Mike Ko
John Hufferd
IBM Corporation
Mallikarjun Chadalapaka
Hewlett-Packard Company
Uri Elzur
Broadcom Corporation
Hemal Shah
Intel Corporation
Patricia Thaler
Agilent Technologies, Inc.

draft-ko-iwarp-iser-v1.0

July, 2003

iSCSI Extensions for RDMA Specification (Version 1.0)

1 Status of this Memo

This document is a Release Specification of the RDMA Consortium. Copies of this document and associated errata may be found at http://www.rdmaconsortium.org.

2 Abstract

iSCSI Extensions for RDMA provides the RDMA data transfer capability to iSCSI [iSCSI] by layering iSCSI on top of the Remote Direct Memory Access Protocol (RDMAP). The iWARP protocol suite provides RDMA Read and Write services, which enable data to be transferred directly into SCSI I/O Buffers without intermediate data copies. This document describes the extensions to the iSCSI protocol to support RDMA services as defined by the iWARP protocol suite.

Table of Contents

Table of Figures

3 Definitions and Acronyms

Some of the following definitions are taken from [RDMAP]. In those definitions, the term ULP refers to the iSER Layer.

3.1 Definitions

Advertisement (Advertised, Advertise, Advertisements, Advertises) – The act of informing a Remote Peer that a local node's buffer is available to it. A Node makes a buffer available for incoming RDMA Read Request Message or incoming RDMA Write Message access by informing its RDMA/DDP peer of the Tagged Buffer identifiers (STag, TO, and buffer length). This Advertisement of Tagged Buffer information is not defined by RDMA/DDP and is left to the ULP. A typical method would be for the Local Peer to embed the Tagged Buffer's STag, TO, and buffer length in a Send Message destined for the Remote Peer.

Completion (Completed, Complete, Completes) – Completion is defined as the process by the iWARP layer to inform the ULP, in this case the iSER Layer, that a particular RDMA Operation has performed all functions specified for the RDMA Operation.

Connection – A connection is a TCP connection. Communication between the initiator and the target occurs over one or more TCP connections. The TCP connections carry control messages, SCSI commands, parameters, and data within iSCSI Protocol Data Units (iSCSI PDUs).

Connection Handle – An information element that identifies the particular iSCSI connection and is unique for a given iSCSI–iSER pair. Every invocation of an Operational Primitive MUST be qualified with the Connection Handle.

Data Sink – The peer receiving a data payload. Note that the Data Sink can be required to both send and receive RDMAP Messages to transfer a data payload.

Data Source – The peer sending a data payload. Note that the Data Source can be required to both send and receive RDMAP Messages to transfer a data payload.

Datamover Interface (DI) – The interface between the iSCSI Layer and the Datamover Layer as described in [DA].

Datamover Layer – A layer that is directly below the iSCSI Layer and above the underlying transport layers. This layer exposes and uses a set of transport independent Operational Primitives for the communication between the iSCSI Layer and itself. The Datamover layer, operating in conjunction with the transport

layers, moves the control and data information on the iSCSI connection. In this specification, the iSER Layer is the Datamover layer.

Datamover Protocol – A Datamover protocol is the wire-protocol that is defined to realize the Datamover layer functionality. In this specification, the iSER protocol is the Datamover protocol.

Event – An indication provided by the RDMAP Layer to the ULP to indicate a Completion or other condition requiring immediate attention.

Inbound RDMA Read Queue Depth (IRD) – The maximum number of incoming outstanding RDMA Read Requests that the RNIC can handle on a particular RDMAP Stream at the Data Source.

Invalidate STag – A mechanism used to prevent the Remote Peer from reusing a previous explicitly Advertised STag, until the Local Peer makes it available through a subsequent explicit Advertisement.

I/O Buffer – A buffer that is used in a SCSI Read or Write operation so SCSI data may be sent from or received into that buffer.

iSCSI – The iSCSI protocol is a mapping of the SCSI remote procedure model of SAM-2 over the TCP, and the protocol itself is defined in [iSCSI].

iSCSI control-type PDU – Any iSCSI PDU that is not an iSCSI data-type PDU and also not a SCSI Data-out PDU carrying solicited data is defined as an iSCSI control-type PDU. Specifically, it is to be noted that SCSI Data-out PDUs for unsolicited data are defined as iSCSI control-type PDUs.

iSCSI data-type PDU – An iSCSI data-type PDU is defined as an iSCSI PDU that causes data transfer, transparent to the remote iSCSI Layer, to take place between the peer iSCSI nodes on a full feature phase iSCSI connection. An iSCSI data-type PDU, when requested for transmission by the sender iSCSI Layer, results in the associated data transfer without the participation of the remote iSCSI Layer, i.e. the PDU itself is not delivered as-is to the remote iSCSI Layer. The following iSCSI PDUs constitute the set of iSCSI data-type PDUs – SCSI Data-In PDU and R2T PDU.

iSCSI Layer – A layer in the protocol stack implementation within an end node that implements the iSCSI protocol and interfaces with the iSER Layer via the Datamover Interface.

iSCSI PDU (iSCSI Protocol Data Unit) – The iSCSI Layer at the initiator and the iSCSI Layer at the target divide their

communications into messages. The term "iSCSI protocol data unit" (iSCSI PDU) is used for these messages.

iSCSI/iSER Connection – An iSER-assisted iSCSI connection.

iSCSI/iSER Session – An iSER-assisted iSCSI session.

iSCSI-iSER Pair – The iSCSI Layer and the underlying iSER Layer.

iSER – iSCSI Extensions for RDMA, the protocol defined in this document.

iSER-assisted – A term generally used to describe the operation of iSCSI when the iSER functionality is also enabled below the iSCSI Layer for the specific iSCSI/iSER connection in question.

iSER-IRD – This variable represents the maximum number of incoming outstanding RDMA Read Requests that the iSER Layer at the initiator declares on a particular RDMAP Stream.

iSER-ORD – This variable represents the maximum number of outstanding RDMA Read Requests that the iSER Layer can initiate on a particular RDMAP Stream. This variable is maintained only by the iSER Layer at the target.

iSER Layer – The layer that implements the iSCSI Extensions for RDMA (iSER) protocol.

iWARP – A suite of wire protocols comprising of [RDMAP], [DDP], and [MPA] when layered above [TCP]. [RDMAP] and [DDP] may be layered above SCTP or other transport protocols.

Local Peer – The RDMAP implementation on the local end of the connection. Used to refer to the local entity when describing protocol exchanges or other interactions between two Nodes.

Node – A computing device attached to one or more links of a network. A Node in this context does not refer to a specific application or protocol instantiation running on the computer. A Node may consist of one or more RNICs installed in a host computer.

Operational Primitive – An Operational Primitive is an abstract functional interface procedure that requests another layer to perform a specific action on the requestor’s behalf or notifies the other layer of some event. The Datamover Interface between an iSCSI Layer and a Datamover layer within an iSCSI end node uses a set of Operational Primitives to define the functional interface between the two layers. Note that not every invocation of an Operational Primitive may elicit a response from the

requested layer. A full discussion of the Operational Primitive types and request-response semantics available to iSCSI and iSER can be found in [DA].

Outbound RDMA Read Queue Depth (ORD) – The maximum number of outstanding RDMA Read Requests that the RNIC can initiate on a particular RDMAP Stream at the Data Sink.

RDMA-enabled Network Interface Controller (RNIC) – A network I/O adapter or embedded controller with iWARP functionality.

RDMA Operation – A sequence of RDMAP Messages, including control Messages, to transfer data from a Data Source to a Data Sink. The following RDMA Operations are defined – RDMA Write Operation, RDMA Read Operation, Send Operation, Send with Invalidate Operation, Send with Solicited Event Operation, Send with Solicited Event and Invalidate Operation, and Terminate Operation.

RDMA Protocol (RDMAP) – A wire protocol that supports RDMA Operations to transfer ULP data between a Local Peer and the Remote Peer as described in [RDMAP].

RDMA Read Operation – An RDMA Operation used by the Data Sink to transfer the contents of a Data Source buffer from the Remote Peer to a Data Sink buffer at the Local Peer. An RDMA Read operation consists of a single RDMA Read Request Message and a single RDMA Read Response Message.

RDMA Read Request – An RDMAP Message used by the Data Sink to request the Data Source to transfer the contents of a buffer. The RDMA Read Request Message describes both the Data Source and the Data Sink buffers.

RDMA Read Response – An RDMAP Message used by the Data Source to transfer the contents of a buffer to the Data Sink, in response to an RDMA Read Request. The RDMA Read Response Message only describes the Data Sink buffer.

RDMA Write Operation – An RDMA Operation used by the Data Source to transfer the contents of a Data Source buffer from the Local Peer to a Data Sink buffer at the Remote Peer. The RDMA Write Message only describes the Data Sink buffer.

RDMAP Message – The sequence of RDMAP packets which represent a single RDMA operation or a part of RDMA Read Operation.

RDMAP Stream – A single bidirectional association between the peer RDMAP layers on two Nodes over a single transport-level stream. For iSER, the association is created when the iSCSI connection

transitions to iSER-assisted mode following a successful iSCSI Login Phase during which iSER support is negotiated.

Remote Direct Memory Access (RDMA) – A method of accessing memory on a remote system in which the local system specifies the remote location of the data to be transferred. Employing an RNIC in the remote system allows the access to take place without interrupting the processing of the CPU(s) on the system.

Remote Peer – The RDMAP implementation on the opposite end of the connection. Used to refer to the remote entity when describing protocol exchanges or other interactions between two Nodes.

SCSI Layer – This layer builds/receives SCSI CDBs (Command Descriptor Blocks) and sends/receives them with the remaining command execute [SAM2] parameters to/from the iSCSI Layer.

Send – An RDMA Operation that transfers the contents of a ULP Buffer from the Local Peer to a Buffer at the Remote Peer.

Send Message Type – A Send Message, Send with Invalidate Message, Send with Solicited Event Message, or Send with Solicited Event and Invalidate Message.

SendInvSE Message – A Send with Solicited Event and Invalidate Message.

SendSE Message – A Send with Solicited Event Message

Sequence Number (SN) – DataSN for a SCSI Data-in PDU and R2TSN for an R2T PDU. The semantics for both types of sequence numbers are as defined in [iSCSI].

Session, iSCSI Session – The group of TCP connections that link an initiator SCSI port with a target SCSI port form an iSCSI session (equivalent to a SCSI I-T nexus). TCP connections can be added to and removed from a session even while the I-T nexus is intact. Across all connections within a session, an initiator sees one and the same target.

Solicited Event (SE) – A facility by which an RDMA Operation sender may cause an Event to be generated at the recipient, if the recipient is configured to generate such an Event, when a Send with Solicited Event or Send with Solicited Event and Invalidate Message is received.

Steering Tag (STag) – An identifier of a Tagged Buffer on a Node as defined in [RDMAP] and [DDP].

Tagged Buffer – A buffer that is explicitly Advertised to a Remote Peer through exchange of an STag, Tagged Offset, and length.

Tagged Offset (TO) – The offset within a Tagged Buffer.

Traditional iSCSI – Refers to the iSCSI protocol defined by [iSCSI] (i.e. without the iSER enhancements).

Untagged Buffer – A buffer that is not explicitly Advertised to the Remote Peer.

3.2 Acronyms

Acronym	Definition
CO	Connection Only
CRC	Cyclic Redundancy Check
DDP	Direct Data Placement Protocol
DI	Datamover Interface
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
I/O	Input – Output
IO	Initialize Only
IP	Internet Protocol
IPsec	Internet Protocol Security
iSER	iSCSI Extensions for RDMA
ITT	Initiator Task Tag
LO	Leading Only
MPA	Marker PDU Aligned Framing for TCP
NOP	No Operation
NSG	Next Stage (during the iSCSI Login Phase)
OS	Operating System

PDU	Protocol Data Unit
R2T	Ready To Transfer
R2TSN	Ready To Transfer Sequence Number
RDMA	Remote Direct Memory Access
RDMAP	Remote Direct Memory Access Protocol
RFC	Request For Comments
RNIC	RDMA-enabled Network Interface Controller
SAM2	SCSI Architecture Model - 2
SCSI	Small Computer Systems Interface
SNACK	Selective Negative Acknowledgment - also Sequence Number Acknowledgement for data
STag	Steering Tag
SW	Session Wide
TCP	Transmission Control Protocol
TO	Tagged Offset
ULP	Upper Level Protocol

4 Overview

4.1 Motivation

The iSCSI protocol ([iSCSI]) is a mapping of the SCSI remote procedure invocation model (see [SAM2]) over the TCP protocol. SCSI commands are carried by iSCSI requests and SCSI responses and status are carried by iSCSI responses. Other iSCSI protocol exchanges and SCSI Data are also transported in iSCSI PDUs.

Out-of-order TCP segments in the traditional iSCSI model have to be stored and reassembled before the iSCSI protocol layer within an end node can place the data in the iSCSI buffers. This reassembly is required because not every TCP segment is likely to contain an iSCSI header to enable its placement and TCP itself does not have a built-in mechanism for signaling ULP message boundaries to aid placement of out-of-order segments. This TCP reassembly at high network speeds is quite counter-productive for the following reasons: wasted memory bandwidth in data copying, need for reassembly memory, wasted CPU cycles in data copying, and the general store-and-forward latency from an application perspective. [iSCSI] itself recognized that TCP reassembly could be a serious issue and had introduced the notion of a "sync and steering layer" that is optional to implement and use. [iSCSI] further defined one specific sync and steering layer – called "markers" – an application-level way of framing iSCSI PDUs within the TCP data stream even when the TCP segments are not yet reassembled to be in-order.

With these [iSCSI] defined techniques, a Network Interface Controller customized for iSCSI (SNIC) could offload the TCP/IP processing and support direct data placement.

Supporting direct data placement is the main function of the iWARP protocol suite. A NIC enhanced with the RDMAP/DDP functions (RNIC) can be used by any application that has been extended to support RDMA.

With the availability of RNICs within a host system, which does not have SNICs, it is appropriate for iSCSI to be able to exploit the direct data placement function of the RNIC like other applications.

iSCSI Extensions for RDMA (iSER) is designed precisely to take advantage of generic RDMA technologies – iSER's goal is to permit iSCSI to employ direct data placement and RDMA capabilities using a generic RNIC. In summary, iSCSI/iSER protocol stack is designed to enable scaling to high speeds by relying on a generic data placement process and RDMA over TCP/IP networking technologies and products, which enable direct data placement of both in-order and out-of-order data.

This document describes iSER as a protocol extension to iSCSI, both for convenience of description and also because it is true in a very strict protocol sense. However, it is to be noted that iSER is in reality extending the connectivity of the iSCSI protocol defined in [iSCSI], and the name iSER reflects this reality.

When the iSCSI protocol defined by [iSCSI] (i.e. without the iSER enhancements) is intended in the rest of the document, the term "traditional iSCSI" is used to make the intention clear.

4.2 Architectural Goals

This section summarizes the architectural goals that guided the design of iSER.

1. Provide iWARP-based data transfer model for iSCSI that enables direct in order or out of order data placement of SCSI data into pre-allocated SCSI buffers while maintaining in order data delivery.

2. Not require any major changes to SCSI Architecture Model (SAM/SAM-2/SAM-3) and SCSI command set standards.

3. Utilize existing traditional iSCSI infrastructure (sometimes referred to as "iSCSI ecosystem") including but not limited to MIB, bootstrapping, negotiation, naming & discovery, and security.

4. Not require iSCSI full feature phase interoperability between an end node operating in traditional iSCSI mode, and an end node operating in iSER-assisted mode.

5. Allow initiator and target implementations that utilize generic RNICs and implement iSCSI and iSER in software (not require iSCSI or iSER specific assists in the iWARP protocol suite or RNIC).

6. Require full and only generic iWARP functionality at both the initiator and the target.

7. Require a session to operate in the traditional iSCSI data transfer mode if iSER is not supported by either the initiator or the target.

8. Implement a light weight Datamover protocol for iSCSI with minimal state maintenance.

4.3 Protocol Overview

Consistent with the architectural goals stated in section 4.2, the iSER protocol does not require changes in the iSCSI ecosystem or any related SCSI specifications. iSER protocol defines the mapping of

iSCSI PDUs to RDMAP Messages in such a way that it is entirely feasible to realize iSCSI/iSER implementations that are based on generic RNICS. The iSER protocol layer requires minimal state maintenance to assist an iSCSI full feature phase connection, besides being oblivious to the notion of an iSCSI session. The crucial protocol aspects of iSER may be summarized thus:

1. iSER-assisted mode is negotiated during the iSCSI login for each connection, but an entire iSCSI session MUST operate in one mode (i.e. one connection in the session cannot operate in iSER-assisted mode while a different connection of the same session is already in full feature mode in the traditional iSCSI mode).

2. Once in iSER-assisted mode, all iSCSI interactions on that connection use RDMAP Messages.

3. A Send Message Type is used for carrying an iSCSI control-type PDU preceded by an iSER header. See section 9.2 for more details on iSCSI control-type PDUs.

4. RDMA Write, RDMA Read Request, and RDMA Read Response Messages are used for carrying control and all data information associated with the iSCSI data-type PDUs. See section 9.1 for more details on iSCSI data-type PDUs.

5. Target drives all data transfer (with the exception of iSCSI unsolicited data) for SCSI writes and SCSI reads, by issuing RDMA Read Requests and RDMA Writes respectively.

