A Study of iSCSI Extensions for RDMA (iSER)

Mallikarjun Chadalapaka
Hewlett-Packard Company
cbm@rose.hp.com

Uri Elzur
Broadcom
uri@broadcom.com

Michael Ko
IBM
mako@almaden.ibm.com

Hemal Shah
Intel Corporation
hemal.shah@intel.com

Patricia Thaler
Agilent Technologies
pat_thaler@agilent.com

Abstract
The iSCSI protocol is the IETF standard that maps the SCSI family of application protocols onto TCP/IP enabling convergence of storage traffic on to standard TCP/IP fabrics. The ability to efficiently transfer and place the data on TCP/IP networks is crucial for this convergence of the storage traffic. The iWARP protocol suite provides Remote Direct Memory Access (RDMA) semantics over TCP/IP networks and enables efficient memory-to-memory data transfers over an IP fabric. This paper studies the design process of iSCSI Extensions for RDMA (iSER), a protocol that maps the iSCSI protocol over the iWARP protocol suite. As part of this study, this paper shows how iSER enables efficient data movement for iSCSI using generic RDMA hardware and then presents a discussion of the iWARP architectural features that were conceived during the iSER design. These features potentially enable highly efficient realizations of other I/O protocols as well.

Categories and Subject Descriptors
C.2.2 [**Computer-Communication Networks**]: Network Protocols - *Applications*.

General Terms
Design, Reliability, Standardization.

Keywords
iSCSI, RDMA, iSER, DA, DI, iWARP, SCSI, RDMAP, DDP, MPA, Datamover, Verbs.

1. Introduction
The iWARP protocol suite provides Remote Direct Memory Access (RDMA) semantics over TCP/IP networks. The iSCSI Extensions for RDMA (iSER) is a protocol that maps the iSCSI protocol over the iWARP protocol suite. This paper analyzes some of the key challenges faced in designing and integrating iSER into the iWARP framework while meeting the expectations of the iSCSI protocol.. As part of this, the paper discusses the key tradeoffs and design choices in the iSER wire protocol, and the role the design of the iSER protocol played in evolving the RNIC (RDMA-enabled Network Interface Controller) architecture and the functionality of iWARP protocol suite.

The organization of the rest of the paper is as follows. Section 2 provides an overview of the iSCSI protocol and the iWARP protocol suite. Section 3 makes a case for the design of the iSER protocol. Section 4 reviews the design tradeoffs behind the major design choices with regard to the layering of the iSER protocol. Section 5 describes some additional design issues in mapping the iSCSI protocol to fit into the iWARP framework. Section 6 describes the extensions to and expectations on iSCSI to harmoniously fit into the iSER and iWARP framework. Section 7 describes the enhancements that were inspired by the iSER protocol and made in the iWARP architecture to better accommodate I/O protocols such as iSER. Section 8 offers the authors' conclusions from this design effort.

2. iSCSI and iWARP

2.1 iSCSI: The Mapping of SCSI over TCP
The iSCSI protocol [4] is a mapping of the SCSI remote procedure invocation model [5] over TCP [1]. SCSI is based on the client-server architecture. Clients in the SCSI model are called "initiators", which issue SCSI "commands" to request services from a server known as the "target". The target is responsible for initiating and pacing the solicited data transfer followed by reporting the completion status back to the initiator. Commands and data are transferred between the initiators and the targets in the iSCSI model in the form of iSCSI PDUs (Protocol Data Units) sent over the TCP/IP network.

ACM SIGCOMM 2003 Workshops, August 25&27, 2003, Karlsruhe, Germany.

The data transfer from the target to the initiator (also referred to as the Read data) is realized in the iSCSI model via a series of SCSI Data-in PDUs. The iSCSI model assumes that the initiator has the necessary buffers ready for receiving the Read data at the time of issuing the SCSI command itself. The iSCSI specification also optionally allows the targets to "collapse" the status PDU into the last SCSI Data-in PDU in the case of a successful command completion – this feature is sometimes also referred to as "phase collapse" since it collapses the data phase and the status phase into one.

The data transfer from the initiator to the target (also referred to as the Write data) is realized in the iSCSI model via two discrete steps – a) an R2T (Ready To Transfer) PDU that announces target's readiness to receive a certain amount of data as specified in the R2T PDU, and b) a series of SCSI Data-out PDUs containing the Write data from the initiator to the target, in response to the R2T PDU. The iSCSI data transfer model also allows the targets and initiators to optionally negotiate an "unsolicited" data transfer up to a limit from the initiators to the targets that skips the first discrete step (R2T PDU) earlier noted.

2.2 iSCSI and TCP: Strengths and Constraints

As an application protocol on top of the TCP, iSCSI enjoys TCP's Internet-friendly congestion control, time-tested reliable transport protocol features, ubiquity and familiarity to name a few. However, it turns out that as an application protocol needing to transact a high volume of storage data transfers, iSCSI also faces certain constraints inherent to TCP's design and usage. The rest of this section analyzes these issues in more detail and describes how the iSCSI protocol addressed them.

The TCP/IP processing overhead and the copy overhead in the TCP/IP stack [2], especially in the receive path, is a well-known problem that affects the utilization of CPU(s) and memory subsystem and prevents scaling to higher speeds. In the iSCSI model, each iSCSI PDU (but not every TCP segment) is self-describing in that it contains data placement information for the payload in the iSCSI header. This allows an implementation which integrates the iSCSI function with a TCP/IP offload engine to offload TCP/IP processing and place commands and data directly in the main memory of the host or storage controller based on the placement information in the iSCSI headers.

Packet reordering [3] is another problem that plagues the traditional TCP/IP stack implementation and requires reassembly of out-of-order TCP segments. Since each TCP segment is not likely to contain an iSCSI header and TCP itself does not have a built-in mechanism for signaling ULP message boundaries to aid in the placement of out-of-order segments, an end node requires a large amount of buffering (typically of the order of TCP bandwidth delay product) to store out-of-order TCP segments, reconstruct the original byte stream and reassemble the iSCSI PDUs. Only then can its iSCSI layer place the data in the iSCSI buffers.

