Network Working Group
Request for Comments: 5040
Category: Standards Track

R. Recio
B. Metzler
IBM Corporation
P. Culley
J. Hilland
Hewlett-Packard Company
D. Garcia

October 2007

A Remote Direct Memory Access Protocol Specification

Status of This Memo

Abstract

This document defines a Remote Direct Memory Access Protocol (RDMAP) that operates over the Direct Data Placement Protocol (DDP protocol). RDMAP provides read and write services directly to applications and enables data to be transferred directly into Upper Layer Protocol (ULP) Buffers without intermediate data copies. It also enables a kernel bypass implementation.

Table of Contents

Table of Figures

1. Introduction

Today, communications over TCP/IP typically require copy operations, which add latency and consume significant CPU and memory resources. The Remote Direct Memory Access Protocol (RDMAP) enables removal of data copy operations and enables reduction in latencies by allowing a local application to read or write data on a remote computer's memory with minimal demands on memory bus bandwidth and CPU processing overhead, while preserving memory protection semantics.

RDMAP is layered on top of Direct Data Placement (DDP) and uses the two buffer models available from DDP. DDP-related terminology is discussed in Section 2.3. As RDMAP builds on DDP, the reader is advised to become familiar with [DDP].

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 RFC 2119 [RFC2119].

1.1. Architectural Goals

RDMAP has been designed with the following high-level architectural goals:

* Provide a data transfer operation that allows a Local Peer to transfer up to 2^32 - 1 octets directly into a previously Advertised Buffer (i.e., Tagged Buffer) located at a Remote Peer without requiring a copy operation. This is referred to as the RDMA Write data transfer operation.

* Provide a data transfer operation that allows a Local Peer to retrieve up to 2^32 - 1 octets directly from a previously Advertised Buffer (i.e., Tagged Buffer) located at a Remote Peer without requiring a copy operation. This is referred to as the RDMA Read data transfer operation.

* Provide a data transfer operation that allows a Local Peer to send up to 2^32 - 1 octets directly into a buffer located at a Remote Peer that has not been explicitly Advertised. This is referred to as the Send (Send with Invalidate, Send with Solicited Event, and Send with Solicited Event and Invalidate) data transfer operation.

* Enable the local ULP to use the Send Operation Type (includes Send, Send with Invalidate, Send with Solicited Event, and Send with Solicited Event and Invalidate) to signal to the remote ULP the Completion of all previous Messages initiated by the local ULP.

* Provide for all operations on a single RDMAP Stream to be reliably transmitted in the order that they were submitted.

* Provide RDMAP capabilities independently for each Stream when the LLP supports multiple data Streams within an LLP connection.

1.2. Protocol Overview

RDMAP provides seven data transfer operations. Except for the RDMA Read operation, each operation generates exactly one RDMA Message. Following is a brief overview of the RDMA Operations and RDMA Messages:

1. Send - A Send operation uses a Send Message to transfer data from the Data Source into a buffer that has not been explicitly Advertised by the Data Sink. The Send Message uses the DDP Untagged Buffer Model to transfer the ULP Message into the Data Sink's Untagged Buffer.

2. Send with Invalidate - A Send with Invalidate operation uses a Send with Invalidate Message to transfer data from the Data Source into a buffer that has not been explicitly Advertised by the Data Sink. The Send with Invalidate Message includes all functionality of the Send Message, with one addition: an STag field is included in the Send with Invalidate Message. After the message has been Placed and Delivered at the Data Sink, the Remote Peer's buffer identified by the STag can no longer be accessed remotely until the Remote Peer's ULP re-enables access and Advertises the buffer.

3. Send with Solicited Event (Send with SE) - A Send with Solicited Event operation uses a Send with Solicited Event Message to transfer data from the Data Source into an Untagged Buffer at the Data Sink. The Send with Solicited Event Message is similar to the Send Message, with one addition: when the Send with Solicited Event Message has been Placed and Delivered, an Event may be generated at the recipient, if the recipient is configured to generate such an Event.

4. Send with Solicited Event and Invalidate (Send with SE and Invalidate) - A Send with Solicited Event and Invalidate operation uses a Send with Solicited Event and Invalidate Message to transfer data from the Data Source into a buffer that has not been explicitly Advertised by the Data Sink. The Send with Solicited Event and Invalidate Message is similar to the Send with Invalidate Message, with one addition: when the Send with

Solicited Event and Invalidate Message has been Placed and Delivered, an Event may be generated at the recipient, if the recipient is configured to generate such an Event.

5. Remote Direct Memory Access Write - An RDMA Write operation uses an RDMA Write Message to transfer data from the Data Source to a previously Advertised Buffer at the Data Sink.

   The ULP at the Remote Peer, which in this case is the Data Sink, enables the Data Sink Tagged Buffer for access and Advertises the buffer's size (length), location (Tagged Offset), and Steering Tag (STag) to the Data Source through a ULP-specific mechanism. The ULP at the Local Peer, which in this case is the Data Source, initiates the RDMA Write operation. The RDMA Write Message uses the DDP Tagged Buffer Model to transfer the ULP Message into the Data Sink's Tagged Buffer. Note: the STag associated with the Tagged Buffer remains valid until the ULP at the Remote Peer invalidates it or the ULP at the Local Peer invalidates it through a Send with Invalidate or Send with Solicited Event and Invalidate.

6. Remote Direct Memory Access Read - The RDMA Read operation transfers data to a Tagged Buffer at the Local Peer, which in this case is the Data Sink, from a Tagged Buffer at the Remote Peer, which in this case is the Data Source. The ULP at the Data Source enables the Data Source Tagged Buffer for access and Advertises the buffer's size (length), location (Tagged Offset), and Steering Tag (STag) to the Data Sink through a ULP-specific mechanism. The ULP at the Data Sink enables the Data Sink Tagged Buffer for access and initiates the RDMA Read operation. The RDMA Read operation consists of a single RDMA Read Request Message and a single RDMA Read Response Message, and the latter may be segmented into multiple DDP Segments.

   The RDMA Read Request Message uses the DDP Untagged Buffer Model to Deliver the STag, starting Tagged Offset, and length for both the Data Source and Data Sink Tagged Buffers to the Remote Peer's RDMA Read Request Queue.

   The RDMA Read Response Message uses the DDP Tagged Buffer Model to Deliver the Data Source's Tagged Buffer to the Data Sink, without any involvement from the ULP at the Data Source.

   Note: the Data Source STag associated with the Tagged Buffer remains valid until the ULP at the Data Source invalidates it or the ULP at the Data Sink invalidates it through a Send with

Invalidate or Send with Solicited Event and Invalidate. The Data Sink STag associated with the Tagged Buffer remains valid until the ULP at the Data Sink invalidates it.

7. Terminate - A Terminate operation uses a Terminate Message to transfer to the Remote Peer information associated with an error that occurred at the Local Peer. The Terminate Message uses the DDP Untagged Buffer Model to transfer the Message into the Data Sink's Untagged Buffer.

1.3. RDMAP Layering

RDMAP is dependent on DDP, subject to the requirements defined in Section 3.1, "Transport Requirements and Assumptions". Figure 1, "RDMAP Layering", depicts the relationship between Upper Layer Protocols (ULPs), RDMAP, DDP protocol, the framing layer, and the transport. For LLP protocol definitions of each LLP, see [MPA], [TCP], and [SCTP].

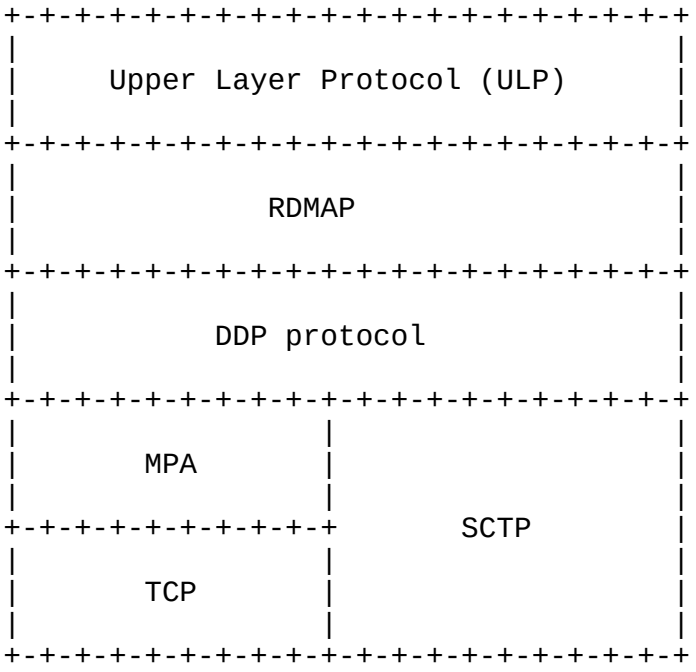

Figure 1: RDMAP Layering

If RDMAP is layered over DDP/MPA/TCP, then the respective headers and ULP Payload are arranged as follows (Note: For clarity, MPA header and CRC fields are included but MPA markers are not shown):

