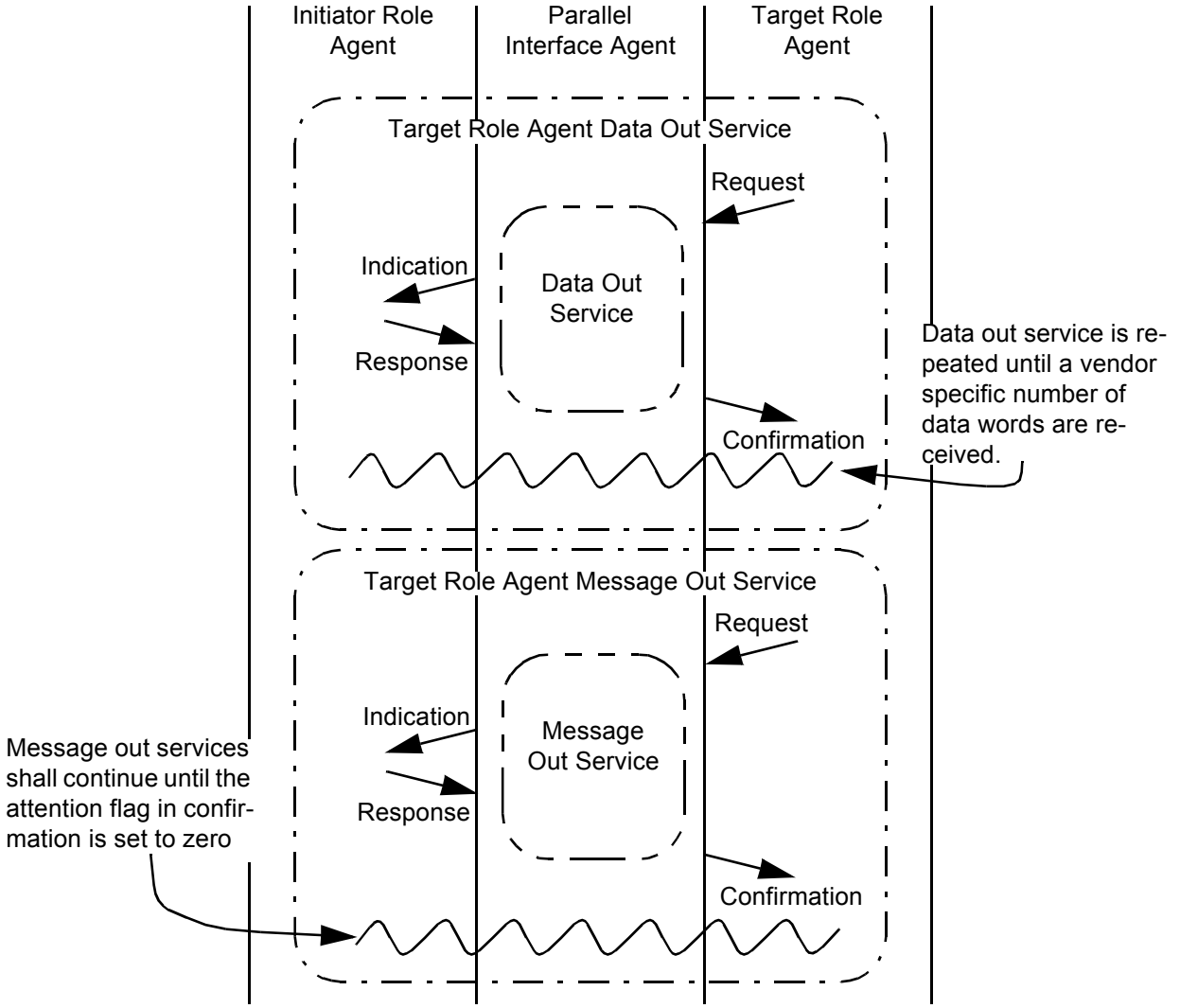


Figure 48 - Target role agent data out service request sequencing (attention flag = 1)

6.3.6 Data in service

The data in service is a four step confirmed service that provides the means to transfer a single data word from the target role agent to the initiator role agent (see figure 49).

6.3.6.1 Data in request

The data in request contains the data word, data bus width, the transfer period, and the REQ/ACK offset. See the SCSI-3 Parallel Interface Standard for the definitions of the data bus width, transfer period, and REQ/ACK offset. The values of the transfer period and the REQ/ACK offset are determined by a negotiation of a synchronous data transfer agreement (see 8.2.12). The value of the data bus width is determined by a negotiation of a wide data transfer agreement (see 8.2.15).

6.3.6.2 Data in indication

The data in indication contains the data word and the parity flag. A parity flag set to zero indicates the data word is valid. A parity flag set to one indicates the data word is not valid.

If the parity flag is set to one the initiator role agent shall set the attention flag to one in the data in

response. When the target role agent generates a message out service the initiator role agent shall send an INITIATOR DETECTED ERROR message (see 8.2.5) to the target role agent. This message notifies device server that one of the data words is invalid. The target role agent shall not retry the data in service.

NOTE 5 - Any data retry attempts are controlled by the device server.

6.3.6.3 Data in response

The data in response contains the attention flag. The attention flag is set to zero to indicate the initiator role agent is not requesting a message out service. The attention flag is set to one to indicate the initiator role agent is requesting that a message out service be generated by the target role initiator, at its discretion.

6.3.6.4 Data in confirmation

The data in confirmation contains the attention flag. An attention flag set to zero indicates to the target role agent that no message out service is being requested. An attention flag set to one indicates to the target role agent that a message out service is being requested by the initiator role agent.

When the attention flag is set to one, the next service request, after the data in service, shall be either a data-in service or a message out service (see figure 50). The target role agent may, however, continue to generate data in services until a vendor specific number of data words have been sent.

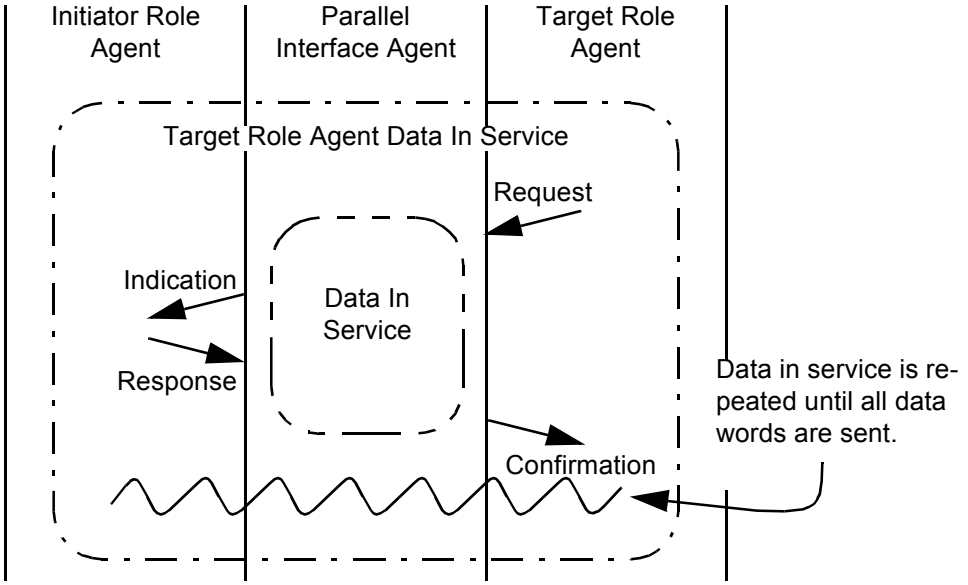


Figure 49 - Target role agent data in service request sequencing

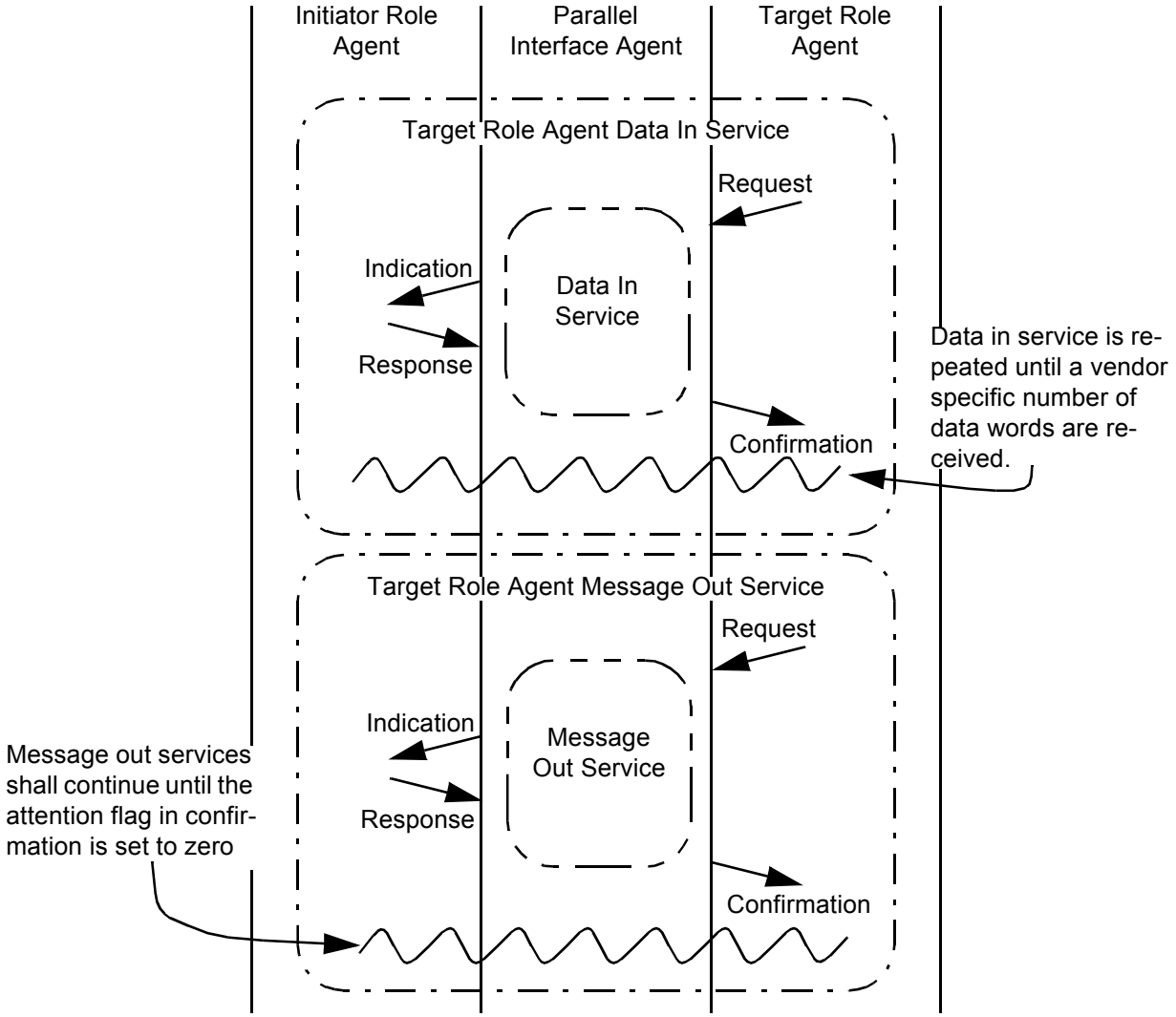


Figure 50 - Target role agent data in service request sequencing (attention flag = 1)

6.3.7 Status service

The status service is a four step confirmed service that provides the means to transfer the status byte from the target role agent to the initiator role agent (see figure 51).