This TCP reassembly at high network speeds is quite counter-productive. If the reassembly function is implemented on a NIC (Network Interface Controller) or HBA (Host Bus Adapter), then the amount of reassembly memory required scales linearly with the bandwidth delay product. At high network speeds, the amount and cost of this memory is likely to create an obstacle to wide deployment. On the other hand, when reassembly is done on the host, it creates wasted memory bandwidth and CPU cycles in data copying. The "RDMA over IP Problem Statement" draft [13] makes a compelling case that the memory subsystem of a server will become the system bottleneck with multiple copies for 10Gb/sec speeds. Even at lower speeds, memory bus utilization is very high for multiple copies, potentially at the expense of other consumers. Furthermore, TCP reassembly - either performed in the NIC memory or in the host memory - suffers from an additional general store-and-forward latency from the application perspective.

To further facilitate locating the iSCSI PDU boundaries in out-of-order TCP segments in order to place data directly and to ease the TCP reassembly buffer burden, iSCSI defined the "sync and steering layer" in the architecture. iSCSI allows using optional fixed interval markers as a framing mechanism for the iSCSI PDUs within the TCP byte stream. When used, the markers can reduce the size of the reassembly buffer but cannot eliminate it, as memory size scales with number of TCP connections supported. Being optional, the markers are not guaranteed to be present in the TCP byte stream. Furthermore, since an iSCSI header is not always present in every TCP segment, an end node must still provide reassembly buffers for out-of-order TCP segments for which the iSCSI header has not yet been located.

2.3 iWARP: The Promise of RDMA

Providing relief to the TCP receive-path copy overhead problem and the reassembly buffer requirement are the motivations behind the iWARP protocol suite. The iWARP protocol suite provides the TCP framing support (Markers PDU Aligned Framing, MPA [6]), the Direct Data Placement support (DDP [7]) and the Remote Direct Memory Access support (RDMAP [8]) in the form of a generalized abstraction at the boundary between the Upper Layer Protocol (ULP) and the transport layer.

DDP/MPA allows for a drastic reduction of the TCP reassembly buffer to a size that can be implemented in an on-chip memory. The "Analysis of MPA over TCP operations" [12] provides a comparative discussion of the

reassembly buffer requirements for TCP depending on framing solutions adopted. The draft presents analysis demonstrating that MPA allows reassembly buffer size to scale as a function of maximum segment size rather than as a function of bandwidth delay product. Another important benefit of DDP/MPA is that it ensures the presence of an iSCSI PDU (or more generally, any upper layer protocol PDU) at a fixed offset in the first DDP segment of a DDP Message. If the location of the iSCSI PDU had no relationship to the TCP segment boundaries, the receiver complexity is largely increased. The "Analysis of MPA over TCP Operations" [12] also provides an illustration for the benefit of header alignment in streamlining the DMA operations and computational load of the receiver.

The iWARP protocol suite is implemented in an RNIC which is capable of providing the direct data placement function without requiring firmware or hardware customizations in the RNIC for each ULP. The iWARP protocol suite is depicted in Figure 1.

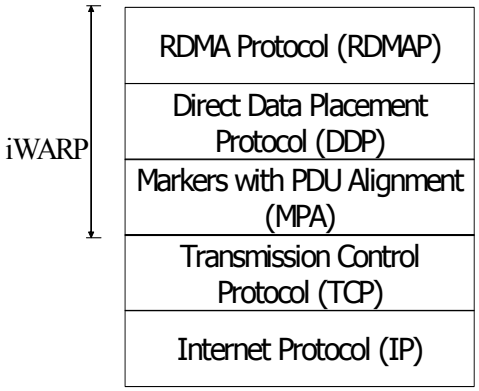

Figure 1 iWARP protocol suite over TCP/IP

The primary unit of information exchange on an iWARP connection is a DDP Segment that is self-describing from a data placement perspective. The MPA framing support is used by the RNIC to locate the DDP control information of the current DDP segment in the TCP stream, whether or not the TCP packets arrive in order. The DDP control information supports the direct data placement and enables the RNIC to steer the payload to its final memory location. The RDMA support is provided in the RDMAP control information and it enables the ULP to specify RDMA Read, RDMA Write, or Send semantics to transfer the data.

A ULP interacts with an RNIC via the RNIC Interface (RI) using a set of functional interfaces called "Verbs" [9]. From this perspective, the RI is also called a "Verbs provider" and the ULP a "Verbs consumer". Verbs, covering various functional aspects of the RI, are defined in the iWARP architecture. The Verbs consumer accesses the RI by creating one or more Queue Pairs (QPs), each of which consists of two Work Queues (WQs): a Send Queue (SQ) and a Receive Queue (RQ) as shown in Figure 2. This QP is associated with a TCP connection (one to one mapping) by the Verbs consumer for carrying out the send and receive operations. Each request to the RI by the consumer takes the form of a Work Request (WR) that the consumer posts to the SQ or RQ as appropriate by invoking Verbs in order to convey the request to the RI. All outbound RDMA operations such as RDMA Read, RDMA Write, and Send are initiated via WRs posted to the SQ. All inbound solicited data carried by the RDMA Write and RDMA Read Response messages is stored by the RNIC directly into the ULP buffer(s). The control and un-solicited data operations such as receiving incoming Send Messages are satisfied via Work Requests posted to the RQ. Each WQ (SQ or RQ) is associated with a Completion Queue (CQ) that notifies the Consumer of the completion of the requested operation via a Work Completion (WC).