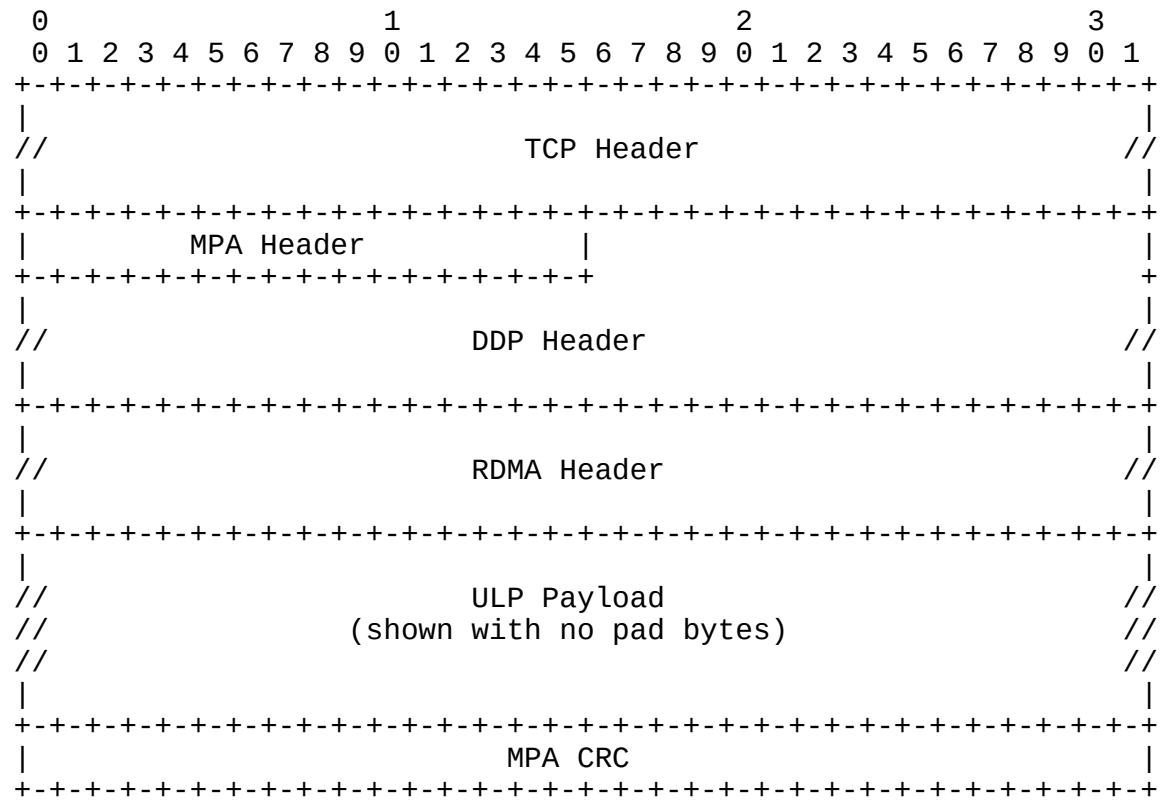

Figure 2: Example of MPA, DDP, and RDMAP Header Alignment over TCP

2. Glossary

2.1. General

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

Completion - Refer to "RDMA Completion" in Section 2.4.

Completed - See "RDMA Completion" in Section 2.4.

Complete - See "RDMA Completion" in Section 2.4.

Completes - See "RDMA Completion" in Section 2.4.

Data Sink - The peer receiving a data payload. Note that the Data Sink can be required to both send and receive RDMA/DDP 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 RDMA/DDP Messages to transfer a data payload.

Data Delivery (Delivery, Delivered, Delivers) - Delivery is defined as the process of informing the ULP or consumer that a particular Message is available for use. This is specifically different from "Placement", which may generally occur in any order, while the order of "Delivery" is strictly defined. See "Data Placement" in Section 2.3.

Delivery - See Data Delivery in Section 2.1.

Delivered - See Data Delivery in Section 2.1.

Delivers - See Data Delivery in Section 2.1.

Fabric - The collection of links, switches, and routers that connect a set of Nodes with RDMA/DDP protocol implementations.

Fence (Fenced, Fences) - To block the current RDMA Operation from executing until prior RDMA Operations have Completed.

iWARP - A suite of wire protocols comprised of RDMAP, DDP, and MPA. The iWARP protocol suite may be layered above TCP, SCTP, or other transport protocols.

Local Peer - The RDMA/DDP protocol implementation on the local end of the connection. Used to refer to the local entity when describing a protocol exchange or other interaction between two Nodes.

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

Placement - See "Data Placement" in Section 2.3.

Placed - See "Data Placement" in Section 2.3.

Places - See "Data Placement" in Section 2.3.

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

RNIC - RDMA Network Interface Controller. In this context, this would be a network I/O adapter or embedded controller with iWARP and Verbs functionality.

RNIC Interface (RI) - The presentation of the RNIC to the Verbs Consumer as implemented through the combination of the RNIC and the RNIC driver.

Termination - See "RDMAP Abortive Termination" in Section 2.4.

Terminated - See "RDMAP Abortive Termination" in Section 2.4.

Terminate - See "RDMAP Abortive Termination" in Section 2.4.

Terminates - See "RDMAP Abortive Termination" in Section 2.4.

ULP - Upper Layer Protocol. The protocol layer above the one currently being referenced. The ULP for RDMA/DDP is expected to be an OS, Application, adaptation layer, or proprietary device. The RDMA/DDP documents do not specify a ULP -- they provide a set of semantics that allow a ULP to be designed to utilize RDMA/DDP.

ULP Payload - The ULP data that is contained within a single protocol segment or packet (e.g., a DDP Segment).

Verbs - An abstract description of the functionality of an RNIC Interface. The OS may expose some or all of this functionality via one or more APIs to applications. The OS will also use some of the functionality to manage the RNIC Interface.

2.2. LLP

LLP - Lower Layer Protocol. The protocol layer beneath the protocol layer currently being referenced. For example, for DDP, the LLP is SCTP, MPA, or other transport protocols. For RDMA, the LLP is DDP.

LLP Connection - Corresponds to an LLP transport-level connection between the peer LLP layers on two Nodes.

LLP Stream - Corresponds to a single LLP transport-level Stream between the peer LLP layers on two Nodes. One or more LLP Streams may map to a single transport-level LLP connection. For transport protocols that support multiple Streams per connection (e.g., SCTP), an LLP Stream corresponds to one transport-level Stream.

MULPDU - Maximum ULPDU. The current maximum size of the record that is acceptable for DDP to pass to the LLP for transmission.

ULPDU - Upper Layer Protocol Data Unit. The data record defined by the layer above MPA.

2.3. Direct Data Placement (DDP)

Data Placement (Placement, Placed, Places) - For DDP, this term is specifically used to indicate the process of writing to a data buffer by a DDP implementation. DDP Segments carry Placement information, which may be used by the receiving DDP implementation to perform Data Placement of the DDP Segment ULP Payload. See "Data Delivery".

DDP Abortive Teardown - The act of closing a DDP Stream without attempting to Complete in-progress and pending DDP Messages.

DDP Graceful Teardown - The act of closing a DDP Stream such that all in-progress and pending DDP Messages are allowed to Complete successfully.

DDP Control Field - A fixed 16-bit field in the DDP Header. The DDP Control Field contains an 8-bit field whose contents are reserved for use by the ULP.

DDP Header - The header present in all DDP segments. The DDP Header contains control and Placement fields that are used to define the final Placement location for the ULP Payload carried in a DDP Segment.

DDP Message - A ULP-defined unit of data interchange, which is subdivided into one or more DDP segments. This segmentation may occur for a variety of reasons, including segmentation to respect the maximum segment size of the underlying transport protocol.

DDP Segment - The smallest unit of data transfer for the DDP protocol. It includes a DDP Header and ULP Payload (if present). A DDP Segment should be sized to fit within the underlying transport protocol MULPDU.

DDP Stream - A sequence of DDP Messages whose ordering is defined by the LLP. For SCTP, a DDP Stream maps directly to an SCTP Stream. For MPA, a DDP Stream maps directly to a TCP connection, and a single DDP Stream is supported. Note that DDP has no ordering guarantees between DDP Streams.

Direct Data Placement - A mechanism whereby ULP data contained within DDP Segments may be Placed directly into its final destination in memory without processing of the ULP. This may occur even when the DDP Segments arrive out of order. Out-of-order Placement support may require the Data Sink to implement the LLP and DDP as one functional block.

Direct Data Placement Protocol (DDP) - Also, a wire protocol that supports Direct Data Placement by associating explicit memory buffer placement information with the LLP payload units.

Message Offset (MO) - For the DDP Untagged Buffer Model, specifies the offset, in bytes, from the start of a DDP Message.

Message Sequence Number (MSN) - For the DDP Untagged Buffer Model, specifies a sequence number that is increasing with each DDP Message.

Queue Number (QN) - For the DDP Untagged Buffer Model, identifies a destination Data Sink queue for a DDP Segment.

Steering Tag - An identifier of a Tagged Buffer on a Node, valid as defined within a protocol specification.

STag - Steering Tag

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

Tagged Buffer Model - A DDP data transfer model used to transfer Tagged Buffers from the Local Peer to the Remote Peer.

Tagged DDP Message - A DDP Message that targets a Tagged Buffer.

Tagged Offset (TO) - The offset within a Tagged Buffer on a Node.

Untagged Buffer - A buffer that is not explicitly Advertised to the Remote Peer. Untagged Buffers support one of the two available data transfer mechanisms called the Untagged Buffer Model. An Untagged Buffer is used to send asynchronous control messages to the Remote Peer for RDMA Read, Send, and Terminate requests. Untagged Buffers handle Untagged DDP Messages.

Untagged Buffer Model - A DDP data transfer model used to transfer Untagged Buffers from the Local Peer to the Remote Peer.

Untagged DDP Message - A DDP Message that targets an Untagged Buffer.

2.4. Remote Direct Memory Access (RDMA)

Completion Queues (CQs) - Logical components of the RNIC Interface that conceptually represent how an RNIC notifies the ULP about the completion of the transmission of data, or the completion of the reception of data; see [RDMASEC].