6. The iWARP protocol suite running on top of TCP guarantees data integrity (iWARP uses a CRC-enhanced framing layer on TCP). For this reason, iSCSI header and data digests are negotiated to "None" for iSCSI/iSER sessions.

7. The iSCSI error recovery hierarchy defined by [iSCSI] is fully supported by iSER.

8. iSER requires no changes to iSCSI authentication, security, and text mode negotiation mechanisms.

Note that traditional iSCSI implementations may have to be adapted to employ iSER. It is expected that the adaptation when required is likely to be centered around the upper layer interface requirements of iSER (section 5).

4.4 RDMA services and iSER

iSER is designed to work with software and/or hardware protocol stacks providing the protocol services defined in [RDMAP]. The

following subsections describe the key protocol elements of RDMAP that iSER relies on.

4.4.1 STag

An STag is the RNIC-unique identifier of an I/O Buffer that the iSER Layer Advertises to the remote iSCSI/iSER node in order to complete a SCSI I/O.

In iSER, Advertisement is the act of informing the target by the initiator that an I/O Buffer is available at the initiator for RDMA Read or RDMA Write access by the target. The initiator Advertises the I/O Buffer by including the STag in the header of an iSER Message containing the SCSI Command PDU to the target. The base Tagged Offset is not explicitly specified, but the target must always assume it as zero. The buffer length is as specified in the SCSI Command PDU.

The iSER Layer at the initiator Advertises the STag for the I/O Buffer of each SCSI I/O to the iSER Layer at the target in the iSER header of the SendSE Message containing the SCSI Command PDU, unless the I/O can be completely satisfied by unsolicited data alone.

The iSER Layer at the target provides the STag for the I/O Buffer that is the Data Sink of an RDMA Read Operation (section 4.4.4) to the RDMAP layer on the initiator node – i.e. this is completely transparent to the iSER Layer at the initiator.

The iSER protocol is defined so that the Advertised STag is automatically invalidated upon a normal completion of the associated task. This automatic invalidation is realized via the SendInvSE Message carrying the SCSI Response PDU. There are two exceptions to this automatic invalidation – bidirectional commands, and abnormal completion of a command. The iSER Layer at the initiator is required to explicitly invalidate the STag in these cases, in addition to sanity checking the automatic invalidation even when that does happen.

4.4.2 Send

Send is the RDMA Operation that is not addressed to an Advertised buffer by the sending side, and thus uses Untagged buffers on the receiving side.

The iSER Layer at the initiator uses the Send Operation to transmit any iSCSI control-type PDU to the target. As an example, the initiator uses Send Operations to transfer iSER Messages containing SCSI Command PDUs to the iSER Layer at the target.

An iSER layer at the target uses the Send Operation to transmit any iSCSI control-type PDU to the initiator. As an example, the target uses Send Operations to transfer iSER Messages containing SCSI Response PDUs to the iSER Layer at the initiator.

4.4.3 RDMA Write

RDMA Write is the RDMA Operation that is used to place data into an Advertised buffer on the receiving side. The sending side addresses the Message using an STag and a Tagged Offset that are valid on the Data Sink.

The iSER Layer at the target uses the RDMA Write Operation to transfer the contents of a local I/O Buffer to an Advertised I/O Buffer at the initiator. The iSER Layer at the target uses the RDMA Write to transfer whole or part of the data required to complete a SCSI Read command.

The iSER Layer at the initiator does not employ RDMA Writes.

4.4.4 RDMA Read

RDMA Read is the RDMA Operation that is used to retrieve data from an Advertised buffer on a remote node. The sending side of the RDMA Read Request addresses the Message using an STag and a Tagged Offset that are valid on the Data Source in addition to providing a valid local STag and Tagged Offset that identify the Data Sink.

The iSER Layer at the target uses the RDMA Read Operation to transfer the contents of an Advertised I/O Buffer at the initiator to a local I/O Buffer at the target. The iSER Layer at the target uses the RDMA Read to fetch whole or part of the data required to complete a SCSI Write.

The iSER Layer at the initiator does not employ RDMA Reads.

4.5 SCSI Read Overview

The iSER Layer at the initiator receives the SCSI Command PDU from the iSCSI Layer. The iSER Layer at the initiator generates an STag for the I/O Buffer of the SCSI Read and Advertises the buffer by including the STag as part of the iSER header for the PDU. The iSER Message is transferred to the target using a SendSE Message.

The iSER Layer at the target uses one or more RDMA Writes to transfer the data required to complete the SCSI Read.

The iSER Layer at the target uses a SendInvSE Message to transfer the SCSI Response PDU back to the iSER Layer at the initiator. The

iSER Layer at the initiator notifies the iSCSI Layer of the availability of the SCSI Response PDU.

4.6 SCSI Write Overview

The iSER Layer at the initiator receives the SCSI Command PDU from the iSCSI Layer. If solicited data transfer is involved, the iSER Layer at the initiator generates an STag for the I/O Buffer of the SCSI Write and Advertises the buffer by including the STag as part of the iSER header for the PDU. The iSER Message is transferred to the target using a SendSE Message.

The iSER Layer at the initiator may optionally send one or more non-immediate unsolicited data PDUs to the target using Send Message Types.

If solicited data transfer is involved, the iSER Layer at the target uses one or more RDMA Reads to transfer the data required to complete the SCSI Write.

The iSER Layer at the target uses a SendInvSE Message to transfer the SCSI Response PDU back to the iSER Layer at the initiator. The iSER Layer at the initiator notifies the iSCSI Layer of the availability of the SCSI Response PDU.

4.7 iSCSI/iSER Layering

Figure 1 iSCSI/iSER Layering in Full Feature Mode depicts the relationship between SCSI, iSCSI, iSCSI Extensions for RDMA (iSER), RDMAP, and the rest of the iWARP stack.

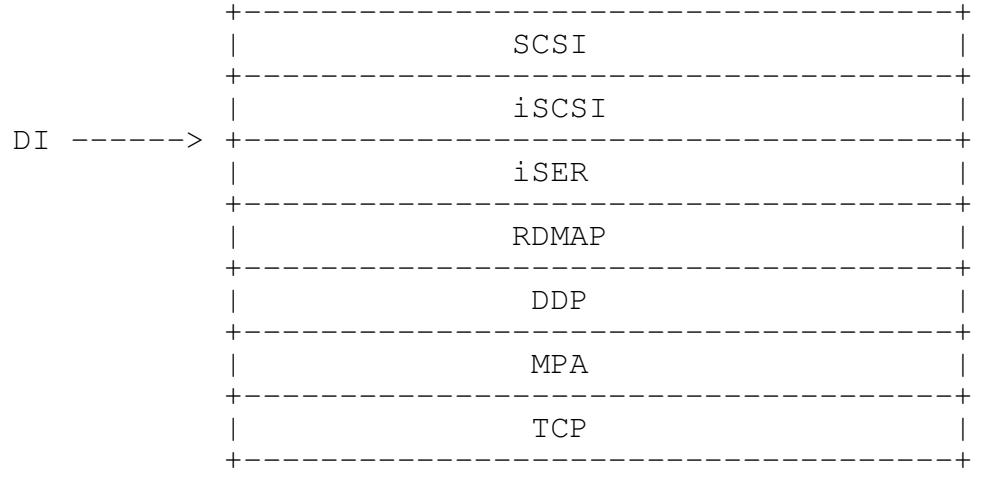

Figure 1 iSCSI/iSER Layering in Full Feature Mode

5 Upper Layer Interface Requirements

This section discusses the upper layer interface requirements in the form of an abstract model of the required interactions between the iSCSI Layer and the iSER Layer. The abstract model used here is derived from the architectural model described in [DA]. The interface requirements are specified by Operational Primitives. An Operational Primitive is an abstract functional interface procedure between the iSCSI Layer and the iSER Layer that requests one layer to perform a specific action on behalf of the other layer or notifies the other layer of some event.

The abstract model and Operational Primitives defined in this section are for the ease of description of iSER protocol. In the rest of the iSER specification, the compliance statements related to the use of these Operational Primitives are only for the purpose of the required interactions between the iSCSI Layer and the iSER Layer. Note that the compliance statements related to Operational Primitives in the rest of this specification only mandate functional equivalence on implementations, but do not put any requirements on the implementation specifics of the interface between the iSCSI Layer and the iSER Layer.

5.1 Operational Primitives offered by iSER

The iSER protocol layer MUST support the following Operational Primitives to be used by the iSCSI protocol layer.

1. Send_Control: The iSCSI Layers at the initiator and the target use this to request the outbound transfer of an iSCSI control-type PDU.
2. Put_Data: The iSCSI Layer at the target uses this to request the outbound transfer of data for a SCSI Data-in PDU.

3. Get_Data: The iSCSI Layer at the target uses this to request the inbound transfer of solicited data requested by an R2T PDU.

4. Allocate_Connection_Resources: The iSCSI Layers at the initiator and the target use this to request the allocation of all iWARP-specific connection resources required for an operational iSCSI/iSER connection.

5. Deallocate_Connection_Resources: The iSCSI Layers at the initiator and the target use this to request the deallocation of all iWARP-specific connection resources that were earlier allocated as a result of a successful Allocate_Connection_Resources invocation.

6. Enable_Datamover: The iSCSI Layers at the initiator and the target use this to request that a specified iSCSI connection be transitioned to iSER-assisted mode.

7. Connection_Terminate: The iSCSI Layers at the initiator and the target use this to request that a specified iSCSI/iSER connection be terminated and all the associated connection and task resources be freed.

8. Notice_Key_Values: The iSCSI Layers at the initiator and the target use this to request that the specified Key-Value pairs are to be taken note of by the local Datamover layer.

9. Deallocate_Task_Resources: The iSCSI Layers at the initiator and the target use this to request the deallocation of all iWARP-specific task resources that may have been allocated as part of the task initiation by the iSER Layer. This Operational Primitive is only used for tasks that did not conclude with a SCSI Response PDU.

5.2 Operational Primitives used by iSER

Note that in the following discussion and in the rest of the document, a PDU is described as "available" to the iSCSI Layer when the iSER Layer notifies the iSCSI Layer of the reception of that inbound PDU, along with an implementation-specific indication as to where the received PDU is.

The iSER layer MUST use the following Operational Primitives offered by the iSCSI protocol layer via DI.

1. Control_Notify: The iSER Layers at both the initiator and the target use this to notify the iSCSI Layer of the availability of an inbound iSCSI control-type PDU.

2. Data_Completion_Notify: The iSER Layer at the target uses this to notify the iSCSI Layer of the completion of inbound/outbound data transfer that was requested by the iSCSI Layer when the request was qualified with Notify_Enable set.

3. Data_ACK_Notify: The iSER Layer at the target uses this to notify the iSCSI Layer of the arrival of the data acknowledgement (as defined in [iSCSI]) requested earlier by the iSCSI Layer for the outbound data transfer (Data-in PDUs).

4. Connection_Terminate_Notify: The iSER Layers at both the initiator and the target use this to notify the iSCSI Layer of the termination of an iSCSI/iSER connection. However, Connection_Terminate_Notify is not invoked when the termination of the connection was earlier requested by the local iSCSI Layer.

5.3 iSCSI Protocol Usage Requirements

An iSER-assisted iSCSI protocol layer should satisfy the following protocol usage requirements from the iSER protocol:

1. The iSCSI Layers at both the initiator and the target MUST negotiate the new RDMAExtensions key (see section 8.3) to "Yes" on the leading connection. If the invocation of the Allocate_Connection_Resources Operational Primitive to the iSER layer fails after this key is negotiated to "Yes", the iSCSI layer MUST fail the iSCSI Login process or terminate the connection as appropriate. See section 12.1.3.1 and 12.1.3.2 for details.

2. The iSCSI Layers at both the initiator and the target MUST negotiate the HeaderDigest key and the DataDigest key to "None" during the login phase for iSER-assisted iSCSI connections.

3. The iSCSI Layer at the initiator MUST set ExpDataSN = 0 in Task Management Function Requests for Task Allegiance Reassignment for read/bidirectional commands, so as to cause the target to send all unacknowledged read data.

4. The iSCSI Layer at the target MUST always return the SCSI status in a separate SCSI Response PDU for read commands, i.e., there MUST NOT be a "phase collapse" in concluding a SCSI Read Command.

5. The iSCSI Layers at both the initiator and the target MUST successfully negotiate the new InitiatorRecvDataSegmentLength key for each iSER-assisted connection, and follow its defined semantics.

6. The iSCSI Layer at both the initiator and the target MUST successfully negotiate the new TargetRecvDataSegmentLength key for each iSER-assisted connection, and follow its defined semantics.

7. The iSCSI Layer at the initiator SHOULD NOT issue proactive (based on time-outs) SNACKs for PDUs that it presumes are lost.

8. The iSCSI Layers at both the initiator and the target MUST negotiate the OFMarker key and the IFMarker key to "No" during the login phase for an iSER-assisted iSCSI connection.

6 Lower Layer Interface Requirements

6.1 Interactions with the iWARP Layer

The iSER protocol layer is layered on top of the iWARP protocol stack (see Figure 1) and the following are the key features that are assumed to be supported by iWARP:

* The RDMAP layer supports all basic RDMAP operations, including RDMA Write Operation, RDMA Read Operation, Send Operation, Send with Invalidate Operation, Send with Solicited Event Operation, Send with Solicited Event & Invalidate Operation, and Terminate Operation.

* The RDMAP/DDP layers provide reliable, in-order message delivery and direct data placement.

* The RDMAP layer encapsulates a single iSER Message into a single RDMAP message on the Data Source side. The RDMAP layer decapsulates the iSER Message before delivering it to the iSER Layer on the Data Sink side.

* When the iSER Layer provides the STag to be remotely invalidated to the RDMAP layer for a SendInvSE Message, the RDMAP layer uses this STag as the STag to be invalidated in the SendInvSE Message.

* The RDMAP layer uses the STag and Tagged Offset provided by the iSER Layer for the RDMA Write and RDMA Read Request Messages.

* When the RDMAP layer delivers the content of an RDMA Send Message Type to the iSER Layer, the RDMAP layer provides the length of the RDMA Send message. This ensures that the iSER Layer does not have to carry a length field in the iSER header.

* When the RDMAP layer delivers the SendSE or SendInvSE Message to the iSER Layer, it notifies the iSER Layer with the mechanism provided on that interface.

* When the RDMAP layer delivers a SendInvSE Message to the iSER Layer, it passes the value of the STag that was invalidated.

* The RDMAP layer propagates all status and error indications to the iSER Layer.

* The iWARP implementation supports the enabling of the iWARP mode after TCP connection establishment.

* Whenever the iSER Layer terminates the RDMAP Stream, the RDMAP layer terminates the associated TCP connection.

6.2 Interactions with the TCP Layer

The iSER Layer does not interface with the TCP layer directly. During connection setup, the iSCSI Layer is responsible for setting up the TCP connection. If the login is successful, the iSCSI Layer invokes the Enable_Datamover Operational Primitive to request the iSER Layer to transition to the iSER-assisted mode for that iSCSI connection. See section 7.1 on iSCSI/iSER Connection Setup. After transitioning to iSER-assisted mode, the iWARP layer is responsible for maintaining the TCP connection and reports to the iSER Layer of any TCP connection failures.

7 Connection Setup and Termination

7.1 iSCSI/iSER Connection Setup

During connection setup, the iSCSI Layer at the initiator is responsible for establishing a TCP connection using the appropriate TCP port as discovered through the traditional iSCSI discovery mechanisms. After the TCP connection is established, the iSCSI Layers at the initiator and the target enter the Login Phase, conducted in TCP stream mode and using the same rules as outlined in [iSCSI]. Transition to iSER-assisted mode occurs when the connection transitions into the iSCSI full feature phase following a successful login negotiation between the initiator and the target in which iSER-assisted mode is negotiated and the necessary iWARP resources have been allocated at both the initiator and the target.

iSER-assisted mode MUST be enabled only if it is negotiated on the leading connection during the LoginOperationalNegotiation Stage of the iSCSI Login Phase. iSER-assisted mode is negotiated using the RDMAExtensions=<boolean-value> key. Both the initiator and the target MUST exchange the RDMAExtensions key with the value set to "Yes" to enable iSER-assisted mode. If both the initiator and the target fail to negotiate the RDMAExtensions key set to "Yes", then the connection MUST continue with the login semantics as defined in [iSCSI].

iSER-assisted mode is defined for a Normal session only and the RDMAExtensions key MUST NOT be negotiated for a Discovery session.

An iSER enabled node is not required to initiate the RDMAExtensions key exchange if its preference is for the traditional iSCSI mode. The RDMAExtensions key, if offered, MUST be sent in the first available Login Response or Login Request PDU in the LoginOperationalNegotiation stage. This is due to the fact that the value of some login parameters might depend on whether iSER-assisted mode is enabled or not.

iSER-assisted mode is a session-wide attribute. If both the initiator and the target negotiated RDMAExtensions="Yes" on the leading connection of a session, then all subsequent connections of the same session MUST enable iSER-assisted mode without having to exchange RDMAExtensions key during the iSCSI Login Phase. Conversely, if both the initiator and the target failed to negotiate RDMAExtensions to "Yes" on the leading connection of a session, then the RDMAExtensions key MUST NOT be negotiated further on any additional subsequent connection of the session.