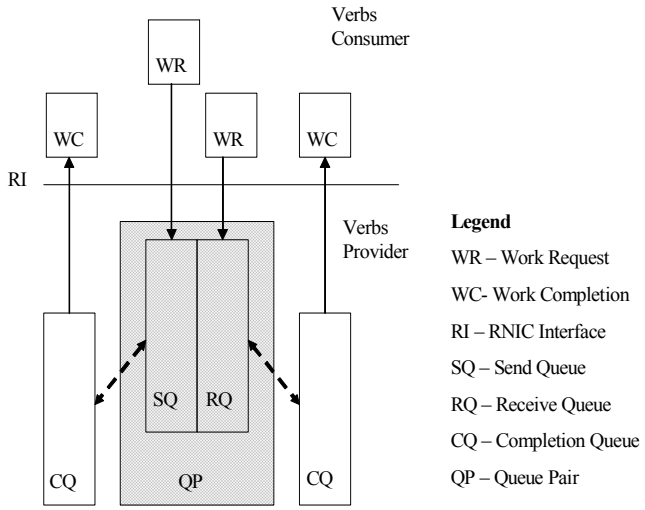

Figure 2 iWARP functional interface

As RNICs support a diversified set of applications, the use of RNICs promotes the convergence on the server hardware and networking requirements. As such it creates larger economic thrust for creating a larger RNIC market than several application-specific markets for iSCSI NICs, IPC NICs, TOE NICs etc. An RNIC employs the iWARP protocol suite defined by the RDMA Consortium that have been accepted as drafts by the IETF Remote Direct Data Placement Working Group (RDDP WG) and are expected to be supported by a broad list of hardware and software vendors.

3. The Case for iSER

Recalling the constraints that were imposed on iSCSI as summarized in section 2.2 and the promise of iWARP as summarized in section 2.3, it is evident that there is a natural convergence of interests between iSCSI and iWARP. The iSCSI Extensions for RDMA (iSER) protocol is borne out of a desire to take advantage of this convergence. Specifically, the iSER protocol design is motivated by the following –

- To have the generic data movement needs of iSCSI be met with the iWARP protocol suite so that the advances in the RDMA technology can continue to naturally improve iSCSI in most efficiently delivering data.
- To have the iWARP protocol suite address the generic issues (yet of crucial importance to iSCSI) such as data placement and copy elimination in high-speed data transfer over TCP/IP, rather than tackle those via iSCSI-specific protocol means such as markers.
- To relieve iSCSI implementations from having to take on transport functionality such as PDU digests, timeouts and retransmissions that iSCSI was forced to incorporate into itself (the data integrity assurance provided by the CRC that is part of the MPA layer in the iWARP protocol suite helps this goal).
- To allow iSCSI to optimally operate in conjunction with the RDMA over TCP/IP technology since RNICs are expected to become pervasive and inexpensive in future, much like the standard ethernet NICs of today.

To summarize, the iSER protocol is designed as a mapping of the iSCSI protocol on to the iWARP protocol suite – i.e. transforming or encapsulating iSCSI PDUs into appropriate RDMAP Messages.

The iSER protocol translates certain iSCSI control and data PDUs into RDMAP messages that can be efficiently handled by a generic RNIC so as to not require a specialized iSCSI HBA. Use of a generic RNIC promotes I/O convergence on the Data Center's server. Such a server can now employ an RNIC that supports both networked storage and interprocess communications (IPC) and potentially other standard sockets applications.

4. iSER Layering – Over DDP or RDMAP?

One of the first design choices in mapping iSCSI over iWARP related to whether iSER should be layered on top of the DDP layer or the RDMAP layer in the protocol stack shown in Figure 1. While the DDP layer provides the data placement functionality adequate for the placement of the SCSI data, the RDMAP layer provides additional RDMA semantics including RDMA Read, RDMA Write and Send that enable iSER, and thus iSCSI, to run on generic RNICs at comparable performance levels to that of iSCSI NICs.

The main reason for deciding on the iSER mapping over RDMAP instead of DDP is to preserve the "single interrupt per I/O" data transfer model that the initiators have come to expect from the I/O adapters, even while using generic RNICs. Traditionally, both parallel SCSI [10] and Fibre Channel for SCSI [11] adapters have delivered this model to initiators, in order to minimize the host interactions per I/O. If iSCSI were mapped over the DDP layer, the R2T PDU (see section 2) soliciting Write data must be handled by the iSCSI layer (outside of the RNIC, since the RNIC is iSCSI-agnostic) at the initiator. In addition, iSCSI does not place a limit on the number of R2Ts that a target may issue to complete one SCSI Write command either – each one of which would require a host processor interaction on the initiator, all in the span of one Write command.

On the other hand, if iSCSI is mapped over the RDMAP layer, an R2T PDU can be transformed into an RDMA Read Operation on the target side in the iSER layer. The RDMA Read Operation can be completely handled internally by the RNIC on the initiator. Thus, layering the iSER protocol on top of RDMAP can preserve the "single interrupt per I/O" model even with generic RNICs. Furthermore, an iSER-over-RDMAP layering minimizes the interactions between the iSCSI/iSER layers and the RDMAP layer during SCSI data transfer, and eliminates the segmentation of SCSI data that should happen at the iSCSI/iSER layers in response to receiving an R2T PDU.

The second major reason for mapping iSER over RDMAP instead of DDP was to not further complicate the generic RNIC design by requiring it to expose two sets of interfaces - one for DDP and one for RDMAP. Verbs (see section 2.3) have been developed that describe the functionality of the RNIC Interface. Layering iSER over DDP would have meant that a different set of Verbs would be required for the DDP interface.

Figure 3 depicts the relationship between SCSI, iSCSI, iSER, and iWARP layers: RDMAP, DDP, MPA, and TCP. The TCP layer provides the byte-stream, full-duplex, reliable delivery service. The MPA layer provides framing and stronger data integrity above TCP. The MPA layer has a containment and alignment property that further enables the DDP layer to have self-describing segments for MPA-aware TCP. The DDP layer provides the segmentation/reassembly of RDMAP/DDP messages, direct data placement, and in-order delivery of messages. For MPA-aware TCP [6], the DDP layer can perform direct data placement for both in-order and out-of-order data. The RDMAP layer provides the RDMA semantics and supports RDMA Read, RDMA Write, and Send Type operations.