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

Invalidate STag - A mechanism used to prevent the Remote Peer from reusing a previous explicitly Advertised STag, until the Local Peer makes it available through a subsequent explicit Advertisement. The STag cannot be accessed remotely until it is explicitly Advertised again.

RDMA Completion (Completion, Completed, Complete, Completes) - For RDMA, Completion is defined as the process of informing the ULP that a particular RDMA Operation has performed all functions specified for the RDMA Operations, including Placement and Delivery. The Completion semantic of each RDMA Operation is distinctly defined.

RDMA Message - A data transfer mechanism used to fulfill an RDMA Operation.

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

RDMA Protocol (RDMAP) - A wire protocol that supports RDMA Operations to transfer ULP data between a Local Peer and the Remote Peer.

RDMAP Abortive Termination (Termination, Terminated, Terminate, Terminates) - The act of closing an RDMAP Stream without attempting to Complete in-progress and pending RDMA Operations.

RDMAP Graceful Termination - The act of closing an RDMAP Stream such that all in-progress and pending RDMA Operations are allowed to Complete successfully.

RDMA Read - An RDMA Operation used by the Data Sink to transfer the contents of a source RDMA buffer from the Remote Peer to 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 RDMA Message used by the Data Sink to request the Data Source to transfer the contents of an RDMA buffer. The RDMA Read Request Message describes both the Data Source and Data Sink RDMA buffers.

RDMA Read Request Queue - The queue used for processing RDMA Read Requests. The RDMA Read Request Queue has a DDP Queue Number of 1.

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

RDMAP Stream - An association between a pair of RDMAP implementations, possibly on different Nodes, which transfer ULP data using RDMA Operations. There may be multiple RDMAP Streams on a single Node. An RDMAP Stream maps directly to a single DDP Stream.

RDMA Write - An RDMA Operation that transfers the contents of a source RDMA Buffer from the Local Peer to a destination RDMA Buffer at the Remote Peer using RDMA. The RDMA Write Message only describes the Data Sink RDMA 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 RNIC in the remote system allows the access to take place without interrupting the processing of the CPU(s) on the system.

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

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

Send Operation Type - A Send Operation, Send with Invalidate Operation, Send with Solicited Event Operation, or Send with Solicited Event and Invalidate Operation.

Solicited Event (SE) - A facility by which an RDMA Operation sender may cause an Event to be generated at the recipient, if the recipient is configured to generate such an Event, when a Send with Solicited Event Message or Send with Solicited Event and Invalidate Message is received. Note: The Local Peer's ULP can use the Solicited Event mechanism to ensure that Messages designated as important to the ULP are handled in an expeditious manner by the Remote Peer's ULP. The ULP at the Local Peer can indicate a given Send Message Type is important by using the Send with Solicited Event Message or Send with Solicited Event and Invalidate Message. The ULP at the Remote Peer can choose to only be notified when valid Send with Solicited Event Messages and/or Send with Solicited Event and Invalidate Messages arrive and handle other valid incoming Send Messages or Send with Invalidate Messages at its leisure.

Terminate - An RDMA Message used by a Node to pass an error indication to the peer Node on an RDMAP Stream. This operation is for RDMAP use only.

ULP Buffer - A buffer owned above the RDMAP layer and Advertised to the RDMAP layer either as a Tagged Buffer or an Untagged ULP Buffer.

ULP Message - The ULP data that is handed to a specific protocol layer for transmission. Data boundaries are preserved as they are transmitted through iWARP.

3. ULP and Transport Attributes

3.1. Transport Requirements and Assumptions

RDMAP MUST be layered on top of the Direct Data Placement Protocol [DDP].

RDMAP requires the following DDP support:

* RDMAP uses three queues for Untagged Buffers:
  * Queue Number 0 (used by RDMAP for Send, Send with Invalidate, Send with Solicited Event, and Send with Solicited Event and Invalidate operations).
  * Queue Number 1 (used by RDMAP for RDMA Read operations).
  * Queue Number 2 (used by RDMAP for Terminate operations).
* DDP maps a single RDMA Message to a single DDP Message.

* DDP uses the STag and Tagged Offset provided by the RDMAP for Tagged Buffer Messages (i.e., RDMA Write and RDMA Read Response).

* When the DDP layer Delivers an Untagged DDP Message to the RDMAP layer, DDP provides the length of the DDP Message. This ensures that RDMAP does not have to carry a length field in its header.

* When the RDMAP layer provides an RDMA Message to the DDP layer, DDP must insert the RsvdULP field value provided by the RDMAP layer into the associated DDP Message.

* When the DDP layer Delivers a DDP Message to the RDMAP layer, DDP provides the RsvdULP field.

* The RsvdULP field must be 1 octet for DDP Tagged Messages and 5 octets for DDP Untagged Messages.

* DDP propagates to RDMAP all operation or protection errors (used by RDMAP Terminate) and, when appropriate, the DDP Header fields of the DDP Segment that encountered the error.

* If an RDMA Operation is aborted by DDP or a lower layer, the contents of the Data Sink buffers associated with the operation are considered indeterminate.

* DDP, in conjunction with the lower layers, provides reliable, in-order Delivery.

3.2. RDMAP Interactions with the ULP

RDMAP provides the ULP with access to the following RDMA Operations as defined in this specification:

* Send

* Send with Solicited Event

* Send with Invalidate

* Send with Solicited Event and Invalidate

* RDMA Write

* RDMA Read

For Send Operation Types, the following are the interactions between the RDMAP layer and the ULP:

* At the Data Source:
  * The ULP passes to the RDMAP layer the following:
    * ULP Message Length
    * ULP Message
    * An indication of the Send Operation Type, where the valid types are: Send, Send with Solicited Event, Send with Invalidate, or Send with Solicited Event and Invalidate.
    * An Invalidate STag, if the Send Operation Type was Send with Invalidate or Send with Solicited Event and Invalidate.
  * When the Send Operation Type Completes, an indication of the Completion results.
* At the Data Sink:
  * If the Send Operation Type Completed successfully, the RDMAP layer passes the following information to the ULP Layer:
    * ULP Message Length
    * ULP Message
    * An Event, if the Data Sink is configured to generate an Event.
    * An Invalidated STag, if the Send Operation Type was Send with Invalidate or Send with Solicited Event and Invalidate.
  * If the Send Operation Type Completed in error, the Data Sink RDMAP layer will pass up the corresponding error information to the Data Sink ULP and send a Terminate Message to the Data Source RDMAP layer. The Data Source RDMAP layer will then pass up the Terminate Message to the ULP.

For RDMA Write operations, the following are the interactions between the RDMAP layer and the ULP:

* At the Data Source:
  * The ULP passes to the RDMAP layer the following:

      * ULP Message Length

      * ULP Message

      * Data Sink STag

      * Data Sink Tagged Offset

      * When the RDMA Write operation Completes, an indication of the Completion results.

   * At the Data Sink:

      * If the RDMA Write completed successfully, the RDMAP layer does not Deliver the RDMA Write to the ULP. It does Place the ULP Message transferred through the RDMA Write Message into the ULP Buffer.

      * If the RDMA Write completed in error, the Data Sink RDMAP layer will pass up the corresponding error information to the Data Sink ULP and send a Terminate Message to the Data Source RDMAP layer. The Data Source RDMAP layer will then pass up the Terminate Message to the ULP.

   For RDMA Read operations, the following are the interactions between the RDMAP layer and the ULP:

   * At the Data Sink:

      * The ULP passes to the RDMAP layer the following:

         * ULP Message Length

         * Data Source STag

         * Data Sink STag

         * Data Source Tagged Offset

         * Data Sink Tagged Offset

      * When the RDMA Read operation Completes, an indication of the Completion results.

   * At the Data Source:

      * If no error occurred while processing the RDMA Read Request, the Data Source will not pass up any information to the ULP.

      * If an error occurred while processing the RDMA Read Request, the Data Source RDMAP layer will pass up the corresponding error information to the Data Source ULP and send a Terminate Message to the Data Sink RDMAP layer. The Data Sink RDMAP layer will then pass up the Terminate Message to the ULP.

   For STags made available to the RDMAP layer, following are the interactions between the RDMAP layer and the ULP:

   * If the ULP enables an STag, the ULP passes the following to the RDMAP layer:

      * STag;

      * range of Tagged Offsets that are associated with a given STag;

      * remote access rights (read, write, or read and write) associated with a given, valid STag; and

      * association between a given STag and a given RDMAP Stream.

   * If the ULP disables an STag, the ULP passes to the RDMAP layer the STag.

   If an error occurs at the RDMAP layer, the RDMAP layer may pass back error information (e.g., the content of a Terminate Message) to the ULP.

4. Header Format

   The control information of RDMA Messages is included in DDP protocol-defined header fields, with the following exceptions:

   * The first octet reserved for ULP usage on all DDP Messages in the DDP Protocol (i.e., the RsvdULP Field) is used by RDMAP to carry the RDMA Message Opcode and the RDMAP version. This octet is known as the RDMAP Control Field in this specification. For Send with Invalidate and Send with Solicited Event and Invalidate, RDMAP uses the second through fifth octets, provided by DDP on Untagged DDP Messages, to carry the STag that will be Invalidated.

   * The RDMA Message length is passed by the RDMAP layer to the DDP layer on all outbound transfers.

   * For RDMA Read Request Messages, the RDMA Read Message Size is included in the RDMA Read Request Header.

* The RDMA Message length is passed to the RDMAP layer by the DDP layer on inbound Untagged Buffer transfers.

* Two RDMA Messages carry additional RDMAP headers. The RDMA Read Request carries the Data Sink and Data Source buffer descriptions, including buffer length. The Terminate carries additional information associated with the error that caused the Terminate.