When the RDMAExtensions key is negotiated to "Yes", the HeaderDigest and the DataDigest keys MUST be negotiated to "None" on all iSCSI/iSER connections participating in that iSCSI session. This is

because, for an iSCSI/iSER connection, the iWARP protocol suite provides a CRC32c-based error detection for all iWARP Messages. Furthermore, all SCSI Read data are sent using RDMA Write Messages instead of the SCSI Data-in PDUs, and all solicited SCSI write data are sent using RDMA Read Response Messages instead of the SCSI Data-out PDUs. HeaderDigest and DataDigest which apply to iSCSI PDUs would not be appropriate for RDMA Read and RDMA Write operations used with iSER.

7.1.1 Initiator Behavior

If the outcome of the iSCSI negotiation is to enable iSER-assisted mode, then on the initiator side, prior to sending the Login Request with the T (Transit) bit set to 1 and the NSG (Next Stage) field set to FullFeaturePhase, the iSCSI Layer MUST invoke the Allocate_Connection_Resources Operational Primitive to request the iSER Layer to allocate the resources necessary to support iWARP. The iWARP resources required are defined by implementation and are outside the scope of this specification. Optionally, the iSCSI Layer MAY invoke the Notice_Key_Values Operational Primitive before invoking the Allocate_Connection_Resources Operational Primitive to request the iSER Layer to take note of the negotiated values of the iSCSI keys for the TCP connection. The specific keys to be passed in as input qualifiers are implementation dependent. These may include, but not limited to, MaxOutstandingR2T, ErrorRecoveryLevel, etc.

Among the iWARP resources allocated at the initiator is the Inbound RDMA Read Queue Depth (IRD). As described in section 11.5.1, R2Ts are transformed by the target into RDMA Read operations. IRD limits the maximum number of simultaneously incoming outstanding RDMA Read Requests per an RDMAP Stream from the target to the initiator. The required value of IRD is outside the scope of the iSER specification. The iSER Layer at the initiator MUST set IRD to 1 or higher if R2Ts are to be used in the connection. However, the iSER Layer at the initiator MAY set IRD to 0 based on implementation configuration which indicates that no R2Ts will be used on that connection. Initially, the iSER-IRD value at the initiator SHOULD be set to the IRD value at the initiator and MUST NOT be more than the IRD value.

On the other hand, the Outbound RDMA Read Queue Depth (ORD) MAY be set to 0 since the iSER Layer at the initiator does not issue RDMA Read Requests to the target.

Failure to allocate the requested iWARP resources locally results in a login failure and its handling is described in section 12.1.3.1. If the iSER Layer at the initiator is successful in allocating the necessary connection resources for iWARP, the following events MUST occur in the specified sequence:

1. The iSER Layer MUST return a success status to the iSCSI Layer in response to the Allocate_Connection_Resources Operational Primitive.

2. After the target returns the Login Response with the T bit set to 1 and the NSG field set to FullFeaturePhase, and a status class of 0 (Success), the iSCSI Layer MUST invoke the Enable_Datamover Operational Primitive with the following qualifiers to request the iSER Layer to transition to iSER-assisted mode (See section 12.1.4.6 for the case when the status class is not Success.):

   a. Connection_Handle that identifies the iSCSI connection.

   b. Transport_Connection_Descriptor which identifies the specific transport connection associated with the Connection_Handle.

3. The iSER Layer MUST enable iWARP and transition the connection to iSER-assisted mode.

4. The iSER Layer MUST send the iSER Hello Message as the first RDMAP message. See Section 7.1.3 on iSER Hello Exchange.

7.1.2 Target Behavior

If the outcome of the iSCSI negotiation is to enable iSER-assisted mode, then on the target side, prior to sending the Login Response with the T (Transit) bit set to 1 and the NSG (Next Stage) field set to FullFeaturePhase, the iSCSI Layer MUST invoke the Allocate_Connection_Resources Operational Primitive to request the iSER Layer to allocate the resources necessary to support iWARP. The iWARP resources required are defined by implementation and are outside the scope of this specification. Optionally, the iSCSI Layer MAY invoke the Notice_Key_Values Operational Primitive before invoking the Allocate_Connection_Resources Operational Primitive to request the iSER Layer to take note of the negotiated values of the iSCSI keys for the TCP connection. The specific keys to be passed in as input qualifiers are implementation dependent. These may include, but not limited to, MaxOutstandingR2T, ErrorRecoveryLevel, etc.

Among the iWARP resources allocated at the target is the Outbound RDMA Read Queue Depth (ORD). As described in section 11.5.1, R2Ts are transformed by the target into RDMA Read operations. The ORD limits the maximum number of simultaneously outstanding RDMA Read Requests per RDMAP Stream from the target to the initiator. Initially, the iSER-ORD value at the target SHOULD be set to the ORD value at the target.

On the other hand, the IRD at the target MAY be set to 0 since the iSER Layer at the target does not expect RDMA Read Requests to be issued by the initiator. Failure to allocate the requested iWARP resources locally is a negotiation failure and is described in section 12.1.3.2.

If the iSER Layer at the target is successful in allocating the necessary iWARP resources, the following events MUST occur in the specified sequence:

1. The iSER Layer MUST return a success status to the iSCSI Layer in response to the Allocate_Connection_Resources Operational Primitive.

2. The iSCSI Layer MUST invoke the Enable_Datamover Operational Primitive with the following qualifiers to request the iSER Layer to transition to iSER-assisted mode:

   a. Connection_Handle that identifies the iSCSI connection.

   b. Transport_Connection_Descriptor which identifies the specific transport connection associated with the Connection_Handle.

   c. The final TCP message containing the Login Response with the T bit set to 1 and the NSG field set to FullFeaturePhase

3. The iSER Layer MUST send the final SCSI Login Response PDU in byte stream mode to conclude the iSCSI Login Phase.

4. After sending the final SCSI Login Response PDU in byte stream mode, the iSER Layer MUST enable iWARP and transition the connection to iSER-assisted mode.

5. After receiving the iSER Hello Message from the initiator, the iSER Layer MUST respond with the iSER HelloReply Message to be sent as the first RDMAP Message. See section 7.1.3 on iSER Hello Exchange for more details.

Note: In the above sequence, the operations as described in the bullets 3 and 4 must be performed atomically. Failure to do this may result in race conditions.

7.1.3 iSER Hello Exchange

After the connection transitions into the iSER-assisted mode, the first RDMAP Message sent by the iSER Layer at the initiator to the target MUST be the iSER Hello Message. The iSER Hello Message is used by the iSER Layer at the initiator to declare iSER parameters

to the target. See section 11.3 on iSER Header Format for iSER Hello Message

In response to the iSER Hello Message, the iSER Layer at the target MUST return the iSER HelloReply Message as the first RDMAP Message sent by the target. The iSER HelloReply Message is used by the iSER Layer at the target to declare iSER parameters to the initiator. See section 11.4 on iSER Header Format for iSER HelloReply Message.

In the iSER Hello Message, the iSER Layer at the initiator declares the iSER-IRD value to the target.

Upon receiving the iSER Hello Message, the iSER Layer at the target MUST set the iSER-ORD value to the minimum of the iSER-ORD value at the target and the iSER-IRD value declared by the initiator. The iSER Layer at the target MAY adjust (lower) its ORD value to match the iSER-ORD value if the iSER-ORD value is smaller than the ORD value at the target in order to free up the unused resources.

In the iSER HelloReply Message, the iSER Layer at the target declares the iSER-ORD value to the initiator.

Upon receiving the iSER HelloReply Message, the iSER Layer at the initiator MAY adjust (lower) its IRD value to match the iSER-ORD value in order to free up the unused resources, if the iSER-ORD value declared by the target is smaller than the iSER-IRD value declared by the initiator.

It is an iSER level negotiation failure if the iSER parameters declared in the iSER Hello Message by the initiator is unacceptable to the target. See section 12.1.3.3 on the handling of the error situation.

7.2 iSCSI/iSER Connection Termination

7.2.1 Normal Connection Termination at the Initiator

The iSCSI Layer at the initiator terminates an iSCSI/iSER connection normally by invoking the Send_Control Operational Primitive qualified with the Logout Request PDU. The iSER Layer at the initiator MUST use a SendSE Message to send the Logout Request PDU to the target. After the iSER Layer at the initiator receives the SendSE Message containing the Logout Response PDU from the target, it MUST notify the iSCSI Layer by invoking the Control_Notify Operational Primitive qualified with the Logout Response PDU.

After the iSCSI logout process is complete, the iSCSI layer at the target is responsible for closing the iSCSI/iSER connection as described in Section 7.2.2. After the RDMAP layer at the initiator reports that the TCP connection has been closed, the iSER Layer at

the initiator MUST deallocate the iWARP resources for the connection, deallocate all the task resources (if any) associated with the connection, invalidate the local mapping(s) (if any) that associate the ITT(s) used on that connection to the local STag(s), and then invoke the Connection_Terminate_Notify Operational Primitive to notify the iSCSI Layer.

7.2.2 Normal Connection Termination at the Target

Upon receiving the SendSE Message containing the Logout Request PDU, the iSER Layer at the target MUST notify the iSCSI Layer at the target by invoking the Control_Notify Operational Primitive qualified with the Logout Request PDU. The iSCSI Layer completes the logout process by invoking the Send_Control Operational Primitive qualified with the Logout Response PDU. The iSER Layer at the target MUST use a SendSE Message to send the Logout Response PDU to the initiator. After the iSCSI logout process is complete, the iSCSI Layer at the target MUST invoke the Connection_Terminate Operational Primitive to request the iSER Layer at the target to terminate the RDMAP Stream.

As part of the termination process, the RDMAP layer MUST close the TCP connection. When the RDMAP layer notifies the iSER Layer after the RDMAP stream and the associated TCP connection are terminated, the iSER Layer MUST deallocate the iWARP resources for the connection. In addition to deallocating the iWARP resources, the iSER Layer at the target MUST deallocate all the task resources (if any) associated with the connection, and invalidate the local and remote mapping(s) (if any) that associate the ITT(s) used on that connection to the local STag(s) and the Advertised STag(s) respectively.

7.2.3 Termination without Logout Request/Response PDUs

7.2.3.1 Connection Termination Initiated by the iSCSI Layer

The Connection_Terminate Operational Primitive MAY be invoked by the iSCSI Layer to terminate the iSCSI/iSER connection without having previously exchanged the Logout Request and Logout Response PDUs between the two iSCSI/iSER nodes. The Connection_Terminate Operational Primitive requests the iSER Layer to terminate the RDMAP Stream. As part of the termination process, the RDMAP layer will close the TCP connection. When the RDMAP layer notifies the iSER Layer after the RDMAP stream and the associated TCP connection are terminated, the iSER Layer MUST perform the following actions.

If the Connection_Terminate Operational Primitive is invoked by the iSCSI Layer at the target, then the iSER Layer at the target MUST deallocate the iWARP resources for the connection, deallocate all the task resources (if any) associated with the connection, and

invalidate the local and remote mappings (if any) that associate the ITT(s) used on the connection to the local STag(s) and the Advertised STag(s) respectively.

If the Connection_Terminate Operational Primitive is invoked by the iSCSI Layer at the initiator, then the iSER Layer at the initiator MUST deallocate the iWARP resources for the connection, deallocate the task resources (if any) associated with the connection, and invalidate the local mapping(s) (if any) that associate the ITT(s) used on the connection to the local STag(s).

7.2.3.2 Connection Termination Notification to the iSCSI Layer

If the iSCSI/iSER connection is terminated without the invocation of Connection_Terminate from the iSCSI Layer, the iSER Layer MUST invoke the Connection_Terminate_Notify Operational Primitive to notify the iSCSI Layer that the iSCSI/iSER connection has been terminated.

Prior to invoking Connection_Terminate_Notify, the iSER Layer at the target MUST deallocate the iWARP resources for the connection, deallocate the task resources (if any) associated with the connection, and invalidate the local and remote mappings (if any) that associate the ITT(s) used on the connection to the local STag(s) and the Advertised STag(s) respectively.

Prior to invoking Connection_Terminate_Notify, the iSER Layer at the initiator MUST deallocate the iWARP resources for the connection, deallocate the task resources (if any) associated with the connection, and invalidate the local mappings (if any) that associate the ITT(s) used on the connection to the local STag(s).

If the remote iSCSI/iSER node initiated the closing of the TCP connection by sending a TCP FIN or TCP RST, the iSER Layer MUST invoke the Connection_Terminate_Notify Operational Primitive to notify the iSCSI Layer after the RDMAP layer reports that the TCP connection is closed.

Another example of a TCP connection termination without a preceding logout is when the iSCSI Layer at the initiator does an implicit logout (connection reinstatement).

8 Login/Text Operational Keys

Certain iSCSI login/text operational keys have restricted usage in iSER, and additional keys are used to support the iSER protocol functionality. All other keys defined by [iSCSI] and not discussed in this section may be used on iSCSI/iSER connections with the same semantics.

8.1 HeaderDigest and DataDigest

If the RDMAExtensions key is negotiated to "Yes" on the leading connection of a session, both HeaderDigest and DataDigest MUST be negotiated to "None" for each connection belonging to that session.

8.2 MaxRecvDataSegmentLength

For an iSCSI connection belonging to a session in which RDMAExtensions=Yes was negotiated on the leading connection of the session, MaxRecvDataSegmentLength need not be declared in the Login Phase. Instead InitiatorRecvDataSegmentLength (as described in section 8.5) and TargetRecvDataSegmentLength (as described in section 8.4) keys are negotiated. The values of the local and remote MaxRecvDataSegmentLength are derived from the InitiatorRecvDataSegmentLength and TargetRecvDataSegmentLength keys even if the MaxRecvDataSegmentLength was declared during the login phase.

In the full feature phase, the initiator MUST consider the value of its local MaxRecvDataSegmentLength (that it would have declared to the target) as having the value of InitiatorRecvDataSegmentLength, and the value of the remote MaxRecvDataSegmentLength (that would have been declared by the target) as having the value of TargetRecvDataSegmentLength. Similarly, the target MUST consider the value of its local MaxRecvDataSegmentLength (that it would have declared to the initiator) as having the value of TargetRecvDataSegmentLength, and the value of the remote MaxRecvDataSegmentLength (that would have been declared by the initiator) as having the value of InitiatorRecvDataSegmentLength.

The MaxRecvDataSegmentLength key is applicable only for iSCSI control-type PDUs.

8.3 RDMAExtensions

Use: LO (leading only)

Senders: Initiator and Target

Scope: SW (session-wide)

RDMAExtensions=<boolean-value>

Irrelevant when: SessionType=Discovery

Default is No

Result function is AND

This key is used by the initiator and the target to negotiate the support for iSER-assisted mode. To enable the use of iSER-assisted mode, both the initiator and the target MUST exchange RDMAExtensions=Yes. iSER-assisted mode MUST NOT be used if either the initiator or the target offers RDMAExtensions=No.

An iSER-enabled node is not required to initiate the RDMAExtensions key exchange if it prefers to operate in the traditional iSCSI mode. However, if the RDMAExtensions key is to be negotiated, it MUST be offered only on the initial Login Request PDU or Login Response PDU of the leading connection, and if offered, the response MUST be sent in the immediately following Login Response or Login Request PDU respectively. The key must precede any other login keys which may be affected by the outcome of the negotiation of the RDMAExtensions key.

8.4 TargetRecvDataSegmentLength

Use: IO (Initialize only)

Senders: Initiator and Target

Scope: CO (connection-only)

Irrelevant when: RDMAExtensions=No

TargetRecvDataSegmentLength=<numerical-value-512-to-(2**24-1)>

Default is 8192 bytes

Result function is minimum

This key is relevant only for the iSCSI connection of an iSCSI session if RDMAExtensions=Yes was negotiated on the leading connection of the session. It is used by the initiator and the target to negotiate the maximum size of the data segment that an initiator may send to the target in an iSCSI control-type PDU. For SCSI Command PDUs and SCSI Data-out PDUs containing non-immediate unsolicited data to be sent by the initiator, the initiator MUST send all non-Final PDUs with a data segment size of exactly TargetRecvDataSegmentLength whenever the PDUs constitute a data sequence whose size is larger than TargetRecvDataSegmentLength.

8.5 InitiatorRecvDataSegmentLength

Use: IO (Initialize only)

Senders: Initiator and Target

Scope: CO (connection-only)

Irrelevant when: RDMAExtensions=No

InitiatorRecvDataSegmentLength=<numerical-value-512-to-(2**24-1)>

Default is 8192 bytes

Result function is minimum

This key is relevant only for the iSCSI connection of an iSCSI session if RDMAExtensions=Yes was negotiated on the leading connection of the session. It is used by the initiator and the target to negotiate the maximum size of the data segment that a target may send to the initiator in an iSCSI control-type PDU.

8.6 OFMarker and IFMarker

If the RDMAExtensions key is negotiated to "Yes" on the leading connection of a session, both OFMarker and IFMarker MUST be negotiated to "No" for each connection belonging to that session if they are negotiated.

9 iSCSI PDU Considerations

When a connection is in the iSER-assisted mode, two types of message transfers are allowed between the iSCSI Layer at the initiator and the iSCSI Layer at the target. These are known as the iSCSI data-type PDUs and the iSCSI control-type PDUs and these terms are described in the following sections.

