Internet Engineering Task Force (IETF)
Request for Comments: 7145
Obsoletes: 5046
Category: Standards Track
ISSN: 2070-1721

M. Ko

A. Nezhinsky
Mellanox

April 2014

Internet Small Computer System Interface (iSCSI) Extensions for the Remote Direct Memory Access (RDMA) Specification

Abstract

Internet Small Computer System Interface (iSCSI) Extensions for Remote Direct Memory Access (RDMA) provides the RDMA data transfer capability to iSCSI by layering iSCSI on top of an RDMA-Capable Protocol. An RDMA-Capable Protocol 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 provided by an RDMA-Capable Protocol.

This document obsoletes RFC 5046.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7145.

Table of Contents

Table of Figures

1. Introduction

1.1. Motivation

The iSCSI protocol ([iSCSI]) is a mapping of the SCSI Architecture Model (see [SAM5] and [iSCSI-SAM]) 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 (Upper Level Protocol) message boundaries to aid placement of out-of-order segments. This TCP reassembly at high network speeds is quite counterproductive 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.

The generic term RDMA-Capable Protocol (RCaP) is used to refer to protocol stacks that provide the Remote Direct Memory Access (RDMA) functionality, such as iWARP and InfiniBand.

With the availability of RDMA-Capable Controllers within a host system, it is appropriate for iSCSI to be able to exploit the direct data placement function of the RDMA-Capable Controller 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 RDMA-Capable Controller. In summary, the iSCSI/iSER protocol stack is designed to enable scaling to high speeds by relying on a generic data placement process and RDMA technologies and products that 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 as defined in [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.

This document obsoletes RFC 5046. See Appendix A for the list of changes from RFC 5046.

1.2. iSCSI/iSER Layering

iSCSI Extensions for RDMA (iSER) is layered between the iSCSI layer and the RCaP layer.

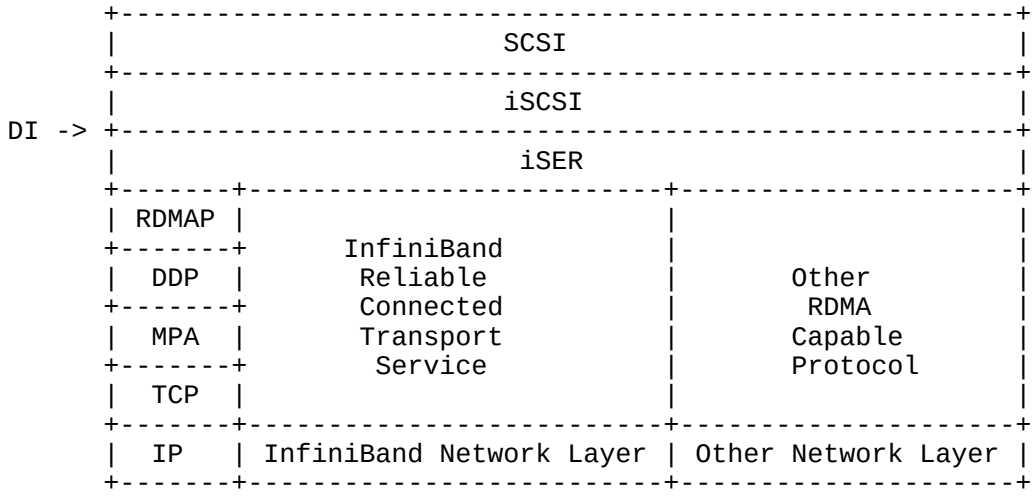

Figure 1: Example of iSCSI/iSER Layering in Full Feature Phase

Figure 1 shows an example of the relationship between SCSI, iSCSI, iSER, and the different RCaP layers. For TCP, the RCaP is iWARP. For InfiniBand, the RCaP is the Reliable Connected Transport Service. Note that the iSCSI layer as described here supports the RDMA Extensions as used in iSER.

1.3. Architectural Goals

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

1. Provide an RDMA 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. Do not require any major changes to the SCSI Architecture Model [SAM5] and SCSI command set standards.

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

4. Enable a session to operate in the Traditional iSCSI data transfer mode if iSER is not supported by either the initiator or the target. (Do 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 to utilize generic RDMA-Capable Controllers such as RNICs or to implement iSCSI and iSER in software. (Do not require iSCSI- or iSER-specific assists in the RCaP implementation or RDMA-Capable Controller.)

6. Implement a lightweight Datamover protocol for iSCSI with minimal state maintenance.

1.4. Protocol Overview

Consistent with the architectural goals stated in Section 1.3, the iSER protocol does not require changes in the iSCSI ecosystem or any related SCSI specifications. The iSER protocol defines the mapping of iSCSI PDUs to RCaP Messages in such a way that it is entirely feasible to realize iSCSI/iSER implementations that are based on generic RDMA-Capable Controllers. The iSER protocol layer requires minimal state maintenance to assist a connection during the iSCSI Full Feature Phase, besides being oblivious to the notion of an iSCSI session. The crucial protocol aspects of iSER may be summarized as follows:

1. iSER-assisted mode is negotiated during the iSCSI login in the leading connection for each session, and an entire iSCSI session can only operate in one mode (i.e., a connection in a session cannot operate in iSER-assisted mode if a different connection of the same session is already in Full Feature Phase in the Traditional iSCSI mode).

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

3. A Send Message is used for carrying an iSCSI control-type PDU preceded by an iSER header. See Section 7.2 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 (i.e., SCSI Data-In PDUs and R2T PDUs). iSER does not use SCSI Data-Out PDUs for solicited data, and SCSI Data-Out PDUs for unsolicited data are not treated as iSCSI data-type PDUs by iSER because RDMA is not used. See Section 7.1 for more details on iSCSI data-type PDUs.

5. The 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. RCaP is responsible for ensuring data integrity. (For example, iWARP includes a CRC-enhanced framing layer called MPA on top of TCP; and for InfiniBand, the CRCs are included in the Reliable Connection mode). For this reason, iSCSI header and data digests are negotiated to "None" for iSCSI/iSER sessions.

7. The iSCSI error recovery hierarchy defined in [iSCSI] is fully supported by iSER. (However, see Section 7.3.11 on the handling of SNACK Request PDUs.)

8. iSER requires no changes to iSCSI 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 3).

1.5. RDMA Services and iSER

iSER is designed to work with software and/or hardware protocol stacks providing the protocol services defined in RCaP documents such as [RDMAP], [IB], etc. The following subsections describe the key protocol elements of RCaP services on which iSER relies.

1.5.1. STag

An STag is the identifier of an I/O Buffer unique to an RDMA-Capable Controller 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 and the Base Offset in the header of an iSER Message containing the SCSI Command PDU to the target. The buffer length is as specified in the SCSI Command PDU.

The iSER layer at the initiator Advertises the STag and the Base Offset for the I/O Buffer of each SCSI I/O to the iSER layer at the target in the iSER header of a Send Message containing the SCSI Command PDU, unless the I/O can be completely satisfied by unsolicited data alone. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

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 1.5.4) to the RCaP layer on the initiator node -- i.e., this is completely transparent to the iSER layer at the initiator.

The iSER layer at the initiator SHOULD invalidate the Advertised STag upon a normal completion of the associated task. The Send with Invalidate Message, if supported by the RCaP layer (e.g., iWARP), can be used for automatic invalidation when it is used to carry 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 SHOULD explicitly invalidate the STag in these two cases. That iSER layer MUST check that STag invalidation has occurred whenever receipt of a Send with Invalidate message is the expected means of causing an STag to be invalidated, and it MUST perform the STag invalidation if the STag has not already been invalidated (e.g., because a Send Message was used instead of Send with Invalidate).

If the Advertised STag is not invalidated as recommended in the foregoing paragraph (e.g., in order to cache the STag for future reuse), the I/O Buffer remains exposed to the network for access by the RCaP. Such an I/O Buffer is capable of being read or written by the RCaP outside the scope of the iSCSI operation for which it was originally established; this fact has both robustness and security considerations. The robustness considerations are that the system containing the iSER initiator may react poorly to an unexpected modification of its memory. For the security considerations, see Section 11.

1.5.2. Send

Send is the RDMA Operation that is not addressed to an Advertised buffer and uses Untagged buffers as the message is received.

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.

For interoperability, iSER implementations SHOULD accept and correctly process SendSE and SendInvSE messages. However, SendSE and SendInvSE messages are to be regarded as optimizations or enhancements to the basic Send Message, and their support may vary by RCaP protocol and specific implementation. In general, these messages SHOULD NOT be used, unless the RCaP requires support for them in all implementations. If these messages are used, the implementation SHOULD be capable of reverting to use of Send in order to work with a receiver that does not support these messages. Attempted use of these messages with a peer that does not support them may result in a fatal error that closes the RCaP connection. For example, these messages SHOULD NOT be used with the InfiniBand RCaP because InfiniBand does not require support for them in all cases. New iSER implementations SHOULD use Send (and not SendSE or SendInvSE) unless there are compelling reasons for doing otherwise. Similarly, iSER implementations SHOULD NOT rely on events triggered by SendSE and SendInvSE, as these messages may not be used.

1.5.3. RDMA Write

RDMA Write is the RDMA Operation that is used to place data into an Advertised buffer at the Data Sink. The Data Source 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 the whole data or part of the data required to complete a SCSI Read command.

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

1.5.4. RDMA Read

RDMA Read is the RDMA Operation that is used to retrieve data from an Advertised buffer at the Data Source. The sender 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 Command.

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

1.6. 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 and the Base Offset as part of the iSER header for the PDU. The iSER Message is transferred to the target using a Send Message. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

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 Send Message to transfer the SCSI Response PDU back to the iSER layer at the initiator. The iSER layer at the initiator invalidates the STag and notifies the iSCSI layer of

the availability of the SCSI Response PDU. The Send with Invalidate Message, if supported by the RCaP layer (e.g., iWARP), can be used for automatic invalidation of the STag.

1.7. 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 and the Base Offset as part of the iSER header for the PDU. The iSER Message is transferred to the target using a Send Message. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

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

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 Send Message to transfer the SCSI Response PDU back to the iSER layer at the initiator. The iSER layer at the initiator invalidates the STag and notifies the iSCSI layer of the availability of the SCSI Response PDU. The Send with Invalidate Message, if supported by the RCaP layer (e.g., iWARP), can be used for automatic invalidation of the STag.

2. Definitions and Acronyms

2.1. Definitions

Advertisement (Advertised, Advertise, Advertisements, Advertises) -- The act of informing a remote iSER (iSCSI Extensions for RDMA) layer 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 the remote iSER layer of the Tagged Buffer identifiers (STag, Base Offset, and buffer length). Note that this Advertisement of Tagged Buffer information is the responsibility of the iSER layer on either end and is not defined by the RDMA-Capable Protocol. A typical method would be for the iSER layer to embed the Tagged Buffer’s STag, Base Offset, and buffer length in a message destined for the remote iSER layer.

Base Offset - A value when added to the Buffer Offset forms the Tagged Offset.

Completion (Completed, Complete, Completes) - Completion is defined as the process by which the RDMA-Capable Protocol layer informs the iSER layer that a particular RDMA Operation has performed all functions specified for the RDMA Operation.

Connection - A connection is a logical bidirectional communication channel between the initiator and the target, e.g., a TCP connection. Communication between the initiator and the target occurs over one or more connections. The 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 layer and the underlying iSER layer. Every invocation of an Operational Primitive is 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 RCaP (RDMA-Capable Protocol) 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 RCaP 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.

Inbound RDMA Read Queue Depth (IRD) - The maximum number of incoming outstanding RDMA Read Requests that the RDMA-Capable Controller can handle on a particular RCaP Stream at the Data Source. For some RDMA-Capable Protocol layers, the term "IRD" may be known by a different name. For example, for InfiniBand, the equivalent to IRD is the Responder Resources.

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 as defined in [iSCSI] is a mapping of the SCSI Architecture Model of SAM-5 over TCP.

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 via RDMA operations at the iSER layer, 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. An iSCSI connection that is not iSER assisted always maps onto a TCP connection at the transport level. But an iSER-assisted iSCSI connection may not have an underlying TCP connection. For some RCaP implementations (e.g., iWARP), an iSER-assisted iSCSI connection has an underlying TCP connection. For other RCaP implementations (e.g., InfiniBand), there is no underlying TCP connection. (In the specific example of InfiniBand [IB], an iSER-assisted iSCSI connection is directly mapped onto the InfiniBand Reliable Connection-based (RC) channel.)

iSCSI/iSER Session - An iSER-assisted iSCSI session. All connections of an iSCSI/iSER session are iSCSI/iSER connections.

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 grants on a particular RCaP Stream.

iSER-ORD - This variable represents the maximum number of outstanding RDMA Read Requests that the iSER layer can initiate on a particular RCaP 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 Mapping - A task state record maintained by the iSER layer that associates the Initiator Task Tag to the Local STag(s). The specifics of the record structure are implementation dependent.

Local Peer - The implementation of the RDMA-Capable Protocol 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 RDMA-Capable Controllers 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 RDMA-Capable Controller can initiate on a particular RCaP Stream at the Data Sink. For some RDMA-Capable Protocol layer, the term "ORD" may be known by a different name. For example, for InfiniBand, the equivalent to ORD is the Initiator Depth.

Phase Collapse - Refers to the optimization in iSCSI where the SCSI status is transferred along with the final SCSI Data-In PDU from a target. See Section 4.2 in [iSCSI].

RCaP Message - One or more packets of the network layer that constitute a single RDMA operation or a part of an RDMA Read Operation of the RDMA-Capable Protocol. For iWARP, an RCaP Message is known as an RDMAP Message.

RCaP Stream - A single bidirectional association between the peer RDMA-Capable Protocol layers on two nodes over a single transport-level stream. For iWARP, an RCaP Stream is known as an RDMAP Stream, and the association is created following a successful Login Phase during which iSER support is negotiated.

RDMA-Capable Protocol (RCaP) - The protocol or protocol suite that provides a reliable RDMA transport functionality, e.g., iWARP, InfiniBand, etc.