6.3.7.1 Status request

The status request contains the status byte.

6.3.7.2 Status indication

The status indication contains the status byte and the parity flag. A parity flag set to zero indicates the status byte is valid. A parity flag set to one indicates the status byte is not valid.

If the parity flag is set to one the initiator role agent shall set the attention flag in the status response. When the target role agent generates a message out service the initiator role agent shall send an INITIATOR DETECTED ERROR message (see 8.2.5) to the target role agent. This message notifies device server that the status byte is invalid.

6.3.7.3 Status response

The status response contains the attention flag. The attention flag is set to zero to indicate the initiator role agent is not requesting a message out service. The attention flag is set to one to indicate the initiator role agent is requesting that a message out service be generated by the target role initiator, at its discretion.

6.3.7.4 Status confirmation

The status confirmation contains the attention flag. An attention flag set to zero indicates to the target role agent that no message out service is being requested. An attention flag set to one indicates to the target role agent that a message out service is being requested by the initiator role agent.

When the attention flag is set to one the next service request, after the status service, shall be a message out service (see figure 52).

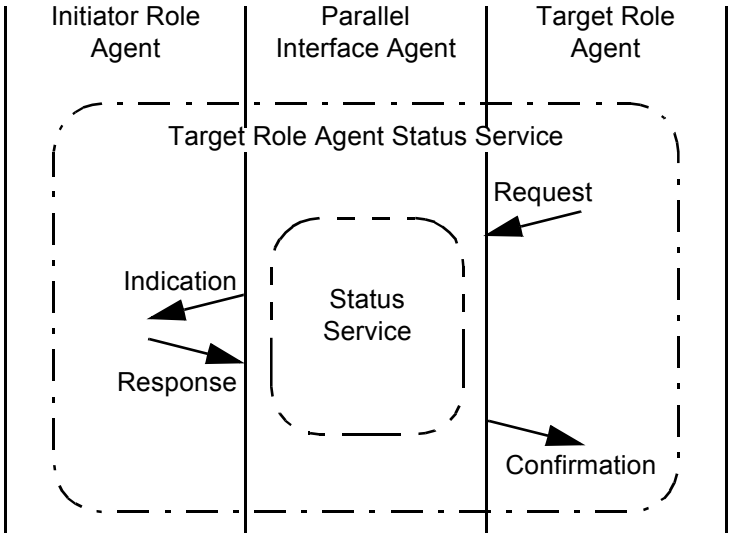


Figure 51 - Target role agent status service request sequencing

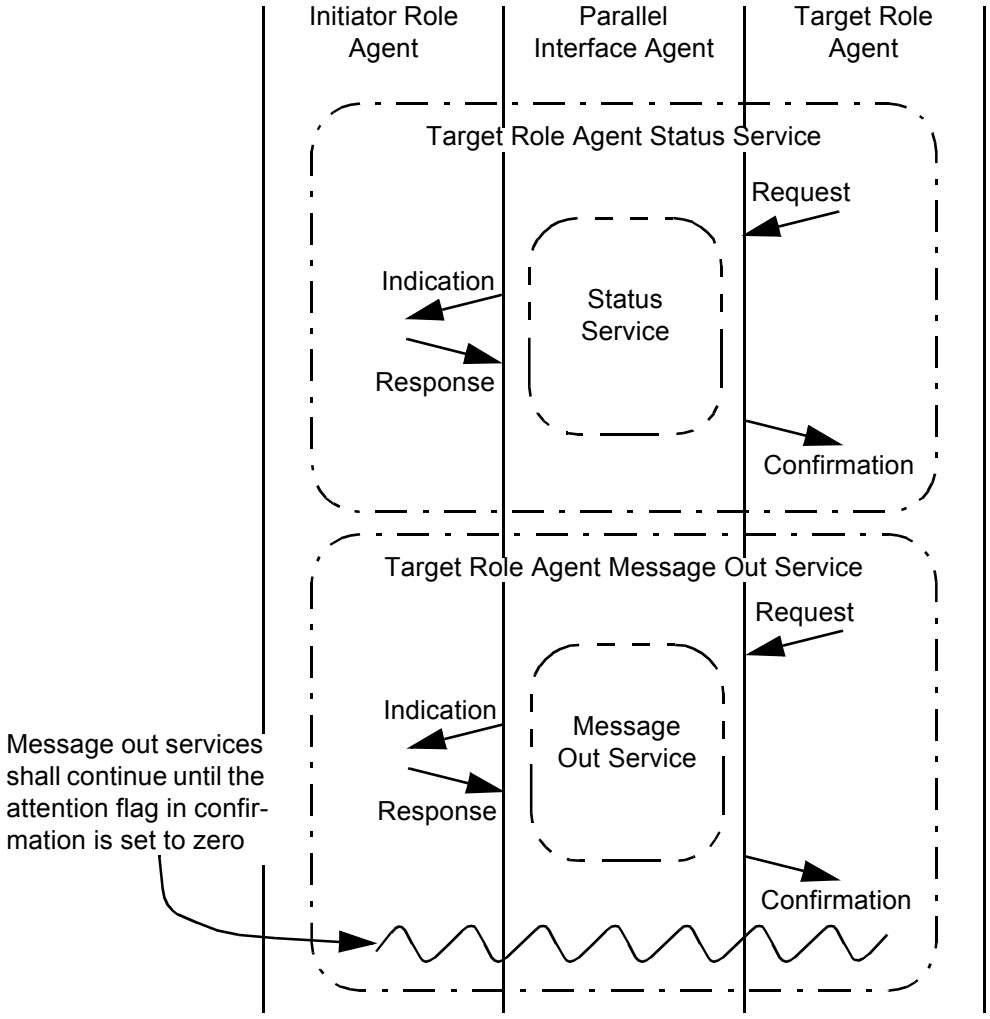


Figure 52 - Target role agent status service request sequencing (attention flag = 1)

6.3.8 Message out service

The message out service is a four step confirmed service that provides the means to transfer a single message out byte from the initiator role agent to the target role agent (see figure 53).

6.3.8.1 Message out request

The message out request contains no parameters.

6.3.8.2 Message out indication

The message out indication contains no parameters.

6.3.8.3 Message out response

The message out response contains a message out byte and the attention flag. The attention flag is set to zero to indicate the initiator role agent is not requesting further message out service. The attention flag is set to one to indicate the initiator role agent is requesting another message out service be generated by the target role agent.

6.3.8.4 Message out confirmation

The message out confirmation contains the message out byte, the attention flag, and the parity flag. An attention flag set to zero indicates to the target role agent that no message out service is being requested. An attention flag set to one indicates to the target role agent that a message out service is being requested by the initiator role agent. A parity flag set to zero indicates the message out byte is valid. A parity flag set to one indicates the message out byte is not valid.

The device server is notified that the message out byte is invalid when the parity flag is set to one. The target role agent may, however, continue to generate message out services until attention flag is set to zero before it notifies the device server. If the device server attempts to retry the message out service the initiator role agent shall resend the entire message(s) in the same order as previously sent during the most recent message out service sequence(s). When a sequence of messages is resent by an initiator role agent because of a parity error, the device server or the target role agent shall not act on any message which it acted on the first time received.

If the device server does not retry the message out service or it exhausts its retry limit it may:

a) return CHECK CONDITION status and set the sense key to ABORTED COMMAND and the additional sense code to MESSAGE ERROR; or
b) indicate an exception condition by performing an unexpected bus free (see 9.5).

When the attention flag is set to one the target role agent shall continue to generate message out service(s) until the attention flag is set to zero (see figure 54), except when rejecting a message.

If the device server receives all of the message byte(s) successfully (i.e., no parity errors) it shall indicate that a retry is not requested by requesting any service request other than a message out service and transferring at least one byte. The device server may also indicate that it has successfully received the message byte(s) by requesting a service that results in a bus free indication (e.g., ABORT TASK SET or TARGET RESET).