9.1 iSCSI Data-Type PDU

An iSCSI data-type PDU is defined as an iSCSI PDU that causes data transfer, transparent to the remote iSCSI layer, to take place between the peer iSCSI nodes in the full feature phase of an iSCSI/iSER connection. An iSCSI data-type PDU, when requested for transmission by the iSCSI Layer in the sending node, results in the data being transferred without the participation of the iSCSI Layers at the sending and the receiving nodes. This is due to the fact that the PDU itself is not delivered as-is to the iSCSI Layer in the receiving node. Instead, the data transfer operations are transformed into the appropriate RDMA operations which are handled by the RNIC. The set of iSCSI data-type PDUs consists of SCSI Data-in PDUs and R2T PDUs.

If the invocation of the Operational Primitive by the iSCSI Layer to request the iSER Layer to process an iSCSI data-type PDU is qualified with Notify_Enable set, then upon completing the RDMA operation, the iSER Layer at the target MUST notify the iSCSI Layer at the target by invoking the Data_Completion_Notify Operational Primitive qualified with ITT and SN. There is no data completion notification at the initiator since the RDMA operations are completely handled by the RNIC at the initiator and the iSER Layer at the initiator is not involved with the data transfer associated with iSCSI data-type PDUs.

If the invocation of the Operational Primitive by the iSCSI Layer to request the iSER Layer to process an iSCSI data-type PDU is qualified with Notify_Enable cleared, then upon completing the RDMA operation, the iSER Layer at the target MUST NOT notify the iSCSI Layer at the target and MUST NOT invoke the Data_Completion_Notify Operational Primitive.

If an operation associated with an iSCSI data-type PDU fails for any reason, the contents of the Data Sink buffers associated with the operation are considered indeterminate.

9.2 iSCSI Control-Type PDU

Any iSCSI PDU that is not an iSCSI data-type PDU and also not a SCSI Data-out PDU carrying solicited data is defined as an iSCSI control-type PDU. The iSCSI Layer invokes the Send_Control Operational

Primitive to request the iSER Layer to process an iSCSI control-type PDU. iSCSI control-type PDUs are transferred using RDMAP Send Message Types. Specifically, it is to be noted that SCSI Data-Out PDUs carrying unsolicited data are defined as iSCSI control-type PDUs. See section 9.3.4 on the treatment of SCSI Data-out PDUs.

When the iSER Layer receives an iSCSI control-type PDU, it MUST notify the iSCSI Layer by invoking the Control_Notify Operational Primitive qualified with the iSCSI control-type PDU.

9.3 iSCSI PDUs

This section describes the handling of each of the iSCSI PDU types by the iSER Layer. The iSCSI Layer requests the iSER Layer to process the iSCSI PDU by invoking the appropriate Operational Primitive. A Connection_Handle MUST qualify each of these invocations. In addition, BHS and the optional AHS of the iSCSI PDU as defined in [iSCSI] MUST qualify each of the invocations. The qualifying Connection_Handle, the BHS and the AHS are not explicitly listed in the subsequent sections.

9.3.1 SCSI Command

The SCSI Command PDU is an iSCSI control-type PDU as described in section 9.2. The iSER Layer at the initiator MUST send the SCSI command in a SendSE Message to the target.

For a SCSI Write or bidirectional command, the iSCSI Layer at the initiator MUST invoke the Send_Control Operational Primitive qualified with ImmediateDataSize, UnsolicitedDataSize, and DataDescriptorOut.

* If there is immediate data to be transferred for the SCSI write or bidirectional command, the qualifier ImmediateDataSize defines the number of bytes of immediate unsolicited data to be sent with the write or bidirectional command, and the qualifier DataDescriptorOut defines the initiator's I/O Buffer containing the SCSI Write data.

* If there is unsolicited data to be transferred for the SCSI Write or bidirectional command, the qualifier UnsolicitedDataSize defines the number of bytes of immediate and non-immediate unsolicited data for the command. The iSCSI Layer will issue one or more SCSI Data-out PDUs for the non-immediate unsolicited data. See Section 9.3.4 on SCSI Data-out.

* If there is solicited data to be transferred for the SCSI Write or bidirectional command, as indicated by the Expected Data Transfer Length in the SCSI Command PDU exceeding the value of

UnsolicitedDataSize, the iSER Layer at the initiator MUST do the following:

a. It MUST allocate a Write STag for the I/O Buffer defined by the qualifier DataDescriptorOut. The DataDescriptorOut describes the I/O buffer starting with the immediate unsolicited data (if any), followed by the non-immediate unsolicited data (if any) and solicited data. This means that the BufferOffset for the SCSI Data-out for this command is equal to the TO. This implies zero TO for this STag points to the beginning of this I/O Buffer.

b. It MUST establish a local mapping that associates the Initiator Task Tag (ITT) to the Write STag.

c. It MUST Advertise the Write STag to the target by sending it as the Write STag in the iSER header of the iSER Message (the payload of the RDMAP SendSE Message) containing the SCSI Write or bidirectional command PDU. See section 11.2 on iSER Header Format for iSCSI Control-Type PDU.

For a SCSI Read or bidirectional command, the iSCSI Layer at the initiator MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorIn which defines the initiator's I/O Buffer for receiving the SCSI Read data. The iSER Layer at the initiator MUST do the following:

a. It MUST allocate a Read STag for the I/O Buffer.

b. It MUST establish a local mapping that associates the Initiator Task Tag (ITT) to the Read STag.

c. It MUST Advertise the Read STag to the target by sending it as the Read STag in the iSER header of the iSER Message (the payload of the RDMAP SendSE Message) containing the SCSI Read or bidirectional command PDU. See section 11.2 on iSER Header Format for iSCSI Control-Type PDU.

If the amount of unsolicited data to be transferred in a SCSI Command exceeds TargetRecvDataSegmentLength, then the iSCSI Layer at the initiator MUST segment the data into multiple iSCSI control-type PDUs, with the data segment length in all PDUs generated except the last one having exactly the size TargetRecvDataSegmentLength. The data segment length of the last iSCSI control-type PDU carrying the unsolicited data can be up to TargetRecvDataSegmentLength.

When the iSER Layer at the target receives the SCSI Command, it MUST establish a remote mapping that associates the ITT to the Advertised Write STag and the Read STag if present in the iSER header. The Write STag is used by the iSER Layer at the target in handling the

data transfer associated with the R2T PDU(s) as described in section 9.3.6. The Read STag is used in handling the SCSI Data-in PDU(s) from the iSCSI Layer at the target as described in section 9.3.5.

9.3.2 SCSI Response

The SCSI Response PDU is an iSCSI control-type PDU as described in section 9.2. The iSCSI Layer at the target MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorStatus which defines the buffer containing the sense and response information. The iSCSI Layer at the target MUST always return the SCSI status for a SCSI command in a separate SCSI Response PDU. "Phase collapse" for transferring SCSI status in a SCSI Data-in PDU MUST NOT be used. The iSER Layer at the target sends the SCSI Response PDU according to the following rules:

* If no STags were Advertised by the initiator in the iSER Message containing the SCSI command PDU, then the iSER Layer at the target MUST send a SendSE Message containing the SCSI Response PDU.

* If the initiator Advertised a Read STag in the iSER Message containing the SCSI Command PDU, then the iSER Layer at the target MUST send a SendInvSE Message containing the SCSI Response PDU. The RDMAP header of the SendInvSE Message MUST carry the Read STag to be invalidated at the initiator.

* If the initiator Advertised only the Write STag in the iSER Message containing the SCSI command PDU, then the iSER Layer at the target MUST send a SendInvSE Message containing the SCSI Response PDU. The RDMAP header of the SendInvSE Message MUST carry the Write STag to be invalidated at the initiator.

When the iSCSI Layer at the target invokes the Send_Control Operational Primitive to send the SCSI Response PDU, the iSER Layer at the target MUST invalidate the remote mapping that associates the ITT to the Advertised STag(s) before transferring the SCSI Response PDU to the initiator.

Upon receiving the SendInvSE Message containing the SCSI Response PDU from the target, the RDMAP layer at the initiator will invalidate the STag specified in the RDMAP header. The iSER Layer at the initiator MUST ensure that the correct STag is invalidated. If both the Read and the Write STags were Advertised earlier by the initiator, then the iSER Layer at the initiator MUST explicitly invalidate the Write STag upon receiving the SendInvSE Message because the RDMAP header of the SendInvSE Message can only carry one STag (in this case the Read STag) to be invalidated.

The iSER Layer at the initiator MUST ensure the invalidation of the STag(s) used in a command before invoking the Control_Notify Operational Primitive qualified with the SCSI Response to notify the iSCSI Layer at the initiator. This precludes the possibility of using the STag(s) after the completion of the command thereby causing data corruption.

When the iSER Layer at the initiator receives the SendSE or the SendInvSE Message containing the SCSI Response PDU, it SHOULD invalidate the local mapping that associates the ITT to the local STag(s). The iSER Layer MUST ensure that all local STag(s) associated with the ITT are invalidated before invoking the Control_Notify Operational Primitive to notify the iSCSI Layer of the SCSI Response PDU.

9.3.3 Task Management Function Request/Response

The Task Management Function Request/Response PDUs are iSCSI control-type PDUs as described in section 9.2. The iSER Layer MUST use a SendSE Message to send the Task Management Function Request /Response PDU.

For the Task Management Function Request with the TASK REASSIGN function, the iSER Layer at the initiator MUST do the following:

* It MUST use the ITT as specified in the Referenced Task Tag from the Task Management Function Request PDU to locate the existing STag(s), if any, in the local mapping(s) that associates the ITT to the local STag(s).

* It MUST invalidate the existing STag(s), if any, and the local mapping(s) that associates the ITT to the local STag(s).

* It MUST allocate a Read STag for the I/O Buffer as defined by the qualifier DataDescriptorIn if the Send_Control Operational Primitive invocation is qualified with DataDescriptorIn.

* It MUST allocate a Write STag for the I/O Buffer as defined by the qualifier DataDescriptorOut if the Send_Control Operational Primitive invocation is qualified with DataDescriptorOut.

* If STags are allocated, it MUST establish new local mapping(s) that associate the ITT to the allocated STag(s).

* It MUST Advertise the STags, if allocated, to the target in the iSER header of the SendSE Message carrying the iSCSI PDU, as described in section 11.2.

For the Task Management Function Request with the TASK REASSIGN function for a SCSI Read or bidirectional command, the iSCSI Layer

at the initiator MUST set ExpDataSN to 0 since the data transfer and acknowledgements happen transparently to the iSCSI Layer at the initiator. This provides the flexibility to the iSCSI Layer at the target to request transmission of only the unacknowledged data as specified in [iSCSI].

When the iSER Layer at the target receives the Task Management Function Request with the TASK REASSIGN function, it MUST do the following:

* It MUST use the ITT as specified in the Referenced Task Tag from the Task Management Function Request PDU to locate the mappings that associate the ITT to the Advertised STag(s) and the local STag(s), if any.

* It MUST invalidate the local STaq(s), if any, associated with the ITT.

* It MUST replace the Advertised STag(s) in the remote mapping that associates the ITT to the Advertised STag(s) with the Write STag and the Read STag if present in the iSER header. The Write STag is used in the handling of the R2T PDU(s) from the iSCSI Layer at the target as described in section 9.3.6. The Read STag is used in the handling of the SCSI Data-in PDU(s) from the iSCSI Layer at the target as described in section 9.3.5.

9.3.4 SCSI Data-out

SCSI Data-out PDUs for unsolicited SCSI Write data are iSCSI control-type PDUs as described in section 9.2. The iSCSI Layer at the initiator MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorOut which defines the initiator's I/O Buffer containing the unsolicited SCSI Write data.

If the amount of unsolicited data to be transferred as SCSI Data-out exceeds TargetRecvDataSegmentLength, then the iSCSI Layer at the initiator MUST segment the data into multiple iSCSI control-type PDUs, with the DataSegmentLength having the value of TargetRecvDataSegmentLength in all PDUs generated except the last one. The DataSegmentLength of the last iSCSI control-type PDU carrying the unsolicited data can be up to TargetRecvDataSegmentLength. The iSCSI Layer at the target MUST perform the reassembly function for the unsolicited data.

For unsolicited data, if the F bit is set to 0 in a SCSI Data-out PDU, the iSER Layer at the initiator MUST use a Send Message to send the SCSI Data-out PDU. If the F bit set to 1, the iSER Layer at the initiator MUST use a SendSE Message to send the SCSI Data-out PDU.

Solicited SCSI Write Data are handled using the R2T mechanism as described in section 9.3.6. Therefore SCSI Data-out PDUs for solicited data should never be requested for transmission by the iSCSI Layer at the initiator. However, if a solicited SCSI Data-out PDU is inadvertently requested (i.e. TTT!=0xffffffff) for transmission by the iSCSI Layer at the initiator, the iSER Layer at the initiator is not required to distinguish it as such. The iSER Layer at the initiator in such a case MAY treat it as an iSCSI control-type PDU and handle it as unsolicited data.

9.3.5 SCSI Data-in

SCSI Data-in PDUs are iSCSI data-type PDUs. When the iSCSI Layer at the target is ready to return the SCSI Read data to the initiator, it MUST invoke the Put_Data Operational Primitive qualified with DataDescriptorIn which defines the SCSI Data-in buffer. See section 9.1 on the general requirement on the handling of iSCSI data-type PDUs. SCSI Data-in PDU(s) are used in SCSI Read data transfer as described in section 11.5.2.

The iSER Layer at the target MUST do the following for each invocation of the Put_Data Operational Primitive:

1. It MUST use the ITT in the SCSI Data-in PDU to locate the remote Read STag in the remote mapping that associates the ITT to Advertised STag(s). The remote mapping was established earlier by the iSER Layer at the target when the SCSI Read Command was received from the initiator.

2. It MUST generate and send an RDMA Write Message containing the read data to the initiator.

    a. It MUST use the remote Read STag as the Data Sink STag of the RDMA Write Message.

    b. It MUST use the Buffer Offset from the SCSI Data-in PDU as the Data Sink Tagged Offset of the RDMA Write Message.

    c. It MUST use DataSegmentLength from the SCSI Data-in PDU to determine the amount of data to be sent in the RDMA Write Message.

3. It MUST associate DataSN and ITT from the SCSI Data-in PDU with the RDMA Write operation. If the Put_Data Operational Primitive invocation was qualified with Notify_Enable set, then when the iSER Layer at the target receives a completion from the RDMAP layer for the RDMA Write Message, the iSER Layer at the target MUST notify the iSCSI Layer by invoking the Data_Completion_Notify Operational Primitive qualified with DataSN and ITT. Conversely, if the Put_Data Operational

Primitive invocation was qualified with Notify_Enable cleared, then the iSER Layer at the target MUST NOT notify the iSCSI Layer on completion and MUST NOT invoke the Data_Completion_Notify Operational Primitive.

When the A-bit is set to 1 in the SCSI Data-in PDU, the iSER Layer at the target MUST notify the iSCSI Layer at the target when the data transfer is complete at the initiator. To perform this additional function, the iSER Layer at the target can take advantage of the operational ErrorRecoveryLevel if previously disclosed by the iSCSI Layer via an earlier invocation of the Notice_Key_Values Operational Primitive. There are two approaches that can be taken:

1. If the iSER Layer at the target knows that the operational ErrorRecoveryLevel is 2, or if the iSER Layer at the target does not know the operational ErrorRecoveryLevel, then the iSER Layer at the target MUST issue a zero-length RDMA Read Message following the RDMA Write Message. When the iSER Layer at the target receives a completion for the RDMA Read Message from the RDMAP layer, implying that the initiator RNIC has completed processing of the RDMA Write Message due to the completion ordering semantics of RDMAP, the iSER Layer at the target MUST invoke the Data_Ack_Notify Operational Primitive qualified with ITT and DataSN to notify the iSCSI Layer at the target.

2. If the iSER Layer at the target knows that the operational ErrorRecoveryLevel is 1, then the iSER Layer at the target MUST do one of the following:

   a. It MUST invoke the Data_Ack_Notify Operational Primitive qualified with ITT and DataSN when it receives the local completion from the RDMAP layer for the RDMA Write Message. This is allowed since digest errors do not occur in iSER (see section 12.1.4.2) and a CRC error will cause the connection to be terminated and the task to be terminated anyway. The local RDMA Write completion from the RDMAP layer guarantees that the RDMAP layer will not access the I/O Buffer again to transfer the data associated with that RDMA Write operation.

   b. Alternatively, it MUST use the same procedure for handling the data transfer completion at the initiator as for ErrorRecoveryLevel 2.

It should be noted that the iSCSI Layer at the target cannot set the A-bit to 1 if the ErrorRecoveryLevel=0.

SCSI status MUST always be returned in a separate SCSI Response PDU. The S bit in the SCSI Data-in PDU MUST always be set to 0. There MUST NOT be a "phase collapse" in the SCSI Data-in PDU.

Since the RDMA Write Message only transfers the data portion of the SCSI Data-in PDU but not the control information in the header, such as ExpCmdSN, if timely updates of such information is crucial, the iSCSI Layer at the initiator MAY issue NOP-out PDUs to request the iSCSI Layer at the target to respond with the information using NOP-in PDUs.