RDMA-Capable Controller - A network I/O adapter or embedded controller with RDMA functionality. For example, for iWARP, this could be an RNIC, and for InfiniBand, this could be a HCA (Host Channel Adapter) or TCA (Target Channel Adapter).

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

RDMA Operation - A sequence of RCaP 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, and Send 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 RCaP 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 RCaP 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.

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 RDMA-Capable Controller in the remote system allows the access to take place without interrupting the processing of the CPU(s) on the system.

Remote Mapping - A task state record maintained by the iSER layer that associates the Initiator Task Tag to the Advertised STag(s) and the Base Offset(s). The specifics of the record structure are implementation dependent.

Remote Peer - The implementation of the RDMA-Capable Protocol 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 [SAM5] parameters to/from the iSCSI layer.

Send - An RDMA Operation that transfers the content of a buffer from the Local Peer to an untagged buffer at the Remote Peer.

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 connections that link an initiator SCSI port with a target SCSI port form an iSCSI session (equivalent to a SCSI Initiator-Target (I-T) nexus). 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.

Steering Tag (STag) - An identifier of a Tagged Buffer on a node (Local or Remote) as defined in [RDMAP] and [DDP]. For other RDMA-Capable Protocols, the Steering Tag may be known by different names but will be referred to herein as STags. For example, for InfiniBand, a Remote STag is known as an R-Key, and a Local STag is known as an L-Key, and both will be considered STags.

Tagged Buffer - A buffer that is explicitly Advertised to the iSER layer at the remote node through the exchange of an STag, Base Offset, and length.

Tagged Offset - The offset within a Tagged Buffer.

Traditional iSCSI - Refers to the iSCSI protocol as defined in [iSCSI] (i.e., without the iSER enhancements).

Untagged Buffer - A buffer that is not explicitly Advertised to the iSER layer at the remode node.

2.2. Acronyms

Acronym	Definition
AHS	Additional Header Segment
BHS	Basic Header Segment
CO	Connection Only
CRC	Cyclic Redundancy Check
DDP	Direct Data Placement Protocol
DI	Datamover Interface
HCA	Host Channel Adapter
IANA	Internet Assigned Numbers Authority

IB	InfiniBand
IETF	Internet Engineering Task Force
I/O	Input - Output
IO	Initialize Only
IP	Internet Protocol
IPoIB	IP over InfiniBand
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)
PDU	Protocol Data Unit
R2T	Ready To Transfer
R2TSN	Ready To Transfer Sequence Number
RCaP	RDMA-Capable Protocol
RDMA	Remote Direct Memory Access
RDMAP	Remote Direct Memory Access Protocol
RFC	Request For Comments
RNIC	RDMA-enabled Network Interface Controller
SAM5	SCSI Architecture Model - 5
SCSI	Small Computer System Interface

SNACK	Selective Negative Acknowledgment - also Sequence Number Acknowledgement for data
STag	Steering Tag
SW	Session Wide
TCA	Target Channel Adapter
TCP	Transmission Control Protocol
TMF	Task Management Function
TTT	Target Transfer Tag
ULP	Upper Level Protocol

2.3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. 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]. [DA] also provides a functional overview of the interactions between the iSCSI layer and the Datamover layer as intended by the Datamover Architecture.

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. Whenever an Operational Primitive in invoked, the Connection_Handle qualifier is used to identify a particular iSCSI connection. For some Operational Primitives, a Data_Descriptor is used to identify the iSCSI/SCSI data buffer associated with the requested or completed operation.

The abstract model and the Operational Primitives defined in this section facilitate the description of the 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 the 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.

Each Operational Primitive is invoked with a set of qualifiers which specify the information context for performing the specific action being requested of the Operational Primitive. While the qualifiers are required, the method of realizing the qualifiers (e.g., by passing synchronously with invocation, or by retrieving from task context, or by retrieving from shared memory, etc.) is implementation dependent.

3.1. Operational Primitives offered by iSER

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

3.1.1. Send_Control

Input qualifiers: Connection_Handle, BHS and AHS (if any) of the iSCSI PDU, PDU-specific qualifiers

Return results: Not specified

This is used by the iSCSI layers at the initiator and the target to request the outbound transfer of an iSCSI control-type PDU (see Section 7.2). Qualifiers that only apply for a particular control-type PDU are known as PDU-specific qualifiers, e.g., ImmediateDataSize for a SCSI Write command. For details on PDU-specific qualifiers, see Section 7.3. The iSCSI layer can only invoke the Send_Control Operational Primitive when the connection is in iSER-assisted mode.

3.1.2. Put_Data

Input qualifiers: Connection_Handle, content of a SCSI Data-In PDU header, Data_Descriptor, Notify_Enable

Return results: Not specified

This is used by the iSCSI layer at the target to request the outbound transfer of data for a SCSI Data-In PDU from the buffer identified by the Data_Descriptor qualifier. The iSCSI layer can only invoke the Put_Data Operational Primitive when the connection is in iSER-assisted mode.

The Notify_Enable qualifier is used to indicate to the iSER layer whether or not it should generate an eventual local completion notification to the iSCSI layer. See Section 3.2.2 on Data_Completion_Notify for details.

3.1.3. Get_Data

Input qualifiers: Connection_Handle, content of an R2T PDU, Data_Descriptor, Notify_Enable

Return results: Not specified

This is used by the iSCSI layer at the target to request the inbound transfer of solicited data requested by an R2T PDU into the buffer identified by the Data_Descriptor qualifier. The iSCSI layer can only invoke the Get_Data Operational Primitive when the connection is in iSER-assisted mode.

The Notify_Enable qualifier is used to indicate to the iSER layer whether or not it should generate the eventual local completion notification to the iSCSI layer. See Section 3.2.2 on Data_Completion_Notify for details.

3.1.4. Allocate_Connection_Resources

Input qualifiers: Connection_Handle, Resource_Descriptor (optional)

Return results: Status

This is used by the iSCSI layers at the initiator and the target to request the allocation of all connection resources necessary to support RCaP for an operational iSCSI/iSER connection. The iSCSI layer may optionally specify the implementation-specific resource requirements for the iSCSI connection using the Resource_Descriptor qualifier.

A return result of Status=success means the invocation succeeded, and a return result of Status=failure means that the invocation failed. If the invocation is for a Connection_Handle for which an earlier invocation succeeded, the request will be ignored by the iSER layer and the result of Status=success will be returned. Only one Allocate_Connection_Resources Operational Primitive invocation can be outstanding for a given Connection_Handle at any time.

3.1.5. Deallocate_Connection_Resources

Input qualifiers: Connection_Handle

Return results: Not specified

This is used by the iSCSI layers at the initiator and the target to request the deallocation of all connection resources that were allocated earlier as a result of a successful invocation of the Allocate_Connection_Resources Operational Primitive.

3.1.6. Enable_Datamover

Input qualifiers: Connection_Handle, Transport_Connection_Descriptor, Final Login_Response_PDU (optional)

Return results: Not specified

This is used by the iSCSI layers at the initiator and the target to request that iSER-assisted mode be used for the connection. The Transport_Connection_Descriptor qualifier is used to identify the specific connection associated with the Connection_Handle. The iSCSI layer can only invoke the Enable_Datamover Operational Primitive when there was a corresponding prior resource allocation.

The Final_Login_Response_PDU input qualifier is applicable only for a target and contains the final Login Response PDU that concludes the iSCSI Login Phase.

3.1.7. Connection_Terminate

Input qualifiers: Connection_Handle

Return results: Not specified

This is used by the iSCSI layers at the initiator and the target to request that a specified iSCSI/iSER connection be terminated and all associated connection and task resources be freed. When this Operational Primitive invocation returns to the iSCSI layer, the iSCSI layer may assume full ownership of all iSCSI-level resources, e.g., I/O Buffers, associated with the connection.

3.1.8. Notice_Key_Values

Input qualifiers: Connection_Handle, number of keys, list of Key-Value pairs

Return results: Not specified

This is used by the iSCSI layers at the initiator and the target to request the iSER layer to take note of the specified Key-Value pairs that were negotiated by the iSCSI peers for the connection.

3.1.9. Deallocate_Task_Resources

Input qualifiers: Connection_Handle, ITT

Return results: Not specified

This is used by the iSCSI layers at the initiator and the target to request the deallocation of all RCaP-specific resources allocated by the iSER layer for the task identified by the ITT qualifier. The iSER layer may require a certain number of RCaP-specific resources associated with the ITT for each new iSCSI task. In the normal course of execution, these task-level resources in the iSER layer are assumed to be transparently allocated on each task initiation and deallocated on the conclusion of each task as appropriate. In exception scenarios where the task does not conclude with a SCSI Response PDU, the iSER layer needs to be notified of the individual task terminations to aid its task-level resource management. This Operational Primitive is used for this purpose and is not needed when a SCSI Response PDU normally concludes a task. Note that RCaP-specific task resources are deallocated by the iSER layer when a SCSI Response PDU normally concludes a task, even if the SCSI status was not success.

3.2. Operational Primitives Used by iSER

The iSER layer MUST use the following Operational Primitives offered by the iSCSI protocol layer when the connection is in iSER-assisted mode.

3.2.1. Control_Notify

Input qualifiers: Connection_Handle, an iSCSI control-type PDU

Return results: Not specified

This is used by the iSER layers at the initiator and the target to notify the iSCSI layer of the availability of an inbound iSCSI control-type PDU. 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.

3.2.2. Data_Completion_Notify

Input qualifiers: Connection_Handle, ITT, SN

Return results: Not specified

This is used by the iSER layer to notify the iSCSI layer of the completion of the outbound data transfer that was requested by the iSCSI layer only if the invocation of the Put_Data Operational Primitive (see Section 3.1.2) was qualified with Notify_Enable set. SN refers to the DataSN associated with the SCSI Data-In PDU.

This is used by the iSER layer to notify the iSCSI layer of the completion of the inbound data transfer that was requested by the iSCSI layer only if the invocation of the Get_Data Operational Primitive (see Section 3.1.3) was qualified with Notify_Enable set. SN refers to the R2TSN associated with the R2T PDU.

3.2.3. Data_ACK_Notify

Input qualifier: Connection_Handle, ITT, DataSN

Return results: Not specified

This is used by the iSER layer at the target 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 via an invocation of the Put_Data Operational Primitive where the A-bit in the SCSI Data-In PDU is set to one. See Section 7.3.5. DataSN refers to the expected DataSN of the next SCSI Data-In PDU that immediately follows the SCSI Data-In PDU with the A-bit set to which this notification corresponds, with semantics as defined in [iSCSI].

3.2.4. Connection_Terminate_Notify

Input qualifiers: Connection_Handle

Return results: Not specified

This is used by the iSER layers at the initiator and the target to notify the iSCSI layer of the unsolicited termination or failure of an iSCSI/iSER connection. The iSER layer MUST deallocate the connection and task resources associated with the terminated connection before the invocation of this Operational Primitive. Note that the Connection_Terminate_Notify Operational Primitive is not invoked when the termination of the connection was earlier requested by the local iSCSI layer.

3.3. iSCSI Protocol Usage Requirements

To operate in iSER-assisted mode, the iSCSI layers at both the initiator and the target MUST negotiate the RDMAExtensions key (see Section 6.3) to "Yes" on the leading connection. If the RDMAExtensions key is not negotiated to "Yes", then iSER-assisted mode MUST NOT be used. If the RDMAExtensons key is negotiated to "Yes", but the invocation of the Allocate_Connection_Resources Operational Primitive to the iSER layer fails, the iSCSI layer MUST fail the iSCSI Login process or terminate the connection as appropriate. See Section 10.1.3.1 for details.

If the RDMAExtensions key is negotiated to "Yes", the iSCSI layer MUST satisfy the following protocol usage requirements from the iSER protocol:

1. The iSCSI layer at the initiator MUST set ExpDataSN to zero 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.

2. 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.

3. The iSCSI layers at both the initiator and the target MUST support the keys as defined in Section 6 on Login/Text Operational Keys. If used as specified, these keys MUST NOT be answered with NotUnderstood, and the semantics as defined MUST be followed for each iSER-assisted connection.

4. The iSCSI layer at the initiator MUST NOT issue SNACKs for PDUs.

4. Lower-Layer Interface Requirements

4.1. Interactions with the RCaP Layer

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

* The RCaP layer supports all basic RDMA operations, including the RDMA Write Operation, RDMA Read Operation, and Send Operation.

* The RCaP layer provides reliable, in-order message delivery and direct data placement.

* When the iSER layer initiates an RDMA Read Operation following an RDMA Write Operation on one RCaP Stream, the RDMA Read Response Message processing on the remote node will be started only after the preceding RDMA Write Message payload is placed in the memory of the remote node.

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

* For an RCaP layer that supports the Send with Invalidate Message (e.g., iWARP), when the iSER layer provides the STag to be remotely invalidated to the RCaP layer for a Send with Invalidate Message, the RCaP layer uses this STag as the STag to be invalidated in the Send with Invalidate Message.

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

* When the RCaP layer delivers the content of an RDMA Send Message to the iSER layer, the RCaP 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 RCaP layer delivers the Send Message to the iSER layer, it notifies the iSER layer with the mechanism provided on that interface.

* For an RCaP layer that supports the Send with Invalidate Message (e.g., iWARP), when the RCaP layer delivers a Send with Invalidate Message to the iSER layer, it passes the value of the STag that was invalidated.

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

* For a transport layer that operates in byte stream mode such as TCP, the RCaP implementation supports the enabling of the RDMA mode after connection establishment and the exchange of Login parameters in byte stream mode. For a transport layer that provides message delivery capability such as [IB], the RCaP implementation supports the direct use of the messaging capability by the iSCSI layer for the Login Phase after connection establishment and before enabling iSER-assisted mode. (In the specific example of InfiniBand [IB], the iSCSI layer uses IB messages to transfer iSCSI PDUs for the Login Phase after connection establishment and before enabling iSER-assisted mode.)

* Whenever the iSER layer terminates the RCaP Stream, the RCaP layer terminates the associated connection.