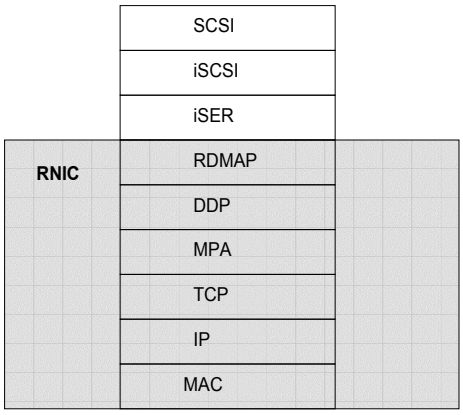

Figure 3 iSCSI/iSER layering in Full Feature Phase

5. The Design of iSER

In this section, we describe three major issues that were encountered in defining a protocol that maps the iSCSI protocol on to the iWARP protocol suite. The challenges addressed here are:

- Compatibility with iSCSI in connection establishment
- Transformation vs. encapsulation of iSCSI PDUs
- Data integrity requirements on iSCSI and iSER

5.1 iSCSI/iSER Connection Setup

One important criterion in the design of iSER is to ensure backward compatibility with iSCSI and to minimize the impact to the existing iSCSI infrastructure (boot, authentication and discovery). Therefore, by using the same well-known port as that defined for iSCSI and using the standard iSCSI login phase with normal TCP connection, an existing iSCSI node is able to engage in a login exchange with an iSER-enabled node. As a consequence of this, there is no change to the standard iSCSI discovery protocol and the existing boot process can still operate without having to engage an RNIC.

During the iSCSI connection setup, the iSCSI layer at the initiator is responsible for establishing a TCP connection to the target using the iSCSI well-known port, or a different port discovered using the standard discovery mechanisms. After the TCP connection is established, the iSCSI layers at the initiator and the target enter the login negotiations using the same rules and operational keys as outlined in the iSCSI specification. Transition to iSER-assisted mode occurs following a successful login negotiation between the initiator and the target in which iSER support is negotiated and the necessary RDMA resources have been allocated. iSER support is negotiated by using a newly defined iSCSI key -"RDMAExtensions". An existing iSCSI node that does not support iSER will not recognize this new key and so the negotiation to use iSER on this connection will fail. The connection then defaults to the iSCSI mode. This ensures backward compatibility with existing iSCSI installations.

An iSCSI session may consist of one or more logically related iSCSI connections acting together to present one "connection" (formally called an "I_T nexus" in [5]) between a SCSI initiator port and a SCSI target port. To simplify SCSI task failover from one connection to the other within an iSCSI session (that the iSCSI specification allows), the applicability of the RDMAExtensions key, which negotiates the iSER support, is made session-wide. So if the "leading connection" (the first iSCSI connection that initiates an iSCSI session) of an iSCSI session supports iSER-assisted mode, then all other connections of that session support iSER-assisted mode.

5.2 Transformation vs. Encapsulation

Since the iSCSI layer operating on an iSER layer is mostly oblivious (except for negotiating the RDMAExtensions login key) to the presence of the iSER layer operating underneath, it may generate any legal iSCSI PDU defined in the iSCSI specification. The best way to deal with each of these PDUs had to be decided during the design of iSER – some of those PDUs needed to be simply encapsulated in iWARP Send Message Types, while some others needed to be transformed into RDMA Reads and RDMA Writes in order to be placed directly in the SCSI buffers. Note that this is a rather unique problem faced by iSER in that a protocol in a multi-protocol stack traditionally simply encapsulates the ULP PDU in its protocol headers, but that simplistic approach of "encapsulating always" in the case of iSER does not achieve the goal of direct data placement of SCSI data.

The design of iSER adopted a fairly straightforward approach to address this question. Those iSCSI PDUs that cause the SCSI data to be moved between the initiator and the target are labeled as "iSCSI data-type PDUs". All other possible iSCSI PDUs are labeled as "iSCSI control-type PDUs". The iSCSI data-type PDUs are transformed by the iSER layer into either RDMA Read or RDMA Write operations. The one exception to this is the SCSI Data-out PDU carrying the so-called "unsolicited data" (see section 2), which is classified as an iSCSI control-type PDU for the simple reason that unsolicited data cannot be placed directly into the SCSI buffers by definition (the data is technically unsolicited by SCSI). All control-type PDUs are simply encapsulated in RDMAP Send Message Types

so that the iSCSI PDU may be delivered as-is on the other end in each case.

The iSER layer on the target applies the following transformations to the two defined iSCSI data-type PDUs: an R2T PDU (see section 2) is transformed into an RDMA Read operation so that the SCSI Write data may be retrieved from the initiator directly into SCSI buffers on the target; a SCSI data-in PDU (see section 2) is transformed into an RDMA Write operation so that the SCSI Read data may be directly placed into the SCSI buffers on the initiator.

5.3 iSER Data Integrity

iSCSI offers separate header and data digests that are intended to provide end-to-end data protection while allowing intermediate devices to offer additional functionality by changing the iSCSI PDU headers (e.g. storage virtualization boxes). The MPA protocol, part of the iWARP protocol suite, provides a single CRC covering the whole PDU, including both the header and the data area. An analysis of these two approaches led to the conclusion that the single CRC in MPA offers the same level of protection as separate header and data digests. The following paragraph summarizes the analysis.