4.1. RDMAP Control and Invalidate STag Field

The version of RDMAP defined by this specification uses all 8 bits of the RDMAP Control Field. The first octet reserved for ULP use in the DDP Protocol MUST be used by the RDMAP to carry the RDMAP Control Field. The ordering of the bits in the first octet MUST be as defined in Figure 3, "DDP Control, RDMAP Control, and Invalidate STag Fields". For Send with Invalidate and Send with Solicited Event and Invalidate, the second through fifth octets of the DDP RsvdULP field MUST be used by RDMAP to carry the Invalidate STag. Figure 3 depicts the format of the DDP Control and RDMAP Control fields. (Note: In Figure 3, the DDP Header is offset by 16 bits to accommodate the MPA header defined in [MPA]. The MPA header is only present if DDP is layered on top of MPA.)

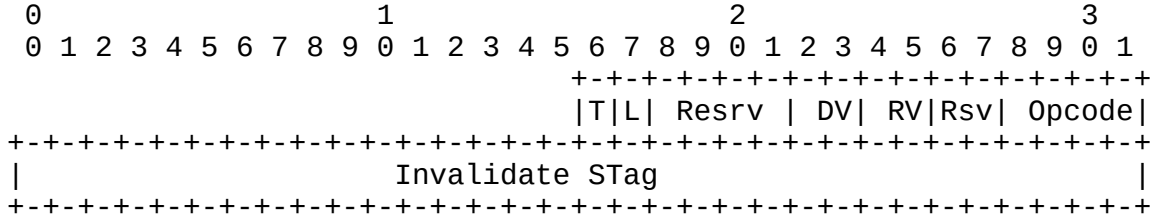

Figure 3: DDP Control, RDMAP Control, and Invalidate STag Fields

All RDMA Messages handed by the RDMAP layer to the DDP layer MUST define the value of the Tagged flag in the DDP Header. Figure 4, "RDMA Usage of DDP Fields", MUST be used to define the value of the Tagged flag that is handed to the DDP layer for each RDMA Message.

Figure 4 defines the value of the RDMA Opcode field that MUST be used for each RDMA Message.

Figure 4 defines when the STag, Queue Number, and Tagged Offset fields MUST be provided for each RDMA Message.

For this version of the RDMAP, all RDMA Messages MUST have:

* Bits 24-25; RDMA Version field: 01b for an RNIC that complies with this RDMA protocol specification. 00b for an RNIC that complies with the RDMA Consortium's RDMA protocol specification. Both version numbers are valid. Interoperability is dependent on MPA protocol version negotiation (e.g., MPA marker and MPA CRC).

* Bits 26-27; Reserved. MUST be set to zero by sender, ignored by the receiver.

* Bits 28-31; OpCode field: see Figure 4.

* Bits 32-63; Invalidate STag. However, this field is only valid for Send with Invalidate and Send with Solicited Event and Invalidate Messages (see Figure 4).

  For Send, Send with Solicited Event, RDMA Read Request, and Terminate, the Invalidate STag field MUST be set to zero on transmit and ignored by the receiver.

RDMA Message OpCode	Message Type	Tagged Flag	STag and TO	Queue Number	Invalidate STag	Message Length Communicated between DDP and RDMAP
0000b	RDMA Write	1	Valid	N/A	N/A	Yes
0001b	RDMA Read Request	0	N/A	1	N/A	Yes
0010b	RDMA Read Response	1	Valid	N/A	N/A	Yes
0011b	Send	0	N/A	0	N/A	Yes
0100b	Send with Invalidate	0	N/A	0	Valid	Yes
0101b	Send with SE	0	N/A	0	N/A	Yes
0110b	Send with SE and Invalidate	0	N/A	0	Valid	Yes
0111b	Terminate	0	N/A	2	N/A	Yes
1000b to 1111b	Reserved	Not Specified				

Figure 4: RDMA Usage of DDP Fields

Note: N/A means Not Applicable.

4.2. RDMA Message Definitions

The following figure defines which RDMA Headers MUST be used on each RDMA Message and which RDMA Messages are allowed to carry ULP Payload:

RDMA Message OpCode	Message Type	RDMA Header Used	ULP Message allowed in the RDMA Message
0000b	RDMA Write	None	Yes
0001b	RDMA Read Request	RDMA Read Request Header	No
0010b	RDMA Read Response	None	Yes
0011b	Send	None	Yes
0100b	Send with Invalidate	None	Yes
0101b	Send with SE	None	Yes
0110b	Send with SE and Invalidate	None	Yes
0111b	Terminate	Terminate Header	No
1000b to 1111b	Reserved	Not Specified	

Figure 5: RDMA Message Definitions

4.3. RDMA Write Header

The RDMA Write Message does not include an RDMAP header. The RDMAP layer passes to the DDP layer an RDMAP Control Field. The RDMA Write Message is fully described by the DDP Headers of the DDP Segments associated with the Message.

See Appendix A for a description of the DDP Segment format associated with RDMA Write Messages.

4.4. RDMA Read Request Header

The RDMA Read Request Message carries an RDMA Read Request Header that describes the Data Sink and Data Source Buffers used by the RDMA Read operation. The RDMA Read Request Header immediately follows the DDP header. The RDMAP layer passes to the DDP layer an RDMAP Control Field. The following figure depicts the RDMA Read Request Header that MUST be used for all RDMA Read Request Messages:

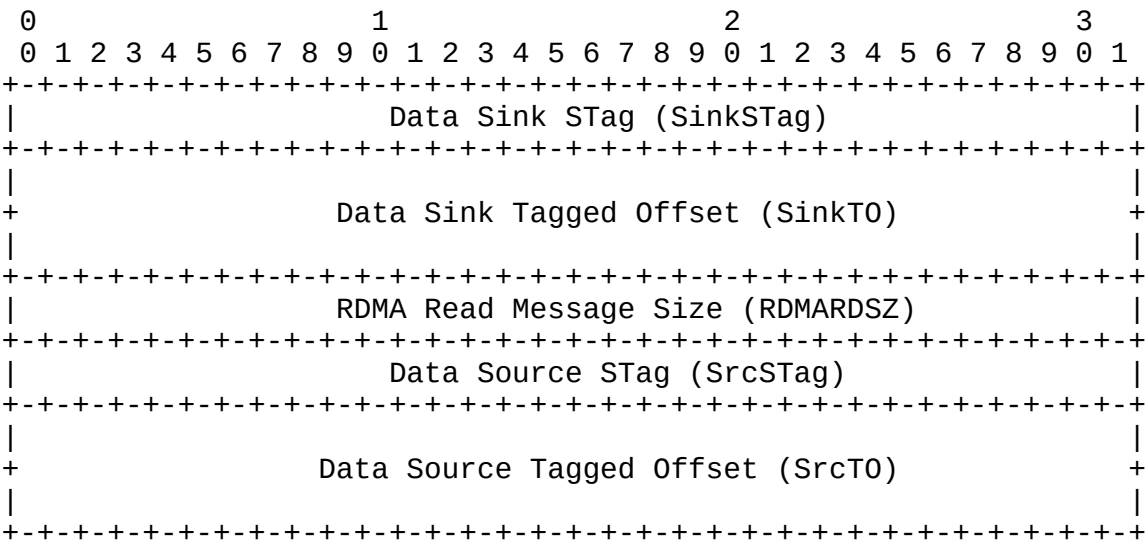

Figure 6: RDMA Read Request Header Format

Data Sink Steering Tag: 32 bits.

The Data Sink Steering Tag identifies the Data Sink's Tagged Buffer. This field MUST be copied, without interpretation, from the RDMA Read Request into the corresponding RDMA Read Response; this field allows the Data Sink to place the returning data. The STag is associated with the RDMAP Stream through a mechanism that is outside the scope of the RDMAP specification.

Data Sink Tagged Offset: 64 bits.

The Data Sink Tagged Offset specifies the starting offset, in octets, from the base of the Data Sink's Tagged Buffer, where the data is to be written by the Data Source. This field is copied from the RDMA Read Request into the corresponding RDMA Read Response and allows the Data Sink to place the returning data. The Data Sink Tagged Offset MAY start at an arbitrary offset.

The Data Sink STag and Data Sink Tagged Offset fields describe the buffer to which the RDMA Read data is written.

Note: the DDP layer protects against a wrap of the Data Sink Tagged Offset.

RDMA Read Message Size: 32 bits.

The RDMA Read Message Size is the amount of data, in octets, read from the Data Source. A single RDMA Read Request Message can retrieve from 0 to 2^32-1 data octets from the Data Source.

Data Source Steering Tag: 32 bits.

The Data Source Steering Tag identifies the Data Source's Tagged Buffer. The STag is associated with the RDMAP Stream through a mechanism that is outside the scope of the RDMAP specification.

Data Source Tagged Offset: 64 bits.

The Tagged Offset specifies the starting offset, in octets, that is to be read from the Data Source's Tagged Buffer. The Data Source Tagged Offset MAY start at an arbitrary offset.

The Data Source STag and Data Source Tagged Offset fields describe the buffer from which the RDMA Read data is read.

See Section 7.2, "Errors Detected at the Remote Peer on Incoming RDMA Messages", for a description of error checking required upon processing of an RDMA Read Request at the Data Source.

4.5. RDMA Read Response Header

The RDMA Read Response Message does not include an RDMAP header. The RDMAP layer passes to the DDP layer an RDMAP Control Field. The RDMA Read Response Message is fully described by the DDP Headers of the DDP Segments associated with the Message.

See Appendix A for a description of the DDP Segment format associated with RDMA Read Response Messages.