4.2. Interactions with the Transport Layer

After the iSER connection is established, the RCaP layer and the underlying transport layer are responsible for maintaining the connection and reporting to the iSER layer any connection failures.

5. Connection Setup and Termination

5.1. iSCSI/iSER Connection Setup

During connection setup, the iSCSI layer at the initiator is responsible for establishing a connection with the target. After the connection is established, the iSCSI layers at the initiator and the target enter the Login Phase using the same rules as outlined in [iSCSI]. The connection transitions into the iSCSI Full Feature Phase in iSER-assisted mode following a successful login negotiation between the initiator and the target in which iSER-assisted mode is negotiated and the connection resources necessary to support RCaP have been allocated at both the initiator and the target. The same connection MUST be used for both the iSCSI Login Phase and the subsequent iSER-assisted Full Feature Phase.

For a transport layer that operates in byte stream mode such as TCP, the RCaP implementation supports the enabling of the RDMA mode after connection establishment and the exchange of Login parameters in byte stream mode. For a transport layer that provides message delivery capability such as [IB], the RCaP implementation supports the use of the messaging capability by the iSCSI layer directly for the Login Phase after connection establishment before enabling iSER-assisted mode.

iSER-assisted mode MUST NOT be enabled unless 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]. If the RDMAExtensions key is not negotiated to Yes, then for some RCaP implementation (such as [IB]), the existing connection may need to be torn down and a new connection may need to be established in TCP-capable mode. (For InfiniBand, this will require a connection like [IPoIB].)

iSER-assisted mode is defined for a Normal session only, and the RDMAExtensions key MUST NOT be negotiated for a Discovery session. Discovery sessions are always conducted using the transport layer as described in [iSCSI].

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 or not iSER-assisted mode is enabled.

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 keys 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, RCaP is responsible for providing error detection that is at least as good as a 32-bit CRC for all iSER 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 that apply to iSCSI PDUs would not be appropriate for RDMA Read and RDMA Write operations used with iSER.

5.1.1. 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 one and the NSG (Next Stage) field set to FullFeaturePhase, the iSCSI layer SHOULD request the iSER layer to allocate the connection resources necessary to support RCaP by invoking the Allocate_Connection_Resources Operational Primitive. The connection resources required are defined by the implementation and are outside the scope of this specification. 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 connection. The specific keys to be passed in as input qualifiers are implementation dependent. These may include, but are not limited to, MaxOutstandingR2T and ErrorRecoveryLevel.

Among the connection resources allocated at the initiator is the Inbound RDMA Read Queue Depth (IRD). As described in Section 9.5.1, R2Ts are transformed by the target into RDMA Read operations. IRD limits the maximum number of simultaneously incoming outstanding RDMA Read Requests per an RCaP 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 zero based on implementation configuration; setting IRD to zero 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 zero since the iSER layer at the initiator does not issue RDMA Read Requests to the target.

Failure to allocate the requested connection resources locally results in a login failure, and its handling is described in Section 10.1.3.1.

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

After the target returns the Login Response with the T bit set to one and the NSG field set to FullFeaturePhase, and a Status-Class of 0x00 (Success), the iSCSI layer MUST invoke the Enable_Datamover Operational Primitive with the following qualifiers. (See Section 10.1.4.6 for the case when the Status-Class is not Success.)

a. Connection_Handle that identifies the iSCSI connection.

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

The iSER layer MUST send the iSER Hello Message as the first iSER Message only if iSERHelloRequired is negotiated to "Yes". See Section 5.1.3 on iSER Hello Exchange.

If the iSCSI layer on the initiator side allocates the connection resources to support RCaP only after it receives the final Login Response PDU from the target, then it may not be able to handle the number of unexpected iSCSI control-type PDUs (as declared by the MaxOutstandingUnexpectedPDUs key from the initiator) that can be sent by the target before the buffer resources are allocated at the initiator side. In this case, the iSERHelloRequired key SHOULD be negotiated to "Yes" so that the initiator can allocate the connection resources before sending the iSER Hello Message. See Section 5.1.3 for more details.

5.1.2. 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 one and the NSG (Next Stage) field set to FullFeaturePhase, the iSCSI layer MUST request the iSER layer to allocate the resources necessary to support RCaP by invoking the Allocate_Connection_Resources Operational Primitive. The connection 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 connection. The specific keys to be passed in as input qualifiers are implementation dependent. These may include, but not limited to, MaxOutstandingR2T and ErrorRecoveryLevel.

Premature allocation of RCaP connection resources can expose an iSER target to a resource exhaustion attack on those resources via multiple iSER connections that progress only to the point at which the implementation allocates the RCaP connection resources. The countermeasure for this attack is initiator authentication; the iSCSI

layer MUST NOT request the iSER layer to allocate the connection resources necessary to support RCaP until the iSCSI layer is sufficiently far along in the iSCSI Login Phase that it is reasonably certain that the peer side is not an attacker. In particular, if the Login Phase includes a SecurityNegotiation stage, the iSCSI layer MUST defer the connection resource allocation (i.e., invoking the Allocate_Connection_Resources Operational Primitive) to the LoginOperationalNegotiation stage ([iSCSI]) so that the resource allocation occurs after the authentication phase is completed.

Among the connection resources allocated at the target is the Outbound RDMA Read Queue Depth (ORD). As described in Section 9.5.1, R2Ts are transformed by the target into RDMA Read operations. The ORD limits the maximum number of simultaneously outstanding RDMA Read Requests per RCaP 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 zero since the iSER layer at the target does not expect RDMA Read Requests to be issued by the initiator.

Failure to allocate the requested connection resources locally results in a login failure, and its handling is described in Section 10.1.3.1.

If the iSER layer at the target is successful in allocating the connection resources necessary to support RCaP, 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:

   a. Connection_Handle that identifies the iSCSI connection.

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

   c. The final transport-layer (e.g., TCP) message containing the Login Response with the T bit set to one and the NSG field set to FullFeaturePhase.

3. The iSER layer MUST send the final Login Response PDU in the native transport mode to conclude the iSCSI Login Phase. If the underlying transport is TCP, then the iSER layer MUST send the final Login Response PDU in byte stream mode.

4. 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 iSER Message if iSERHelloRequired is negotiated to "Yes". If the iSER layer receives an iSER Hello Message when iSERHelloRequired is negotiated to "No", then this MUST be treated as an iSER protocol error. See Section 5.1.3 on iSER Hello Exchange for more details.

Note: In the above sequence, the operations as described in items 3 and 4 MUST be performed atomically for iWARP connections. Failure to do this may result in race conditions.

5.1.3. iSER Hello Exchange

If iSERHelloRequired is negotiated to "Yes", the first iSER 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 9.3 on iSER Header Format for iSER Hello Message. Conversely, if iSERHelloRequired is negotiated to "No", then the iSER layer at the initiator MUST NOT send an 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 iSER Message sent by the target if iSERHelloRequired is negotiated to "Yes". The iSER HelloReply Message is used by the iSER layer at the target to declare iSER parameters to the initiator. See Section 9.4 on iSER Header Format for iSER HelloReply Message. If the iSER layer receives an iSER Hello Message when iSERHelloRequired is negotiated to "No", then this MUST be treated as an iSER protocol error. See Section 10.1.3.4 on iSER Protocol Errors on for more details.

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. In order to free up the unused resources, 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 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 are unacceptable to the target. This includes the following:

* 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.

See Section 10.1.3.2 on the handling of the error situation.

An initiator that conforms to [RFC5046] allocates connection resources before sending the Login Request with the T (Transit) bit set to one and the NSG (Next Stage) field set to FullFeaturePhase. (For brevity, this is referred to as "early" connection allocation.) The current iSER specification relaxes this requirement to allow an initiator to allocate connection resources after it receives the final Login Response PDU from the target. (For brevity, this is referred to as "late" connection allocation.) An initiator that employs "late" connection allocation may encounter problems (e.g., RCaP connection closure) with a target that sends unexpected iSCSI PDUs immediately upon transitioning to Full Feature Phase, as allowed by the negotiated value of the MaxOutstandingUnexpectedPDUs key. The only way to prevent this situation in full generality is to use iSER Hello Messages, as they enable the initiator to allocate its connection resources before sending its iSER Hello Message. The iSERHelloRequired key is used by the initiator to determine if it is dealing with a target that supports the iSER Hello exchanges. Fortunately, known iSER target implementations do not take full advantage of the number of allowed unexpected PDUs immediately upon transitioning into Full Feature Phase, thus enabling an initiator workaround that involves a smaller quantity of connection resources prior to Full Feature Phase, as explained further below.

In the following summary, where "late" connection allocation is practiced, an initiator that follows [RFC5046] is referred to as an "old" initiator; otherwise, it is referred to as a "new" initiator. Similarly, a target that does not support the iSERHelloRequired key

(and responds with "NotUnderstood" when negotiating the iSERHelloRequired key) is referred to as an "old" target; otherwise, it is referred to as a "new" target. Note that an "old" target can still support the iSER Hello exchanges, but this fact is not known by the initiator. A "new" target can also respond with "No" when negotiating the iSERHelloRequired key. In this case, its behavior with respect to "late" connection allocation is similar to an "old" target.

A "new" initiator will work fine with a "new" target.

For an "old" initiator and an "old" target, the failure by the initiator to handle the number of unexpected iSCSI control-type PDUs that are sent by the target before the buffer resources are allocated at the initiator can result in the failure of the iSER session caused by closure of the underlying RCaP connection. For the "old" target, there is a known implementation that sends one unexpected iSCSI control-type PDU after sending the final Login Response and then waits awhile before sending the next one. This tends to alleviate somewhat the buffer allocation problem at the initiator.

For a "new" initiator and an "old" target, the failure by the initiator to handle the number of unexpected iSCSI control-type PDUs that are sent by the target before the buffer resources are allocated at the initiator can result in the failure of the iSER session caused by closure of the underlying RCaP connection. A "new" initiator MAY choose to terminate the connection; otherwise, it SHOULD do one of the following:

1. Allocate the connection resources before sending the final Login Request PDU.

2. Allocate one or more buffers for receiving unexpected control-type PDUs from the target before sending the final Login Request PDU. This reduces the possibility of the unexpected control-type PDUs causing the RCaP connection to close before the connection resources have been allocated.

For an "old" initiator and a "new" target, if the iSERHelloRequired key is not negotiated, a "new" target MUST still respond with the iSER HelloReply Message when it receives the iSER Hello Message. If the iSERHelloRequired key is negotiated to "No" or "NotUnderstood", a "new" target MAY choose to terminate the connection; otherwise, it SHOULD delay sending any unexpected control-type PDUs until one of the following events has occurred:

1. A PDU is received from the initiator after it sends the final Login Response PDU.

2. A system-configurable timeout period (say, one second) has expired.

5.2. iSCSI/iSER Connection Termination

5.2.1. 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 Send Message to send the Logout Request PDU to the target. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP). After the iSER layer at the initiator receives the Send 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 5.2.2. After the RCaP layer at the initiator reports that the connection has been closed, the iSER layer at the initiator MUST deallocate all connection and task resources (if any) associated with the connection, and invalidate the Local Mappings (if any) before notifying the iSCSI layer by invoking the Connection_Terminate_Notify Operational Primitive.

5.2.2. Normal Connection Termination at the Target

Upon receiving the Send 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 Send Message to send the Logout Response PDU to the initiator. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP). After the iSCSI logout process is complete, the iSCSI layer

at the target MUST request the iSER layer at the target to terminate the RCaP Stream by invoking the Connection_Terminate Operational Primitive.

As part of the termination process, the RCaP layer MUST close the connection. When the RCaP layer notifies the iSER layer after the RCaP Stream and the associated connection are terminated, the iSER layer MUST deallocate all connection and task resources (if any) associated with the connection, and invalidate the Local and Remote Mappings (if any).

5.2.3. Termination without Logout Request/Response PDUs

5.2.3.1. Connection Termination Initiated by the iSCSI layer

The Connection_Terminate Operational Primitive MAY be invoked by the iSCSI layer to request the iSER layer to terminate the RCaP Stream without having previously exchanged the Logout Request and Logout Response PDUs between the two iSCSI/iSER nodes. As part of the termination process, the RCaP layer will close the connection. When the RCaP layer notifies the iSER layer after the RCaP Stream and the associated 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 all connection and task resources (if any) associated with the connection, and invalidate the Local and Remote Mappings (if any).

If the Connection_Terminate Operational Primitive is invoked by the iSCSI layer at the initiator, then the iSER layer at the initiator MUST deallocate all connection and task resources (if any) associated with the connection, and invalidate the Local Mappings (if any).

5.2.3.2. 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 notify the iSCSI layer that the iSCSI/iSER connection has been terminated by invoking the Connection_Terminate_Notify Operational Primitive.

Prior to invoking Connection_Terminate_Notify, the iSER layer at the target MUST deallocate all connection and task resources (if any) associated with the connection, and invalidate the Local and Remote Mappings (if any).

Prior to invoking Connection_Terminate_Notify, the iSER layer at the initiator MUST deallocate all connection and task resources (if any) associated with the connection, and invalidate the Local Mappings (if any).

If the remote iSCSI/iSER node initiated the closing of the connection (e.g., by sending a TCP FIN or TCP RST), the iSER layer MUST notify the iSCSI layer after the RCaP layer reports that the connection is closed by invoking the Connection_Terminate_Notify Operational Primitive.

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

6. 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 in [iSCSI] and not discussed in this section may be used on iSCSI/iSER connections with the same semantics.

6.1. HeaderDigest and DataDigest

Irrelevant when: RDMAExtensions=Yes

Negotiations resulting in RDMAExtensions=Yes for a session imply HeaderDigest=None and DataDigest=None for all connections in that session and override the settings, whether default or configured.

6.2. 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, and MUST be ignored if it is declared. Instead, InitiatorRecvDataSegmentLength (as described in Section 6.5) and TargetRecvDataSegmentLength (as described in Section 6.4) keys are negotiated. The values of the local and remote MaxRecvDataSegmentLength are derived from the InitiatorRecvDataSegmentLength and TargetRecvDataSegmentLength keys.

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.