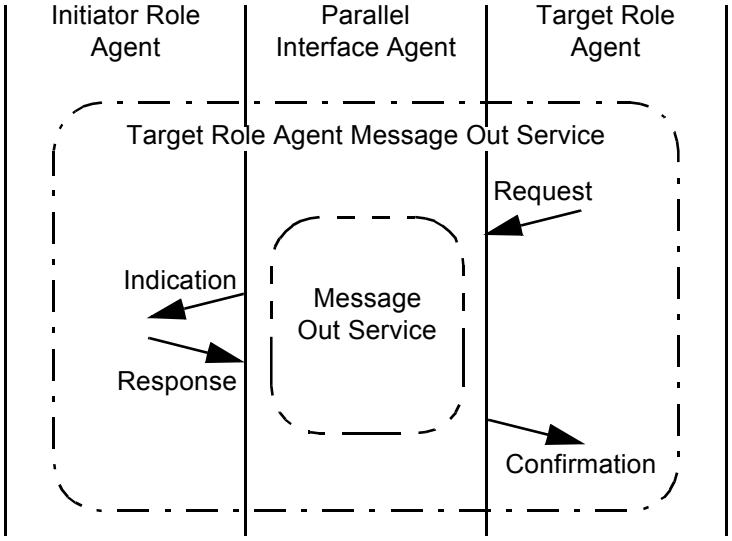


Figure 53 - Target role agent message out service request sequencing

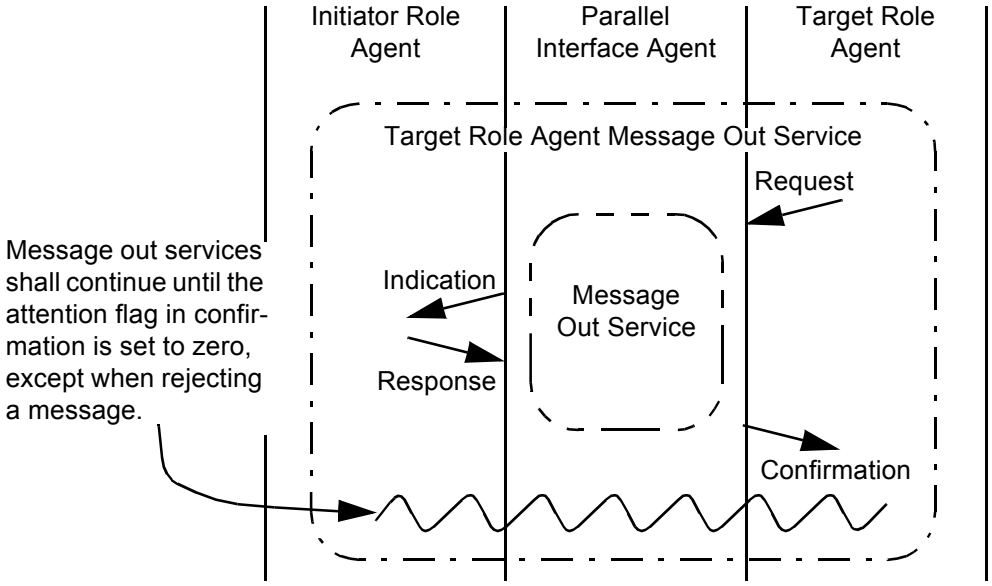


Figure 54 - Target role agent message out service request sequencing (attention flag = 1)

6.3.9 Message in service

The message in service is a four step confirmed service that provides the means to transfer the message in byte from the target role agent to the initiator role agent (see figure 55).

6.3.9.1 Message in request

The message in request contains the message in byte.

6.3.9.2 Message in indication

The message in indication contains the message in byte and the parity flag. A parity flag set to zero indicates the message in byte is valid. A parity flag set to one indicates the message in byte is not valid.

If the parity flag is set to one the initiator role agent shall set the attention flag in the message in response. When the target role agent generates a message out service the initiator role agent shall send a MESSAGE PARITY ERROR message (see 8.2.6) to the target role agent. This message notifies the target role agent that the message in byte is invalid.

6.3.9.3 Message in response

The message in response contains the attention flag. The attention flag is set to zero to indicate the initiator role agent is not requesting a message out service. The attention flag is set to one to indicate the initiator role agent is requesting that a message out service be generated by the target role initiator, at its discretion.

6.3.9.4 Message in confirmation

The message in confirmation contains the attention flag. An attention flag set to zero indicates to the target role agent that no message out service is being requested. An attention flag set to one indicates to the target role agent that a message out service is being requested by the initiator role agent.

When the attention flag is set to one the next service request, after the complete message is received, shall be a message out service (see figure 56).

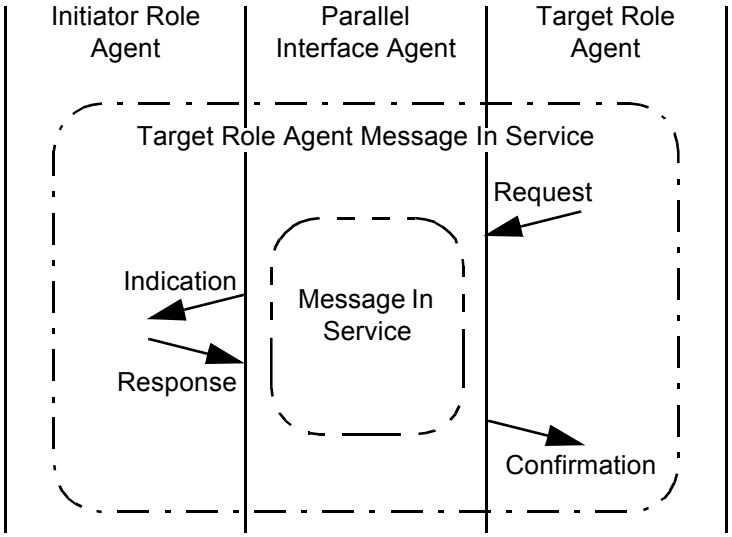


Figure 55 - Target role agent message in service request sequencing

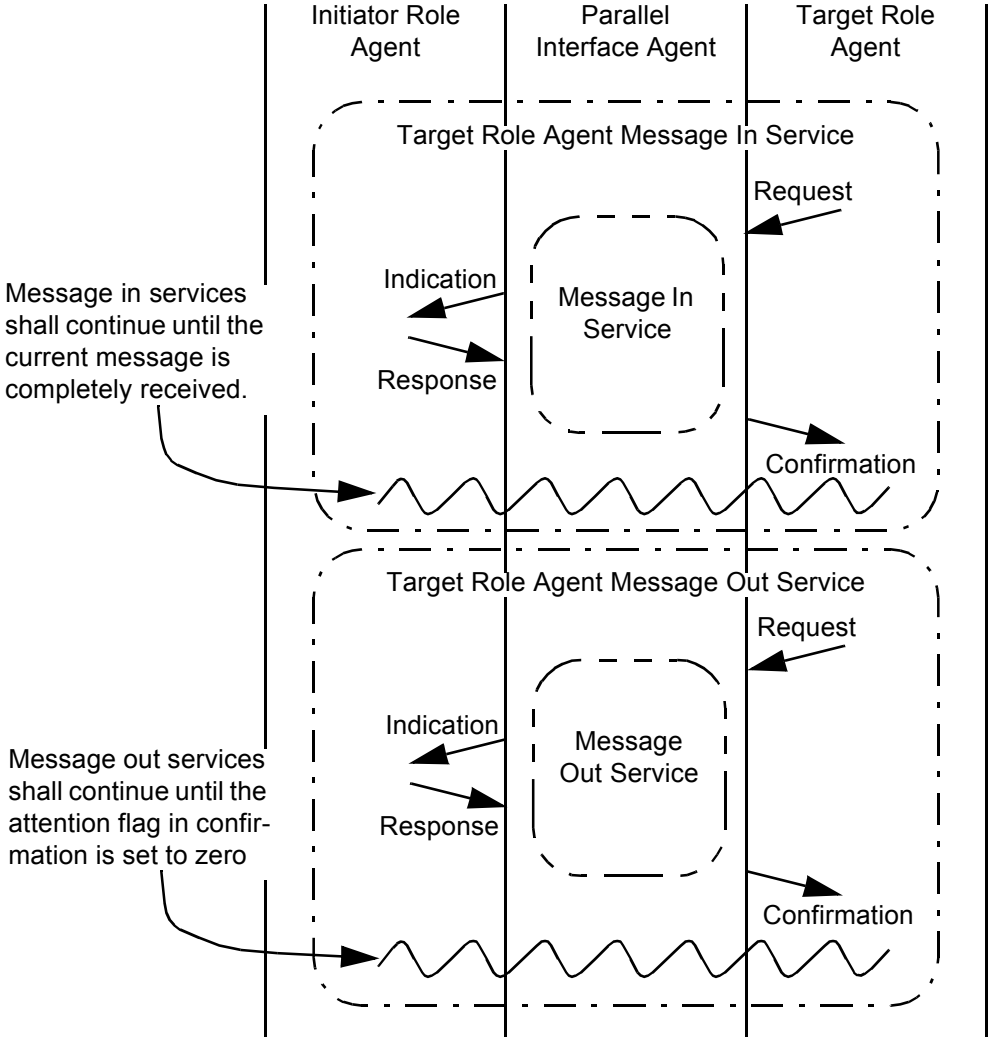


Figure 56 - Target role agent message in service request sequencing (attention flag = 1)

7 SCSI pointers

The SCSI Interlocked Protocol provides for a set of three pointers for each task, called the saved pointers. The set of three pointers consist of one for the command, one for the data, and one for the status. When a send command service is received from an application client, the task's three saved pointers are copied into the initiator role agent's set of three active pointers. There is only one set of active pointers in each initiator role agent. The active pointers point to the next command, data, or status byte to be transferred between the initiator role agent and the target role agent. The saved and active pointers reside in the initiator role agent.

The saved command pointer always points to the start of the command descriptor block for the task. The saved status pointer always points to the start of the status area for the task. The saved data pointer points to the start of the data area until the target role agent sends a SAVE DATA POINTER message for the task.

In response to the SAVE DATA POINTER message, the initiator role agent stores the value of the active data pointer into the saved data pointer for that task. The target role agent may restore the active pointers to the saved pointer values for the current task by sending a RESTORE POINTERS message to the initiator role agent. The initiator role agent then copies the set of saved pointers into the set of active pointers. Whenever a target role agent disconnects from the bus, only the set of saved pointers are retained. The set of active pointers is restored from the set of saved pointers upon reconnection of the task.

Since the data pointer value may be modified by the target role agent before the task ends, it should not be used to test for actual transfer length because the value may no longer be valid.

8 SCSI Interlocked Protocol messages

SCSI Interlocked Protocol messages allow communication between an initiator role agent and a target role agent for the purpose of link management. The link management messages used for this purpose are defined within this standard and their use is confined to this standard. Other SCSI Interlocked Protocol messages allow communication between the application client and the task manager for the purpose of task management. The task management messages are defined in the SCSI-3 Architecture Model Standard, however, their binary values for the SCSI-3 Interlocked Protocol Standard are defined by this standard.

8.1 Message protocols and formats

8.1.1 Message service protocol rules

One or more messages may be communicated between agents in consecutive message in services or message out services, but a single message shall be communicated in consecutive message in services or message out services.