4.6. Send Header and Send with Solicited Event Header

The Send and Send with Solicited Event Messages do not include an RDMAP header. The RDMAP layer passes to the DDP layer an RDMAP Control Field. The Send and Send with Solicited Event Messages are fully described by the DDP Headers of the DDP Segments associated with the Messages.

See Appendix A for a description of the DDP Segment format associated with Send and Send with Solicited Event Messages.

4.7. Send with Invalidate Header and Send with SE and Invalidate Header

The Send with Invalidate and Send with Solicited Event and Invalidate Messages do not include an RDMAP header. The RDMAP layer passes to the DDP layer an RDMAP Control Field and the Invalidate STag field (see section 4.1 RDMAP Control and Invalidate STag Field). The Send with Invalidate and Send with Solicited Event and Invalidate Messages are fully described by the DDP Headers of the DDP Segments associated with the Messages.

See Appendix A for a description of the DDP Segment format associated with Send and Send with Solicited Event Messages.

4.8. Terminate Header

The Terminate Message carries a Terminate Header that contains additional information associated with the cause of the Terminate. The Terminate Header immediately follows the DDP header. The RDMAP layer passes to the DDP layer an RDMAP Control Field. The following figure depicts a Terminate Header that MUST be used for the Terminate Message:

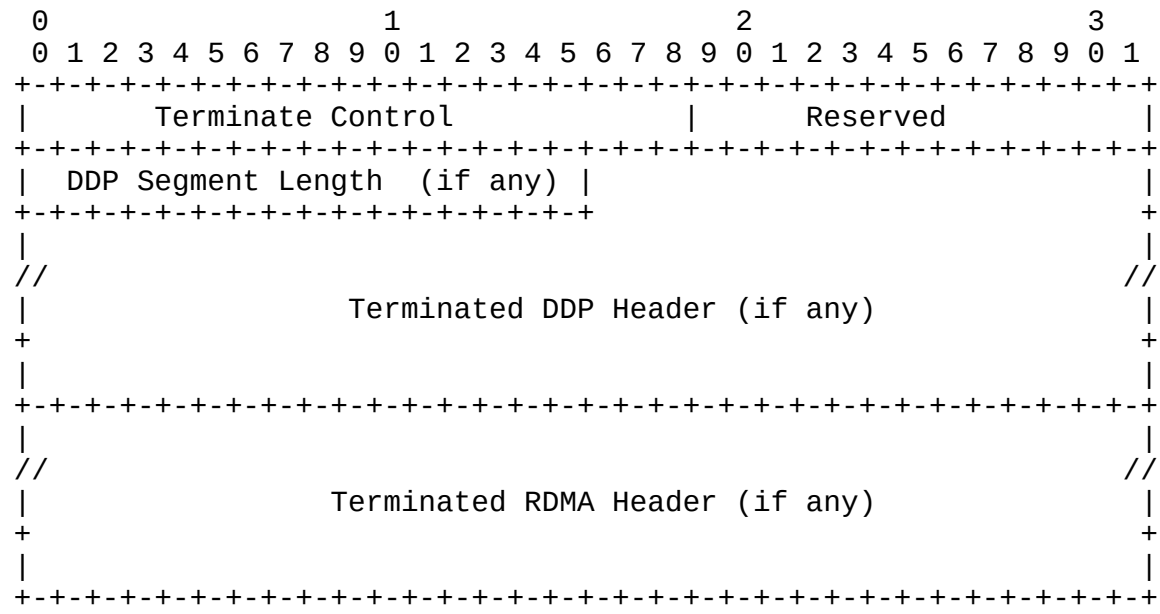

Figure 7: Terminate Header Format

Terminate Control: 19 bits.

The Terminate Control field MUST have the format defined in Figure 8 below.

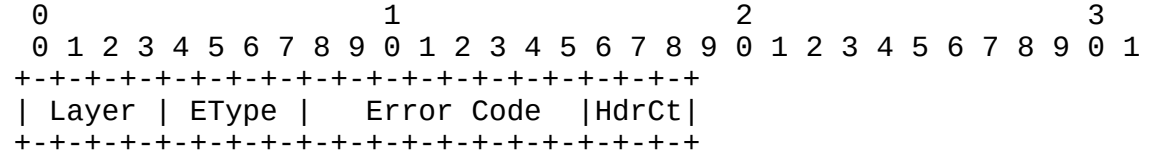

Figure 8: Terminate Control Field

* Figure 9, "Terminate Control Field Values", defines the valid values that MUST be used for this field.

  * Layer: 4 bits.

    Identifies the layer that encountered the error.

  * EType (RDMA Error Type): 4 bits.

    Identifies the type of error that caused the Terminate. When the error is detected at the RDMAP layer, the RDMAP layer inserts the Error Type into this field. When the error is detected at an LLP layer, an LLP layer creates the Error Type

and the DDP layer passes it up to the RDMAP layer, and the RDMAP layer inserts it into this field.

* Error Code: 8 bits.

  This field identifies the specific error that caused the Terminate. When the error is detected at the RDMAP layer, the RDMAP layer creates the Error Code. When the error is detected at an LLP layer, the LLP layer creates the Error Code, the DDP layer passes it up to the RDMAP layer, and the RDMAP layer inserts it into this field.

* HdrCt: 3 bits.

  Header control bits:

  * M: bit 16. DDP Segment Length valid. See Figure 10 for when this bit SHOULD be set.

  * D: bit 17. DDP Header Included. See Figure 10 for when this bit SHOULD be set.

  * R: bit 18. RDMAP Header Included. See Figure 10 for when this bit SHOULD be set.

Layer	Layer Name	Error Type	Error Type Name	Error Code	Error Code Name
0000b	RDMA	0000b	Local Catastrophic Error	None	None - This error type does not have an error code. Any value in this field is acceptable.
		0001b	Remote Protection Error	00X	Invalid STag
				01X	Base or bounds violation
				02X	Access rights violation
				03X	STag not associated with RDMAP Stream
				04X	TO wrap
				09X	STag cannot be Invalidated
				FFX	Unspecified Error
		0010b	Remote Operation Error	05X	Invalid RDMAP version
				06X	Unexpected OpCode
				07X	Catastrophic error, localized to RDMAP Stream
				08X	Catastrophic error, global
				09X	STag cannot be Invalidated
				FFX	Unspecified Error

0001b	DDP	See DDP Specification [DDP] for a description of the values and names.			
0010b	LLP (e.g., MPA)	For MPA, see MPA Specification [MPA] for a description of the values and names.			

Figure 9: Terminate Control Field Values

Reserved: 13 bits. This field MUST be set to zero on transmit, ignored on receive.

DDP Segment Length: 16 bits

The length handed up by the DDP layer when the error was detected. It MUST be valid if the M bit is set. It MUST be present when the D bit is set.

Terminated DDP Header: 112 bits for Tagged Messages and 144 bits for Untagged Messages.

The DDP Header of the incoming Message that is associated with the Terminate. The DDP Header is not present if the Terminate Error Type is a Local Catastrophic Error. It MUST be present if the D bit is set.

Terminated RDMA Header: 224 bits.

The Terminated RDMA Header is only sent back if the terminate is associated with an RDMA Read Request Message. It MUST be present if the R bit is set.

If the terminate occurs before the first RDMA Read Request byte is processed, the original RDMA Read Request Header is sent back.

If the terminate occurs after the first RDMA Read Request byte is processed, the RDMA Read Request Header is updated to reflect the current location of the RDMA Read operation that is in process:

* Data Sink STag = Data Sink STag originally sent in the RDMA Read Request.

* Data Sink Tagged Offset = Current offset into the Data Sink Tagged Buffer. For example, if the RDMA Read Request was terminated after 2048 octets were sent, then the Data Sink Tagged Offset = the original Data Sink Tagged Offset + 2048.

* Data Message size = Number of bytes left to transfer.

* Data Source STag = Data Source STag in the RDMA Read Request.

* Data Source Tagged Offset = Current offset into the Data Source Tagged Buffer. For example, if the RDMA Read Request was terminated after 2048 octets were sent, then the Data Source Tagged Offset = the original Data Source Tagged Offset + 2048.

Note: if a given LLP does not define any termination codes for the RDMAP Termination message to use, then none would be used for that LLP.

Figure 10, "Error Type to RDMA Message Mapping", maps layer name and error types to each RDMA Message type:

Layer Name	Error Type Name	Terminate Includes DDP Header and DDP Segment Length	Terminate Includes RDMA Header	What type of RDMA Message can cause the error
	Local Catastrophic Error	No	No	Any
RDMA	Remote Protection Error	Yes, if possible	Yes	Only RDMA Read Request, Send with Invalidate, and Send with SE and Invalidate
	Remote Operation Error	Yes, if possible	No	Any
DDP	See DDP Spec [DDP]	Yes	No	Any
LLP	See LLP Spec (e.g., MPA)	No	No	Any

Figure 10: Error Type to RDMA Message Mapping

5. Data Transfer

5.1. RDMA Write Message

An RDMA Write is used by the Data Source to transfer data to a previously Advertised Tagged Buffer at the Data Sink. The RDMA Write Message has the following semantics:

* An RDMA Write Message MUST reference a Tagged Buffer. That is, the Data Source RDMAP layer MUST request that the DDP layer mark the Message as Tagged.

* A valid RDMA Write Message MUST NOT be delivered to the Data Sink's ULP (i.e., it is placed by the DDP layer).

* At the Remote Peer, when an invalid RDMA Write Message is delivered to the Remote Peer's RDMAP layer, an error is surfaced (see Section 7.1, "RDMAP Error Surfacing").

* The Tagged Offset of a Tagged Buffer MAY start at a non-zero value.