Note that RFC 3720 requires that when a target receives a NOP-Out request with a valid Initiator Task Tag, it responds with a NOP-In with the same Initiator Task Tag that was provided in the NOP-Out request. Furthermore, it returns the first MaxRecvDataSegmentLength bytes of the initiator-provided Ping Data. Since there is no MaxRecvDataSegmentLength common to the initiator and the target in iSER, the length of the data sent with the NOP-Out request MUST NOT exceed InitiatorMaxRecvDataSegmentLength.

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

6.3. 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, an initiator MUST offer the key in the first Login Request PDU in the LoginOperationalNegotiation stage of the leading connection, and a target MUST offer the key in the first Login Response PDU with which it is allowed to do so (i.e., the first Login Response PDU issued

after the first Login Request PDU with the C bit set to zero) in the LoginOperationalNegotiation stage of the leading connection. In response to the offered key=value pair of RDMAExtensions=yes, an initiator MUST respond in the next Login Request PDU with which it is allowed to do so, and a target MUST respond in the next Login Response PDU with which it is allowed to do so.

Negotiating the RDMAExtensions key first enables a node to negotiate the optimal value for other keys. Certain iSCSI keys such as MaxBurstLength, MaxOutstandingR2T, ErrorRecoveryLevel, InitialR2T, ImmediateData, etc., may be negotiated differently depending on whether the connection is in Traditional iSCSI mode or iSER-assisted mode.

6.4. 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 in the Full Feature Phase. 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.

6.5. 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 in the Full Feature Phase.

6.6. OFMarker and IFMarker

Irrelevant when: RDMAExtensions=Yes

Negotiations resulting in RDMAExtensions=Yes for a session imply OFMarker=No and IFMarker=No for all connections in that session and override the settings, whether default or configured.

6.7. MaxOutstandingUnexpectedPDUs

Use: LO (leading only), Declarative

Senders: Initiator and Target

Scope: SW (session-wide)

Irrelevant when: RDMAExtensions=No

MaxOutstandingUnexpectedPDUs=
<numerical-value-from-2-to-(2**32-1) | 0>

Default is 0

This key is used by the initiator and the target to declare the maximum number of outstanding "unexpected" iSCSI control-type PDUs that it can receive in the Full Feature Phase. It is intended to allow the receiving side to determine the amount of buffer resources needed beyond the normal flow control mechanism available in iSCSI. An initiator or target should select a value such that it would not impose an unnecessary constraint on the iSCSI layer under normal circumstances. The value of 0 is defined to indicate that the declarer has no limit on the maximum number of outstanding "unexpected" iSCSI control-type PDUs that it can receive. See Sections 8.1.1 and 8.1.2 for the usage of this key. Note that iSER Hello and HelloReply Messages are not iSCSI control-type PDUs and are not affected by this key.

For interoperability with implementations based on [RFC5046], this key SHOULD be negotiated because the default value of 0 in [RFC5046] is problematic for most implementations as it does not impose a bound on resources consumable by unexpected PDUs.

6.8. MaxAHSLength

Use: LO (leading only), Declarative

Senders: Initiator and Target

Scope: SW (session-wide)

Irrelevant when: RDMAExtensions=No

MaxAHSLength=<numerical-value-from-2-to-(2**32-1) | 0>

Default is 256

This key is used by the initiator and target to declare the maximum size of AHS in an iSCSI control-type PDU that it can receive in the Full Feature Phase. It is intended to allow the receiving side to determine the amount of resources needed for receive buffering. An initiator or target should select a value such that it would not impose an unnecessary constraint on the iSCSI layer under normal circumstances. The value of 0 is defined to indicate that the declarer has no limit on the maximum size of AHS in iSCSI control-type PDUs that it can receive.

For interoperability with implementations based on [RFC5046], an initiator or target MAY terminate the connection if it anticipates MaxAHSLength to be greater than 256 and the key is not understood by its peer.

6.9. TaggedBufferForSolicitedDataOnly

Use: LO (leading only), Declarative

Senders: Initiator

Scope: SW (session-wide)

RDMAExtensions=<boolean-value>

Irrelevant when: RDMAExtensions=No

Default is No

This key is used by the initiator to declare to the target the usage of the Write Base Offset in the iSER header of an iSCSI control-type PDU. When set to No, the Base Offset is associated with an I/O buffer that contains all the write data, including both unsolicited and solicited data. When set to Yes, the Base Offset is associated with an I/O buffer that only contains solicited data.

6.10. iSERHelloRequired

Use: LO (leading only), Declarative

Senders: Initiator

Scope: SW (session-wide)

RDMAExtensions=<boolean-value>

Irrelevant when: RDMAExtensions=No

Default is No

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 to declare to the target whether the iSER Hello Exchange is required. When set to Yes, the iSER layers MUST perform the iSER Hello Exchange as described in Section 5.1.3. When set to No, the iSER layers MUST NOT perform the iSER Hello Exchange.

7. 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.

7.1. 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's transfer 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 RDMA-Capable Controller. 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 the ITT and SN. There is no data completion notification at the initiator since the RDMA operations are completely handled by the RDMA-Capable Controller 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.

7.2. 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 Send Messages of RCaP. Specifically, it is to be noted that SCSI Data-Out PDUs carrying unsolicited data are defined as iSCSI control-type PDUs. See Section 7.3.4 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.

7.3. 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, the 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.

7.3.1. SCSI Command

Type: control-type PDU

PDU-specific qualifiers (for SCSI Write or bidirectional command): ImmediateDataSize, UnsolicitedDataSize, DataDescriptorOut

PDU-specific qualifiers (for SCSI Read or bidirectional command): DataDescriptorIn

The iSER layer at the initiator MUST send the SCSI command in a Send Message to the target. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

For a SCSI Write or bidirectional command, the iSCSI layer at the initiator MUST invoke the Send_Control Operational Primitive as follows:

* If there is immediate data to be transferred for the SCSI write or bidirectional command, the qualifier ImmediateDataSize MUST be used to define the number of bytes of immediate unsolicited data to be sent with the write or bidirectional command, and the qualifier DataDescriptorOut MUST be used to define 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 MUST be used to define 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 7.3.4 on SCSI Data-Out.

* If there is solicited data to be transferred for the SCSI Write or bidirectional command, as indicated when the Expected Data Transfer Length in the SCSI Command PDU exceeds 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. 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. When TaggedBufferForSolicitedDataOnly is negotiated to No, the Base Offset is associated with this I/O Buffer. When TaggedBufferForSolicitedDataOnly is negotiated to Yes, the Base Offset is associated with an I/O Buffer that contains only solicited data.

  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 and the Base Offset to the target by sending them in the iSER header of the iSER Message (the payload of the Send Message of RCaP) containing the SCSI Write or bidirectional command PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP). See Section 9.2 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 and note the Base Offset for this 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 and the Base Offset to the target by sending them in the iSER header of the iSER Message (the payload of the Send Message of RCaP) containing the SCSI Read or bidirectional command PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP). See Section 9.2 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 generated PDUs (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 Base Offset(s) and the Advertised STag(s) 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 7.3.6. The Read STag is used in handling the SCSI Data-In PDU(s) from the iSCSI layer at the target as described in Section 7.3.5.

7.3.2. SCSI Response

Type: control-type PDU

PDU-specific qualifiers: DataDescriptorStatus

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 Send Message containing the SCSI Response PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

* 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 Send Message containing the SCSI Response PDU. The header of the Send Message MUST carry the Read STag to be invalidated at the initiator. The Send with Invalidate Message, if supported by the RCaP layer (e.g., iWARP), can be used for the automatic invalidation of the STag.

* 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 Send Message containing the SCSI Response PDU. The header of the Send Message MUST carry the Write STag to be invalidated at the initiator. The Send with Invalidate Message, if supported by the RCaP layer (e.g., iWARP), can be used for the automatic invalidation of the STag.

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 before transferring the SCSI Response PDU to the initiator.

Upon receiving a Send Message containing the SCSI Response PDU from the target, the iSER layer at the initiator MUST invalidate the STag(s) specified in the header. (If a Send with Invalidate Message is supported by the RCaP layer (e.g., iWARP) and is used to carry the SCSI Response PDU, the RCaP layer at the initiator will invalidate the STag. 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 Send with Invalidate Message because the header of the Send with Invalidate 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 notifying the iSCSI layer at the initiator by invoking the Control_Notify Operational Primitive qualified with the SCSI Response. This precludes the possibility of

using the STag(s) after the completion of the command; such use would cause data corruption.

When the iSER layer at the initiator receives a Send Message containing the SCSI Response PDU, it SHOULD invalidate the Local Mapping. The iSER layer MUST ensure that all local STag(s) associated with the ITT are invalidated before notifying the iSCSI layer of the SCSI Response PDU by invoking the Control_Notify Operational Primitive qualified with the SCSI Response PDU.

7.3.3. Task Management Function Request/Response

Type: control-type PDU

PDU-specific qualifiers (for TMF Request): DataDescriptorOut, DataDescriptorIn

The iSER layer MUST use a Send Message to send the Task Management Function Request/Response PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

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 STags (if any) in the Local Mappings.

* It MUST invalidate the existing STags (if any) and the Local Mappings.

* It MUST allocate a Read STag for the I/O Buffer and note the Base Offset associated with 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 and note the Base Offset associated with 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 and the Base Offsets, if allocated, to the target in the iSER header of the Send Message carrying the iSCSI PDU, as described in Section 9.2. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

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 zero 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 Local and Remote Mappings (if any).

* It MUST invalidate the local STags (if any) associated with the ITT.

* It MUST replace the Base Offset(s) and the Advertised STag(s) in the Remote Mapping with the Base Offset(s) and the Advertised STag(s) 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 7.3.6. 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 7.3.5.

7.3.4. SCSI Data-Out

Type: control-type PDU

PDU-specific qualifiers: DataDescriptorOut

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 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, where the DataSegmentLength has the value of TargetRecvDataSegmentLength in all generated PDUs 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, the iSER layer at the initiator MUST use a Send Message to send the SCSI Data-Out PDU. If the F bit is set to 1, the SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

Note that for solicited data, the SCSI Data-Out PDUs are not used since R2T PDUs are not delivered to the iSCSI layer at the initiator; instead, R2T PDUs are transformed by the iSER layer at the target into RDMA Read operations. (See Section 7.3.6.)

7.3.5. SCSI Data-In

Type: data-type PDU

PDU-specific qualifiers: DataDescriptorIn

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 7.1 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 9.5.2.

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 and the Base Offset in the Remote Mapping. 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 add the Buffer Offset from the SCSI Data-In PDU to the Base Offset from the Remote Mapping 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 the 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 RCaP 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 the 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 one 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 Request Message following the RDMA Write Message. When the iSER layer at the target receives a completion for the RDMA Read Request Message from the RCaP layer, implying that the RDMA-Capable Controller at the initiator has completed processing the RDMA Write Message due to the completion ordering semantics of RCaP, the iSER layer at the target MUST notify the iSCSI layer at the target by invoking the Data_ACK_Notify Operational Primitive qualified with ITT and DataSN (see Section 3.2.3).

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 notify the iSCSI layer at the target by invoking the Data_ACK_Notify Operational Primitive qualified with ITT and DataSN (see Section 3.2.3) when it receives the local completion from the RCaP layer for the RDMA Write Message. This is allowed since digest errors do not occur in iSER (see Section 10.1.4.2) 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 RCaP layer guarantees that the RCaP 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 zero. 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 are 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.

7.3.6. Ready To Transfer (R2T)

Type: data-type PDU

PDU-specific qualifiers: DataDescriptorOut

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 7.1 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. 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 and the Base Offset in the Remote Mapping. The Remote Mapping was established earlier by the iSER layer at the target when the iSER Message containing the Advertised Write STag, the Base Offset, 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 zero, the iSER layer at the target MUST terminate the connection and free up the resources associated with the connection (as described in Section 5.2.3) 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 by invoking 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 the 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 add the Buffer Offset from the R2T PDU to the Base Offset from the Remote Mapping as the Data Source Tagged Offset of the RDMA Read Request Message.

5. It MUST associate the 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 RCaP 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 the 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 RCaP 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 RCaP layer at the 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 are 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 are 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.

7.3.7. Asynchronous Message

Type: control-type PDU

PDU-specific qualifiers: DataDescriptorSense

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 Send Message to send the Asynchronous Message PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

7.3.8. Text Request and Text Response

Type: control-type PDU

PDU-specific qualifiers: DataDescriptorTextOut (for Text Request), DataDescriptorIn (for Text Response)

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 Send Messages to send the Text Request (or Text Response PDUs). The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

7.3.9. Login Request and Login Response

During the login negotiation, the iSCSI layer interacts with the transport layer directly, and the iSER layer is not involved. See Section 5.1 on iSCSI/iSER Connection Setup. If the underlying transport is TCP, 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.

The iSCSI layer MUST NOT send a Login Request (or a Login Response) PDU during the Full Feature Phase. A Login Request (or a Login Response) PDU, if used, MUST be treated as an iSCSI protocol error. The iSER layer MAY reject such a PDU from the iSCSI layer with an appropriate error code. If a Login Request PDU is received by the iSCSI layer at the target, it MUST respond with a Reject PDU with a reason code of "protocol error".

7.3.10. Logout Request and Logout Response

Type: control-type PDU

PDU-specific qualifiers: None

The iSER layer MUST use a Send Message to send the Logout Request or Logout Response PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP). Sections 5.2.1 and 5.2.2 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.

7.3.11. 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 RCaP 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 MUST NOT send SNACK Request PDUs. A SNACK Request PDU, if used, MUST be treated as an iSCSI protocol error. The iSER layer MAY reject such a PDU from the iSCSI layer with an appropriate error code. If a SNACK Request PDU is received by the iSCSI layer at the target, it MUST respond with a Reject PDU with a reason code of "protocol error".