8.1.2 Message protocol rules

The first message sent by the initiator role agent after a successful selection service shall be an IDENTIFY, ABORT TASK SET, or TARGET RESET message. If a target role agent receives any other message it shall generate a bus free service (see Unexpected Bus Free, 9.5).

If the first message is an IDENTIFY message, then it may be immediately followed by other messages, such as the first of a pair of SYNCHRONOUS DATA TRANSFER REQUEST messages. With tagged queuing a task attribute shall immediately follow the IDENTIFY message, then more messages may immediately follow. The IDENTIFY message establishes a logical connection between the initiator and the specified logical unit within the target known as an I_T_L nexus. After the reselection service, the target role agent’s first message shall be IDENTIFY. This allows the I_T_L nexus to be re-established. Only one logical unit shall be identified for any connection; if a target role agent receives a second IDENTIFY message with a different logical unit number during a connection, it shall cause an unexpected bus free by generating a bus free service (see 9.5).

All initiators shall implement the mandatory messages tabulated in the "Initiator" column of tables 8, 16, and 21. All targets shall implement the mandatory messages tabulated in the "Target" column of tables 8, 16, and 21.

The initiator role agent is required to notify the target role agent that a consecutive series of message out services are ending by clearing the attention flag in the message out response of the last message byte when it sends certain messages identified in tables 8, 16, and 21. These messages are identified by a “Yes” entry in the column headed “Negate ATN Before Dropping ACK”.

Whenever an I_T_L nexus is established by an initiator that is allowing disconnection, the initiator role agent shall ensure that the active pointers are equal to the saved pointers for that particular logical unit. An implied restore pointers operation shall occur as a result of a reconnection service.

8.1.3 Message formats

One-byte, Two-byte, and Extended message formats are defined. The first byte of the message determines the format as defined in table 6.

Table 6 - Message format

Code	Message format
00h	One-byte message (TASK COMPLETE)
01h	Extended messages
02h - 13h	One-byte messages
14h - 15h	Reserved One-byte messages
16h - 17h	One-byte messages
18h - 1Fh	Reserved One-byte messages
20h - 24h	Two-byte messages
25h - 2Fh	Reserved Two-byte messages
30h - 7Fh	Reserved
80h - FFh	One-byte message (IDENTIFY)

8.1.3.1 One-byte messages

One-byte messages consist of a single byte transferred during a message in service or a message out service. The code of the byte determines which message is to be performed as defined in tables 8, 16, and 21.

8.1.3.2 Two-byte messages

Two-byte messages consist of two consecutive bytes transferred during two consecutive message in services or two consecutive message out services. The code of the first byte determines which message is to be performed as defined in tables 8, 16, and 21. The second byte is a parameter byte which is used as defined in the message description.

8.1.3.3 Extended messages

A value of 01h in the first byte of a message indicates the beginning of a multiple-byte extended message. The minimum number of bytes sent for an extended message is three. All of the extended message bytes shall be transferred in consecutive message in services or consecutive message out services. The extended message format is shown in table 7.

Table 7 - Extended message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01h)							
1	EXTENDED MESSAGE LENGTH (n-1)							
2	EXTENDED MESSAGE CODE (y)							
3-n	EXTENDED MESSAGE ARGUMENTS							

The EXTENDED MESSAGE LENGTH specifies the length in bytes of the EXTENDED MESSAGE CODE plus the extended message arguments to follow. Therefore, the total length of the message is equal to the EXTENDED MESSAGE LENGTH plus two. A value of zero for the EXTENDED MESSAGE LENGTH indicates 256

bytes follow.

The EXTENDED MESSAGE CODEs are listed in table 8.

The EXTENDED MESSAGE ARGUMENTS are specified within the extended message descriptions (see 8.2.8, 8.2.12, and 8.2.15).

8.2 Link control messages

Table 8 - Link control message codes

Code	Support		Message Name	Direction		Negate ATN before last ACK
	Initiator	Target				
12h	O	O	CONTINUE TASK		Out	Yes
04h	O	O	DISCONNECT	In		n/a
04h	O	O	DISCONNECT		Out	Yes
80h+	M	O	IDENTIFY	In		n/a
80h+	M	M	IDENTIFY		Out	Not required
23h	O	O	IGNORE WIDE RESIDUE	In		n/a
05h	M	M	INITIATOR DETECTED ERROR		Out	Yes
09h	M	M	MESSAGE PARITY ERROR		Out	Yes
07h	M	M	MESSAGE REJECT	In	Out	Yes
***	O	O	MODIFY DATA POINTER	In		n/a
08h	M	M	NO OPERATION		Out	Yes
03h	O	O	RESTORE POINTERS	In		n/a
02h	O	O	SAVE DATA POINTER	In		n/a
***	O	O	SYNCHRONOUS DATA TRANSFER REQUEST	In	Out	Yes
13h	O	O	TARGET TRANSFER DISABLE		Out	Yes
00h	M	M	TASK COMPLETE	In		n/a
***	O	O	WIDE DATA TRANSFER REQUEST	In	Out	Yes

Key: M=Mandatory support, O=Optional support
In=Target role agent to initiator role agent, Out=Initiator role agent to target role agent
Yes=Initiator parallel interface agent shall negate ATN before last ACK of message (see SCSI-3 Parallel Interface Standard)
Not required=Initiator parallel interface agent may or may not negate ACK before last ACK of message (see SCSI-3 Parallel Interface Standard)
n/a=Not applicable
***=Extended message
80h+=Codes 80h through FFh are used for IDENTIFY messages

8.2.1 CONTINUE TASK

The CONTINUE TASK message is sent from the initiator role agent to the target role agent to reconnect to a task. This message shall be sent as one of the messages within the consecutive message out services sent after the IDENTIFY message.

Thus the messages within the consecutive message out services following a selection service consists of the IDENTIFY, task attribute (if any), and CONTINUE TASK messages.

The purpose of the CONTINUE TASK message is to distinguish a valid initiator role agent attempt at a reconnection service from an incorrect initiator reconnection service.

If the target role agent expects a significant delay before it will be ready to continue processing the reconnected task, it may attempt to free the SCSI bus by sending a DISCONNECT message to the initiator role agent. The initiator role agent may reject the disconnection attempt by responding with MESSAGE REJECT message.

If the CONTINUE TASK message occurs on an initial connection then the target role agent shall generate a bus free service.

If the CONTINUE TASK message occurs on a subsequent connection then the target role agent may either treat this as a dynamic head-of-queue request or it may reject the message with a MESSAGE REJECT message.

An initiator role agent that gets rejected should set the attention flag to one and send an ABORT TAG message on the resulting message out service. Otherwise, the target role agent may treat the connection as an overlapped command (see 9.4).

Initiator role agents should avoid sending this message to target role agents that have not implemented this message. Such target role agents may not respond as described in this section. An application client can determine whether a device server implements this message by examining the TRANDIS bit in the standard INQUIRY data (see SCSI-3 Primary Commands Standard). The application client shall inform the initiator role agent to use the CONTINE TASK message by issuing a TARGET TRANSFER DISABLE link control function in the send SCSI command service.

8.2.2 DISCONNECT

The DISCONNECT message is sent from a target role agent to inform an initiator role agent that the present connection is going to be ended and that a later reconnect will be required in order to complete the task. This message shall not cause the initiator role agent to save the data pointer. After successfully sending this message, the target role agent shall generate a bus free service. The target role agent shall consider the message transmission to be successful when it receives a message in confirmation with the attention flag cleared.

Target role agents that are requested to break data transfers into multiple connections shall end each successful connection (except possibly the last) with a SAVE DATA POINTER - DISCONNECT message sequence.

This message may also be sent from an initiator role agent to a target role agent to instruct the target role agent to disconnect from the SCSI bus. If this option is supported, and after the DISCONNECT message is received, the target role agent shall generate a message in service, send the DISCONNECT message to the initiator role agent (possibly preceded by SAVE DATA POINTER message), and then generate a bus free service. After issuing the bus free service the target role agent shall not generate a reselection service for at least a disconnect delay or the time limit specified in the DISCONNECT TIME LIMIT mode parameter (see 9.7) whichever is greater. If this option is not supported or the target role agent cannot disconnect at the time when it receives the DISCONNECT message from the initiator role agent, the target role agent shall respond by sending a MESSAGE REJECT message to the initiator role agent.

8.2.3 IDENTIFY

The IDENTIFY message (see table 9) is sent by either the initiator role agent or the target role agent to establish an I_T_L nexus.

Table 9 - IDENTIFY message format

Bit / Byte	7	6	5	4	3	2	1	0
0	IDENTIFY	DISCPRIV	LUN					

The IDENTIFY bit shall be set to one to specify that this is an IDENTIFY message.

A disconnect privilege (DISCPRIV) bit of one specifies that the initiator role agent has granted the target role agent the privilege of disconnecting. A DISCPRIV bit of zero specifies that the target role agent shall not disconnect. This bit is not defined and shall be set to zero when an IDENTIFY message is sent by a target role agent.

The LUN field specifies a logical unit number.

Only one logical unit number shall be identified per task. The initiator role agent may send one or more IDENTIFY messages during a connection. A second IDENTIFY message with a different value in the LUN field shall not be issued before a bus free indication has occurred; if a target role agent receives a second IDENTIFY message with a different value in this field, it shall cause an unexpected bus free (see 9.5) by generating a bus free service. Thus an initiator role agent may change the DISCPRIV bit, but may not attempt to switch to another task. (See the DTDC field of the disconnect/reconnect mode page in the SCSI-3 Primary Commands Standard for additional controls over disconnection.)

An implied RESTORE POINTERS message shall be performed by the initiator role agent prior to issuing the message in response to the message in indication of the IDENTIFY message sent following a reconnection service.

8.2.4 IGNORE WIDE RESIDUE

The IGNORE WIDE RESIDUE message (see table 10) shall be sent from a target role agent to indicate that the number of valid bytes sent in the last data word of a data in service is less than the negotiated transfer width. This message shall be sent immediately following that data in service and prior to any other messages.

Table 10 - IGNORE WIDE RESIDUE message format

Bit / Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (23h)							
1	NUMBER OF BYTES TO IGNORE (01h, 02h, 03h)							

The NUMBER OF BYTES TO IGNORE field indicates the number of invalid data bytes transferred.