9.3.6 Ready To Transfer (R2T)

The R2T PDU is an iSCSI data-type PDU. In order to send an R2T PDU, the iSCSI Layer at the target MUST invoke the Get_Data Operational Primitive qualified with DataDescriptorOut which defines the I/O Buffer for receiving the SCSI Write data from the initiator. See section 9.1 on the general requirements on the handling of iSCSI data-type PDUs.

The iSER Layer at the target MUST do the following for each invocation of the Get_Data Operational Primitive:

1. It MUST ensure a valid local STag for the I/O Buffer and a valid local mapping that associates the Initiator Task Tag (ITT) to the local STag. This may involve allocating a valid local STag and establishing a local mapping.

2. It MUST use the ITT in the R2T to locate the remote Write STag in the remote mapping that associates the ITT to Advertised STag(s). The remote mapping was established earlier by the iSER Layer at the target when the iSER Message containing the Advertised Write STag and the SCSI Command PDU for a SCSI Write or bidirectional command was received from the initiator.

3. If the iSER-ORD value at the target is set to 0, the iSER Layer at the target MUST terminate the connection and free up the resources associated with the connection (as described in 7.2.3) if it received the R2T PDU from the iSCSI Layer at the target. Upon termination of the connection, the iSER Layer at the target MUST notify the iSCSI Layer at the target using the Connection_Terminate_Notify Operational Primitive.

4. If the iSER-ORD value at the target is set to greater than 0, the iSER Layer at the target MUST transform the R2T PDU into an RDMA Read Request Message. While transforming the R2T PDU, the iSER Layer at the target MUST ensure that the number of outstanding RDMA Read Request Messages does not exceed iSER-ORD value. To transform the R2T PDU, the iSER Layer at the target:

   a. MUST derive the local STag and local Tagged Offset from the DataDescriptorOut that qualified the Get_Data invocation.

b. MUST use the local STag as the Data Sink STag of the RDMA Read Request Message.

c. MUST use the local Tagged Offset as the Data Sink Tagged Offset of the RDMA Read Request Message.

d. MUST use the Desired Data Transfer Length from the R2T PDU as the RDMA Read Message Size of the RDMA Read Request Message.

e. MUST use the remote Write STag as the Data Source STag of the RDMA Read Request Message.

f. MUST use the Buffer Offset from the R2T PDU as the Data Source Tagged Offset of the RDMA Read Request Message.

5. It MUST associate R2TSN and ITT from the R2T PDU with the RDMA Read operation. If the Get_Data Operational Primitive invocation was qualified with Notify_Enable set, then when the iSER Layer at the target receives a completion from the RDMAP layer for the RDMA Read operation, the iSER Layer at the target MUST notify the iSCSI Layer by invoking the Data_Completion_Notify Operational Primitive qualified with R2TSN and ITT. Conversely, if the Get_Data Operational Primitive invocation was qualified with Notify_Enable cleared, then the iSER Layer at the target MUST NOT notify the iSCSI Layer on completion and MUST NOT invoke the Data_Completion_Notify Operational Primitive.

When the RDMAP layer at the initiator receives a valid RDMA Read Request Message, it will return an RDMA Read Response Message containing the solicited write data to the target. When the RDMAP layer at target receives the RDMA Read Response Message from the initiator, it will place the solicited data in the I/O Buffer referenced by the Data Sink STag in the RDMA Read Response Message.

Since the RDMA Read Request Message from the target does not transfer the control information in the R2T PDU such as ExpCmdSN, if timely updates of such information is crucial, the iSCSI Layer at the initiator MAY issue NOP-out PDUs to request the iSCSI Layer at the target to respond with the information using NOP-in PDUs.

Similarly, since the RDMA Read Response Message from the initiator only transfers the data but not the control information normally found in the SCSI Data-out PDU, such as ExpStatSN, if timely updates of such information is crucial, the iSCSI Layer at the target MAY issue NOP-in PDUs to request the iSCSI Layer at the initiator to respond with the information using NOP-out PDUs.

9.3.7 Asynchronous Message

The Asynchronous Message PDU is an iSCSI control-type PDU as described in section 9.2. The iSCSI Layer MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorSense which defines the buffer containing the sense and iSCSI Event information. The iSER Layer MUST use a SendSE Message to send the Asynchronous Message PDU.

9.3.8 Text Request & Text Response

The Text Request and Text Response PDUs are iSCSI control-type PDUs as described in section 9.2. The iSCSI Layer MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorTextOut (or DataDescriptorIn) which defines the Text Request (or Text Response) buffer. The iSER Layer MUST use SendSE Messages to send the Text Request and Text Response PDUs.

9.3.9 Login Request & Login Response

The Login Request PDUs and the Login Response PDUs are exchanged when the connection between the initiator and the target is still in the byte stream mode. During the login negotiation, the iSCSI Layer interacts with the TCP layer directly and the iSER Layer is not involved. See section 7.1 on iSCSI/iSER Connection Setup.

If the iSCSI Layer attempts to send a Login Request (or a Login Response) PDU during the full feature phase, it MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorLoginRequest (or DataDescriptorLoginResponse) which defines the Login Request (or Login Response) buffer. The iSER Layer MUST handle it as an iSCSI control-type PDU as described in section 9.2, and use SendSE Messages to send the Login Request and Login Response PDUs.

9.3.10 Logout Request & Logout Response

The Logout Request and Logout Response PDUs are iSCSI control-type PDUs as described in section 9.2. The iSER Layer MUST use a SendSE Message to send the Logout Request or Logout Response PDU. Section 7.2.1 and 7.2.2 describe the handling of the Logout Request and the Logout Response at the initiator and the target and the interactions between the initiator and the target to terminate a connection.

9.3.11 SNACK Request

Since HeaderDigest and DataDigest must be negotiated to "None", there are no digest errors when the connection is in iSER-assisted mode. Also since RDMAP delivers all messages in the order they were sent, there are no sequence errors when the connection is in iSER-

assisted mode. Therefore the iSCSI Layer SHOULD NOT send SNACK Request PDUs. In particular, the Proactive (Time out) SNACK SHOULD NOT be issued. If the iSCSI Layer invokes the Send_Control Operational Primitive to request the iSER Layer to send a SNACK Request, the iSER Layer MUST handle it as an iSCSI control-type PDU as described in section 9.2, and use a SendSE Message to send the SNACK Request PDU. Upon receiving the iSER Message containing the SNACK PDU, the iSER Layer notifies the iSCSI Layer using the Control_Notify Operational Primitive.

9.3.12 Reject

The Reject PDU is an iSCSI control-type PDU as described in section 9.2. The iSCSI Layer MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorReject which defines the Rejct buffer. The iSER Layer MUST use a SendSE Message to send the Reject PDU.

9.3.13 NOP-Out & NOP-In

The NOP-Out and NOP-In PDUs are iSCSI control-type PDUs as described in section 9.2. The iSCSI Layer MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorNOPOut (or DataDescriptorNOPIn) which defines the Ping (or Return Ping) data buffer. The iSER Layer MUST use SendSE Messages to send the NOP-Out and NOP-In PDUs.

10 Flow Control and STag Management

10.1 Flow Control for RDMA Send Message Types

RDMAP Send Message Types are used by the iSER Layer to transfer iSCSI control-type PDUs. Each RDMAP Send Message Type consumes an Untagged Buffer at the Data Sink. However, neither the RDMAP layer nor the iSER Layer provides an explicit flow control mechanism for the RDMAP Send Message Types. Therefore, the iSER Layer SHOULD provision enough Untagged buffers for handling incoming RDMAP Send Message Types to prevent a buffer underrun condition at the RDMAP layer. If a buffer underrun happens, it may result in the termination of the connection. An implementation may choose to satisfy this requirement by using a common buffer pool shared across multiple connections, with usage limits on a per connection basis and usage limits on the buffer pool itself. In such an implementation, exceeding the buffer usage limit for a connection or the buffer pool itself may trigger interventions from the iSER Layer to replenish the buffer pool and/or to isolate the connection causing the problem.

10.2 Flow Control for RDMA Read Resources

The total number of RDMA Read operations that can be active simultaneously on an iSCSI/iSER connection depends on the amount of resources allocated as declared in the iSER Hello exchange described in section 7.1.3. Exceeding the number of RDMA Read operations allowed on a connection will result in the connection being terminated by the RDMAP layer. The iSER Layer at the target maintains the iSER-ORD to keep track of the maximum number of RDMA Read Requests that can be issued by the iSER Layer on a particular RDMAP Stream.

During connection setup (see section 7.1), iSER-IRD is known at the initiator and iSER-ORD is known at the target after the iSER Layers at the initiator and the target have respectively allocated the iWARP resources for the connection, as directed by the Allocate_Connection_Resources Operational Primitive from the iSCSI Layer before the end of the iSCSI Login Phase. In the full feature phase, the first message sent by the initiator is the iSER Hello Message (see section 11.3) which contains the value of iSER-IRD. In response to the iSER Hello Message, the target sends the iSER HelloReply Message (see section 11.4) which contains the value of iSER-ORD. The iSER Layer at both the initiator and the target MAY adjust (lower) the iWARP resources associated with iSER-IRD and iSER-ORD respectively to match the iSER-ORD value declared in the HelloReply Message. The iSER Layer at the target MUST flow control the RDMA Read Request Messages to not exceed the iSER-ORD value at the target.

10.3 STag Management

An STag, as defined in [RDMAP], is an identifier of a Tagged Buffer used in an RDMA operation. The allocation and the subsequent invalidation of the STags are specified in this document if the STags are exposed on the wire by being Advertised in the iSER header or declared in the RDMAP header of an iWARP Message.

10.3.1 Allocation of STags

When the iSCSI Layer at the initiator invokes the Send_Control Operational Primitive to request the iSER Layer at the initiator to process a SCSI Command, zero, one, or two STags may be allocated by the iSER Layer. See section 9.3.1 for details. The number of STags allocated depends on whether the command is unidirectional or bidirectional and whether solicited write data transfer is involved or not.

When the iSCSI Layer at the initiator invokes the Send_Control Operational Primitive to request the iSER Layer at the initiator to process a Task Management Function Request with the TASK REASSIGN function, besides allocating zero, one, or two STags, the iSER Layer MUST invalidate the existing STags, if any, associated with the ITT. See section 9.3.3 for details.

The iSER Layer at the target allocates a local Data Sink STag when the iSCSI Layer at the target invokes the Get_Data Operational Primitive to request the iSER Layer to process an R2T PDU. See section 9.3.6 for details.

10.3.2 Invalidation of STags

The invalidation of the STags at the initiator at the completion of a unidirectional or bidirectional command when the associated SCSI Response PDU is sent by the target is described in section 9.3.2.

When a unidirectional or bidirectional command concludes without the associated SCSI Response PDU being sent by the target, the iSCSI Layer at the initiator MUST invoke the Deallocate_Task_Resources Operational Primitive qualified with ITT. In response, the iSER Layer at the initiator MUST locate the STag(s) (if any) in the local mapping that associates the ITT to the local STag(s). The iSER Layer at the initiator MUST invalidate the STag(s) (if any) and the local mapping.

For an RDMA Read operation used to realize a SCSI Write data transfer, the iSER Layer at the target SHOULD invalidate the Data Sink STag at the conclusion of the RDMA Read operation referencing the Data Sink STag (to permit the immediate reuse of buffer resources).

For an RDMA Write operation used to realize a SCSI Read data transfer, the Data Source STag at the target is not declared to the initiator and is not exposed on the wire. Invalidation of the STag is thus not specified.

When a unidirectional or bidirectional command concludes without the associated SCSI Response PDU being sent by the target, the iSCSI Layer at the target MUST invoke the Deallocate_Task_Resources Operational Primitive qualified with ITT. In response, the iSER Layer at the target MUST locate the local STag(s) (if any) in the local mapping that associates the ITT to the local STag(s). The iSER Layer at the target MUST invalidate the local STag(s) (if any) and the mapping.

11 iSER Control and Data Transfer

For iSCSI data-type PDUs (see section 9.1), the iSER Layer uses RDMA Read and RDMA Write operations to transfer the solicited data. For iSCSI control-type PDUs (see section 9.2), the iSER Layer uses RDMAP Send Message Types.

11.1 iSER Header Format

An iSER header MUST be present in every RDMAP Send Message Type. The iSER header is located in the first 12 bytes of the message payload of the RDMAP Send Message Type, as shown in Figure 2.

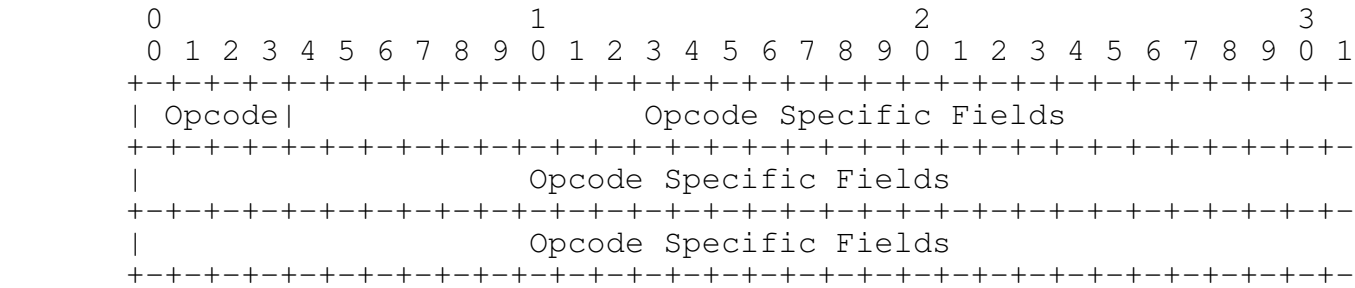

Figure 2 iSER Header Format

Opcode – Operation Code: 4 bits

The Opcode field identifies the type of iSER Messages:

0001b = iSCSI control-type PDU

0010b = iSER Hello Message

0011b = iSER HelloReply Message

All other opcodes are reserved.

11.2 iSER Header Format for iSCSI Control-Type PDU

The iSER Layer uses RDMAP Send Message Types to transfer iSCSI control-type PDUs (see section 9.2). The message payload of each of the RDMAP Send Message Types used for transferring an iSER Message contains an iSER Header followed by an iSCSI control-type PDU.

The iSER header in an RDMAP Send Message Type carrying an iSCSI control-type PDU MUST have the format as described in Figure 3.

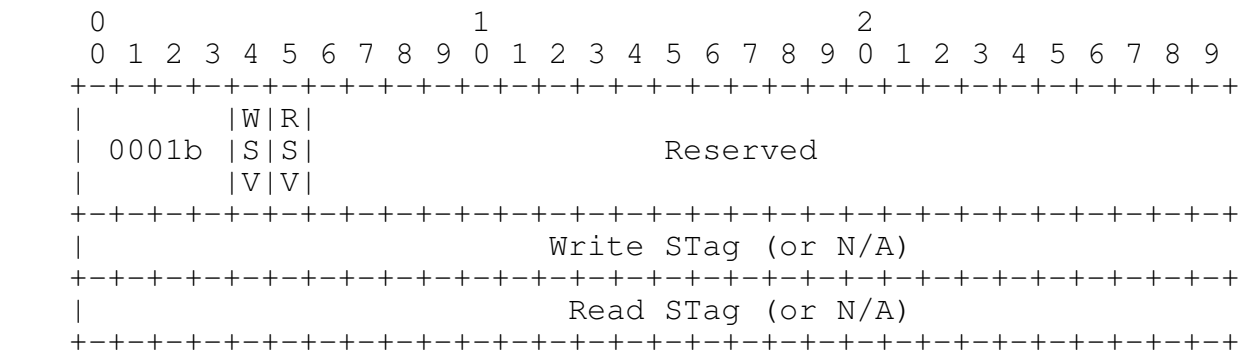

Figure 3 iSER Header Format for iSCSI Control-Type PDU

WSV – Write STag Valid flag: 1 bit

This flag indicates the validity of the Write STag field of the iSER Header. If set to one, the Write STag field in this iSER Header is valid. If set to zero, the Write STag field in this iSER Header MUST be ignored at the receiver. The Write STag Valid flag is set to one when there is solicited data to be transferred for a SCSI Write or bidirectional command, or when there are non-immediate unsolicited and solicited data to be transferred for the referenced task specified in a Task Management Function Request with the TASK REASSIGN function.

RSV – Read STag Valid flag: 1 bit

This flag indicates the validity of the Read STag field of the iSER Header. If set to one, the Read STag field in this iSER Header is valid. If set to zero, the Read STag field in this iSER Header MUST be ignored at the receiver. The Read STag Valid flag is set to one for a SCSI Read or bidirectional command, or a Task Management Function Request with the TASK REASSIGN function.

Write STag – Write Steering Tag: 32 bits.

This field contains the Write STag when the Write STag Valid flag is set to one. For a SCSI Write or bidirectional command, the Write STag is used to Advertise the initiator’s I/O Buffer containing the solicited data. For a Task Management Function Request with the TASK REASSIGN function, the Write STag is used to Advertise the initiator's I/O Buffer containing the non-immediate unsolicited data and solicited data. This Write STag is used as the Data Source STag in the resultant RDMA Read operation(s). When the Write STag Valid flag is set to zero, this field MUST be set to zero.

Read STag - Read Steering Tag: 32 bits.