7.3.12. Reject

Type: control-type PDU

PDU-specific qualifiers: DataDescriptorReject

The iSCSI layer MUST invoke the Send_Control Operational Primitive qualified with DataDescriptorReject, which defines the Reject buffer. The iSER layer MUST use a Send Message to send the Reject PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

7.3.13. NOP-Out and NOP-In

Type: control-type PDU

PDU-specific qualifiers: DataDescriptorNOPOut (for NOP-Out), DataDescriptorNOPIn (for NOP-In)

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 Send Messages to send the NOP-Out (or NOP-In) PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP).

8. Flow Control and STag Management

8.1. Flow Control for RDMA Send Messages

Send Messages in RCaP are used by the iSER layer to transfer iSCSI control-type PDUs. Each Send Message in RCaP consumes an Untagged Buffer at the Data Sink. However, neither the RCaP layer nor the iSER layer provides an explicit flow control mechanism for the Send Messages. Therefore, the iSER layer SHOULD provision enough Untagged buffers for handling incoming Send Messages to prevent buffer exhaustion at the RCaP layer. If buffer exhaustion occurs, it may result in the termination of the connection.

An implementation may choose to satisfy the buffer 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.

iSER also provides the MaxOutstandingUnexpectedPDUs key to be used by the initiator and the target to declare the maximum number of outstanding "unexpected" control-type PDUs that it can receive. It is intended to allow the receiving side to determine the amount of buffer resources needed beyond the normal flow control mechanism available in iSCSI.

The buffer resources required at both the initiator and the target as a result of control-type PDUs sent by the initiator are described in Section 8.1.1. The buffer resources required at both the initiator and target as a result of control-type PDUs sent by the target are described in Section 8.1.2.

8.1.1. Flow Control for Control-Type PDUs from the Initiator

The control-type PDUs that can be sent by an initiator to a target can be grouped into the following categories:

1. Regulated: Control-type PDUs in this category are regulated by the iSCSI CmdSN window mechanism, and the immediate flag is not set.

2. Unregulated but Expected: Control-type PDUs in this category are not regulated by the iSCSI CmdSN window mechanism but are expected by the target.

3. Unregulated and Unexpected: Control-type PDUs in this category are not regulated by the iSCSI CmdSN window mechanism and are "unexpected" by the target.

8.1.1.1. Control-Type PDUs from the Initiator in the Regulated Category

Control-type PDUs that can be sent by the initiator in this category are regulated by the iSCSI CmdSN window mechanism, and the immediate flag is not set.

The queuing capacity required of the iSCSI layer at the target is described in Section 4.2.2.1 of [iSCSI]. For each of the control-type PDUs that can be sent by the initiator in this category, the initiator MUST provision for the buffer resources required for the corresponding control-type PDU sent as a response from the target. The following is a list of the PDUs that can be sent by the initiator and the PDUs that are sent by the target in response:

a. When an initiator sends a SCSI Command PDU, it expects a SCSI Response PDU from the target.

b. When the initiator sends a Task Management Function Request PDU, it expects a Task Management Function Response PDU from the target.

c. When the initiator sends a Text Request PDU, it expects a Text Response PDU from the target.

d. When the initiator sends a Logout Request PDU, it expects a Logout Response PDU from the target.

e. When the initiator sends a NOP-Out PDU as a ping request with ITT != 0xffffffff and TTT = 0xffffffff, it expects a NOP-In PDU from the target with the same ITT and TTT as in the ping request.

The response from the target for any of the PDUs enumerated here may alternatively be in the form of a Reject PDU sent before the task is active, as described in Section 7.3 of [iSCSI].

8.1.1.2. Control-Type PDUs from the Initiator in the Unregulated but Expected Category

For the control-type PDUs in the Unregulated but Expected category, the amount of buffering resources required at the target can be predetermined. The following is a list of the PDUs in this category:

a. SCSI Data-Out PDUs are used by the initiator to send unsolicited data. The amount of buffer resources required by the target can be determined using FirstBurstLength. Note that SCSI Data-Out PDUs are not used for solicited data since the R2T PDU, which is used for solicitation, is transformed into RDMA Read operations by the iSER layer at the target. See Section 7.3.4.

b. A NOP-Out PDU with TTT != 0xffffffff is sent as a ping response by the initiator to the NOP-In PDU sent as a ping request by the target.

8.1.1.3. Control-Type PDUs from the Initiator in the Unregulated and Unexpected Category

PDUs in the Unregulated and Unexpected category are PDUs with the immediate flag set. The number of PDUs that are in this category and can be sent by an initiator is controlled by the value of MaxOutstandingUnexpectedPDUs declared by the target. (See Section 6.7.) After a PDU in this category is sent by the initiator, it is outstanding until it is retired. At any time, the number of outstanding unexpected PDUs MUST NOT exceed the value of MaxOutstandingUnexpectedPDUs declared by the target.

The target uses the value of MaxOutstandingUnexpectedPDUs that it declared to determine the amount of buffer resources required for control-type PDUs in this category that can be sent by an initiator. For the initiator, for each of the control-type PDUs that can be sent in this category, the initiator MUST provision for the buffer resources if required for the corresponding control-type PDU that can be sent as a response from the target.

An outstanding PDU in this category is retired as follows. If the CmdSN of the PDU sent by the initiator in this category is x, the PDU is outstanding until the initiator sends a non-immediate control-type

PDU on the same connection with CmdSN = y (where y is at least x) and the target responds with a control-type PDU on any connection where ExpCmdSN is at least y+1.

When the number of outstanding unexpected control-type PDUs equals MaxOutstandingUnexpectedPDUs, the iSCSI layer at the initiator MUST NOT generate any unexpected PDUs, which otherwise it would have generated, even if the unexpected PDU is intended for immediate delivery.

8.1.2. Flow Control for Control-Type PDUs from the Target

Control-type PDUs that can be sent by a target and are expected by the initiator are listed in the Regulated category. (See Section 8.1.1.1.)

For the control-type PDUs that can be sent by a target and are unexpected by the initiator, the number is controlled by MaxOutstandingUnexpectedPDUs declared by the initiator. (See Section 6.7.) After a PDU in this category is sent by a target, it is outstanding until it is retired. At any time, the number of outstanding unexpected PDUs MUST NOT exceed the value of MaxOutstandingUnexpectedPDUs declared by the initiator. The initiator uses the value of MaxOutstandingUnexpectedPDUs that it declared to determine the amount of buffer resources required for control-type PDUs in this category that can be sent by a target. The following is a list of the PDUs in this category and the conditions for retiring the outstanding PDU:

a. For an Asynchronous Message PDU with StatSN = x, the PDU is outstanding until the initiator sends a control-type PDU with ExpStatSN set to at least x+1.

b. For a Reject PDU with StatSN = x, which is sent after a task is active, the PDU is outstanding until the initiator sends a control-type PDU with ExpStatSN set to at least x+1.

c. For a NOP-In PDU with ITT = 0xffffffff and StatSN = x, the PDU is outstanding until the initiator responds with a control-type PDU on the same connection where ExpStatSN is at least x+1. But if the NOP-In PDU is sent as a ping request with TTT != 0xffffffff, the PDU can also be retired when the initiator sends a NOP-Out PDU with the same ITT and TTT as in the ping request. Note that when a target sends a NOP-In PDU as a ping request, it must provision a buffer for the NOP-Out PDU sent as a ping response from the initiator.

When the number of outstanding unexpected control-type PDUs equals MaxOutstandingUnexpectedPDUs, the iSCSI layer at the target MUST NOT generate any unexpected PDUs, which otherwise it would have generated, even if its intent is to indicate an iSCSI error condition (e.g., Asynchronous Message, Reject). Task timeouts, as in the initiator's waiting for a command completion or other connection and session-level exceptions, will ensure that correct operational behavior will result in these cases despite not generating the PDU. This rule overrides any other requirements elsewhere that require that a Reject PDU MUST be sent.

(Implementation note: SCSI task timeout and recovery can be a lengthy process and hence SHOULD be avoided by proper provisioning of resources.)

(Implementation note: To ensure that the initiator has a means to inform the target that outstanding PDUs have been retired, the target should reserve the last unexpected control-type PDU allowable by the value of MaxOutstandingUnexpectedPDUs declared by the initiator for sending a NOP-In ping request with TTT != 0xffffffff to allow the initiator to return the NOP-Out ping response with the current ExpStatSN.)

8.2. Flow Control for RDMA Read Resources

If iSERHelloRequired is negotiated to "Yes", then 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 5.1.3. Exceeding the number of RDMA Read operations allowed on a connection will result in the connection being terminated by the RCaP 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 RCaP Stream.

During connection setup (see Section 5.1), 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 connection resources necessary to support RCaP, 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, if iSERHelloRequired is negotiated to "Yes", then the first message sent by the initiator is the iSER Hello Message (see Section 9.3), which contains the value of iSER-IRD. In response to the iSER Hello Message, the target sends the iSER HelloReply Message (see Section 9.4), which contains the value of iSER-ORD. The iSER layer at both the initiator and the target MAY adjust (lower) the 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 control the flow of the RDMA Read Request Messages so that it does not exceed the iSER-ORD value at the target.

If iSERHelloRequired is negotiated to "No", then the maximum number of RDMA Read operations that can be active is negotiated via other means outside the scope of this document. For example, in InfiniBand, iSER connection setup uses InfiniBand Connection Manager (CM) Management Datagrams (MADs), with additional iSER information exchanged in the private data.

8.3. STag Management

An STag is an identifier of a Tagged Buffer used in an RDMA operation. If the STags are exposed on the wire by being Advertised in the iSER header or declared in the header of an RCaP Message, then the allocation and the subsequent invalidation of the STags are as specified in this document.

8.3.1. 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 7.3.1 for details. The number of STags allocated depends on whether the command is unidirectional or bidirectional and whether or not solicited write data transfer is involved.

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 7.3.3 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 7.3.6 for details.

8.3.2. 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 7.3.2.

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 request the iSER layer at the initiator to invalidate the STags by invoking the Deallocate_Task_Resources Operational Primitive qualified with ITT. In response, the iSER layer at the initiator MUST locate the STags (if any) in the Local Mapping. The iSER layer at the initiator MUST invalidate the STags (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 request the iSER layer at the target to invalidate the STags by invoking the Deallocate_Task_Resources Operational Primitive qualified with ITT. In response, the iSER layer at the target MUST locate the local STags (if any) in the Local Mapping. The iSER layer at the target MUST invalidate the local STags (if any) and the Local Mapping.

9. iSER Control and Data Transfer

For iSCSI data-type PDUs (see Section 7.1), the iSER layer uses RDMA Read and RDMA Write operations to transfer the solicited data. For iSCSI control-type PDUs (see Section 7.2), the iSER layer uses Send Messages of RCaP.

9.1. iSER Header Format

An iSER header MUST be present in every Send Message of RCaP. The iSER header is located in the first 28 bytes of the message payload of the Send Message of RCaP, as shown in Figure 2.

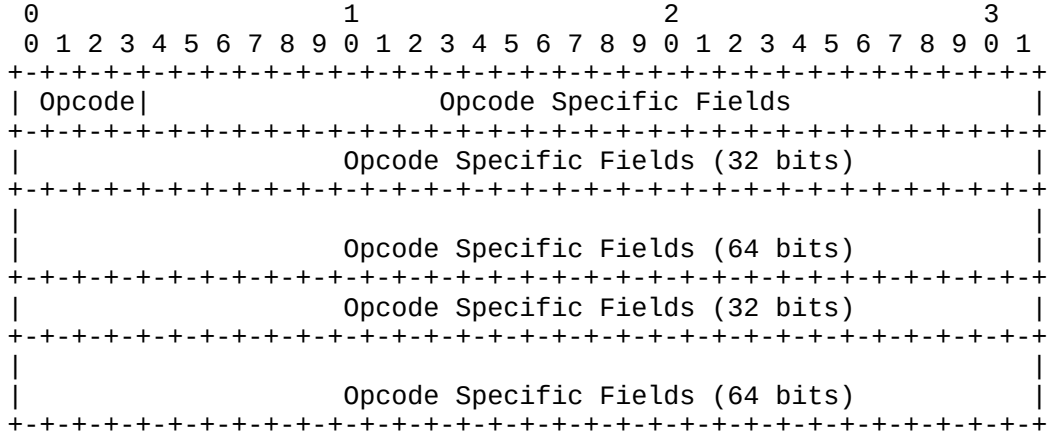

Figure 2: iSER Header Format

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 unassigned.

9.2. iSER Header Format for iSCSI Control-Type PDU

The iSER layer uses Send Messages of RCaP to transfer iSCSI control-type PDUs (see Section 7.2). The message payload of each of the Send Messages of RCaP used for transferring an iSER Message contains an iSER Header followed by an iSCSI control-type PDU.

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

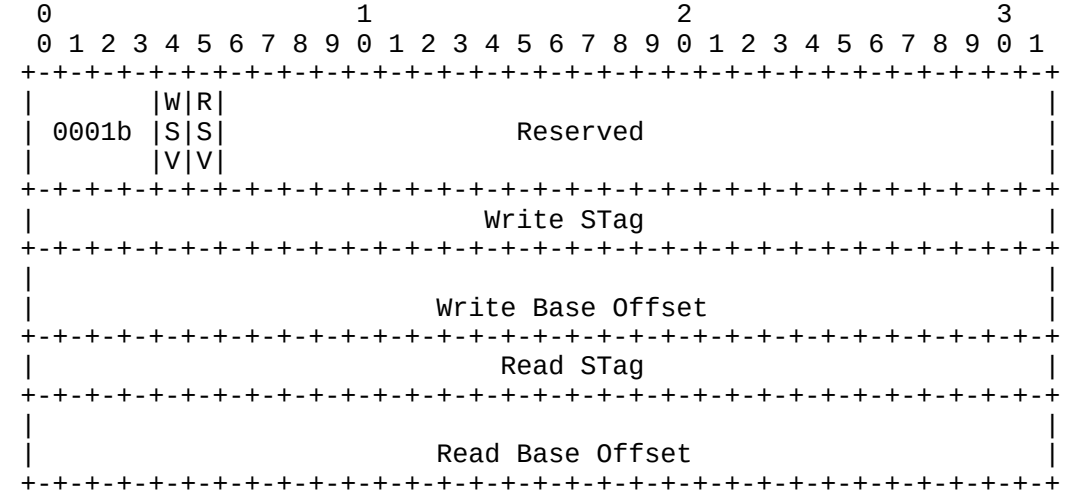

Figure 3: iSER Header Format for iSCSI Control-Type PDU

WSV - Write STag Valid flag: 1 bit

This flag indicates the validity of the Write STag field and the Write Base Offset field of the iSER Header. If set to one, the Write STag field and the Write Base Offset field in this iSER Header are valid. If set to zero, the Write STag field and the Write Base Offset 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 and the Read Base Offset field of the iSER Header. If set to one, the Read STag field and the Read Base Offset field in this iSER Header

are valid. If set to zero, the Read STag field and the Read Base Offset 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 and ignored on receive.