8.2.5 INITIATOR DETECTED ERROR

The INITIATOR DETECTED ERROR message is sent from an initiator role agent to inform a target role agent that an error has occurred that does not preclude the target role agent from retrying the task. The source of the error may either be related to previous activities on the SCSI parallel interface interconnect or may be internal to the initiator role agent and unrelated to any previous SCSI parallel interface interconnect activity. Although the integrity of the currently active pointers in not assured, a RESTORE POINTERS message or a disconnect followed by a reconnect, shall cause the pointers to be restored to their defined prior state.

8.2.6 MESSAGE PARITY ERROR

The MESSAGE PARITY ERROR message is sent from the initiator role agent to the target role agent to indicate that it received a message in indication with the parity flag set.

In order to indicate its intentions of sending this message, the initiator role agent shall set the attention flag in the message in response of the message in indication which has the parity flag set. This provides an interlock so that the target role agent can determine which message byte has the parity error. If the target role agent receives this message under any other circumstance, it shall cause an unexpected bus free (see 9.5) by generating a bus free service.

If after receiving the MESSAGE PARITY ERROR message the target role agent generates a message in service before generating any other services, the target role agent shall resend the entire message that had the parity error.

8.2.7 MESSAGE REJECT

The MESSAGE REJECT message is sent from either the initiator role agent or target role agent to indicate that the last message or message byte it received was inappropriate or has not been implemented.

In order to indicate its intentions of sending this message, the initiator role agent shall set the attention flag in the message in response of the message in indication that is to be rejected. If the target role agent receives this message under any other circumstance, it shall reject this message.

When a target role agent sends this message, it shall generate a message in service and send this message prior to generating any additional message out services. This provides an interlock so that the initiator role agent can determine which message byte is rejected.

If the attention flag is set in the MESSAGE REJECT confirmation then the target role agent shall return to the message out service. The initiator role agent shall begin any subsequent message out services with the first byte of a message.

8.2.8 MODIFY DATA POINTER

The MODIFY DATA POINTER message (see table 11) is sent from the target role agent to the initiator role agent and requests that the signed ARGUMENT be added (two's complement) to the value of the current data pointer. The Enable Modify Data Pointer (EMDP) bit in the Disconnect-Reconnect mode page (see SCSI-3 Primary Commands Standard) indicates whether or not the target role agent is permitted to issue the MODIFY DATA POINTER message.

Table 11 - MODIFY DATA POINTER message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01h)							
1	EXTENDED MESSAGE LENGTH (05h)							
2	MODIFY DATA POINTER (00h)							
3	(MSB)							
4				ARGUMENT				
5								
6								(LSB)

8.2.9 NO OPERATION

The NO OPERATION message is sent from an initiator role agent in response to a target role agent's request for a message when the initiator role agent does not currently have any other valid message to send.

For example, if the target role agent does not respond to the attention flag until a later service and at that time the original message is no longer valid the initiator role agent may send the NO OPERATION message when the target role agent generates a message out service.

8.2.10 RESTORE POINTERS

The RESTORE POINTERS message is sent from a target role agent to direct the initiator role agent to copy the most recently saved command, data, and status pointers for the task to the corresponding active pointers. The command and status pointers shall be restored to the beginning of the present command and status areas. The data pointer shall be restored to the value at the beginning of the data area in the absence of a SAVE DATA POINTER message or to the value at the point at which the last SAVE DATA POINTER message occurred for that task.

8.2.11 SAVE DATA POINTER

The SAVE DATA POINTER message is sent from a target role agent to direct the initiator role agent to copy the current data pointer to the saved data pointer for the current task.

8.2.12 SYNCHRONOUS DATA TRANSFER REQUEST

SYNCHRONOUS DATA TRANSFER REQUEST (SDTR) messages (see table 12) are used to negotiate a synchronous data transfer agreement between two SCSI devices.

Table 12 - SYNCHRONOUS DATA TRANSFER message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01h)							
1	EXTENDED MESSAGE LENGTH (03h)							
2	SYNCHRONOUS DATA TRANSFER REQUEST (01h)							
3	TRANSFER PERIOD FACTOR							
4	REQ/ACK OFFSET							

The TRANSFER PERIOD FACTOR times four is the value of the transfer period, with one exception. A value of 12 (0Ch) denotes a transfer period value of 50. This is to accommodate the fact that 50 is not evenly divisible by 4, yet is a value that must be supported since it represents the maximum allowable speed within the Fast-20 timings specified for the SCSI-3 parallel interface. Faster timings may be allowed by future SCSI Parallel Interface Standards.

The REQ/ACK OFFSET is the maximum number of data words allowed to be outstanding before a data in or data out response is received. The size of a data word may be 1, 2, or 4 bytes depending on what values, if any, have been previously negotiated through an exchange of WIDE DATA TRANSFER REQUEST messages. The REQ/ACK OFFSET value is chosen to prevent overflow conditions in the device's reception buffer and offset counter. A REQ/ACK OFFSET value of zero shall indicate asynchronous data transfer mode; a value of FFh shall indicate unlimited REQ/ACK offset.

The transfer period and the REQ/ACK offset are passed to the parallel interface agent in data out and data in requests. (see 5.2.1.1 and 5.2.2.1).

An SDTR agreement applies to all logical units of the two SCSI devices that negotiated agreement. That is, if SCSI device A, acting as an initiator role agent negotiates a synchronous data transfer agreement with SCSI device B (in the target role), then the same data transfer agreement applies to SCSI devices A and B even if SCSI device B changes to an initiator role.

A synchronous data transfer agreement only applies to the two SCSI devices that negotiate the agreement. Separate synchronous data transfer agreements are negotiated for each pair of SCSI devices.

An SDTR message exchange shall be initiated by an initiator role agent or a target role agent whenever a previously arranged data transfer agreement may have become invalid. The agreement becomes invalid after any condition which may leave the data transfer agreement in an indeterminate state such as:

a) after a reset indication;
b) after a TARGET RESET message;
c) after a power cycle and;
d) after a successful WDTR message exchange.

In addition, an initiator role agent or a target role agent may initiate an SDTR message exchange whenever it is appropriate to negotiate a new data transfer agreement (either synchronous or asynchronous). Initiator role agents or target role agents that are capable of synchronous data transfers shall not respond to an SDTR message with a MESSAGE REJECT message.

Renegotiation after every selection service is not recommended, since a significant performance impact is likely.

The SDTR message exchange establishes the permissible transfer periods and the REQ/ACK offsets for all logical units on the two SCSI devices. This agreement only applies to data in services and data out services.

The originating initiator role agent or target role agent (the agent that sends the first of the pair of SDTR messages) sets its values according to the rules above to permit it to receive data successfully. If the responding agent can also receive data successfully with these values (or smaller transfer periods or larger REQ/ACK offsets or both), it returns the same values in its SDTR message. If it requires a larger transfer period, a smaller REQ/ACK offset, or both in order to receive data successfully, it substitutes values in its SDTR message as required, returning unchanged any value not required to be changed. The initiator role agent and target role agent when transmitting data shall respect the limits set by the other's SDTR message, but it is permitted to transfer data with larger transfer periods, smaller REQ/ACK offsets, or both than specified in the other's SDTR message. The successful completion of an exchange of SDTR messages implies an agreement as shown in table 13.

Table 13 - SDTR messages implied agreements

Responding agent SDTR response	Implied agreement
Non-zero REQ/ACK OFFSET	Synchronous transfer (i.e., the initiator role agent and the target role agent transmit data with a transfer period equal to or greater than and a REQ/ACK offset equal to or less than the values received in the responding device's SDTR message).
REQ/ACK OFFSET equal to zero	Asynchronous transfer
MESSAGE REJECT message	Asynchronous transfer

The implied synchronous agreement shall remain in effect until a TARGET RESET message is received, until a reset indication occurs, or until either the initiator role agent or the target role agent elects to modify the agreement. The default data transfer mode is asynchronous data transfer mode. The default data transfer mode is entered at power on, after a TARGET RESET message, after a successful WDTR exchange, or after a reset indication.

8.2.12.1 Target role agent initiated SDTR negotiation

If the target role agent recognizes that SDTR negotiation is required, it sends an SDTR message to the initiator role agent.

The initiator role agent shall set the attention flag for the message in response for one of the bytes of the SDTR message. In the following message out service the initiator role agent shall respond with an SDTR message or with a MESSAGE REJECT message.

If an abnormal condition prevents the initiator role agent from issuing a message out response both devices shall return to their default agreement.

Following an initiator role agent's responding SDTR message, an implied agreement for synchronous operation shall not be considered to exist until;

a) the target role agent receives a message out confirmation with the last byte of the SDTR message, the parity flag set to zero, and the attention flag set to zero, indicating that the initiator role agent has accepted the SDTR negotiation;
b) and the initiator role agent receives an indication from any service other than a message out service.

After a vendor-specific number of retry attempts (greater than zero), if the target role agent has not received the initiator role agent's responding SDTR message, it shall generate a bus free service without any further information transfer attempt (see 9.5). This indicates that a catastrophic error condition has occurred. The initiator role agent and the target role agent shall use the asynchronous data transfer mode for data transfers between the two devices.

If, following an initiator role agent's responding SDTR message, the target role agent generates a message in service and the first message in is MESSAGE REJECT, the implied agreement shall be considered to be negated and both agents shall use the asynchronous data transfer mode for data transfers between the two devices.

8.2.12.2 Initiator role agent initiated SDTR negotiation

If the initiator role agent recognizes that SDTR negotiation is required, it sets the attention flag in the response of the next service indication.

In the subsequent message out service the initiator role agent shall respond with an SDTR message.

If an abnormal condition prevents the target role agent from generating a message out service with an SDTR message both the initiator role agent and the target role agent shall return to their default agreement.

Following a target role agent's responding SDTR message, an implied agreement for synchronous operation shall not be considered to exist until;

a) the initiator role agent receives a message in indication with the last byte of the SDTR message and the parity flag set to zero, indicating that the target role agent has accepted the SDTR negotiation;
b) and the target role agent receives a message in confirmation for the last byte of the SDTR message with the attention flag set to zero.