MPA, like iSCSI, uses the CRC-32c algorithm. It provides end-to-end protection while both the initiator and the target are within the iWARP fabric. When either the initiator or the target resides outside of the iWARP fabric, the CRC is replaced as part of the protocol conversion operation. If the protocol conversion box cannot be trusted to protect the data, the data may reach the destination corrupted. Similar exposure is exhibited by a middle box (e.g. a storage virtualization box) in the traditional iSCSI model employing two digests. The middle box may change the content of the iSCSI header in the process of creating a new CRC for it. In such a case, though the data was protected by a separate CRC, the whole PDU can be sent to an incorrect destination, thereby creating data corruption. The crux of the matter is the trust placed into any intermediate box determines the functionality of the digest coverage scheme. Therefore, the CRC provided by the MPA protocol was deemed adequate for iSCSI/iSER and no additional digest was required at the level of iSCSI. Furthermore, when used on top of an RNIC, neither the iSCSI layer nor the iSER layer participates in the data movement, let alone verifies the digests. Therefore, both the HeaderDigest and the DataDigest are negotiated to “None” for iSER-assisted mode.

By relying on the CRC protection offered by the MPA layer in place of header and data digests means that in iSER-assisted mode, iSCSI does not incur digest errors.

6. Extensions to iSCSI

Designing the iSER protocol that allows iSCSI to map to RDMAP requires a certain amount of adaptation of iSCSI itself. Some of these extensions can be viewed as primarily being made in the iSCSI *implementations* rather than in the iSCSI protocol. Others are enhancements added to the iSCSI protocol in order to support the iWARP protocol.

The required extensions to iSCSI included the following:

- Generalize the iSCSI to iSER interface for the benefit of any future Datamover protocol (such as iSER)
- Extensions that are added to the iSCSI protocol to support the iWARP protocol suite
- Constraints/expectations that are placed on the iSCSI protocol to support the iWARP protocol suite

6.1 Extensions to the iSCSI Interface

The iSCSI protocol was written to interface directly with the TCP layer which is responsible for transporting iSCSI PDUs to the peer iSCSI node. During the design of iSER, there was a strong desire to generalize and extend the functionality required of the transport layer so as to allow for the operation of iSCSI protocol on other Datamover protocols in the future. This desire led to the definition of a Datamover layer that provides all data transport functionality to the iSCSI layer. Besides data transfer, a Datamover layer also provides functions such as placing SCSI data into the data buffers or picking up SCSI data for transmission, etc. One such Datamover is the iSER protocol layer operating in conjunction with the iWARP protocol suite. In layering iSCSI over the iSER layer, it is desirable to minimize the impact to existing iSCSI code. On the other hand, the iSCSI layer needs to be cognizant of the presence of the Datamover layer so that control and data information can be exchanged effectively. In order to minimize the effort for developers and make iSCSI amenable to interfacing with other Datamover layers in the future, the architecture of the iSCSI and the Datamover layers was defined as a generalized Datamover Architecture (DA). The architecturally required set of interactions between the iSCSI layer and the Datamover layer are also defined in the Datamover Architecture, under the name of Datamover Interface (DI).

DI defines an abstract procedural interface of the interaction between the iSCSI layer and the underlying Datamover layer. It relies on a defined set of operational primitives provided by each layer to the other in order to carry out the request-response interactions. It models the architecturally required minimum interactions between an operational iSCSI layer and a Datamover layer to realize the iSCSI-transparent data movement.

Of the operational primitives defined in DI, some are used for specifying control and data transfers for iSCSI PDUs,

while others are defined to allow the iSCSI layer to control the RDMA functionality in iSER. All operational primitives are generic in nature in that they can be used for any Datamover and not specifically for the iSER layer.

6.2 Extensions to the iSCSI Protocol

During the iSER design phase, each extension to the iSCSI protocol was made only when it was backed by a compelling rationale. The following extensions significantly enhanced the functionality of the iSER-assisted mode and thus were deemed necessary..

- As summarized in section 5.1, there was a desire to be able to completely leverage the full iSCSI infrastructure (boot, authentication and discovery) via relying on original iSCSI architectural elements to bootstrap into the iSER mode. In keeping with this desire, a new key, RDMAExtensions, was added to allow the iSCSI peers in a login negotiation to determine if both sides support the iSER-assisted mode. By negotiating the iSER support during the login negotiation, the backward compatibility for an iSER-assisted node is provided so that it may interoperate with an existing iSCSI node by negotiating on the connection in the traditional iSCSI mode.
- Each iWARP Send Message Type, big or small, consumes a full posted receive buffer on the respective Receive Queue (see Figure 2). Requiring the sender to send PDUs with full-sized segments whenever possible obviously improves the utilization of buffering resources at the receiver. This realization drove the creation of two new iSCSI keys called InitiatorRecvDataSegmentLength and TargetRecvDataSegmentLength that require the initiator and the target to generate full-sized iSCSI control-type PDUs (except possibly the last PDU in a PDU sequence).

6.3 Expectations on the iSCSI Protocol

In order to support iWARP, certain features in the iSCSI protocol are not allowed, or strongly discouraged. Note that some of these restricted features are optional in the iSCSI protocol anyway.

- As discussed in section 5.3, the MPA CRC satisfies the data integrity expectations of iSCSI. For this reason, the HeaderDigest and DataDigest keys are negotiated to "None".
- In the iSER/RDMAP model, the iSCSI layer is not privy to data movement, so in general it cannot possibly be "expecting" the next data PDU to be received at any time. In view of this, ExpDataSN (a sequential data sequence number that informs the target what the initiator is expecting to arrive next) is set to 0 in reassigning tasks from a failed iSCSI connection to an operational iSCSI connection within the same session.
- The phase-collapsed (see section 2.1) PDU contains Read data to be transformed as well as status to be encapsulated while iSER's policy is to choose either transformation or encapsulation (see section 5.2) in order to facilitate direct Read data placement. The decision to use RDMA Write for Read data does not allow the status information and Read data to be combined in a single RDMAP Message. For this reason, phase collapse is not used while returning SCSI status for a SCSI Read command.
- Digest errors are by definition non-existent in iSCSI/iSER because digests are not used in the iSER-enabled mode of iSCSI. This led to the recommendation that SNACKs and iSCSI-level command retries should not be used by initiators. Both these features are essentially PDU recovery mechanisms meant to recover iSCSI PDUs lost due to digest errors at the iSCSI layer.