Write Base Offset: 64 bits

This field contains the Base Offset associated with the I/O Buffer for the SCSI Write command when the Write STag Valid flag is set to one. When the Write STag Valid flag is set to zero, this field MUST be set to zero and ignored on receive.

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 and ignored on receive.

Read Base Offset: 64 bits

This field contains the Base Offset associated with the I/O Buffer for the SCSI Read command when the Read STag Valid flag is set to one. When the Read STag Valid flag is set to zero, this field MUST be set to zero and ignored on receive.

Reserved:

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

9.3. 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. If iSERHelloRequired is negotiated to "Yes", then iSER Hello Message is the first iSER Message sent on the RCaP Stream from the iSER layer at the initiator to the iSER layer at the target.

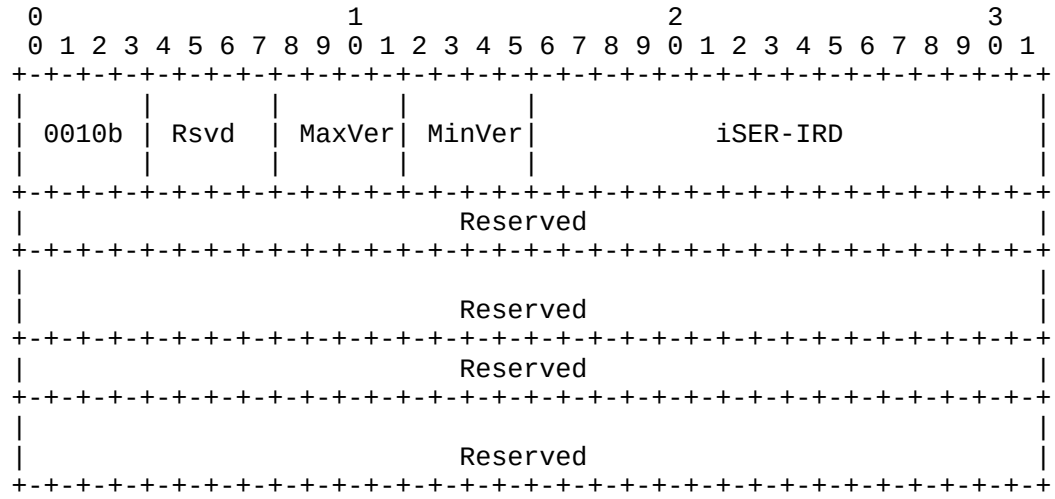

Figure 4: iSER Header Format for iSER Hello Message

MaxVer - Maximum Version: 4 bits

This field specifies the maximum version of the iSER protocol supported. It MUST be set to 10 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 10 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.

9.4. 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. If iSERHelloRequired is negotiated to "Yes", then the iSER HelloReply Message is the first iSER Message sent on the RCaP Stream from the iSER layer at the target to the iSER layer at the initiator.

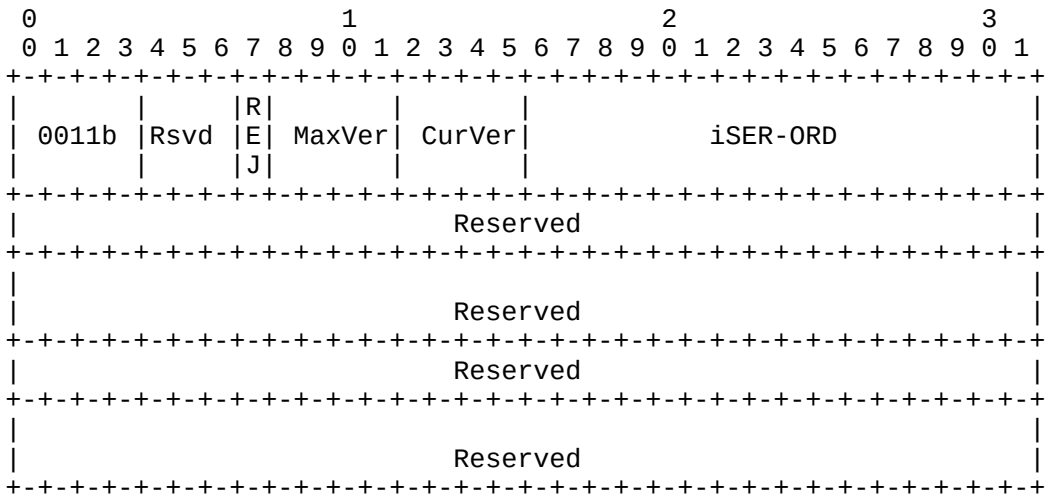

Figure 5: iSER Header Format for iSER HelloReply Message

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 10 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 10 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.

9.5. 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.

9.5.1. 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 RCaP layer to transmit a Send Message with the message payload consisting of the iSER header followed by the SCSI Command PDU and immediate data (if any). The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP). If there is solicited data, the iSER layer MUST Advertise the Write STag and the Base Offset in the iSER header of the Send Message, as described in Section 9.2. Upon receiving the Send 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 7.3.1 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 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 7.3.4 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 7.3.6 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 Send 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 7.3.2 for details on the handling of the SCSI Response PDU.

9.5.2. 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 RCaP layer to transmit a Send Message with the message payload consisting of the iSER header followed by the SCSI Command PDU. The SendSE Message should be used if supported by the RCaP layer (e.g., iWARP). The iSER layer at the initiator MUST Advertise the Read STag and the Base Offset in the iSER header of the Send Message, as described in Section 9.2. Upon receiving the Send 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 7.3.1 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 7.3.5 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. The SendInvSE Message should be used if supported by the RCaP layer (e.g., iWARP). Upon receiving the Send 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 7.3.2 for details on the handling of the SCSI Response PDU.

9.5.3. Bidirectional Operation

The initiator and the target handle the SCSI Write and the SCSI Read portions of this bidirectional operation the same as described in Sections 9.5.1 and 9.5.2, respectively.

10. iSER Error Handling and Recovery

RCaP provides the iSER layer with reliable in-order delivery. Therefore, the error management needs of an iSER-assisted connection are somewhat different than those of a Traditional iSCSI connection.

10.1. Error Handling

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

1. Those originating at the transport layer (e.g., TCP).

2. Those originating at the RCaP layer.

3. Those originating at the iSER layer.

4. Those originating at the iSCSI layer.

10.1.1. Errors in the Transport Layer

If the transport layer is TCP, then TCP packets with detected 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 transport layer will cause the iSCSI/iSER connection to be terminated as well.

10.1.1.1. Failure in the Transport Layer Before RCaP Mode is Enabled

If the 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 connection is lost or terminated after the iSCSI layer has invoked the Allocate_Connection_Resources Operational Primitive, then the iSCSI layer MUST request the iSER layer to deallocate all connection resources by invoking the Deallocate_Connection_Resources Operational Primitive.

10.1.1.2. Failure in the Transport Layer After RCaP Mode is Enabled

If the 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 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 5.2.3.2.

10.1.2. Errors in the RCaP Layer

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

10.1.2.1. Errors Detected in the Local RCaP Layer

If an error is encountered at the local RCaP layer, the RCaP layer MAY send a Send Message to the Remote Peer to report the error if possible. (For iWARP, see [RDMAP] for the list of errors where a Terminate Message is sent.) The RCaP layer is responsible for terminating the connection. After the RCaP layer notifies the iSER layer that the 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 5.2.3.2.

10.1.2.2. Errors Detected in the RCaP Layer at the Remote Peer

If an error is encountered at the RCaP layer at the Remote Peer, the RCaP layer at the Remote Peer may send a Send Message to report the error if possible. If it is unable to send a Send Message, the connection is terminated. This is treated the same as a failure in the transport layer after RDMA is enabled, as described in Section 10.1.1.2.

If an error is encountered at the RCaP layer at the Remote Peer and it is able to send a Send Message, the RCaP layer at the Remote Peer is responsible for terminating the connection. After the local RCaP layer notifies the iSER layer that the 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 5.2.3.2.

10.1.3. Errors in the iSER Layer

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

10.1.3.1. Insufficient Connection Resources to Support RCaP 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 connection resources necessary to support RCaP, it MUST return a status of failure to the iSCSI layer at the initiator. The iSCSI layer at the initiator MUST terminate the connection as described in Section 5.2.3.1.

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 connection resources necessary to support RCaP, 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 0x03 (Target Error), and a Status-Code of 0x02 (Out of Resources). The iSCSI layers at the initiator and the target MUST terminate the connection as described in Section 5.2.3.1.

10.1.3.2. iSER Negotiation Failures

If iSERHelloRequired is negotiated to "Yes" and the RCaP or iSER related parameters declared by the initiator in the iSER Hello Message are unacceptable to the iSER layer at the target, the iSER layer at the target MUST set the Reject (REJ) flag, as described in Section 9.4, 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 RCaP layer to send the iSER HelloReply Message, the handling of the error situation is the same as that for iSER format errors as described in Section 10.1.3.3.

10.1.3.3. 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 Sections 9.3 and 9.4)

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

1. The iSER layer MUST request the RCaP layer to terminate the RCaP Stream. The RCaP layer MUST terminate the associated connection.

2. The iSER layer MUST notify the iSCSI layer of the connection termination 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 5.2.3.2.

10.1.3.4. iSER Protocol Errors

If iSERHelloRequired is negotiated to "Yes", then the first iSER Message sent by the iSER layer at the initiator MUST be the iSER Hello Message (see Section 9.3). In this case the first iSER Message sent by the iSER layer at the target MUST be the iSER HelloReply Message (see Section 9.4). Failure to send the iSER Hello or HelloReply Message, as indicated by the wrong Opcode in the iSER header, is a protocol error. Conversely, if the iSER Hello Message is sent by the iSER layer at the initiator when iSERHelloRequired is negotiated to "No", the iSER layer at the target MAY treat this as a protocol error or respond with an iSER HelloReply Message. The handling of iSER protocol errors is the same as that for iSER format errors as described in Section 10.1.3.3.

If the sending side of an iSER-enabled connection acts in a manner not permitted by the negotiated or declared login/text operational key values as described in Section 6, this is a protocol error and the receiving side MAY handle this the same as for iSER format errors as described in Section 10.1.3.3.

10.1.4. 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 10.2.

10.1.4.1. iSCSI Format Errors

When an iSCSI format error is detected, the iSCSI layer MUST request the iSER layer to terminate the RCaP Stream by invoking the Connection_Terminate Operational Primitive. For more details on connection termination, see Section 5.2.3.1.

10.1.4.2. 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".

10.1.4.3. 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 RCaP layer will deliver all messages in the order they were sent, sequence errors will not occur in iSER-assisted mode.

10.1.4.4. iSCSI Protocol Error

When the iSCSI layer handles certain protocol errors by dropping the connection, the error handling is the same as that for iSCSI format errors as described in Section 10.1.4.1.

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.

10.1.4.5. SCSI Timeouts and Session Errors

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

10.1.4.6. 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 request the iSER layer to deallocate all connection

resources by invoking the Deallocate_Connection_Resources Operational Primitive. The iSCSI layer at the initiator MUST terminate the 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 has not yet invoked the Enable_Datamover Operational Primitive, the iSCSI layer at the target MUST request the iSER layer at the target to deallocate all connection resources by invoking the Deallocate_Connection_Resources Operational Primitive. The iSCSI layer at both the initiator and the target MUST terminate the 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 request the iSER layer to deallocate all connection resources by invoking the Deallocate_Connection_Resources Operational Primitive. The iSCSI layer MUST terminate the 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.

10.2. 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 10.2.1 on PDU Recovery.

* For ErrorRecoveryLevel 2, see Section 10.2.2 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 5.1.1 and 5.1.2.

10.2.1. PDU Recovery

As described in Sections 10.1.4.2 and 10.1.4.3, digest and sequence errors will not occur in the iSER-assisted mode. If the RCaP layer detects an error, it will close the iSCSI/iSER connection, as described in Section 10.1.2. Therefore, PDU recovery is not useful in the iSER-assisted mode.

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

10.2.2. Connection Recovery

The iSCSI layer at the initiator MAY reassign connection allegiance for non-immediate commands that 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 7.3.3 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 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 request the iSER layer to process SCSI Data-In for all unacknowledged data by invoking the Put_Data Operational Primitive. See Section 7.3.5 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 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 by invoking the Get_Data Operational Primitive. See Section 7.3.6 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.

11. Security Considerations

When iSER is layered on top of an RCaP layer and provides the RDMA extensions to the iSCSI protocol, the security considerations of iSER are the same as that of the underlying RCaP layer. For iWARP, this is described in [RDMAP] and [RDDPSEC], plus the updates to both of those RFCs that are contained in [IPSEC-IPS].

Since iSER-assisted iSCSI protocol is still functionally iSCSI from a security considerations perspective, all of the iSCSI security requirements as described in [iSCSI] apply. If iSER is layered on top of a non-IP-based RCaP layer, all the security protocol mechanisms applicable to that RCaP layer are also applicable to an iSCSI/iSER connection. If iSER is layered on top of a non-IP protocol, the IPsec mechanism as specified in [iSCSI] MUST be implemented at any point where the iSER protocol enters the IP network (e.g., via gateways), and the non-IP protocol SHOULD implement (optional to use) a packet-by-packet security protocol equal in strength to the IPsec mechanism specified by [iSCSI].