This field contains the Read STag when the Read STag Valid flag is set to one. The Read STag is used to Advertise the initiator's Read I/O Buffer of a SCSI Read or bidirectional command, or a Task Management Function Request with the TASK REASSIGN function. This Read STag is used as the Data Sink STag in the resultant RDMA Write operation(s). When the Read STag Valid flag is zero, this field MUST be set to zero.

Reserved:

Reserved fields MUST be set to zero on transmit and MUST be ignored on receive.

11.3 iSER Header Format for iSER Hello Message

An iSER Hello Message MUST only contain the iSER header which MUST have the format as described in Figure 4. iSER Hello Message is the first RDMAP Message sent on the RDMAP Stream from the iSER Layer at the initiator to the iSER Layer at the target.

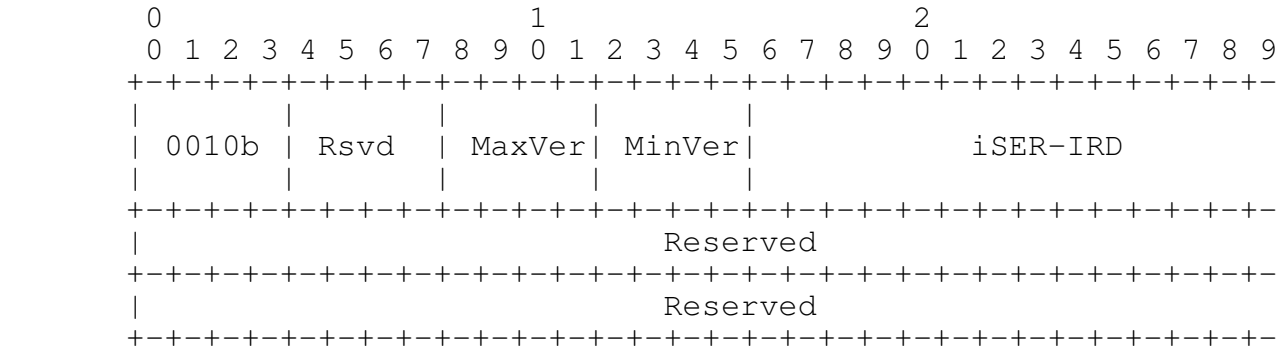

Figure 4 iSER Header Format for iSER Hello Message

MaxVer - Maximum Version: 4 bits

This field specifies the maximum version of the iSER protocol supported. It MUST be set to 0 to indicate the version of the specification described in this document.

MinVer - Minimum Version: 4 bits

This field specifies the minimum version of the iSER protocol supported. It MUST be set to 0 to indicate the version of the specification described in this document.

iSER-IRD: 16 bits

This field contains the value of the iSER-IRD at the initiator.

Reserved (Rsvd):

Reserved fields MUST be set to zero on transmit, and MUST be ignored on receive.

11.4 iSER Header Format for iSER HelloReply Message

An iSER HelloReply Message MUST only contain the iSER header which MUST have the format as described in Figure 5. The iSER HelloReply Message is the first RDMAP Message sent on the RDMAP Stream from the iSER Layer at the target to the iSER Layer at the initiator.

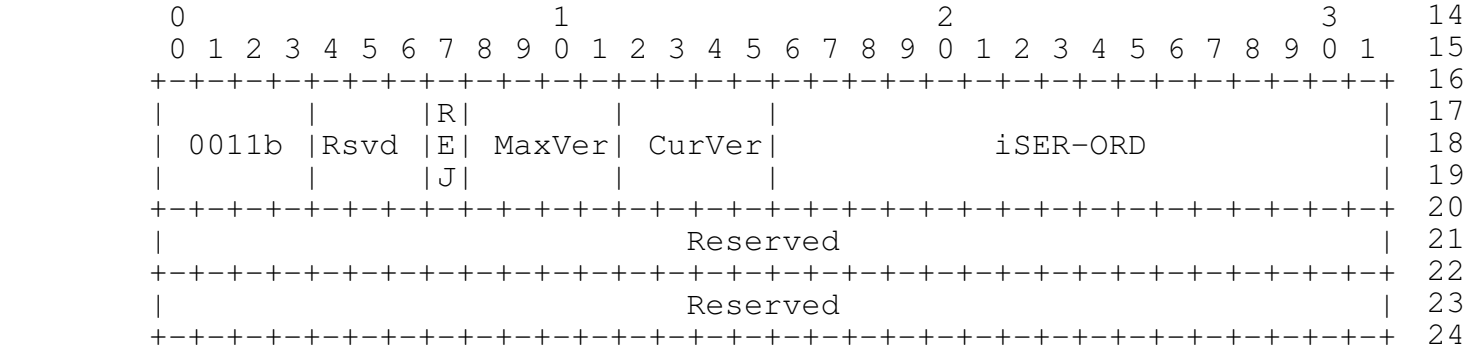

Figure 5 iSER Header Format for iSER HelloReply Message

REJ – Reject flag: 1 bit

This flag indicates whether the target is rejecting this connection. If set to one, the target is rejecting the connection.

MaxVer – Maximum Version: 4 bits

This field specifies the maximum version of the iSER protocol supported. It MUST be set to 0 to indicate the version of the specification described in this document.

CurVer – Current Version: 4 bits

This field specifies the current version of the iSER protocol supported. It MUST be set to 0 to indicate the version of the specification described in this document.

iSER-ORD: 16 bits

This field contains the value of the iSER-ORD at the target.

Reserved (Rsvd):

Reserved fields MUST be set to zero on transmit, and MUST be ignored on receive.

11.5 SCSI Data Transfer Operations

The iSER Layer at the initiator and the iSER Layer at the target handle each SCSI Write, SCSI Read, and bidirectional operation as described below.

11.5.1 SCSI Write Operation

The iSCSI Layer at the initiator MUST invoke the Send_Control Operational Primitive to request the iSER Layer at the initiator to send the SCSI Write Command. The iSER Layer at the initiator MUST request the RDMAP layer to transmit a SendSE Message with the message payload consisting of the iSER header followed by the SCSI Command PDU and immediate data (if any). If there is solicited data, the iSER Layer MUST Advertise the Write STag in the iSER header of the SendSE Message, as described in section 11.2. Upon receiving the SendSE Message, the iSER Layer at the target MUST notify the iSCSI Layer at the target by invoking the Control_Notify Operational Primitive qualified with the SCSI Command PDU. See section 9.3.1 for details on the handling of the SCSI Write Command.

For the non-immediate unsolicited data, the iSCSI Layer at the initiator MUST invoke a Send_Control Operational Primitive qualified with the SCSI Data-out PDU. Upon receiving each Send or SendSE Message containing the non-immediate unsolicited data, the iSER Layer at the target MUST notify the iSCSI Layer at the target by invoking the Control_Notify Operational Primitive qualified with the SCSI Data-out PDU. See section 9.3.4 for details on the handling of the SCSI Data-out PDU.

For the solicited data, when the iSCSI Layer at the target has an I/O Buffer available, it MUST invoke the Get_Data Operational Primitive qualified with the R2T PDU. See section 9.3.6 for details on the handling of the R2T PDU.

When the data transfer associated with this SCSI Write operation is complete, the iSCSI Layer at the target MUST invoke the Send_Control Operational Primitive when it is ready to send the SCSI Response PDU. Upon receiving a SendSE or SendInvSE Message containing the SCSI Response PDU, the iSER Layer at the initiator MUST notify the iSCSI Layer at the initiator by invoking the Control_Notify Operational Primitive qualified with the SCSI Response PDU. See section 9.3.2 for details on the handling of the SCSI Response PDU.

11.5.2 SCSI Read Operation

The iSCSI Layer at the initiator MUST invoke the Send_Control Operational Primitive to request the iSER Layer at the initiator to send the SCSI Read Command. The iSER Layer at the initiator MUST request the RDMAP layer to transmit a SendSE Message with the message payload consisting of the iSER header followed by the SCSI Command PDU. The iSER Layer at the initiator MUST Advertise the Read STag in the iSER header of the SendSE Message, as described in section 11.2. Upon receiving the SendSE Message, the iSER Layer at the target MUST notify the iSCSI Layer at the target by invoking the Control_Notify Operational Primitive qualified with the SCSI Command PDU. See section 9.3.1 for details on the handling of the SCSI Read Command.

When the requested SCSI data is available in the I/O Buffer, the iSCSI Layer at the target MUST invoke the Put_Data Operational Primitive qualified with the SCSI Data-in PDU. See section 9.3.5 for details on the handling of the SCSI Data-in PDU.

When the data transfer associated with this SCSI Read operation is complete, the iSCSI Layer at the target MUST invoke the Send_Control Operational Primitive when it is ready to send the SCSI Response PDU. Upon receiving the SendInvSE Message containing the SCSI Response PDU, the iSER Layer at the initiator MUST notify the iSCSI Layer at the initiator by invoking the Control_Notify Operational Primitive qualified with the SCSI Response PDU. See section 9.3.2 for details on the handling of the SCSI Response PDU.

11.5.3 Bidirectional Operation

The initiator and the target handle the SCSI Write and the SCSI Read portions of this bidirectional operation in a similar manner as described in Section 11.5.1 and Section 11.5.2 respectively.

12 iSER Error Handling and Recovery

[RDMAP] and the protocols below it provide the iSER Layer with reliable in-order delivery. Therefore, the error management needs of an iSCSI/iSER connection are somewhat different than those of traditional iSCSI running directly over TCP.

12.1 Error Handling

iSER error handling is described in the following sections, classified loosely based on the sources of errors:

1. Those originating at the TCP layer.

2. Those originating at the RDMAP layer.

3. Those originating at the iSER Layer.

4. Those originating at the iSCSI Layer.

12.1.1 Errors in the TCP Layer

TCP packets with errors are silently dropped by the TCP layer and result in retransmission at the TCP layer. This has no impact on the iSER Layer. However, connection loss (e.g., link failure) and unexpected termination (e.g., TCP graceful or abnormal close without the iSCSI Logout exchanges) at the TCP layer will cause the iSCSI/iSER connection to be terminated as well.

12.1.1.1 TCP Failure Before iWARP is Enabled

If the TCP connection is lost or terminated before the iSCSI Layer invokes the Allocate_Connection_Resources Operational Primitive, the login process is terminated and no further action is required.

If the TCP connection is lost or terminated after the iSCSI Layer has invoked the Allocate_Connection_Resources Operational Primitive, then the iSCSI Layer MUST invoke the Deallocate_Connection_Resources Operational Primitive to request the iSER Layer to deallocate the iWARP resources for the connection.

12.1.1.2 TCP Failure After iWARP is Enabled

If the TCP connection is lost or terminated after the iSCSI Layer has invoked the Enable_Datamover Operational Primitive, the iSER Layer MUST notify the iSCSI Layer of the TCP connection loss by invoking the Connection_Terminate_Notify Operational Primitive. Prior to invoking the Connection_Terminate_Notify Operational Primitive, the iSER layer MUST perform the actions described in Section 7.2.3.2.

12.1.2 Errors in the iWARP protocol suite

The RDMAP layer does not have error recovery operations built in. If errors are detected at the RDMAP layer, the RDMAP layer will terminate the RDMAP Stream and the associated TCP connection.

12.1.2.1 Errors Detected in the Local RDMAP Layer

If an error is encountered at the local RDMAP layer, the RDMAP layer MAY send a Terminate Message to the Remote Peer to report the error if possible. (See [RDMAP] for the list of errors where a Terminate Message is sent.) The RDMAP layer is responsible for terminating the TCP connection. After the RDMAP layer notifies the iSER Layer that the TCP connection is terminated, the iSER Layer MUST notify the iSCSI Layer by invoking the Connection_Terminate_Notify Operational Primitive. Prior to invoking the Connection_Terminate_Notify Operational Primitive, the iSER layer MUST perform the actions described in Section 7.2.3.2.

12.1.2.2 Errors Detected in the RDMAP Layer at the Remote Peer

If an error is encountered at the RDMAP layer at the Remote Peer, the RDMAP layer at the Remote Peer may send a Terminate Message to report the error if possible. If it is unable to send the Terminate Message, the TCP connection is terminated. This is treated similar to a TCP failure after iWARP is enabled as described in section 12.1.1.2.

If an error is encountered at the RDMAP layer at the Remote Peer and it is able to send a Terminate Message, the RDMAP layer at the Remote Peer is responsible for terminating the TCP connection. After the local RDMAP layer notifies the iSER Layer that the TCP connection is terminated, the iSER Layer MUST notify the iSCSI Layer by invoking the Connection_Terminate_Notify Operational Primitive. Prior to invoking the Connection_Terminate_Notify Operational Primitive, the iSER layer MUST perform the actions described in Section 7.2.3.2.

12.1.3 Errors in the iSER Layer

The error handling due to errors at the iSER Layer is described in the following sections.

12.1.3.1 Insufficient iWARP Resources at the Initiator at Connection Setup

After the iSCSI Layer at the initiator invokes the Allocate_Connection_Resources Operational Primitive during the iSCSI

login negotiation phase, if the iSER Layer at the initiator fails to allocate the necessary iWARP resources, it MUST return a status of failure to the iSCSI Layer at the initiator. The iSCSI Layer at the initiator MUST terminate the TCP connection as described in Section 7.2.3.1.

12.1.3.2 Insufficient iWARP Resources at the Target at Connection Setup

After the iSCSI Layer at the target invokes the Allocate_Connection_Resources Operational Primitive during the iSCSI login negotiation phase, if the iSER Layer at the target fails to allocate the necessary iWARP resources, it MUST return a status of failure to the iSCSI Layer at the target. The iSCSI Layer at the target MUST send a Login Response with a status class of 3 (Target Error), and a status code of "0302" (Out of Resources). The iSCSI Layers at the initiator and the target MUST terminate the TCP connection as described in Section 7.2.3.1.

12.1.3.3 iSER Negotiation Failures

If the iWARP or iSER related parameters declared by the initiator in the iSER Hello Message is unacceptable to the iSER Layer at the target, the iSER Layer at the target MUST set the Reject (REJ) flag, as described in section 11.4, in the iSER HelloReply Message. The following are the cases when the iSER Layer MUST set the REJ flag to 1 in the HelloReply Message:

* The initiator-declared iSER-IRD value is greater than 0 and the target-declared iSER-ORD value is 0.

* The initiator-supported and the target-supported iSER protocol versions do not overlap.

After requesting the RDMAP layer to send the iSER HelloReply Message, the handling of the error situation is similar to that for iSER format errors, as described in section 12.1.3.4.

12.1.3.4 iSER Format Errors

The following types of errors in an iSER header are considered format errors:

* Illegal contents of any iSER header field

* Inconsistent field contents in an iSER header

* Length error for an iSER Hello or HelloReply Message (see section 11.3 and 11.4)

When a format error is detected, the following events MUST occur in the specified sequence:

1. The iSER Layer MUST request the RDMAP layer to terminate the RDMAP Stream. The RDMAP layer MUST terminate the associated TCP connection.

2. The iSER Layer MUST notify the iSCSI Layer by invoking the Connection_Terminate_Notify Operational Primitive. Prior to invoking the Connection_Terminate_Notify Operational Primitive, the iSER layer MUST perform the actions described in Section 7.2.3.2.

12.1.3.5 iSER Protocol Errors

The first iSER Message sent by the iSER Layer at the initiator after transitioning into iSER-assisted mode MUST be the iSER Hello Message (see section 11.3). Likewise, the first iSER Message sent by the iSER Layer at the target after transitioning into iSER-assisted mode MUST be the iSER HelloReply Message (see section 11.4). Failure to send the iSER Hello or HelloReply Message, as indicated by the wrong Opcode in the iSER header, is a protocol error.

The handling of an iSER protocol error is similar to that for iSER format errors, as described in section 12.1.3.4.

12.1.4 Errors in the iSCSI Layer

The error handling due to errors at the iSCSI Layer is described in the following sections. For error recovery, see section 12.2.

12.1.4.1 iSCSI Format Errors

When an iSCSI format error is detected, the iSCSI Layer MUST invoke the Connection_Terminate Operational Primitive to request the iSER Layer to terminate the RDMAP Stream. For more details on the connection termination, see Section 7.2.3.1.

12.1.4.2 iSCSI Digest Errors

In the iSER-assisted mode, the iSCSI Layer will not see any digest error because both the HeaderDigest and the DataDigest keys are negotiated to "None".

12.1.4.3 iSCSI Sequence Errors

For traditional iSCSI, sequence errors are caused by dropped PDUs due to header or data digest errors. Since digests are not used in iSER-assisted mode and the RDMAP layer will deliver all messages in

the order they were sent, sequence errors will not occur in iSER-assisted mode.

12.1.4.4 iSCSI Protocol Error

When the iSCSI Layer handles certain protocol errors by dropping the connection, the error handling is similar to that for iSCSI format errors as described in section 12.1.4.1

When the iSCSI Layer uses the iSCSI Reject PDU and response codes to handle certain other protocol errors, no special handling at the iSER Layer is required.

12.1.4.5 SCSI Timeouts and Session Errors

This is handled at the iSCSI Layer and no special handling at the iSER Layer is required.

12.1.4.6 iSCSI Negotiation Failures

For negotiation failures that happen during the Login Phase at the initiator after the iSCSI Layer has invoked the Allocate_Connection_Resources Operational Primitive and before the Enable_Datamover Operational Primitive has been invoked, the iSCSI Layer MUST invoke the Deallocate_Connection_Resources Operational Primitive for the iSER Layer to deallocate the iWARP resources for the connection. The iSCSI Layer at the initiator MUST terminate the TCP connection.