7. iWARP Enhancements for iSER

After having made the significant design choice in the early stages of the design that the iSER protocol is to be layered on top of the RDMAP protocol, the designers of iSER focused their attention on potential features that could be added to the RDMAP protocol (that also was concurrently in its final stages of development) and iWARP Verbs (see section 2.3) to further enhance the envisioned iSER data transfer model. The crucial design challenge here was to identify the iWARP protocol features that will significantly benefit an iSER implementation (and actually many other applications as well) running on a generic RNIC, even while those iWARP protocol features themselves are completely iSER-agnostic. A similar design challenge awaited the iSER design process in proposing a set of iSER-agnostic Verbs features that would immensely benefit iSER implementations.

A Steering Tag (STag) is an RNIC-unique identifier for an I/O buffer and it is used in a SCSI I/O operation. The iSER layer at the initiator advertises the STag(s) for the I/O buffer(s) of each SCSI command to the iSER layer at the target unless the SCSI command can be completely satisfied by unsolicited data alone. Once the I/O operation is completed, remote access to the buffer is to be disabled so that further inadvertent remote access of the memory associated with the STag cannot be made while the memory is in use. Some applications require many small SCSI Reads and SCSI Writes. Therefore, it is essential that the overhead of registering, advertising and invalidating STags be minimized to allow efficient operation. Consequently, a principal focus during the iSER designer process was to fine-tune the iWARP features that had to do with one of these STag management aspects.

To summarize the iWARP architectural innovations made to help the iSER protocol, there are fundamentally two categories:

- Enhancements made in iWARP wire protocols
- Enhancements made in iWARP Verbs

The following sections describe each of these innovations in more detail.

7.1 Enhancements in iWARP protocols

This section describes the iWARP protocol innovations which enable a highly efficient invalidation of STags at the initiator. At the end of an SCSI Read or SCSI Write, the STag at the initiator is made inaccessible to the target by invalidating the STag(s) used in that I/O operation. The Send with Invalidate (SendInv) and Send with Invalidate and Solicited Event (SendInvSE) RDMAP Messages were created to allow an iSER target to cause an efficient invalidation of the STag on the initiator. The iSER layer at the target uses a SendInvSE Message which auto-invalidates the STag at the initiator to encapsulate the SCSI Response delivered after the conclusion of the data transfer, thus cleanly invalidating the advertised STag on the initiator without requiring a separate local operation at the initiator.

The notable thing in this model is that it aligns quite nicely with the transaction-oriented nature of SCSI I/O operations. The target sends the status in an SCSI response PDU at the end of each SCSI Read or SCSI Write, thus concluding the transaction. Whenever an STag was associated with the SCSI Read or SCSI Write, the iSER layer at the target can use a SendInvSE to send the SCSI response PDU and invalidate the STag at the initiator. When the RDMAP layer at the initiator receives a SendInv or SendInvSE message from the target, the message automatically invalidates the STag carried in the iWARP header of the message.

Bidirectional SCSI commands are a special case because they may have two STags, viz. one for the read I/O buffer and one for the write I/O buffer. A bidirectional SCSI command always has a Read STag. It will also have a Write STag unless none of the data was solicited. For bidirectional SCSI commands, the associated Read STag is invalidated via the SendInvSE sent by the target. The initiator is responsible for invalidating the Write STag locally if one was used.

7.2 Enhancements in iWARP Verbs

The innovations in iWARP Verbs made for iSER were primarily of two categories - the aforementioned fine-tuning of STag management Verbs, and creating better buffer management techniques.

7.2.1 Efficient Invalidation of STags at the Target

The iSER layer at the target does not explicitly advertise any STags in the course of completing a SCSI Read or Write operation. However, the iSER layer at the target exposes its local STag of the Data Sink in an RDMA Read operation. In order to efficiently invalidate that local STag automatically without target-local RNIC interactions, a new type of Work Request (see section 2.3) was proposed during the design of iSER that would significantly benefit the targets. The iSER layer at the target is expected to use this new Work Request called the "RDMA Read with Invalidate Local STag" Work Request. It combines two work requests, RDMA Read, and Invalidate Local STag, into one. At the conclusion of the RDMA Read operation, the RNIC at the target will automatically invalidate the local STag used in the RDMA Read operation, thus saving a round trip to the hardware for the iSER layer.

7.2.2 Efficient Registration of STags

Typically, an I/O buffer is described by the length of the buffer, a list of starting physical addresses of fixed sized blocks, and an initial offset into the first block. Memory registration traditionally allowed the Verbs Consumer to register an I/O Buffer in order to enable the RNIC to directly access the I/O Buffer via an RNIC-construct called a Memory Region (MR). An STag is returned to Verbs Consumer at the end of the memory registration process. The RNIC maintains a Translation and Protection Table (TPT) that contains the data structures to control buffer access and translate the pair consisting of an STag identifying a buffer and Tagged Offset (TO) within the buffer into a local memory address directly accessible by the RNIC.

In iWARP, the standard synchronous memory registration Verb allowed the Consumer to register a memory region with the RNIC and obtain an STag. This verb combines the allocation of TPT resources and initializing those resources in corresponding table entries. Due to the synchronous nature of this verb and the overhead of the memory registration operation, it was deemed undesirable during the iSER design process to have the memory registration in the critical path of the data transfer. For iSCSI on the other hand, the mapping of an I/O buffer to physical memory locations is known only during the critical path of the command execution.