* An RDMA Write Message MAY target all or part of a previously Advertised Buffer.

* The RDMAP does not define how the buffer(s) are used by an outbound RDMA Write or how they are addressed. For example, an implementation of RDMA may choose to allow a gather-list of non-contiguous data blocks to be the source of an RDMA Write. In this case, the data blocks would be combined by the Data Source and sent as a single RDMA Write Message to the Data Sink.

* The Data Source RDMAP layer MUST issue RDMA Write Messages to the DDP layer in the order they were submitted by the ULP.

* At the Data Source, a subsequent Send (Send with Invalidate, Send with Solicited Event, or Send with Solicited Event and Invalidate) Message MAY be used to signal Delivery of previous RDMA Write Messages to the Data Sink, if the ULP chooses to signal Delivery in this fashion.

* If the Local Peer wishes to write to multiple Tagged Buffers on the Remote Peer, the Local Peer MUST use multiple RDMA Write Messages. That is, a single RDMA Write Message can only write to one remote Tagged Buffer.

* The Data Source MAY issue a zero-length RDMA Write Message.

5.2. RDMA Read Operation

The RDMA Read operation MUST consist of a single RDMA Read Request Message and a single RDMA Read Response Message.

5.2.1. RDMA Read Request Message

An RDMA Read Request is used by the Data Sink to transfer data from a previously Advertised Tagged Buffer at the Data Source to a Tagged Buffer at the Data Sink. The RDMA Read Request Message has the following semantics:

* An RDMA Read Request Message MUST reference an Untagged Buffer. That is, the Local Peer’s RDMAP layer MUST request that the DDP mark the Message as Untagged.

* One RDMA Read Request Message MUST consume one Untagged Buffer.

* The Remote Peer's RDMAP layer MUST process an RDMA Read Request Message. A valid RDMA Read Request Message MUST NOT be delivered to the Data Sink's ULP (i.e., it is processed by the RDMAP layer).

* At the Remote Peer, when an invalid RDMA Read Request Message is delivered to the Remote Peer's RDMAP layer, an error is surfaced (see Section 7.1, "RDMAP Error Surfacing").

* An RDMA Read Request Message MUST reference the RDMA Read Request Queue. That is, the Local Peer's RDMAP layer MUST request that the DDP layer set the Queue Number field to one.

* The Local Peer MUST pass to the DDP layer RDMA Read Request Messages in the order they were submitted by the ULP.

* The Remote Peer MUST process the RDMA Read Request Messages in the order they were sent.

* If the Local Peer wishes to read from multiple Tagged Buffers on the Remote Peer, the Local Peer MUST use multiple RDMA Read Request Messages. That is, a single RDMA Read Request Message MUST only read from one remote Tagged Buffer.

* AN RDMA Read Request Message MAY target all or part of a previously Advertised Buffer.

* If the Data Source receives a valid RDMA Read Request Message, it MUST respond with a valid RDMA Read Response Message.

* The Data Sink MAY issue a zero-length RDMA Read Request Message by setting the RDMA Read Message Size field to zero in the RDMA Read Request Header.

* If the Data Source receives a non-zero-length RDMA Read Message Size, the Data Source RDMAP MUST validate the Data Source STag and Data Source Tagged Offset contained in the RDMA Read Request Header.

* If the Data Source receives an RDMA Read Request Header with the RDMA Read Message Size set to zero, the Data Source RDMAP:

  * MUST NOT validate the Data Source STag and Data Source Tagged Offset contained in the RDMA Read Request Header, and

  * MUST respond with a zero-length RDMA Read Response Message.

5.2.2. RDMA Read Response Message

The RDMA Read Response Message uses the DDP Tagged Buffer Model to Deliver the contents of a previously requested Data Source Tagged Buffer to the Data Sink, without any involvement from the ULP at the Remote Peer. The RDMA Read Response Message has the following semantics:

* The RDMA Read Response Message for the associated RDMA Read Request Message travels in the opposite direction.

* An RDMA Read Response Message MUST reference a Tagged Buffer. That is, the Data Source RDMAP layer MUST request that the DDP mark the Message as Tagged.

* The Data Source MUST ensure that a sufficient number of Untagged Buffers are available on the RDMA Read Request Queue (Queue with DDP Queue Number 1) to support the maximum number of RDMA Read Requests negotiated by the ULP.

* The RDMAP layer MUST Deliver the RDMA Read Response Message to the ULP.

* At the Remote Peer, when an invalid RDMA Read Response Message is delivered to the Remote Peer’s RDMAP layer, an error is surfaced (see Section 7.1, "RDMAP Error Surfacing").

* The Tagged Offset of a Tagged Buffer MAY start at a non-zero value.

* The Data Source RDMAP layer MUST pass RDMA Read Response Messages to the DDP layer, in the order that the RDMA Read Request Messages were received by the RDMAP layer, at the Data Source.

* The Data Sink MAY validate that the STag, Tagged Offset, and length of the RDMA Read Response Message are the same as the STag, Tagged Offset, and length included in the corresponding RDMA Read Request Message.

* A single RDMA Read Response Message MUST write to one remote Tagged Buffer. If the Data Sink wishes to read multiple Tagged Buffers, the Data Sink can use multiple RDMA Read Request Messages.

5.3. Send Message Type

The Send Message Type uses the DDP Untagged Buffer Model to transfer data from the Data Source into an Untagged Buffer at the Data Sink.

* A Send Message Type MUST reference an Untagged Buffer. That is, the Local Peer's RDMAP layer MUST request that the DDP layer mark the Message as Untagged.

* One Send Message Type MUST consume one Untagged Buffer.

  * The ULP Message sent using a Send Message Type MAY be less than or equal to the size of the consumed Untagged Buffer. The RDMAP layer communicates to the ULP the size of the data written into the Untagged Buffer.

  * If the ULP Message sent via Send Message Type is larger than the Data Sink's Untagged Buffer, it is an error (see Section 9.1, "RDMAP Error Surfacing").

* At the Remote Peer, the Send Message Type MUST be Delivered to the Remote Peer's ULP in the order they were sent.

* After the Send with Solicited Event or Send with Solicited Event and Invalidate Message is Delivered to the ULP, the RDMAP MAY generate an Event, if the Data Sink is configured to generate such an Event.

* At the Remote Peer, when an invalid Send Message Type is Delivered to the Remote Peer's RDMAP layer, an error is surfaced (see Section 7.1, "RDMAP Error Surfacing").

* The RDMAP does not specify the structure of the buffer(s) used by an outbound RDMA Write nor does it specify how the buffer(s) are addressed. For example, an implementation of RDMA may choose to allow a gather-list of non-contiguous data blocks to be the source of a Send Message Type. In this case, the data blocks would be combined by the Data Source and sent as a single Send Message Type to the Data Sink.

* For a Send Message Type, the Local Peer's RDMAP layer MUST request that the DDP layer set the Queue Number field to zero.

* The Local Peer MUST issue Send Message Type Messages in the order they were submitted by the ULP.

* The Data Source MAY pass a zero-length Send Message Type. A zero-length Send Message Type MUST consume an Untagged Buffer at the Data Sink. A Send with Invalidate or Send with Solicited Event and Invalidate Message MUST reference an STag. That is, the Local Peer's RDMAP layer MUST pass the RDMA control field and the STag that will be Invalidated to the DDP layer.

* When the Send with Invalidate and Send with Solicited Event and Invalidate Message are Delivered to the Remote Peer's RDMAP layer, the RDMAP layer MUST:

  * Verify the STag that is associated with the RDMAP Stream; and

  * Invalidate the STag if it is associated with the RDMAP Stream; or issue a Terminate Message with the STag Cannot be Invalidated Terminate Error Code, if the STag is not associated with the RDMAP Stream.

5.4. Terminate Message

The Terminate Message uses the DDP Untagged Buffer Model to transfer-error-related information from the Data Source into an Untagged Buffer at the Data Sink and then ceases all further communications on the underlying DDP Stream. The Terminate Message has the following semantics:

* A Terminate Message MUST reference an Untagged Buffer. That is, the Local Peer's RDMAP layer MUST request that the DDP layer mark the Message as Untagged.

* A Terminate Message references the Terminate Queue. That is, the Local Peer's RDMAP layer MUST request that the DDP layer set the Queue Number field to two.

* One Terminate Message MUST consume one Untagged Buffer.

* On a single RDMAP Stream, the RDMAP layer MUST guarantee placement of a single Terminate Message.

* A Terminate Message MUST be Delivered to the Remote Peer's RDMAP layer. The RDMAP layer MUST Deliver the Terminate Message to the ULP.

* At the Remote Peer, when an invalid Terminate Message is delivered to the Remote Peer's RDMAP layer, an error is surfaced (see Section 7.1 "RDMAP Error Surfacing").

* The RDMAP layer Completes in error all ULP operations that have not been provided to the DDP layer.

* After sending a Terminate Message on an RDMAP Stream, the Local Peer MUST NOT send any more Messages on that specific RDMAP Stream.

* After receiving a Terminate Message on an RDMAP Stream, the Remote Peer MAY stop sending Messages on that specific RDMAP Stream.

5.5. Ordering and Completions

It is important to understand the difference between Placement and Delivery ordering since RDMAP provides quite different semantics for the two.

Note that many current protocols, both as used in the Internet and elsewhere, assume that data is both Placed and Delivered in order. Taking advantage of this fact allowed applications to take a variety of shortcuts. For RDMAP, many of these shortcuts are no longer safe to use, and could cause application failure.