If during the SDTR message the target role agent receives a message in confirmation with an attention flag set to one and then the first message of the message out service either a MESSAGE PARITY ERROR or MESSAGE REJECT is received the synchronous operation shall be considered to be negated by both the initiator role agent and the target role agent. In this case, both the initiator role agent and the target role agent shall return to their default agreement.

For the MESSAGE PARITY ERROR case, the implied agreement shall be reinstated if a retransmitted of the second of the pair of messages is successfully accomplished.

After a vendor-specific number of retry attempts (greater than zero), if the target role agent continues to receive MESSAGE PARITY ERROR messages, it shall terminate the retry activity. This may be done either by generating any service other than a message in or message out service and transferring at least one byte of information or by generating a bus free service. The initiator role agent shall accept such action as aborting the SDTR negotiation, and both the initiator role agent and the target role agent shall return to their default agreement.

8.2.13 TARGET TRANSFER DISABLE

The TARGET TRANSFER DISABLE message is sent from an initiator role agent to a target role agent to request that subsequent reconnections for data transfer be done by the initiator role agent instead of the target role agent. The target role agent might reconnect for other purposes, but shall not generate any data in service or data out services after a target role agent reconnection. SCSI devices that implement this message shall also implement the CONTINUE TASK message.

This message shall be sent as the last message of the series of consecutive message out services of an initial connection. The target role agent may continue the task, including any data out services on the initial

connection, until the target role agent would normally disconnect, but the target role agent shall not reconnect to transfer data. That is, the target role agent shall not generate a data in service on the initial connection and the target role agent shall not generate any data in or data out services on any subsequent target role agent reconnection for the task.

When the target role agent is ready to transfer data for a disconnected task for which a TARGET TRANSFER DISABLE message has been sent, the target role agent shall reconnect to the initiator role agent for the task (via a reselection service and consecutive message in services containing an IDENTIFY message, and an optional SIMPLE TASK message), send a DISCONNECT message, and, if the initiator role agent does not respond with a MESSAGE REJECT message, generate a bus free service. This connection serves to notify the initiator role agent that the task is ready for data transfer. If the initiator role agent rejects the DISCONNECT message, the target role agent may generate a data in or data out service; otherwise, the initiator role agent may reconnect to the task as described in the CONTINUE TASK message to perform the data transfer.

Initiator role agents should avoid sending the TARGET TRANSFER DISABLE message to target role agents that have not implemented this message. Such target role agents may not respond as described in this section. An application client can determine whether a device server implements this message by examining the (TRANDIS) bit in the standard INQUIRY data (see SCSI-3 Primary Commands Standard). The application client shall inform the initiator role agent to use the CONTINE TASK message by issuing a TARGET TRANSFER DISABLE link control function in the send SCSI command service.

8.2.14 TASK COMPLETE

The TASK COMPLETE message is sent from a target role agent to an initiator role agent to indicate that a task has completed and that valid status has been sent to the initiator role agent. After successfully sending this message, the target role agent generates a bus free service. The target role agent shall consider the message transmission to be successful on receipt of a message in confirmation with the attention flag cleared.

The task may have completed successfully or unsuccessfully as indicated in the status.

8.2.15 WIDE DATA TRANSFER REQUEST

WIDE DATA TRANSFER REQUEST (WDTR) messages (see table 14) are used to negotiate a wide data transfer agreement between two SCSI devices.

Table 14 - WIDE DATA TRANSFER message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01h)							
1	EXTENDED MESSAGE LENGTH (02h)							
2	WIDE DATA TRANSFER REQUEST (03h)							
3	TRANSFER WIDTH EXPONENT (m)							

The TRANSFER WIDTH EXPONENT defines the transfer width to be used during data in services and date out services. The transfer width that is established applies to all logical units on both SCSI devices. Valid transfer widths are 8 bits (m=00h), 16 bits (m=01h), and 32 bits (m=02h). Values of m greater than 02h are reserved.

The transfer width is passed to the parallel interface agent as the data bus width in data out and data in requests. (see 5.2.1.1 and 5.2.2.1).

A WDTR agreement applies to all logical units of the two SCSI devices that negotiated agreement. That is, if SCSI device A, acting as an initiator role agent negotiates a wide data transfer agreement with SCSI device B (in the target role), then the same transfer width agreement applies to SCSI devices A and B even if SCSI device B changes to an initiator role.

A wide data transfer agreement only applies to the two SCSI devices that negotiate the agreement. Separate wide transfer agreements are negotiated for each pair of SCSI devices.

A WDTR message exchange shall be initiated by an initiator role agent or a target role agent whenever a previously arranged wide transfer agreement may have become invalid. The agreement becomes invalid after any condition which may leave the wide transfer agreement in an indeterminate state such as:

a) after a reset indication;
b) after a TARGET RESET message; and
c) after a power cycle.

In addition, an initiator role agent or a target role agent may initiate a WDTR message exchange whenever it is appropriate to negotiate a new wide transfer agreement. Initiator role agents or target role agents that are capable of wide data transfers (greater than 8 bits) shall not respond to a WDTR message with a MESSAGE REJECT message.

Renegotiation after every selection service is not recommended, since a significant performance impact is likely.

The WDTR message exchange establishes an agreement between the two SCSI devices on the width of the data path to be used during data in and data out services between the two SCSI devices. This agreement only applies to data in services and data out services. All other services shall use an eight-bit data path.

If a SCSI device implements both wide data transfer option and synchronous data transfer option, then it shall negotiate the wide data transfer agreement prior to negotiating the synchronous data transfer agreement. If a synchronous data transfer agreement is in effect, then:

a) if a WDTR message is rejected with a MESSAGE REJECT message the prior synchronous data transfer agreement shall remain intact; or
b) if a WDTR message is not rejected with a MESSAGE REJECT message a WDTR message shall reset the synchronous data transfer agreement to asynchronous mode.

The originating initiator role agent or target role agent (the agent that sends the first of the pair of WDTR messages) sets its transfer width value to the maximum data path width it elects to accommodate. If the responding agent can also accommodate this transfer width, it returns the same value in its WDTR message. If it requires a smaller transfer width, it substitutes the smaller value in its WDTR message. The successful completion of an exchange of WDTR messages implies an agreement as shown in table 15.

Table 15 - WDTR messages implied agreements

Responding agent WDTR response	Implied agreement
Non-zero TRANSFER WIDTH EXPONENT	Wide transfer (i.e., the initiator role agent and the target role agent transmit data with a transfer width equal to the responding device's transfer width). If the initiating agent does not support the responding agents TRANSFER WIDTH EXPONENT then the initiating agent shall MESSAGE REJECT the WDTR message (see 8.2.15.1 and 8.2.15.2).
TRANSFER WIDTH equal to zero	Eight-bit data transfer
MESSAGE REJECT message	Eight-bit data transfer

The implied transfer width agreement shall remain in effect until a TARGET RESET message is received, until a reset indication occurs, or until either the initiator role agent or the target role agent elects to modify the agreement. The default data transfer width is eight-bit data transfer mode. The default data transfer width is entered at power on, after a TARGET RESET message, or after a reset indication.

8.2.15.1 Target role agent initiated WDTR negotiation

If the target role agent recognizes that WDTR negotiation is required, it sends a WDTR message to the initiator role agent.

The initiator role agent shall set the attention flag for the message in response for one of the bytes of the WDTR message. In the following message out service the initiator role agent shall respond with a WDTR message or with a MESSAGE REJECT message.

If an abnormal condition prevents the initiator role agent from issuing a message out response both devices shall return to their default agreement.

Following an initiator role agent's responding WDTR message, an implied agreement for wide data transfers operation shall not be considered to exist until;

a) the target role agent receives a message out confirmation with the last byte of the WDTR message, the parity flag set to zero, and the attention flag set to zero, indicating that the initiator role agent has accepted the WDTR negotiation; and
b) the initiator role agent receives an indication from any service other than a message out service.

After a vendor-specific number of retry attempts (greater than zero), if the target role agent has not received the initiator role agent's responding WDTR message, it shall generate a bus free service without any further information transfer attempt (see 9.5). This indicates that a catastrophic error condition has occurred. The initiator role agent and the target role agent shall use the asynchronous data transfer mode for data transfers between the two agents.

If the target role agent does not support the initiator role agent's responding TRANSFER WIDTH EXPONENT the target role agent shall generate a message in service and the first message is MESSAGE REJECT, the implied agreement shall be considered to be negated and both agents shall use the eight-bit data transfer mode for data transfers between the two devices.

If, following an initiator role agent's responding WDTR message, the target role agent generates a message in service and the first message is MESSAGE REJECT, the implied agreement shall be considered to be negated and both agents shall use the eight-bit data transfer mode for data transfers

between the two devices.

8.2.15.2 Initiator role agent initiated WDTR negotiation

If the initiator role agent recognizes that WDTR negotiation is required, it sets the attention flag in the response of the next service indication.

In the following message out service the initiator role agent shall respond with a WDTR message.

If an abnormal condition prevents the target role agent from generating a message out service with a WDTR message both the initiator role agent and the target role agent shall return to their default agreement.

Following a target role agent's responding WDTR message, an implied agreement for wide data transfers shall not be considered to exist until;

a) the initiator role agent receives a message in indication with the last byte of the WDTR message and the parity flag set to zero, indicating that the target role agent has accepted the WDTR negotiation; and
b) the target role agent receives a message in confirmation for the last byte of the WDTR message with the attention flag set to zero.

If the initiator role agent does not support the target role agent's responding TRANSFER WIDTH EXPONENT the initiator role agent shall set the attention flag for the message in response for one of the bytes of the WDTR message. In the following message out service the initiator role agent shall respond with a MESSAGE REJECT message.

If during the WDTR message the target role agent receives a message in confirmation with an attention flag set to one and then the first message of the message out service is either a MESSAGE PARITY ERROR or MESSAGE REJECT message the wide data transfers shall be considered to be negated by both the initiator role agent and the target role agent. In this case, both the initiator role agent and the target role agent shall use the eight-bit data transfer mode for data transfers between the two devices.

For the MESSAGE PARITY ERROR case, the implied agreement shall be reinstated if a retransmission of the second of the pair of messages is successfully accomplished.