For negotiation failures during the Login Phase at the target, the iSCSI Layer can use a Login Response with a status class other than 0 (success) to terminate the Login Phase. If the iSCSI Layer has invoked the Allocate_Connection_Resources Operational Primitive and before the Enable_Datamover Operational Primitive has been invoked, the iSCSI Layer at the target MUST invoke the Deallocate_Connection_Resources Operational Primitive to request the iSER Layer at the target to deallocate the iWARP resources for the connection. The iSCSI Layer at both the initiator and the target MUST terminate the TCP connection.

During the iSCSI Login Phase, if the iSCSI Layer at the initiator receives a Login Response from the target with a status class other than 0 (Success) after the iSCSI Layer at the initiator has invoked the Allocate_Connection_Resources Operational Primitive, the iSCSI Layer MUST invoke the Deallocate_Connection_Resources Operational Primitive to request the iSER Layer to deallocate all iWARP resources for the connection. The iSCSI Layer MUST terminate the TCP connection in this case.

For negotiation failures during the full feature phase, the error handling is left to the iSCSI Layer and no special handling at the iSER Layer is required.

12.2 Error Recovery

Error recovery requirements of iSCSI/iSER are the same as that of traditional iSCSI. All three ErrorRecoveryLevels as defined in [iSCSI] are supported in iSCSI/iSER.

* For ErrorRecoveryLevel 0, session recovery is handled by iSCSI and no special handling by the iSER Layer is required.

* For ErrorRecoveryLevel 1, see section 12.2.1 on SNACK Handling and PDU Recovery.

* For ErrorRecoveryLevel 2, see section 12.2.2 on Connection Recovery.

The iSCSI Layer MAY invoke the Notice_Key_Values Operational Primitive during connection setup to request the iSER Layer to take note of the value of the operational ErrorRecoveryLevel, as described in sections 7.1.1 and 7.1.2.

12.2.1 SNACK Handling and PDU Recovery

As described in sections 12.1.4.2 and 12.1.4.3, digest and sequence errors will not occur in the iSER-assisted mode. If the RDMAP layer detects an error, it will close the iSCSI/iSER connection, as described in section 12.1.2. Therefore, PDU recovery is not useful in the iSER-assisted mode.

The iSCSI Layer at the initiator SHOULD disable timeout-driven proactive SNACKs. If the iSCSI Layer at the target receives a SNACK, it MUST respond to it as required by [iSCSI].

The iSCSI Layer at the initiator SHOULD disable iSCSI timeout-driven PDU retransmissions.

12.2.2 Connection Recovery

The iSCSI Layer at the initiator MAY reassign connection allegiance for non-immediate commands which are still in progress and are associated with the failed connection by using a Task Management Function Request with the TASK REASSIGN function. See section 9.3.3 for more details.

When the iSCSI Layer at the initiator does a task reassignment for a SCSI Write command, it MUST qualify the Send_Control Operational Primitive invocation with DataDescriptorOut which defines the I/O

Buffer for both the non-immediate unsolicited data and the solicited data. This allows the iSCSI Layer at the target to use recovery R2Ts to request for data originally sent as unsolicited and solicited from the initiator.

When the iSCSI Layer at the target accepts a reassignment request for a SCSI Read command, it MUST invoke the Put_Data Operational Primitive to request the iSER Layer to process SCSI Data-in for all unacknowledged data. See section 9.3.5 on the handling of SCSI Data-in.

When the iSCSI Layer at the target accepts a reassignment request for a SCSI Write command, it MUST invoke the Get_Data Operational Primitive to request the iSER Layer to process a recovery R2T for any non-immediate unsolicited data and any solicited data sequences that have not been received. See section 9.3.6 on the handling of Ready To Transfer (R2T).

The iSCSI Layer at the target MUST NOT issue recovery R2Ts on an iSCSI/iSER connection for a task for which the connection allegiance was never reassigned. The iSER Layer at the target MAY reject such a recovery R2T received via the Get_Data Operational Primitive invocation from the iSCSI Layer at the target, with an appropriate error code.

The iSER Layer at the target will process the requests invoked by the Put_Data and Get_Data Operational Primitives for a reassigned task in the same way as for the original commands.

13 Security Considerations

Since iSER is layered on top of the iWARP layer and provides the RDMA extensions to the iSCSI protocol, the security considerations of iSER are similar to that of the underlying RDMAP layer as described in [RDMAP].

All the security protocol mechanisms described in [iSCSI] MAY be deployed for an iSCSI/iSER connection. If the IPsec mechanism is used, then it MUST be established before the connection transitions from the traditional iSCSI mode to the iSER-assisted mode.

14 IANA Considerations

The login operational keys RDMAExtensions, InitiatorRecvDataSegmentLength, and TargetRecvDataSegmentLength will be registered with IANA before this draft is approved to become an RFC.

15 References

15.1 Normative References

[RDMAP] R. Recio et al., "An RDMA Protocol Specification", RDMA Consortium Draft Specification draft-recio-iwarp-rdmap-v1.0, October 2002

[DDP] H. Shah et al., "Direct Data Placement over Reliable Transports", RDMA Consortium Draft Specification draft-shah-iwarp-ddp-v1.0, October 2002

[MPA] P. Culley et al., "Marker PDU Aligned Framing for TCP Specification", RDMA Consortium Draft Specification draft-culley-iwarp-mpa-v1.0, October 2002

[DA] M. Chadalapaka et al., "Datamover Architecture for iSCSI", RDMA Consortium Draft Specification draft-chadalapaka-iwarp-da-v1.0, July 2003

[TCP] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981

[iSCSI] J. Satran et al., "iSCSI", IETF Internet-draft draft-ietf-ips-iSCSI-20.txt (work in progress), January 2003

15.2 Informative References

[IPSEC] S. Kent et al., "Security Architecture for the Internet Protocol", RFC 2401, November 1998

[SAM2] T10/1157D, SCSI Architecture Model – 2 (SAM-2)

[VERBS] J. Hilland et al., "RDMA Protocol Verbs Specification", RDMAC Consortium Draft Specification draft-hilland-iwarp-verbs-v1.0-RDMAC, April 2003

16 Appendix

16.1 iWARP Message Format for iSER

This section is for information only and is NOT part of the standard. It simply depicts the iWARP Message format for the various iSER Messages.

16.1.1 iWARP Message Format for iSER Hello Message

The following figure depicts an iSER Hello Message encapsulated in an iWARP SendSE Message.

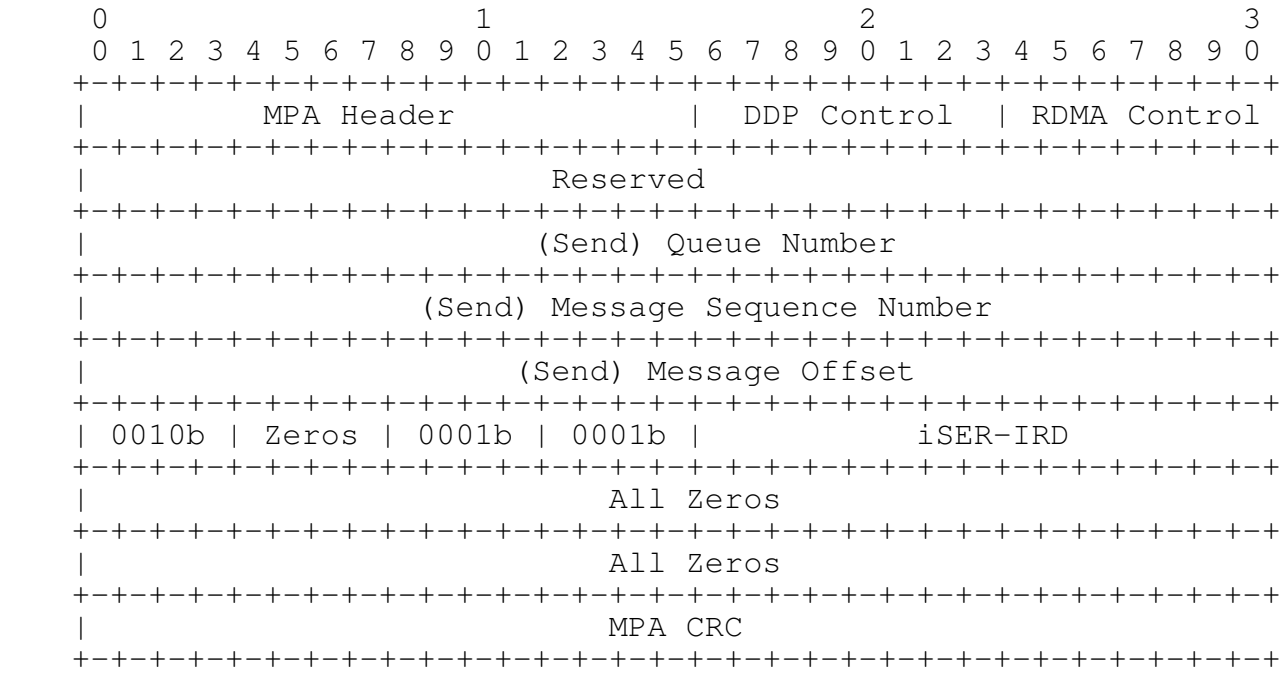

Figure 6 SendSE Message containing an iSER Hello Message

16.1.2 iWARP Message Format for iSER HelloReply Message

The following figure depicts an iSER HelloReply Message encapsulated in an iWARP SendSE Message. The Reject (REJ) flag is set to 0.

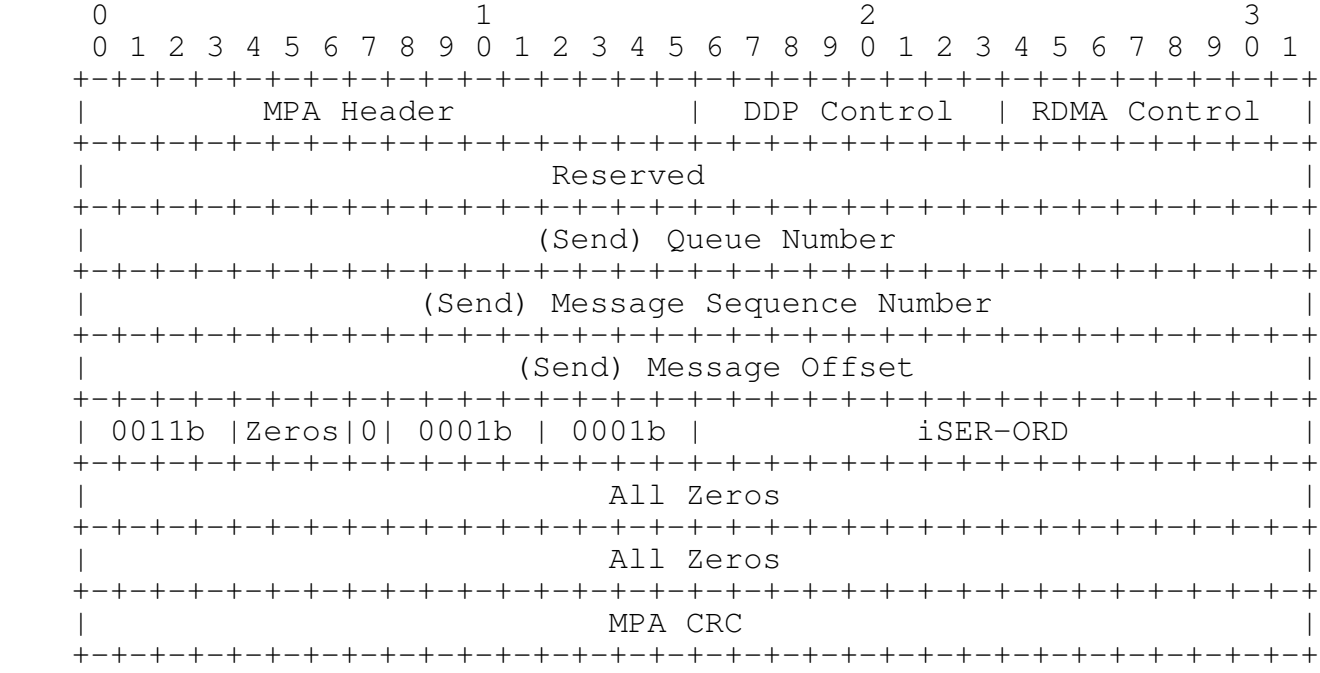

Figure 7 SendSE Message containing an iSER HelloReply Message

16.1.3 iWARP Message Format for SCSI Read Command PDU

The following figure depicts a SCSI Read Command PDU embedded in an iSER Message encapsulated in an iWARP SendSE Message. For this particular example, in the iSER header, the Write STag Valid flag is set to zero, the Read STag Valid flag is set to one, the Write STag field is set to all zeros, and the Read STag field contains a valid Read STag.

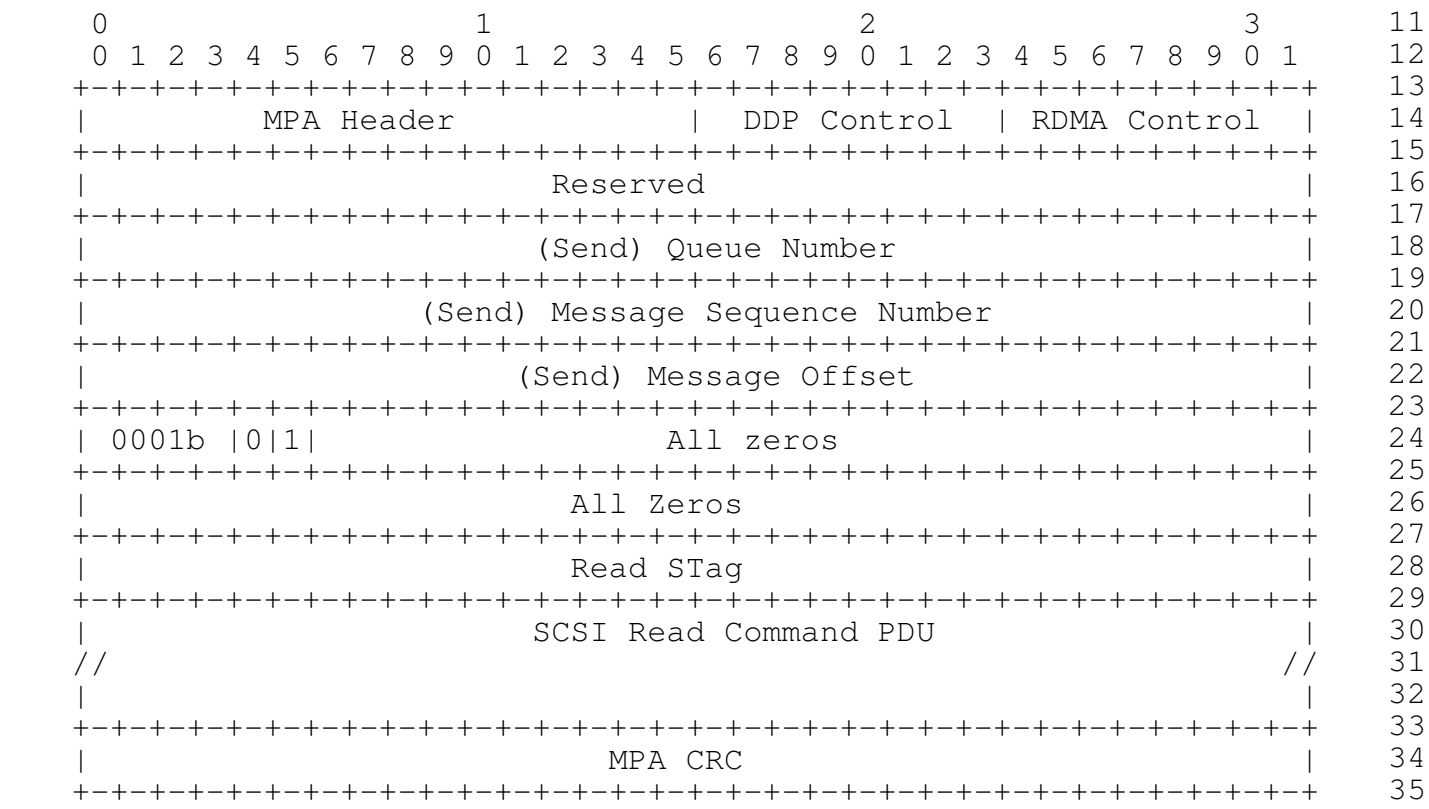

Figure 8 SendSE Message containing a SCSI Read Command PDU

16.1.4 iWARP Message Format for SCSI Read Data

The following figure depicts an iWARP RDMA Write Message carrying SCSI Read data in the payload:

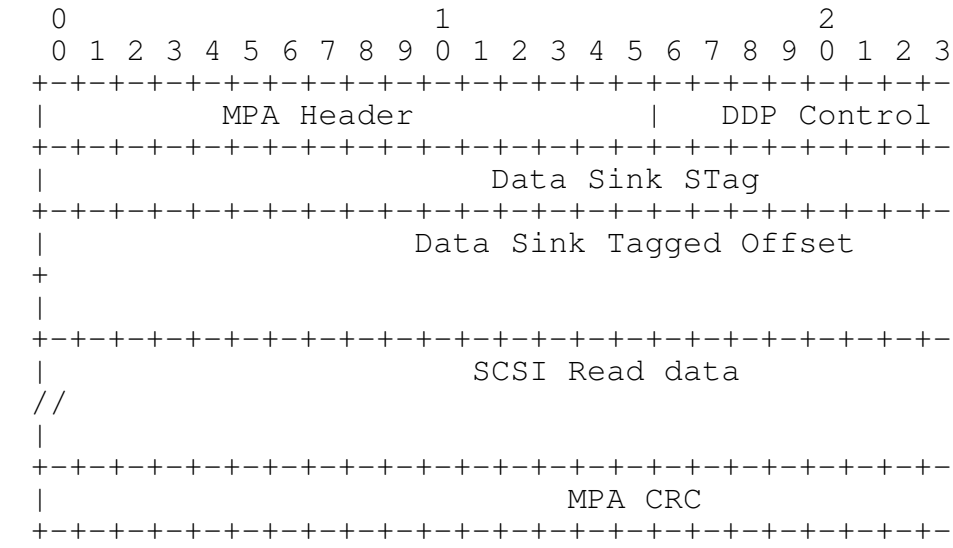

Figure 9 RDMA Write Message containing SCSI Read Data

16.1.5 iWARP Message Format for SCSI Write Command PDU

The following figure depicts a SCSI Write Command PDU embedded in an iSER Message encapsulated in an iWARP SendSE Message. For this particular example, in the iSER header, the Write STag Valid flag is set to one, the Read STag Valid flag is set to zero, the Write STag field contains a valid Write STag, and the Read STag field is set to all zeros since it is not used.