To accommodate the iSCSI needs, the RDMA verbs enhanced the memory registration mechanism by adding allocation and "fast-register" Verbs – these two new Verbs essentially are the two discrete steps of the traditional integrated memory registration Verb. The fast-register is an asynchronous memory registration operation that fills in the TPT data structures associated with a memory region that was pre-allocated to an STag. This allows the iSER Consumer to use pre-allocated STag in performing memory

registration operations in the critical path of the command execution via fast-register. Since the required RNIC resources are allocated in a discrete step outside of the critical path and so the value of the STag is known to the iSER layer, the iSER layer does not have to wait for the result of the fast-register operation. By separating the allocation and the fast-registration steps, the iSER layer does not have to wait for the result of the registration when it needs to post work requests to the work queue – i.e. a "synchronous wait" model has been transformed into a "asynchronous post and complete" model. This is a significant performance win for iSER implementations.

7.2.3 Overhead Analysis

The efficient invalidation of STags at the initiator and the target and the efficient registration of STags result in the reduction in the processing overhead per command. The use of SendInvSE to carry SCSI Response PDU from the target to the initiator eliminates an explicit invalidation of the STag per command on the initiator. This results in the savings of one work request and one work completion per command on the initiator. Similarly, the use of "RDMA Read with Invalidate Local STag" work request on the target results in the savings of a work request and a work completion for invalidating the local STag per R2T request. Finally, the fast-register and invalidate STag work requests allow both the initiator and target, to avoid the overhead of the synchronous memory registration/deregistration operations, and efficiently pipeline asynchronous memory registration/STag invalidation operations with other SQ operations.

7.2.4 STag Re-use

Allocated STags may be frequently re-used. It is therefore desirable to provide protection against protocol errors that might cause data from an old operation to be written into an STag that has been re-used by a current operation. To provide this protection, the 32-bit STag is partitioned into an 8-bit key and a 24-bit index. The index is assigned by the RNIC when the STag is allocated. When the iSER layer registers an STag, the iSER layer identifies the key by the index and supplies a value for the key. Access to the advertised buffer and invalidation of the STag, however, is checked against the full 32-bit STag. By a judicious cycling through the 8-bit key values, the iSER layer can greatly contain the exposure from broken iSER peers that illegally try to make a remote access with an STag that was advertised to them sometime ago.

7.2.5 The Buffer Management Problem

Buffers for send operations are not advertised. A receive queue (RQ) of buffers is posted to the RNIC for receiving Send messages. In previous RDMA architectures such as Infiniband, each connection had its own queue of receive buffers.

For iSCSI, there may be many connections and usage of each connection may be bursty. Providing sufficient buffers in a peak operation of each connection in separate RQs would require an excessive amount of memory. This is exacerbated by the fact that the iSCSI architecture doesn't specifically limit certain types of control messages (e.g. task management PDUs) the initiator can send to the target.

Shared receive queues (S-RQ) were invented to provide a more efficient way of providing the required buffers across multiple iSCSI/iSER connections. Described differently, the S-RQ was deemed to be the most efficient solution to the "free space flow" problem across the connections sharing it.

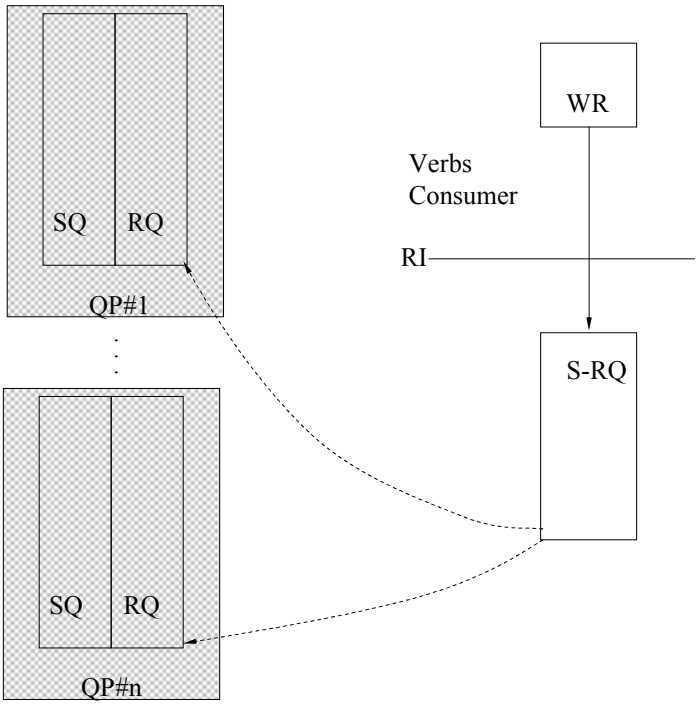

Figure 4 The S-RQ Operational Model

Figure 4 shows the logical operational model of an S-RQ. It can be seen that the S-RQ is acting like a repository of Work Requests and a Verbs Consumer (such as iSER) is posting Work Requests only to it. Two Queue Pairs (marked as QP#1 and QP#n) shown in the picture are sharing the S-RQ, thus "drawing" the Work Requests (with the receive buffer locations) as each Queue Pair requires a new receive buffer. In this operational model, there is no practical limit on the number of Queue Pairs that can share one S-RQ.

7.2.5.1 iSER and S-RQ

Before creating the notion of S-RQ, an iSCSI or iSER level flow control on receive buffers was considered as a way to reduce memory requirements. In iSCSI, the target provides a command window to the initiator. This provides a limited form of flow control by limiting the number of outstanding command PDUs. However, there are other PDUs that use send type operations such as Data-Out PDUs for non-immediate unsolicited write data and immediate commands (i.e. not accounted for by the iSCSI command window) that are not strictly flow controlled by iSCSI on a PDU basis. Because of these reasons, the iSCSI command window does not provide the positive flow control to resolve the memory usage issues of separate receive queues.