In order to protect target RCaP connection resources from possible resource exhaustion attacks, allocation of such resources for a new connection MUST be delayed until it is reasonably certain that the new connection is not part of a resource exhaustion attack (e.g., until after the SecurityNegotiation stage of Login); see Section 5.1.2.

A valid STag exposes I/O Buffer resources to the network for access via the RCaP. The security measures for the RCAP and iSER described in the above paragraphs can be used to protect data in an I/O buffer from undesired disclosure or modification, and these measures are of heightened importance for implementations that retain (e.g., cache) STags for use in multiple tasks (e.g., iSCSI I/O operations) because the resources are exposed to the network for a longer period of time.

A complementary means of controlling I/O Buffer resource exposure is invalidation of the STag after completion of the associated task, as specified in Section 1.5.1. The use of Send with Invalidate messages (which cause remote STag invalidation) is OPTIONAL, therefore the iSER layer MUST NOT rely on use of a Send with Invalidate by its Remote Peer to cause local STag invalidation. If an STag is expected to be invalid after completion of a task, the iSER layer MUST check the STag and invalidate it if it is still valid.

12. IANA Considerations

IANA has added the following entries to the "iSCSI Login/Text Keys" registry:

MaxAHSLength, RFC 7145

TaggedBufferForSolicitedDataOnly, RFC 7145

iSERHelloRequired, RFC 7145

IANA has updated the following entries in the "iSCSI Login/Text Keys" registry to reference this RFC.

InitiatorRecvDataSegmentLength

MaxOutstandingUnexpectedPDUs

RDMAExtensions

TargetRecvDataSegmentLength

IANA has also changed the reference to RFC 5046 for the "iSCSI Login/Text Keys" registry to refer to this RFC.

IANA has updated the registrations of the iSER Opcodes 1-3 in the "iSER Opcodes" registry to reference this RFC. IANA has also changed the reference to RFC 5046 for the "iSER Opcodes" registry to refer to this RFC.

13. References

13.1. Normative References

[RFC5046] Ko, M., Chadalapaka, M., Hufferd, J., Elzur, U., Shah, H., and P. Thaler, "Internet Small Computer System Interface (iSCSI) Extensions for Remote Direct Memory Access (RDMA)", RFC 5046, October 2007.

[iSCSI] Chadalapaka, M., Satran, J., Meth, K., and D. Black, "Internet Small Computer System Interface (iSCSI) Protocol (Consolidated)", RFC 7143, April 2014.

[RDMAP] Recio, R., Metzler, B., Culley, P., Hilland, J., and D. Garcia, "A Remote Direct Memory Access Protocol Specification", RFC 5040, October 2007.

[DDP] Shah, H., Pinkerton, J., Recio, R., and P. Culley, "Direct Data Placement over Reliable Transports", RFC 5041, October 2007.

[MPA] Culley, P., Elzur, U., Recio, R., Bailey, S., and J. Carrier, "Marker PDU Aligned Framing for TCP Specification", RFC 5044, October 2007.

[RDDPSEC] Pinkerton, J. and E. Deleganes, "Direct Data Placement Protocol (DDP) / Remote Direct Memory Access Protocol (RDMAP) Security", RFC 5042, October 2007.

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[IPSEC-IPS] Black, D. and P. Koning, "Securing Block Storage Protocols over IP: RFC 3723 Requirements Update for IPsec v3", RFC 7146, April 2014.

13.2. Informative References

[SAM5] INCITS Technical Committee T10, "SCSI Architecture Model - 5 (SAM-5)", T10/BSR INCITS 515 rev 04, Committee Draft.

[iSCSI-SAM] Knight, F. and M. Chadalapaka, "Internet Small Computer System Interface (iSCSI) SCSI Features Update", RFC 7144, April 2014.

[DA] Chadalapaka, M., Hufferd, J., Satran, J., and H. Shah, "DA: Datamover Architecture for the Internet Small Computer System Interface (iSCSI)", RFC 5047, October 2007.

[IB] InfiniBand Architecture Specification Volume 1 Release 1.2, October 2004

[IPoIB] Chu, J. and V. Kashyap, "Transmission of IP over InfiniBand (IPoIB)", RFC 4391, April 2006.

Appendix A. Summary of Changes from RFC 5046

All changes are backward compatible with RFC 5046 except for item #8, which reflects all known implementations of iSER, each of which has implemented this change, despite its absence in RFC 5046. As a result, a hypothetical implementation based on RFC 5046 will not interoperate with an implementation based on this version of the specification.

1. Removed the requirement that a connection be opened in "normal" TCP mode and transitioned to zero-copy mode. This allows the specification to conform to existing implementations for both InfiniBand and iWARP. Changes were made in Sections 1, 3.1.6, 4.2, 5.1, 5.1.1, 5.1.2, 5.1.3, 10.1.3.4, and 11.

2. Added a clause in Section 6.2 to clarify that MaxRecvDataSegmentLength must be ignored if it is declared in the Login Phase.

3. Added a clause in Section 6.2 to clarify that the initiator must not send more than InitiatorMaxRecvDataSegmentLength worth of data when a NOP-Out request is sent with a valid Initiator Task Tag. Since InitiatorMaxRecvDataSegmentLength can be smaller than TargetMaxRecvDataSegmentLength, returning the original data in the NOP-Out request in this situation can overflow the receive buffer unless the length of the data sent with the NOP-Out request is less than InitiatorMaxRecvDataSegmentLength.

4. Added a SHOULD negotiate recommendation for MaxOutstandingUnexpectedPDUs in Section 6.7.

5. Added MaxAHSLength key in Section 6.8 to set a limit on the AHS Length. This is useful when posting receive buffers in knowing what the maximum possible message length is in a PDU that contains AHS.

6. Added TaggedBufferForSolicitedDataOnly key in Section 6.9 to indicate how the memory region will be used. An initiator can treat the memory regions intended for unsolicited and solicited data differently and can use different registration modes. In contrast, RFC 5046 treats the memory occupied by the data as a contiguous (or virtually contiguous, by means of scatter-gather mechanisms) and homogenous region. Adding a new key will allow different memory models to be accommodated. Changes were also made in Section 7.3.1.

7. Added iSERHelloRequired key in Section 6.10 to allow an initiator to allocate connection resources after the login process by requiring the use of the iSER Hello messages before sending iSCSI PDUs. The default is "No" since iSER Hello messages have not been implemented and are not in use. Changes were made in Sections 5.1.1, 5.1.2, 5.1.3, 8.2, 9.3, 9.4, 10.1.3.2, and 10.1.3.4.

8. Added two 64-bit fields in iSER header in Section 9.2 for the Read Base Offset and the Write Base Offset to accommodate a non-zero Base Offset. This allows one implementation such as the Open Fabrics Enterprise Distribution (OFED) stack to be used in both the InfiniBand and the iWARP environment.

   Changes were made in the definitions of Base Offset, Advertisement, and Tagged Buffer. Changes were also made in Sections 1.5.1, 1.6, 1.7, 7.3.1, 7.3.3, 7.3.5, 7.3.6, 9.1, 9.3, 9.4, 9.5.1, and 9.5.2. This change is not backward compatible with RFC 5046, but it was part of all known implementations of iSER at the time this document was developed.

9. Remove iWARP-specific behavior. Changes were made in the definitions of RDMA Operation and Send Message Type.

   Clarifications were added in Section 1.5.2 on the use of SendSE and SendInvSE. These clarifications reflect a removal of the requirements in RFC 5046 for the use of these messages, as implementations have not followed RFC 5046 in this area. Changes affecting Send with Invalidate were made in Sections 1.5.1, 1.6, 1.7, 4.1, and 7.3.2. Changes affecting Terminate were made in Sections 10.1.2.1 and 10.1.2.2. Changes were made in Appendix B to remove iWARP headers.

10. Removed denial-of-service descriptions for the initiator in Section 5.1.1 since they are applicable for the target only.

11. Clarified in Section 1.5.1 that STag invalidation is the initiator’s responsibility for security reasons, and the initiator cannot rely on the target using an Invalidate version of Send. Added text in Section 11 on Stag invalidation.

Appendix B. Message Format for iSER

This section is for information only and is NOT part of the standard.

B.1. iWARP Message Format for iSER Hello Message

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

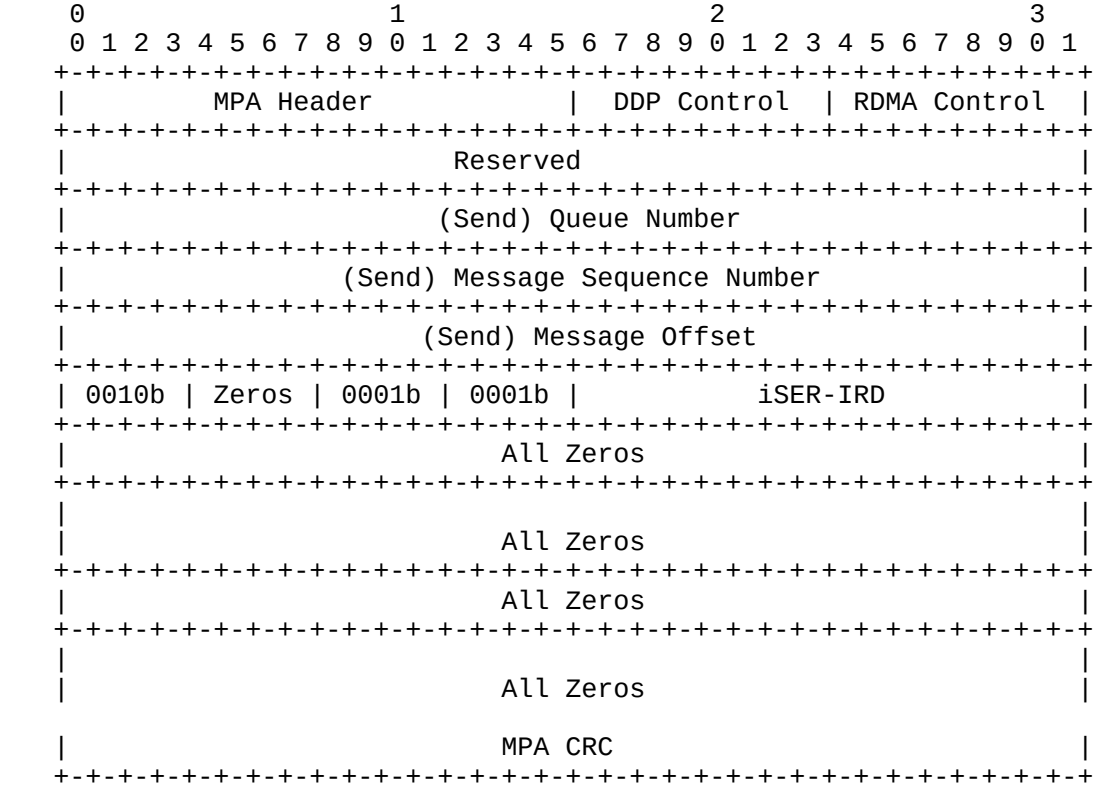

Figure 6: SendSE Message Containing an iSER Hello Message

B.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 zero.

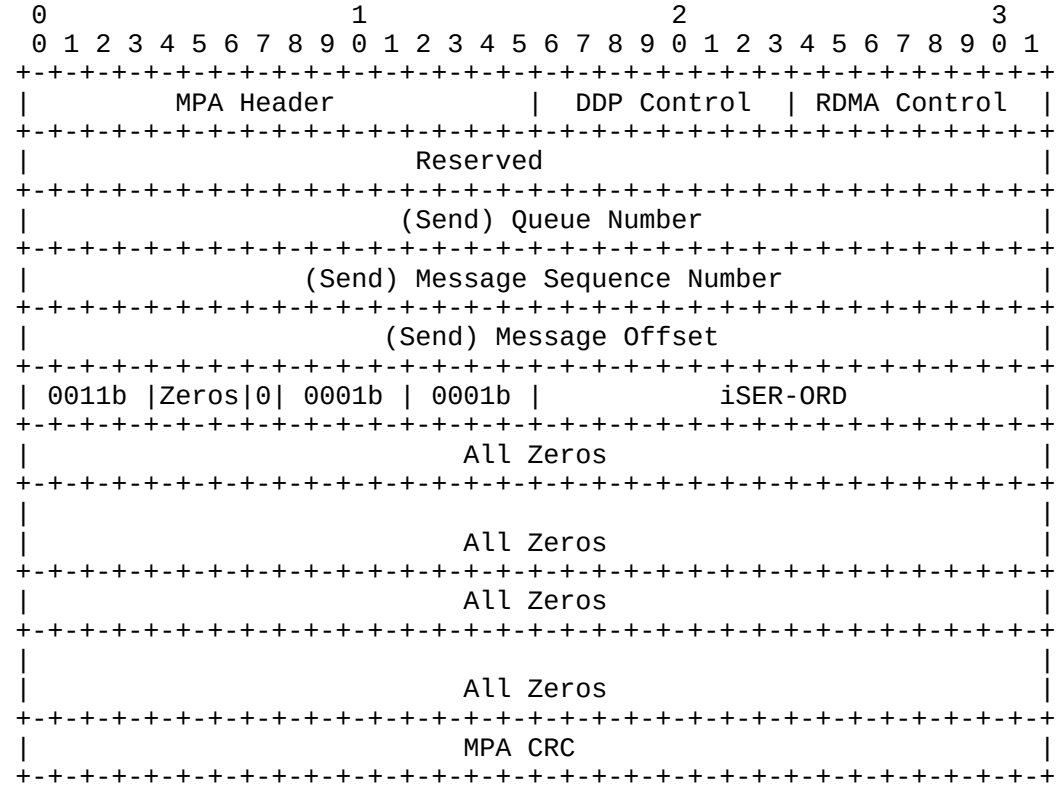

Figure 7: SendSE Message Containing an iSER HelloReply Message

B.3. iSER Header Format for SCSI Read Command PDU

The following figure depicts a SCSI Read Command PDU embedded in an iSER 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, the Write Base Offset field is set to all zeros, the Read STag field contains a valid Read STag, and the Read Base Offset field contains a valid Base Offset for the Read Tagged Buffer.