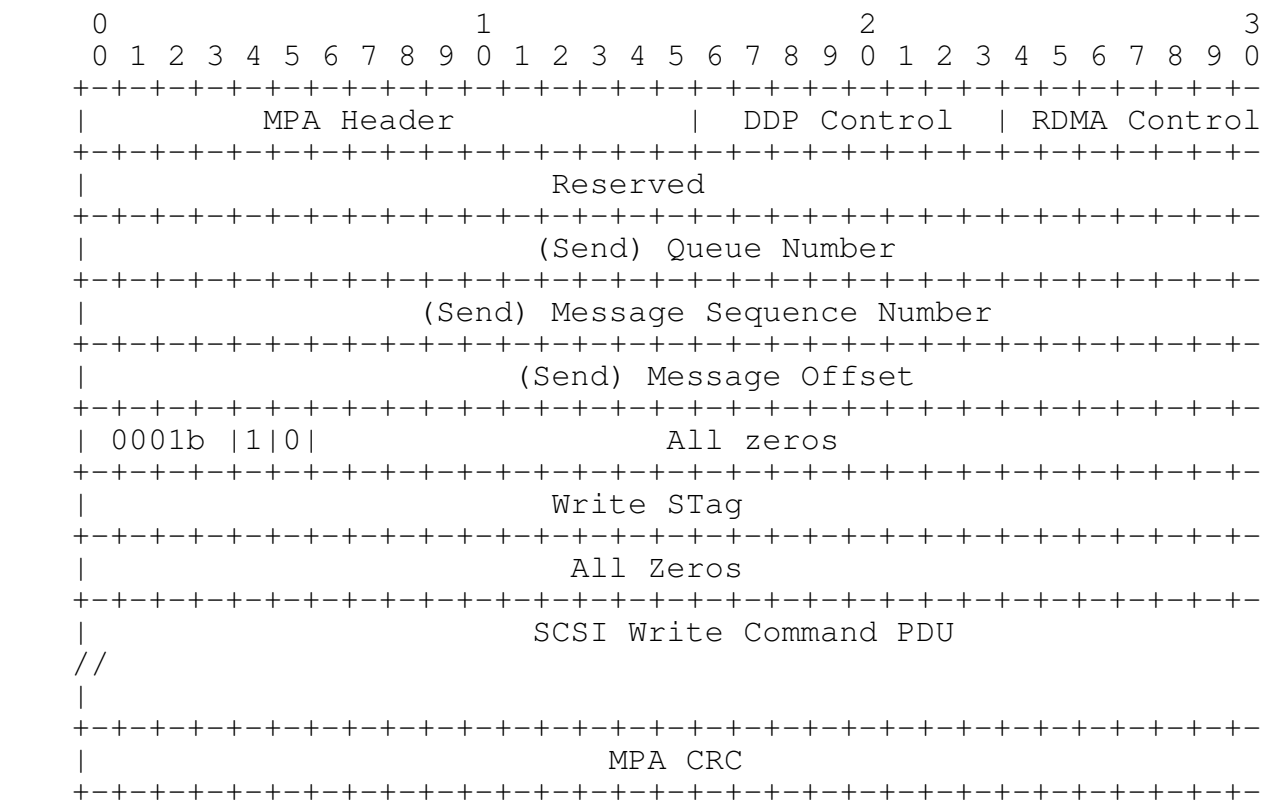

Figure 10 SendSE Message containing a SCSI Write Command PDU

16.1.6 iWARP Message Format for RDMA Read Request

An iSCSI R2T is transformed into an iWARP RDMA Read Request Message. The following figure depicts an iWARP RDMA Read Request Message:

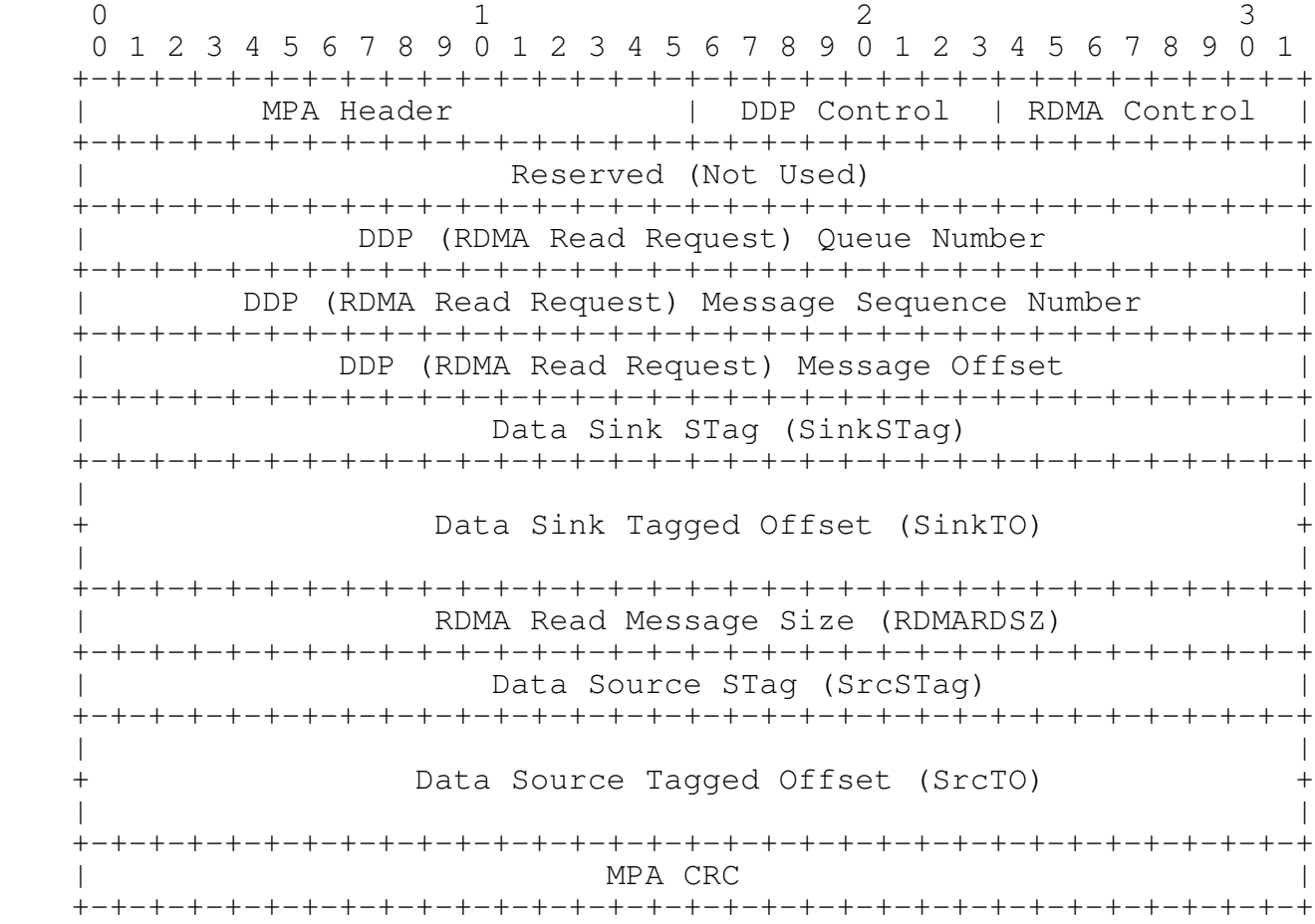

Figure 11 RDMA Read Request Message

16.1.7 iWARP Message Format for Solicited SCSI Write Data

The following figure depicts an iWARP RDMA Read Response Message carrying the solicited SCSI Write data in the payload:

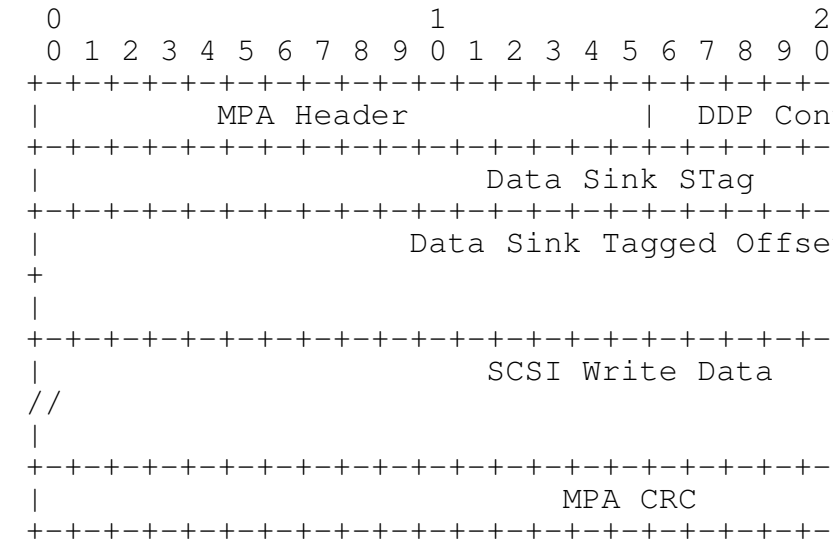

Figure 12 RDMA Read Response Message containing SCSI Write Data

16.1.8 iWARP Message Format for SCSI Response PDU

The following figure depicts a SCSI Response PDU embedded in an iSER Message encapsulated in an iWARP SendInvSE Message:

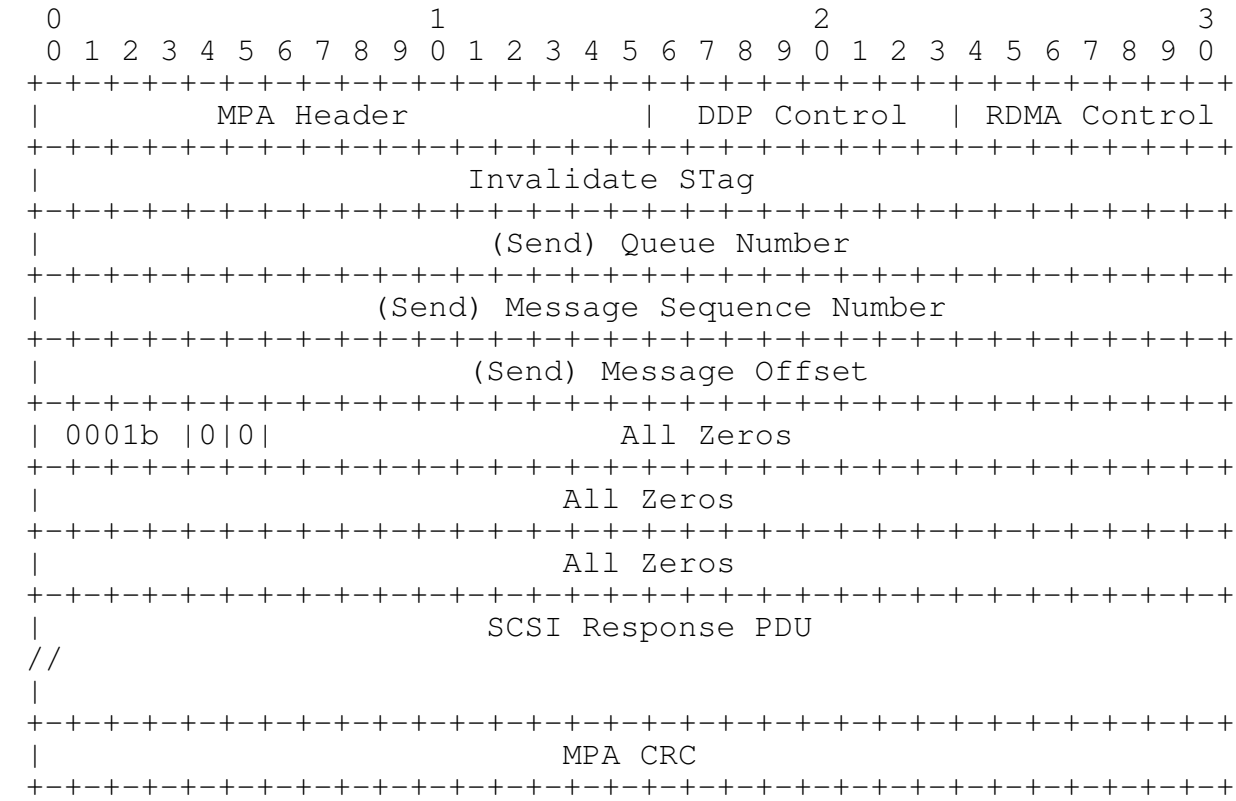

Figure 13 SendInvSE Message containing SCSI Response PDU

17 Author's Address

Mallikarjun Chadalapaka
Hewlett-Packard Company
8000 Foothills Blvd.
Roseville, CA 95747-5668, USA
Phone: +1-916-785-5621
Email: cbm@rose.hp.com

Uri Elzur
Broadcom Corporation
16215 Alton Parkway
Irvine, California 92619-7013, USA
Phone: +1-949-926-6432
Email: Uri@Broadcom.com

John Hufferd
IBM Corp.
5600 Cottle Rd.
San Jose, CA 95120, USA
Phone: +1-408-256-0403
Email: hufferd@us.ibm.com

Mike Ko
IBM Corp.
650 Harry Rd.
San Jose, CA 95120, USA
Phone: +1-408-927-2085
Email: mako@us.ibm.com

Hemal Shah
Intel Corporation
MS AN1-PTL1
1501 South Mopac Expressway, #400
Austin, Texas 78746, USA
Phone: +1-512-732-3963
Email: hemal.shah@intel.com

Patricia Thaler
Agilent Technologies, Inc.
1101 Creekside Ridge Drive, #100
M/S-RG10
Roseville, CA 95678, USA
Phone: +1-916-788-5662
email: pat_thaler@agilent.com

18 Acknowledgments

Dwight Barron
Hewlett-Packard Company
20555 SH.249
Houston, TX 77070-2698, USA
Phone: +1-281-514-2769
Email: Dwight.Barron@Hp.com

John Carrier
Adaptec, Inc.
691 S. Milpitas Blvd.
Milpitas, CA 95035, USA
Phone: +1-360-378-8526
Email: john_carrier@adaptec.com

Ted Compton
EMC Corporation
Research Triangle Park, NC 27709, USA
Phone: +1-919-248-6075
Email: compton_ted@emc.com

Paul R. Culley
Hewlett-Packard Company
20555 SH 249
Houston, Tx. 77070-2698, USA
Phone: +1-281-514-5543
Email: paul.culley@hp.com

Jeff Hilland
Hewlett-Packard Company
20555 SH 249
Houston, Tx. 77070-2698, USA
Phone: +1-281-514-9489
Email: jeff.hilland@hp.com

Mike Krause
Hewlett-Packard Company
43LN
19410 Homestead Road
Cupertino, CA 95014, USA
Phone: +1-408-447-3191
Email: krause@cup.hp.com

Jim Pinkerton
Microsoft, Inc.
One Microsoft Way
Redmond, WA, 98052, USA
Email: jpink@windows.microsoft.com

Renato J. Recio
IBM Corp.
11501 Burnett Road
Austin, TX 78758, USA
Phone: +1-512-838-3685
Email: recio@us.ibm.com

Julian Satran
IBM Corp.
Haifa Research Lab
Haifa University Campus - Mount Carmel
Haifa 31905, Israel
Phone: +972-4-829-6264
Email: Julian_Satran@il.ibm.com

Tom Talpey
Network Appliance
375 Totten Pond Road
Waltham, MA 02451, USA
Phone: +1-781-768-5329
EMail: thomas.talpey@netapp.com

Jim Wendt
Hewlett-Packard Company
8000 Foothills Boulevard MS 5668
Roseville, CA 95747-5668, USA
Phone: +1-916-785-5198
Email: jim_wendt@hp.com

19 Full Copyright Statement