After a vendor-specific number of retry attempts (greater than zero), if the target role agent continues to receive MESSAGE PARITY ERROR messages, it shall terminate the retry activity. This may be done either by generating any service other than a message in or message out service and transferring at least one byte of information or by generating a bus free service. The initiator role agent shall accept such action as aborting the WDTR negotiation, and both the initiator role agent and the target role agent shall return to their default agreement.

8.3 Task attribute messages

Two byte task attribute messages are used to specify an identifier, called a tag, for a task which establishes the I_T_L_Q nexus. The tag field is an 8-bit unsigned integer assigned by the application client and sent to the initiator role agent in the send SCSI command service. The tag for every task for each I_T_L nexus should be unique. If the task manager receives a tag that is currently in use for the I_T_L nexus, then it shall respond as defined in 9.4. A tag becomes available for reassignment when the task ends. The numeric value of a tag is arbitrary, providing there are no outstanding duplicates, and has no effect on the order of execution.

For each logical unit on each target, each application client has up to 256 tags to assign to tasks. Thus a target with eight logical units could have up to 14 336 tasks concurrently in existence if there were seven initiators on the bus.

After an initiator role agent generates a successful selection service, the appropriate task attribute message shall be sent immediately following the IDENTIFY message to establish the I_T_L_Q nexus for the task. Only one I_T_L_Q nexus may be established during a connection. If a task attribute message is not sent, then only an I_T_L nexus is established for the task (i.e., an untagged command).

Whenever a target role agent generates a reconnection service to continue a tagged task, the SIMPLE QUEUE message shall be sent immediately following the IDENTIFY message to resume the I_T_L_Q nexus for the task. Only one I_T_L_Q nexus may occur during a reconnection. If the SIMPLE TAG message is not sent, then only an I_T_L nexus occurs for the task (i.e., an untagged command).

If a target attempts to reconnect using an invalid tag, then the initiator role agent should set the attention flag to one in the message in confirmation. After the corresponding message out service the initiator role agent shall respond with an ABORT TASK message.

If a target does not implement tagged queuing and a queue tag message is received the target role agent shall generate a message in service with a MESSAGE REJECT message and accept the task as if it were untagged provided there are no outstanding untagged tasks from that initiator role agent.

See SCSI-3 Architecture Model Standard for the task set management rules.

Table 16 - Task attribute message codes

Code	Support		Message Name	Direction		Negate ATN before last ACK
	Initiator	Target				
24h	O	O	ACA		Out	Not required
21h	Q	Q	HEAD OF QUEUE		Out	Not required
0Ah	O	O	LINKED COMMAND COMPLETE	In		n/a
0Bh	O	O	LINKED COMMAND COMPLETE (WITH FLAG)	In		n/a
22h	Q	Q	ORDERED		Out	Not required
20h	Q	Q	SIMPLE	In	Out	Not required

Key: M=Mandatory support, O=Optional support, Q=Mandatory if tagged queuing is implemented
In=Target role agent to initiator role agent, Out=Initiator role agent to target role agent
Yes=Initiator parallel interface agent shall negate ATN before last ACK of message (see SCSI-3 Parallel Interface Standard)
Not required=Initiator parallel interface agent may or may not negate ACK before last ACK of message (see SCSI-3 Parallel Interface Standard)
n/a=Not applicable
***=Extended message

8.3.1 ACA

See table 17 for the format of the ACA message.

Table 17 - ACA message format

Bit / Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (24h)							
1	TAG (00h-FFh)							

The ACA message specifies that the task shall be placed in the task set as an ACA task. The rules used by the task manager to handle ACA tasks within a task set are defined in the SCSI-3 Architecture Model Standard.

8.3.2 HEAD OF QUEUE

See table 18 for the format of the HEAD OF QUEUE message.

Table 18 - HEAD OF QUEUE message format

Bit / Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (21h)							
1	TAG (00h-FFh)							

The HEAD OF QUEUE message specifies that the task shall be placed in the task set as a HEAD OF QUEUE task. The rules used by the device server to handle HEAD OF QUEUE tasks within a task set are defined in the SCSI-3 Architecture Model Standard.

8.3.3 LINKED COMMAND COMPLETE

The LINKED COMMAND COMPLETE message is sent from a target role agent to an initiator role agent to indicate that a linked command has completed and that status has been sent. The initiator role agent shall then set the pointers to the initial state for the next linked command.

8.3.4 LINKED COMMAND COMPLETE (WITH FLAG)

The LINKED COMMAND COMPLETE (WITH FLAG) message is sent from a target role agent to an initiator role agent to indicate that a linked command (with the flag bit set to one) has completed and that status has been sent. The initiator role agent shall then set the pointers to the initial state of the next linked command. Typically this message would be used to cause an interrupt in the application client between two linked commands.

8.3.5 ORDERED

See table 19 for the format of the ORDERED message.

Table 19 - ORDERED message format

Bit / Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (22h)							
1	TAG (00h-FFh)							

The ORDERED message specifies that the task shall be placed in the task set as an ORDERED task. The rules used by the task manager to handle ORDERED tasks within a task set are defined in the SCSI-3 Architecture Model Standard.

8.3.6 SIMPLE

See table 20 for the format of the SIMPLE message.

Table 20 - SIMPLE message format

Bit / Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (20h)							
1	TAG (00h-FFh)							

The SIMPLE message specifies that the task shall be placed in the task set as a SIMPLE task. The rules used by the task manager to handle SIMPLE tasks within a task set are defined in the SCSI-3 Architecture Model Standard.

8.4 Task management messages

Table 21 - Task management message codes

Code	Support		Message Name	Direction		Negate ATN before last ACK
	Initiator	Target				
0Dh	Q	Q	ABORT TASK		Out	Yes
06h	O	M	ABORT TASK SET		Out	Yes
16h	O	O	CLEAR ACA		Out	Not required
0Eh	Q	Q	CLEAR TASK SET		Out	Yes
17h	O	O	LOGICAL UNIT RESET (Note)		Out	Yes
0Ch	O	M	TARGET RESET		Out	Yes
11h	O	O	TERMINATE TASK		Out	Yes

Key: M=Mandatory support, O=Optional support, Q=Mandatory if tagged queuing is implemented
In=Target role agent to initiator role agent, Out=Initiator role agent to target role agent
Yes=Initiator parallel interface agent shall negate ATN before last ACK of message (see SCSI-3 Parallel Interface Standard)
Not required=Initiator parallel interface agent may or may not negate ACK before last ACK of message (see SCSI-3 Parallel Interface Standard)
n/a=Not applicable
***=Extended message

Note-The LOGICAL UNIT RESET message is mandatory if hierarchical addressing (see SCSI-3 Controller Command Standard) is implemented by the target.

8.4.1 ABORT TASK

The ABORT TASK message is defined in the SCSI-3 Architecture Model Standard.

The task management function services for the ABORT TASK function are defined in 5.3.1.2.

If the ABORT TASK message is received by the target role agent as one of the messages in the consecutive message out services following a reselection service, the ABORT TASK message aborts the current task.

8.4.2 ABORT TASK SET

The ABORT TASK SET message is defined in the SCSI-3 Architecture Model Standard.

The task management function services for the ABORT TASK SET function are defined in 5.3.1.2.

If only an I_T nexus has been established, the target role agent shall generate a bus free service. No status or message shall be sent for the current task and no other task shall be affected.

The ABORT TASK SET message in the case of only an I_T nexus is useful to an initiator role agent that cannot get an IDENTIFY message through to the target role agent due to parity errors and just needs to end the current connection. No pending data, status, or tasks are affected.

It is not an error to issue this message to an I_T_L nexus that does not have any pending or current tasks.

8.4.2.1 CLEAR ACA

The CLEAR ACA message is defined in the SCSI-3 Architecture Model Standard.

The task management function services for the CLEAR ACA function are defined in 5.3.1.2.

It is not an error to issue a CLEAR ACA message when no ACA condition is in effect.

8.4.2.2 CLEAR TASK SET

The CLEAR TASK SET message is defined in the SCSI-3 Architecture Model Standard.

The task management function services for the CLEAR TASK SET function are defined in 5.3.1.2.

8.4.2.3 LOGICAL UNIT RESET

The LOGICAL UNIT RESET message is defined in the SCSI-3 Architecture Model Standard.

The task management function service for the LOGICAL UNIT RESET function are defined in 5.3.1.2.

If only an I_T nexus has been established the LOGICAL UNIT RESET shall be performed as a TARGET RESET.

8.4.2.4 TARGET RESET

The TARGET RESET message is defined in the SCSI-3 Architecture Model Standard.

The task management function services for the TARGET RESET function are defined in 5.3.1.2.

The TARGET RESET message shall not change any SCSI IDs.

8.4.2.5 TERMINATE TASK

The TERMINATE TASK message is defined in the SCSI-3 Architecture Model Standard.

The task management function services for the TERMINATE TASK function are defined in 5.3.1.2.

If the task manager does not support this message or is unable to stop the current task, it shall send a MESSAGE REJECT message to the initiator role agent and continue the task in a normal manner.

9 Command processing considerations and exception conditions

The following subclauses describe some aspects of command processing, including exception conditions and error handling which are specific to the SCSI-3 Interlocked Protocol Standard.

9.1 Asynchronous event notification

Notification of an asynchronous event is performed using the SEND command with the AER bit set to one. The information identifying the condition being reported shall be returned during the data-out delivery service of the SEND command (see SCSI-3 Primary Commands Standard).

An error condition or unit attention condition shall be reported once per occurrence of the event causing it. The target may choose to use an asynchronous event notification or to return CHECK CONDITION status on a subsequent command, but not both. Notification of command-related error conditions shall be sent only to the initiator that requested the task.

The asynchronous event notification protocol can be used to notify processor devices that a system resource has become available. If a target chooses to use this method, the sense key in the sense data sent to the processor device shall be set to UNIT ATTENTION.

The asynchronous event notification protocol shall be used only with SCSI devices that return processor device type with an AERC bit of one in response to an INQUIRY command. The INQUIRY command should be sent to logical unit zero of each SCSI device responding to selection. This procedure shall be conducted prior to the first asynchronous event notification and shall be repeated whenever the device deems it appropriate or when an event occurs that may invalidate the current information. (See SYNCHRONOUS DATA TRANSFER REQUEST message (8.2.12) for examples of these events.)