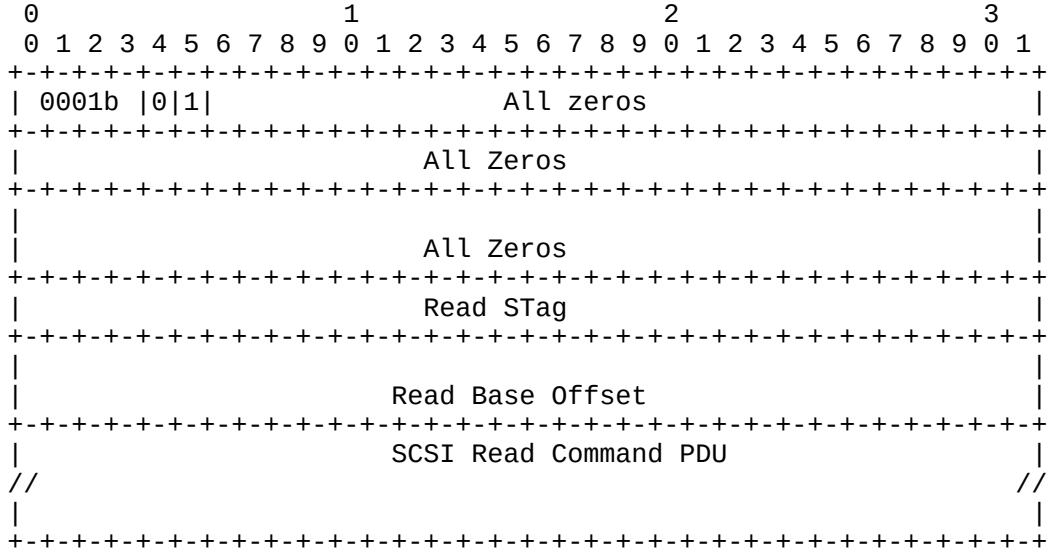

Figure 8: iSER Header Format for SCSI Read Command PDU

B.4. iSER Header Format for SCSI Write Command PDU

The following figure depicts a SCSI Write Command PDU embedded in an iSER 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, the Write Base Offset field contains a valid Base Offset for the Write Tagged Buffer, the Read STag field is set to all zeros since it is not used, and the Read Base Offset field is set to all zeros.

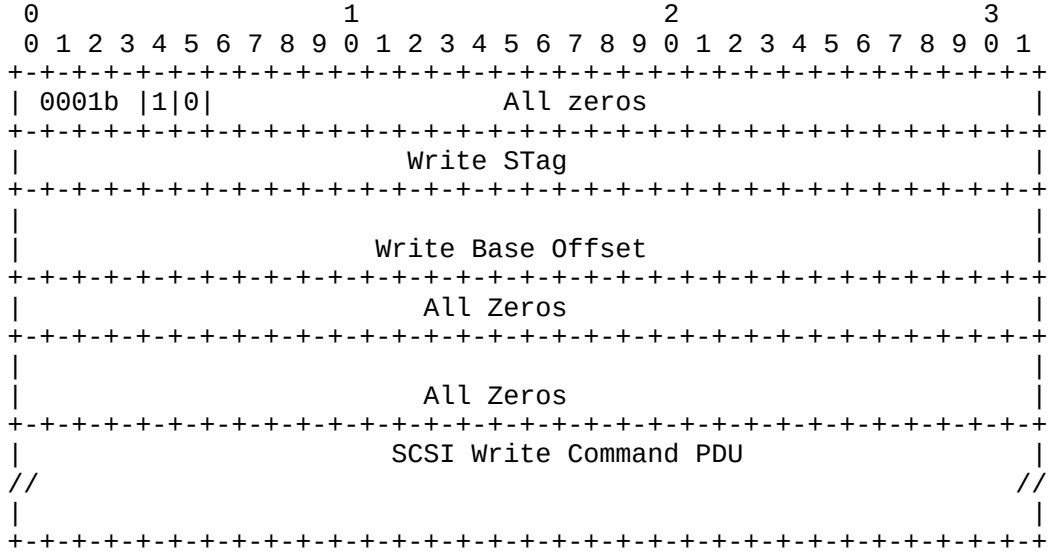

Figure 9: iSER Header Format for SCSI Write Command PDU

B.5. iSER Header Format for SCSI Response PDU

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

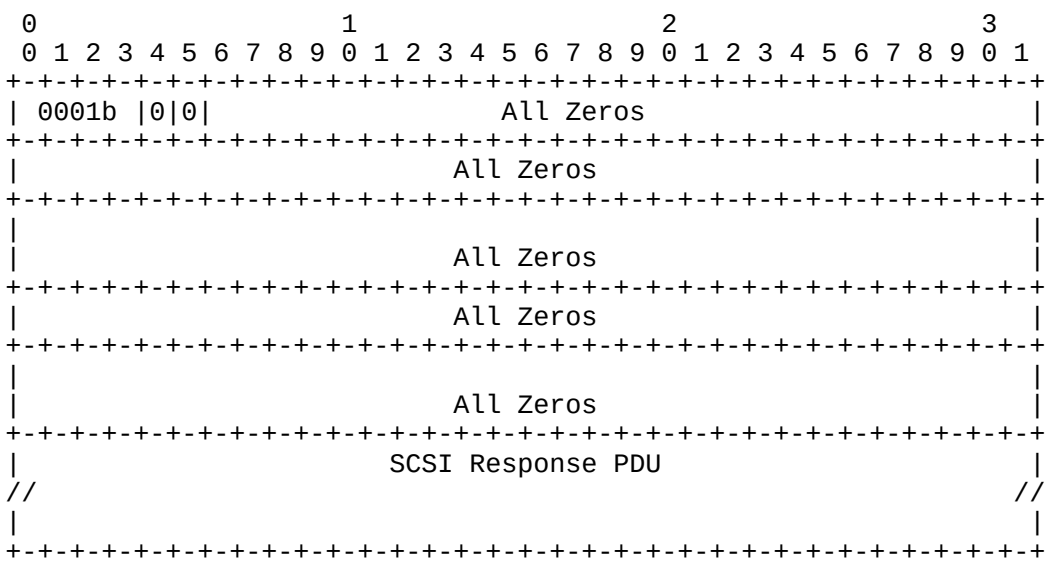

Figure 10: iSER Header Format for SCSI Response PDU

Appendix C. Architectural Discussion of iSER over InfiniBand

This section explains how an InfiniBand network (with Gateways) would be structured. It is informational only and is intended to provide insight on how iSER is used in an InfiniBand environment.

C.1. Host Side of iSCSI and iSER Connections in InfiniBand

Figure 11 defines the topologies in which iSCSI and iSER will be able to operate on an InfiniBand Network.

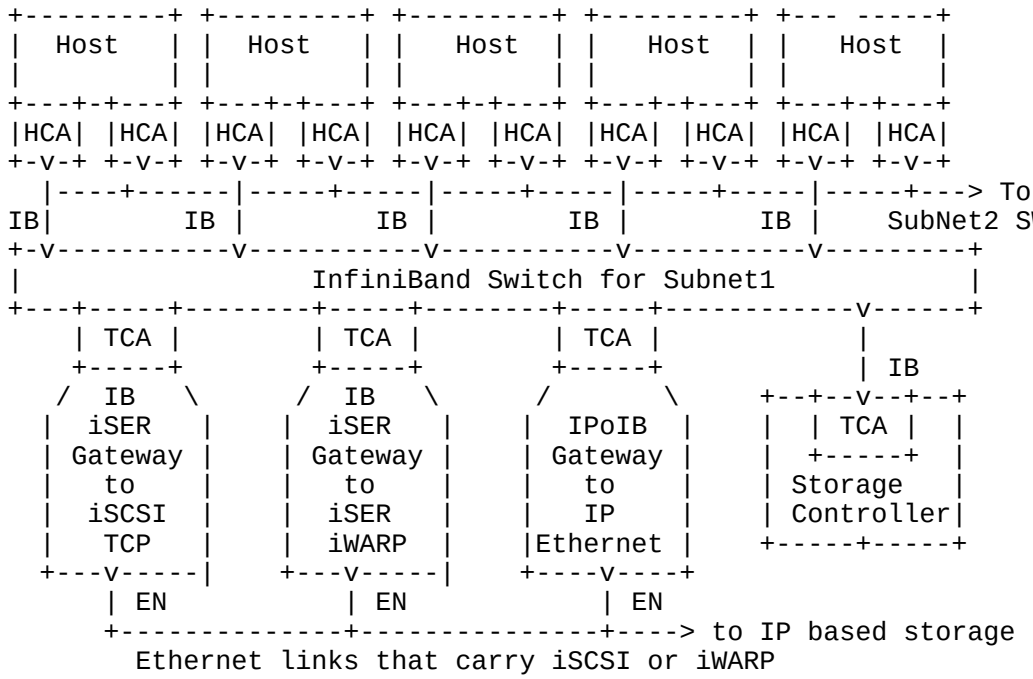

Figure 11: iSCSI and iSER on IB

In Figure 11, the Host systems are connected via the InfiniBand Host Channel Adapters (HCAs) to the InfiniBand links. With the use of IB switch(es), the InfiniBand links connect the HCA to InfiniBand Target Channel Adapters (TCAs) located in gateways or Storage Controllers. An iSER-capable IB-IP Gateway converts the iSER Messages encapsulated in IB protocols to either standard iSCSI, or iSER Messages for iWARP. An [IPoIB] Gateway converts the InfiniBand [IPoIB] protocol to IP protocol, and in the iSCSI case, permits iSCSI to be operated on an IB Network between the Hosts and the [IPoIB] Gateway.

C.2. Storage Side of iSCSI and iSER Mixed Network Environment

Figure 12 shows a storage controller that has three different portal groups: one supporting only iSCSI (TPG-4), one supporting iSER/iWARP or iSCSI (TPG-2), and one supporting iSER/IB (TPG-1). Here, "TPG" stands for "Target Portal Group".

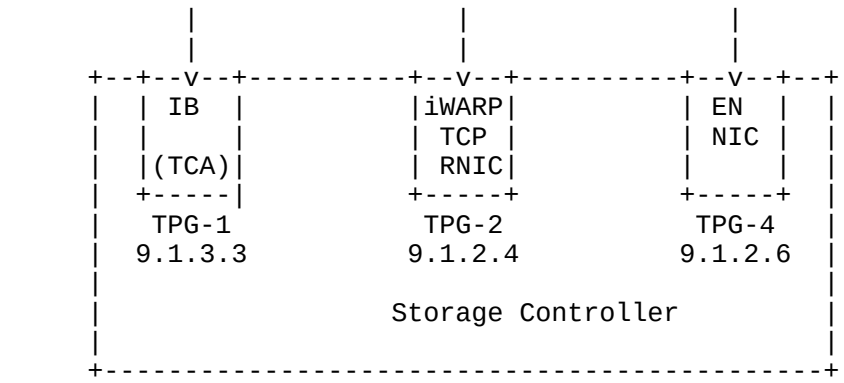

Figure 12: Storage Controller with TCP, iWARP, and IB Connections

The normal iSCSI portal group advertising processes (via the Service Location Protocol (SLP), Internet Storage Name Service (iSNS), or SendTargets) are available to a Storage Controller.

C.3. Discovery Processes for an InfiniBand Host

An InfiniBand Host system can gather portal group IP addresses from SLP, iSNS, or the SendTargets discovery processes by using TCP/IP via [IPoIB]. After obtaining one or more remote portal IP addresses, the Initiator uses the standard IP mechanisms to resolve the IP address to a local outgoing interface and the destination hardware address (Ethernet MAC or InfiniBand Global Identifier (GID) of the target or a gateway leading to the target). If the resolved interface is an [IPoIB] network interface, then the target portal can be reached through an InfiniBand fabric. In this case, the Initiator can establish an iSCSI/TCP or iSCSI/iSER session with the Target over that InfiniBand interface, using the hardware address (InfiniBand GID) obtained through the standard Address Resolution Protocol (ARP) processes.

If more than one IP address is obtained through the discovery process, the Initiator should select a Target IP address that is on the same IP subnet as the Initiator, if one exists. This will avoid a potential overhead of going through a gateway when a direct path exists.

In addition, a user can configure manual static IP route entries if a particular path to the target is preferred.

C.4. IBTA Connection Specifications

It is outside the scope of this document, but it is expected that the InfiniBand Trade Association (IBTA) has or will define:

* The iSER ServiceID

* A means for permitting a Host to establish a connection with a peer InfiniBand end-node, and that peer indicating when that end-node supports iSER, so the Host would be able to fall back to iSCSI/TCP over [IPoIB].

* A means for permitting the Host to establish connections with IB iSER connections on storage controllers or IB iSER-connected Gateways in preference to IPoIB-connected Gateways/Bridges or connections to Target Storage Controllers that also accept iSCSI via [IPoIB].

* A means for combining the IB ServiceID for iSER and the IP port number such that the IB Host can use normal IB connection processes, yet ensure that the iSER target peer can actually connect to the required IP port number.

Appendix D. Acknowledgments

The authors acknowledge the following individuals for identifying implementation issues and/or suggesting resolutions to the issues clarified in this document: Robert Russell, Arne Redlich, David Black, Mallikarjun Chadalapaka, Tom Talpey, Felix Marti, Robert Sharp, Caitlin Bestler, Hemal Shah, Spencer Dawkins, Pete Resnick, Ted Lemon, Pete McCann, and Steve Kent. Credit also goes to the authors of the original iSER Specification [RFC5046], including Michael Ko, Mallikarjun Chadalapaka, John Hufferd, Uri Elzur, Hemal Shah, and Patricia Thaler. This document benefited from all of their contributions.

Authors' Addresses

Michael Ko

EMail: mkosjc@gmail.com

Alexander Nezhinsky
Mellanox Technologies
13 Zarchin St.
Raanana 43662
Israel

Phone: +972-74-712-9000
EMail: alexandern@mellanox.com, nezhinsky@gmail.com