The following rules apply to implementations of the RDMAP protocol. Note that in these rules, Send includes Send, Send with Invalidate, Send with Solicited Event, and Send with Solicited Event and Invalidate:

1. RDMAP does not provide ordering among Messages on different RDMAP Streams.

2. RDMAP does not provide ordering between operations that are generated from the two ends of an RDMAP Stream.

3. RDMA Messages that use Tagged and Untagged Buffers MAY be Placed in any order. If an application uses overlapping buffers (points different Messages or portions of a single Message at the same buffer), then it is possible that the last incoming write to the Data Sink buffer will not be the last outgoing data sent from the Data Source.

4. For a Send operation, the contents of an Untagged Buffer at the Data Sink MAY be indeterminate until the Send is Delivered to the ULP at the Data Sink.

5. For an RDMA Write operation, the contents of the Tagged Buffer at the Data Sink MAY be indeterminate until a subsequent Send is Delivered to the ULP at the Data Sink.

6. For an RDMA Read operation, the contents of the Tagged Buffer at the Data Sink MAY be indeterminate until the RDMA Read Response Message has been Delivered at the Local Peer.

Statements 4, 5, and 6 imply "no peeking" at the data to see if it is done. It is possible for some data to arrive before logically earlier data does, and peeking may cause unpredictable application failure.

7. If the ULP or Application modifies the contents of Tagged or Untagged Buffers, which are being modified by an RDMA Operation while the RDMAP is processing the RDMA Operation, the state of the Buffers is indeterminate.

8. If the ULP or Application modifies the contents of Tagged or Untagged Buffers, which are read by an RDMA Operation while the RDMAP is processing the RDMA Operation, the results of the read are indeterminate.

9. The Completion of an RDMA Write or Send Operation at the Local Peer does not guarantee that the ULP Message has yet reached the Remote Peer ULP Buffer or been examined by the Remote ULP.

10. Send Messages MUST be Delivered to the ULP at the Remote Peer after they are Delivered to RDMAP by DDP and in the order that they were Delivered to RDMAP.

    Note that DDP ordering rules ensure that this will be the same order that they were submitted at the Local Peer and that any prior RDMA Writes have been submitted for ordered Placement at the Remote Peer. This means that when the ULP sees the Delivery of the Send, the memory buffers targeted by any preceding RDMA Writes and Sends are available to be accessed locally or remotely as authorized. If the ULP overlaps its buffers for different operations, the data from the RDMA Write or Send may be overwritten by subsequent RDMA Operations before the ULP receives and processes the Delivery.

11. RDMA Read Response Messages MUST be Delivered to the ULP at the Remote Peer after they are Delivered to RDMAP by DDP and in the order that the they were Delivered to RDMAP.

    DDP ordering rules ensure that this will be the same order that they were submitted at the Local Peer. This means that when the ULP sees the Delivery of the RDMA Read Response, the memory buffers targeted by the RDMA Read Response are available to be accessed locally or remotely as authorized. If the ULP overlaps

its buffers for different operations, the data from the RDMA Read Response may be overwritten by subsequent RDMA Operations before the ULP receives and processes the Delivery.

12. RDMA Read Request Messages, including zero-length RDMA Read Requests, MUST NOT start processing at the Remote Peer until they have been Delivered to RDMAP by DDP.

    Note: the ULP is assured that data written can be read back. For example, if

    a) an RDMA Read Request is issued by the local peer,
    b) the Request targets the same ULP Buffer as a preceding Send or RDMA Write (in the same direction as the RDMA Read Request), and
    c) there are no other sources of update for the ULP Buffer,

    then the Remote Peer will send back the data written by the Send or RDMA Write. That is, for this example, the ULP Buffer is Advertised for use on a series of RDMA Messages, is only valid on the RDMAP Stream for which it is Advertised, and is not locally updated while the series of RDMAP Messages are performed. For this example, order rule (12) assures that subsequent local or remote accesses to the ULP Buffer contain the data written by the Send or RDMA Write.

    RDMA Read Response Messages MAY be generated at the Remote Peer after subsequent RDMA Write Messages or Send Messages have been Placed or Delivered. Therefore, when an application does an RDMA Read Request followed by an RDMA Write (or Send) to the same buffer, it may get the data from the later RDMA Write (or Send) in the RDMA Read Response Message, even though the operations completed in order at the Local Peer. If this behavior is not desired, the Local Peer ULP must Fence the later RDMA write (or Send) by withholding the RDMA Write Message until all outstanding RDMA Read Responses have been Delivered.

13. The RDMAP layer MUST submit RDMA Messages to the DDP layer in the order the RDMA Operations are submitted to the RDMAP layer by the ULP.

14. A Send or RDMA Write Message MUST NOT be considered Complete at the Local Peer (Data Source) until it has been successfully completed at the DDP layer.

15. RDMA Operations MUST be Completed at the Local Peer in the order that they were submitted by the ULP.

16. At the Data Sink, an incoming Send Message MUST be Delivered to the ULP only after the DDP Message has been Delivered to the RDMAP layer by the DDP layer.

17. RDMA Read Response Message processing at the Remote Peer (reading the specified Tagged Buffer) MUST be started only after the RDMA Read Request Message has been Delivered by the DDP layer (thus, all previous RDMA Messages have been properly submitted for ordered Placement).

18. Send Messages MAY be Completed at the Remote Peer (Data Sink) before prior incoming RDMA Read Request Messages have completed their response processing.

19. An RDMA Read operation MUST NOT be Completed at the Local Peer until the DDP layer Delivers the associated incoming RDMA Read Response Message.

20. If more than one outstanding RDMA Read Request Messages are supported by both peers, the RDMA Read Response Messages MUST be submitted to the DDP layer on the Remote Peer in the order the RDMA Read Request Messages were Delivered by DDP, but the actual read of the buffer contents MAY take place in any order at the Remote Peer.

    This simplifies Local Peer Completion processing for RDMA Reads in that a Delivered RDMA Read Response MUST be sufficient to Complete the RDMA Read operation.

6. RDMAP Stream Management

RDMAP Stream management consists of RDMAP Stream Initialization and RDMAP Stream Termination.

6.1. Stream Initialization

RDMAP Stream initialization occurs after the LLP Stream has been created (e.g., for DDP/MPA over TCP, the first TCP Segment after the SYN, SYN/ACK exchange). The ULP is responsible for transitioning the LLP Stream into RDMA-enabled mode. The switch to RDMA mode typically occurs sometime after LLP Stream setup. Once in RDMA enabled mode, an implementation MUST send only RDMA Messages across the transport Stream until the RDMAP Stream is torn down.

For each direction of an RDMAP Stream:

* For a given RDMAP Stream, the number of outstanding RDMA Read Requests is limited per RDMAP Stream direction.

* It is the ULP's responsibility to set the maximum number of outstanding, inbound RDMA Read Requests per RDMAP Stream direction.

* The RDMAP layer MUST provide the maximum number of outstanding, inbound RDMA Read Requests per RDMAP Stream direction that were negotiated between the ULP and the Local Peer's RDMAP layer. The negotiation mechanism is outside the scope of this specification.

* It is the ULP's responsibility to set the maximum number of outstanding, outbound RDMA Read Requests per RDMAP Stream direction.

* The RDMAP layer MUST provide the maximum number of outstanding, outbound RDMA Read Requests for the RDMAP Stream direction that were negotiated between the ULP and the Local Peer's RDMAP layer. The negotiation mechanism is outside the scope of this specification.

* The Local Peer's ULP is responsible for negotiating with the Remote Peer's ULP the maximum number of outstanding RDMA Read Requests for the RDMAP Stream direction. It is recommended that the ULP set the maximum number of outstanding, inbound RDMA Read Requests equal to the maximum number of outstanding, outbound RDMA Read Requests for a given RDMAP Stream direction.

* For outbound RDMA Read Requests, the RDMAP layer MUST NOT exceed the maximum number of outstanding, outbound RDMA Read Requests that were negotiated between the ULP and the Local Peer's RDMAP layer.

* For inbound RDMA Read Requests, the RDMAP layer MUST NOT exceed the maximum number of outstanding, inbound RDMA Read Requests that were negotiated between the ULP and the Local Peer's RDMAP layer.

6.2. Stream Teardown

There are three methods for terminating an RDMAP Stream: ULP Graceful Termination, RDMAP Abortive Termination, and LLP Abortive Termination.

The ULP is responsible for performing ULP Graceful Termination. After a ULP Graceful Termination, either side of the Stream can initiate LLP Graceful Termination, using the graceful termination mechanism provided by the LLP.

RDMAP Abortive Termination allows the RDMAP to issue a Terminate Message describing the reason the RDMAP Stream was terminated. The next section (6.2.1, "RDMAP Abortive Termination") describes the RDMAP Abortive Termination in detail.

LLP Abortive Termination results due to an LLP error and causes the RDMAP Stream to be torn down midstream, without an RDMAP Terminate Message. While this last method is highly undesirable, it is possible, and the ULP should take this into consideration.

6.2.1. RDMAP Abortive Termination

RDMAP defines a Terminate operation that SHOULD be invoked when either an RDMAP error is encountered or an LLP error is surfaced to the RDMAP layer by the LLP.

It is not always possible to send the Terminate Message. For example, certain LLP errors may occur that cause the LLP Stream to be torn down a) before RDMAP is aware of the error, b) before RDMAP is able to send the Terminate Message, or c) after RDMAP has posted the Terminate Message to the LLP, but it has not yet been transmitted by the LLP.

Note that an RDMAP Abortive Termination may entail loss of data. In general, when a Terminate Message is received, it is impossible to tell for sure what unacknowledged RDMA Messages were Completed successfully at the Remote Peer. Thus, the state of all outstanding RDMA Messages is indeterminate, and the Messages SHOULD be considered Completed in error.