Each SCSI device that returns processor device type with an AERC bit of one shall be issued a TEST UNIT READY command to determine that the SCSI device is ready to receive an asynchronous event notification. A SCSI device returning CHECK CONDITION status is issued a REQUEST SENSE command. This clears any pending unit attention condition. A SCSI device that returns processor device type with an AERC bit of one and returns GOOD status when issued a TEST UNIT READY command shall accept a SEND command with an AER bit of one.

> NOTE 6 - A SCSI device which can use asynchronous event notification at initialization time should provide means to defeat these notifications. This can be done with a switch or jumper wire. Devices which implement saved parameters may alternatively save the asynchronous event notification permissions either on a per SCSI device basis or as a system wide option. In any case, a device conducts a survey with INQUIRY commands to be sure that the devices on the SCSI bus are appropriate destinations for SEND commands with an AER bit of one. (The devices on the bus or the SCSI ID assignments may have changed.)

See asynchronous event reporting in the SCSI-3 Architecture Model Standard for more information on asynchronous event notification.

9.2 Attention condition

The attention condition allows the initiator role agent to inform a target role agent that the initiator role agent has a message ready. The target role agent may get this message by issuing a message out service.

The initiator role agent creates an attention condition by setting the attention flag to one during a:

a) selection service request,
b) command service response,
c) data out service response,

d) data in service response,
e) status service response,
f) message out service response,
g) message in service response.

The initiator role agent shall set the attention flag to zero during the message out service response of the last byte of the messages indicated with a Yes in tables 8 and 21. If the target role agent detects that an initiator role agent failed to meet this requirement, then the target role agent shall issue a bus free service (see unexpected bus free, 9.5).

A target role agent shall respond with a message out service request as follows:

a) If the attention flag is set to one on a command service confirmation, the target role agent shall issue a message out service request after transferring part or all of the command descriptor block bytes.
b) If the attention flag is set to one on a data in or data out service confirmation, the target role agent shall issue a message out service request at the target role agent's earliest convenience (often, but not necessarily on a logical block boundary). The initiator role agent shall continue to respond to data out services until it receives a message out service indication.
c) If the attention flag is set to one on a status service confirmation, the target role agent shall issue a message out service request after the status service confirmation.
d) If the attention flag is set to one on a message out service confirmation, the target role agent shall issue another message out service request before issuing any other service requests, except when rejecting a message.
e) If the attention flag is set to one on a message in service confirmation, the target role agent shall issue a message out service request before issuing any other service requests after all the message byte(s) from the current message are received. This permits a MESSAGE PARITY ERROR message from the initiator role agent to be associated with the appropriate message.
f) If the attention flag is set to one on a selection service indication, the target role agent shall issue a message out service request before issuing any other service requests.

9.3 Hard reset

In addition to the hard reset characteristics defined in the SCSI-3 Architecture Model Standard the SCSI-3 Interlocked Protocol Standard has the following hard reset characteristics:

a) may or may not change the SCSI ID

9.4 Overlapped commands

Overlapped commands occur on a reconnection if an initiator role agent attempts to reconnect to a task, and then the initiator role agent does not send an ABORT TASK SET, ABORT TASK, TARGET RESET, CLEAR TASK SET, CONTINUE TASK, or TERMINATE TASK message as one of the messages within the consecutive message out services sent after the IDENTIFY message.

A task manager that detects an incorrect initiator connection shall abort all tasks for the initiator and the associated logical unit and shall return CHECK CONDITION status. The sense key shall be set to ABORTED COMMAND and the additional sense code shall be set to TAGGED OVERLAPPED COMMANDS with the additional sense code qualifier set to the value of the duplicate tag.

NOTE 7 - An incorrect initiator connection may be indicative of a serious error and, if not detected, could result in a task operating with a wrong set of pointers. This is considered a catastrophic failure on the part of the initiator. Therefore, vendor-specific error recovery procedures may be required to guarantee the data integrity on the medium. The target may return additional sense data to aid in this error recovery procedure (e.g., sequential-access devices may return the residue of blocks remaining to be written or read at the time the second command was received).

NOTE 8 - Some targets may not detect an incorrect initiator connection until after the command descriptor block has been received.

See the SCSI-3 Architecture Model Standard for more information on overlapped commands.

9.5 Unexpected bus free

An unexpected bus free occurs after a bus free service indication occurs and the initiator role agent does not expect an indication of a bus free service. Initiator role agents only expect an indication of a bus free service to occur after one of the following occurs:

a) reset service,
b) task management function service
c) the following link control messages:
   a) DISCONNECT message when sent from an initiator role agent,
   b) TASK COMPLETE,
d) unsuccessful selection or reselection.

The target role agent uses an unexpected bus free to inform the initiator role agent of a protocol error. The target role agent may generate a bus free request at any time.

The target role agent shall terminate the task that was the current task before the bus free request by clearing all data and status for that task.

The device server may optionally prepare sense data that may be retrieved by a REQUEST SENSE command. However, an unexpected bus free does not create an exception condition.

The initiator role agent shall terminate the task that was the current task before the bus free indication was received and set the service response within the send SCSI command confirmation to SERVICE DELIVERY OR TARGET FAILURE.

9.6 Unexpected reselection

An unexpected reselection occurs if a task manger attempts to reconnect to a task for which a nexus does not exist. An initiator role agent should respond to an unexpected reselection by sending an ABORT TASK message.

9.7 Use of disconnect-reconnect page parameters

The disconnect-reconnect page (see SCSI-3 Primary Commands Standard) provides the application client the means to tune the performance of the service delivery subsystem. The following subclause defines the fields in the disconnect-reconnect mode page of the MODE SENSE or MODE SELECT command that are used by target role agents.

The application client passes the fields used to control the service delivery subsystem to a device server by means of a MODE SELECT command. The device server then communicates the field values to the target role agent. The field values are communicated from the device server to the target role agent in a vendor specific manor.

SCSI-3 interlaced protocol devices shall only use disconnect-reconnect page parameter fields defined below. If any other fields within the disconnect-reconnect page of the MODE SELECT command contain a non-zero value, the device server shall return CHECK CONDITION status for that MODE SELECT command. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN PARAMETER LIST.

The BUFFER FULL RATIO field and BUFFER EMPTY RATIO FIELD are used as described in the SCSI-3 Primary Commands Standard.

The BUS INACTIVITY LIMIT field indicates the maximum time in 100 µs increments that the target role agent is permitted to delay between the receipt of a service confirmation and issuance of the next service request. If the bus inactivity limit is exceeded the target role agent shall attempt to disconnect (see 8.2.2) if the initiator role agent has granted the disconnect privilege (see 8.2.3) and it is not restricted by DTDC. This value may be rounded as defined in the SCSI-3 Primary Commands Standard. A value of zero indicates that there is no bus inactivity limit.

The DISCONNECT TIME LIMIT field indicates the minimum time in 100 µs increments that the target role agent shall wait after issuing a bus free service before attempting a reselection service. This value may be rounded as defined in the SCSI-3 Primary Commands Standard. A value of zero indicates that there is no disconnect time limit.

The CONNECT TIME LIMIT field indicates the maximum time in 100 µs increments that the target role agent is allowed to issue service requests before disconnecting, if the initiator role agent has granted the disconnect privilege (see 8.2.3) and it is not restricted by DTDC. This value may be rounded as defined in the SCSI-3 Primary Commands Standard. A value of zero indicates that there is no connect time limit.

The MAXIMUM BURST SIZE field indicates the maximum number of consecutive data-in services or data-out data services that the target role agent shall issue before disconnecting if the initiator role agent has granted the disconnect privilege (see 8.2.3). This value is expressed in increments of 512 bytes (e.g. a value of one means 512 bytes, two means 1024 bytes, etc.). A value of zero indicates there is no limit on the amount of data-in or data-out services per task.

The enable modify data pointers (EMDP) bit indicates whether or not the initiator role agent allows the MODIFY DATA POINTERS message to be issued by the target role agent. If the EMDP bit is a zero, the target role agent shall not issue the MODIFY DATA POINTERS message. If the EMDP bit is a one, the target role agent is allowed to issue MODIFY DATA POINTERS messages.

If the EMDP bit is a one and the initiator role agent responds to a MODIFY DATA POINTER message with a MESSAGE REJECT, then the device server shall request a status service with the status set to CHECK CONDITION. The sense key shall be set to ABORTED COMMAND and the sense code shall be set to INVALID MESSAGE ERROR.

A disconnect immediate (DIMM) bit of zero indicates that the target role agent may request data-in or data-out services following a command service without attempting a disconnect (see 8.2.2). A DIMM bit of one indicates that the target role agent shall attempt a disconnect (see 8.2.2) between a command service request and subsequent data-in or data-out service request. The DIMM bit only applies when the initiator role agent has granted the disconnect privilege (see 8.2.3).

The DATA TRANSFER DISCONNECT CONTROL (DTDC) field (see table 22) defines further restrictions on when a disconnect is permitted.

TABLE 22 - DATA TRANSFER DISCONNECT CONTROL

DTDC	Description
000b	DATA TRANSFER DISCONNECT CONTROL is not used. Disconnect is controlled by the other fields in this page.
001b	Once it has issued a data-in service or data-out service request, the target role agent shall not disconnect until the last data-in service or data-out service request has been issued for the task. The connect time limit and bus inactivity limit are ignored during the data transfer.
010b	Reserved
011b	Once it has issued a data-in service or data-out service request, the target role agent shall not disconnect until the device server issues the send SCSI command service response (the task is complete). The connect time limit and bus inactivity limit are ignored once data transfer has begun.
100b-111b	Reserved

If DTDC is non-zero and the maximum burst size is non-zero, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN PARAMETER LIST.

Annex A (informative)

A.0 SCSI-3 to SCSI-2 terminology mapping

This annex contains a mapping of terminology used in SCSI-2 to the terminology used in this SCSI-3 standard (see table A.1).

Table A.1 - SCSI-3 to SCSI-2 terminology mapping

SCSI-3 equivalent term	SCSI-2 term
abort task	abort tag
abort task set	abort
clear task set	clear queue
head of queue	head of queue tag
ordered	ordered queue tag
overlapped commands	incorrect initiator connection
service (SIP/SPI services only)	phase
simple	simple queue tag
target reset	bus device reset
task	I/O process
task complete	command complete
task set	queue