The iSER architects have considered designing a flow control mechanism as part of the iSER protocol. Such a flow control mechanism could allow a quiet connection to be provisioned with just a few buffers. When the connection becomes active, more buffers can be provisioned and notice of the credit can be sent to the peer. This would resolve the buffer provisioning problem but has significant disadvantages. When a connection transitions from quiet to active, there would be a delay while buffers are provisioned and flow control messages are changed before it can operate at full speed. When a connection transitions from active to quiet, it may have many buffers in its RQ and a method would need to be provided to reclaim the credits and free the buffers. This complicates the flow control mechanism, and thus the whole iSER protocol.

Because of these disadvantages, and given that implementations can have the option of using S-RQ to considerably alleviate the problems, it was decided to not design in the Send Message flow control into the iSER protocol. This decision significantly simplified the iSER protocol design, besides making it quite similar to other ULPs that run on TCP/IP and do not enforce precise flow control.

7.2.5.2 Shared Receive Queue (S-RQ)

An S-RQ provides a pool of buffers that can be used by many connections. When a Send Message Type arrives on one of the connections, the connection pulls a buffer from the S-RQ to hold the received message. Buffers can be provisioned to the S-RQ to handle the expected load across the aggregated connections. This is similar to the way that many Fibre Channel and non-RDMA iSCSI implementations provide buffering.

The S-RQ has a low-water mark. When the RNIC detects that the available buffers have fallen below the low-water mark, it will alert the ULP (iSER). The ULP can then supply more buffers. This low-water mark can also provide an early detection of an attack trying to deplete buffers from the S-RQ.

Even if many connections simultaneously become active, the link rate limits the rate at which messages can arrive. One way of deciding how many buffers to provision is to consider the link rate and the time it takes to respond to a request for more buffering.

7.2.5.3 Comparison of Queue Memory Requirements

This section analyzes the memory requirements for implementations using RQs versus S-RQs. Consider an RNIC that is handling 1000 iSCSI connections. Let us further assume that the iSCSI connections have negotiated to allow 64 Kbytes of unsolicited data to be sent for each SCSI Write command and the maximum data size allowed in each PDU is 8 Kbytes (These are fairly typical values). Therefore, each SCSI Write command in the command window requires up to 9 PDUs.

If dedicated RQs are used and the command window is modest (e.g. 4 outstanding commands) then the target would have to provide about 40 buffers on the RQ (allowing for 4 Write commands with the full amount of unsolicited data plus some buffers for PDU types not limited by the command window). For 1000 connections, the RQs would then contain 40,000 buffers of 8,252 bytes each (8K bytes of data plus headers). This is a total of more than 330 Mbytes.

If S-RQs are used, one would want to provide enough buffering to allow time to provision more buffers even when minimum size PDUs are arriving at link rate. At 1 Gbit/s, it takes 1.26 micro seconds to receive a minimum size iSCSI PDU. This includes Ethernet framing overhead and headers for IP, TCP, MPA, DDP, iSER and iSCSI. If the time to provision more buffers is 1 ms, then the low-water mark would need to be set at about 1000 buffers. The S-RQ would require about 8 Mbytes for a 1 Gbit/s link and about 80 Mbytes for a 10 Gbit/s link.

8. Conclusion

In summary, we have described some of the challenges and the issues that the protocol designers faced in architecting the iSER protocol. The authors believe that the goal of minimizing impacts to the iSCSI code and keeping the rest of the iSCSI infrastructure (boot, authentication and discovery) intact was fully met in the iSER design, even while enabling iSCSI to take advantage of RDMA capabilities. The iSER protocol was designed to allow iSCSI to use the direct data placement and RDMA capabilities using a generic RNIC. Use of a generic RNIC promotes convergence on the Data Center's server. Such a server can now employ an RNIC that supports both networked storage and interprocess communications (IPC) and potentially other standard sockets applications. The iSER design effort has also enhanced the iSCSI architecture by generalizing the architectural framework for functionality distribution between iSCSI and iSER so other

types of Datamovers may be employed in the future. The iSER design effort also significantly enriched the iWARP protocols and RDMA verbs by introducing new features and requirements that not only benefit iSER, but also other applications that will be using the RNIC in the future.

Acknowledgments

This protocol was developed by a design team that, in addition to the authors, includes the members of the storage group in the RDMA Consortium. The authors acknowledge the contributions of the entire design team.

9. References

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

[2] D. Clark et al, "An analysis of TCP processing overhead", IEEE Communications, vol. 27, Issue 6, June 1989, pp 23-29

[3] Bennett J., Partridge C., and Sheetman N., “Packet Reordering is Not Pathological Network Behavior”. IEEE/ACM Transactions on Networking, 7(6), December 1999, 789-798.

[4] J. Satran et al., "iSCSI", Internet Draft draft-ietf-ips-iscsi-20.txt, February 2003

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

[6] P. Culley et al., "Markers PDU Aligned Framing for TCP Specification", Internet Draft draft-ietf-iwarp-mpa-02.txt, February 2003

[7] H. Shah et al., "Direct Data Placement over Reliable Transports", Internet Draft draft-ietf-iwarp-ddp-01.txt, February 2003

[8] R. Recio et al., “An RDMA Protocol Specification”, Internet Draft draft-ietf-iwarp-rdma-01.txt, February 2003

[9] J. Hilland et al., "RDMA Protocol Verbs Specification", Internet Draft draft-hilland-rddp-verbs-00.txt, April 2003

[10] SCSI Parallel Interface-4 (SPI-4), http://www.t10.org/ftp/t10/drafts/spi4/spi4r10.pdf

[11] SCSI Fibre Channel Protocol-2 (FCP-2), http://www.t10.org/ftp/t10/drafts/fcp2/fcp2r08.pdf

[12] U. Elzur et al., “Analysis of MPA over TCP operations”, IETF Draft draft-elzur-tsvwg-mpa-tcp-analysis-00.txt, February 2003

[13] A. Romanow et al., "RDMA over IP Problem Statement", IETF Draft draft-ietf-rddp-problem-statement-01.txt, February 2003