When a peer sends or receives a Terminate Message, it MAY immediately tear down the LLP Stream. The peer SHOULD perform a graceful LLP teardown to ensure the Terminate Message is successfully Delivered.

See Section 4.8, "Terminate Header", for a description of the Terminate Message and its contents. See Section 5.4, "Terminate Message", for a description of the Terminate Message semantics.

7. RDMAP Error Management

The RDMAP protocol does not have RDMAP- or DDP-layer error recovery operations built in. If everything is working, the LLP guarantees will ensure that the Messages are arriving at the destination.

If errors are detected at the RDMAP or DDP layer, then the RDMAP, DDP, and LLP Streams are Abortively Terminated (see Section 4.8, "Terminate Header").

In general, poor implementations or improper ULP programming cause the errors detected at the RDMAP and DDP layers. In these cases, returning a diagnostic termination error Message and closing the RDMAP Stream is far simpler than attempting to maintain the RDMAP Stream, particularly when the cause of the error is not known.

If an LLP does not support teardown of a Stream independent of other Streams, and an RDMAP error results in the Termination of a specific Stream, then the LLP MUST label the Stream as an erroneous Stream and MUST NOT allow any further data transfer on that Stream after RDMAP requests the Stream to be torn down.

For a specific LLP connection, when all Streams are either gracefully torn down or are labeled as erroneous Streams, the LLP connection MUST be torn down.

Since errors are detected at the Remote Peer (possibly long) after RDMA Messages are passed to the DDP and the LLP at the Local Peer and after the RDMA Operations conveyed by the Messages are Completed, the sender cannot easily determine which of its Messages have been received. (RDMA Reads are an exception to this rule.)

For a list of errors returned to the Remote Peer as a result of an Abortive Termination, see Section 4.8, "Terminate Header".

7.1. RDMAP Error Surfacing

If an error occurs at the Local Peer, the RDMAP layer MUST attempt to inform the local ULP that the error has occurred.

The Local Peer MUST send a Terminate Message for each of the following cases:

1. For errors detected while creating RDMA Write, Send, Send with Invalidate, Send with Solicited Event, Send with Solicited Event and Invalidate, or RDMA Read Requests, or other reasons not directly associated with an incoming Message, the Terminate Message and Error code are sent instead of the request. In this case, the Error Type and Error Code fields are included in the Terminate Message, but the Terminated DDP Header and Terminated RDMA Header fields are set to zero.

2. For errors detected on an incoming RDMA Write, Send, Send with Invalidate, Send with Solicited Event, Send with Solicited Event and Invalidate, or Read Response Message (after the Message has been Delivered by DDP), the Terminate Message is sent at the earliest possible opportunity, preferably in the next outgoing RDMA Message. In this case, the Error Type, Error Code, ULP PDU

Length, and Terminated DDP Header fields are included in the Terminate Message, but the Terminated RDMA Header field is set to zero.

3. For errors detected on an incoming RDMA Read Request Message (after the Message has been Delivered by DDP), the Terminate Message is sent at the earliest possible opportunity, preferably in the next outgoing RDMA Message. In this case, the Error Type, Error Code, ULP PDU Length, Terminated DDP Header, and Terminated RDMA Header fields are included in the Terminate Message.

4. If more than one error is detected on incoming RDMA Messages, before the Terminate Message can be sent, then the first RDMA Message (and its associated DDP Segment) that experienced an error MUST be captured by the Terminate Message, in accordance with rules 2 and 3 above.

7.2. Errors Detected at the Remote Peer on Incoming RDMA Messages

On incoming RDMA Writes, RDMA Read Response, Sends, Send with Invalidate, Send with Solicited Event, Send with Solicited Event and Invalidate, and Terminate Messages, the following must be validated:

1. The DDP layer MUST validate all DDP Segment fields.

2. The RDMA OpCode MUST be valid.

3. The RDMA Version MUST be valid.

   Additionally, on incoming Send with Invalidate and Send with Solicited Event and Invalidate Messages, the following must also be validated:

4. The Invalidate STag MUST be valid.

5. The STag MUST be associated to this RDMAP Stream.

On incoming RDMA Request Messages, the following must be validated:

1. The DDP layer MUST validate all Untagged DDP Segment fields.

2. The RDMA OpCode MUST be valid.

3. The RDMA Version MUST be valid.

4. For non-zero length RDMA Read Request Messages:

   a. The Data Source STag MUST be valid.

b. The Data Source STag MUST be associated to this RDMAP Stream.

c. The Data Source Tagged Offset MUST fall in the range of legal offsets associated with the Data Source STag.

d. The sum of the Data Source Tagged Offset and the RDMA Read Message Size MUST fall in the range of legal offsets associated with the Data Source STag.

e. The sum of the Data Source Tagged Offset and the RDMA Read Message Size MUST NOT cause the Data Source Tagged Offset to wrap.

8. Security Considerations

This section references the resources that discuss protocol- specific security considerations and implications of using RDMAP with existing security services. A detailed analysis of the security issues around implementation and use of the RDMAP can be found in [RDMASEC].

[RDMASEC] introduces the RDMA reference model and discusses how the resources of this model are vulnerable to attacks and the types of attack these vulnerabilities are subject to. It also details the levels of Trust available in this peer-to-peer model and how this defines the nature of resource sharing.

The IPsec requirements for RDDP are based on the version of IPsec specified in RFC 2401 [RFC2401] and related RFCs, as profiled by RFC 3723 [RFC3723], despite the existence of a newer version of IPsec specified in RFC 4301 [RFC4301] and related RFCs [RFC4303], [RFC4306], [RFC4835]. One of the important early applications of the RDDP protocols is their use with iSCSI [iSER]; RDDP's IPsec requirements follow those of IPsec in order to facilitate that usage by allowing a common profile of IPsec to be used with iSCSI and the RDDP protocols. In the future, RFC 3723 may be updated to the newer version of IPsec, and the IPsec security requirements of any such update should apply uniformly to iSCSI and the RDDP protocols.

8.1. Summary of RDMAP-Specific Security Requirements

[RDMASEC] defines the security requirements for the implementation of the components of the RDMA reference model, namely the RDMA enabled NIC (RNIC) and the Privileged Resource Manager. An RDMAP implementation conforming to this specification MUST conform to these requirements.

8.1.1. RDMAP (RNIC) Requirements

RDMAP provides several countermeasures for all types of attacks as introduced in [RDMASEC]. In the following, this specification lists all security requirements that MUST be implemented by the RNIC. A more detailed discussion of RNIC security requirements can be found in Section 5 of [RDMASEC].

1. An RNIC MUST ensure that a specific Stream in a specific Protection Domain cannot access an STag in a different Protection Domain.

2. An RNIC MUST ensure that if an STag is limited in scope to a single Stream, no other Stream can use the STag.

3. An RNIC MUST ensure that a Remote Peer is not able to access memory outside of the buffer specified when the STag was enabled for remote access.

4. An RNIC MUST provide a mechanism for the ULP to establish and revoke the association of a ULP Buffer to an STag and TO range.

5. An RNIC MUST provide a mechanism for the ULP to establish and revoke read, write, or read and write access to the ULP Buffer referenced by an STag.

6. An RNIC MUST ensure that the network interface can no longer modify an Advertised Buffer after the ULP revokes remote access rights for an STag.

7. An RNIC MUST ensure that a Remote Peer is not able to invalidate an STag enabled for remote access, if the STag is shared on multiple streams.

8. An RNIC MUST choose the value of STags in a way difficult to predict. It is RECOMMENDED to sparsely populate them over the full available range.

9. An RNIC MUST NOT enable sharing a Completion Queue (CQ) across ULPs that do not share partial mutual trust.

10. An RNIC MUST ensure that if a CQ overflows, any Streams that do not use the CQ MUST remain unaffected.

11. An RNIC implementation SHOULD provide a mechanism to cap the number of outstanding RDMA Read Requests.

12. An RNIC MUST NOT enable firmware to be loaded on the RNIC directly from an untrusted Local Peer or Remote Peer, unless the Peer is properly authenticated*, and the update is done via a secure protocol, such as IPsec.

    * by a mechanism outside the scope of this specification. The mechanism presumably entails authenticating that the remote ULP has the right to perform the update.

8.1.2. Privileged Resource Manager Requirements

With RDMAP, all reservations of local resources are initiated from local ULPs. To protect from local attacks including unfair resource distribution and gaining unauthorized access to RNIC resources, a Privileged Resource Manager (PRM) must be implemented, which manages all local resource allocation. Note that the PRM must not be provided as an independent component, and its functionality can also be implemented as part of the privileged ULP or as part of the RNIC itself.

A PRM implementation must meet the following security requirements (a more detailed discussion of PRM security requirements can be found in Section 5 of [RDMASEC]):

1. All Non-Privileged ULP interactions with the RNIC Engine that could affect other ULPs MUST be done using the Resource Manager as a proxy.

2. All ULP resource allocation requests for scarce resources MUST also be done using a Privileged Resource Manager.

3. The Privileged Resource Manager MUST NOT assume that different ULPs share Partial Mutual Trust unless there is a mechanism to ensure that the ULPs do indeed share partial mutual trust.

4. If Non-Privileged ULPs are supported, the Privileged Resource Manager MUST verify that the Non-Privileged ULP has the right to access a specific Data Buffer before allowing an STag for which the ULP has access rights to be associated with a specific Data Buffer.

5. The Privileged Resource Manager MUST control the allocation of CQ entries.

6. The Privileged Resource Manager SHOULD prevent a Local Peer from allocating more than its fair share of resources.

7. RDMA Read Request Queue resource consumption MUST be controlled by the Privileged Resource Manager such that RDMAP/DDP Streams that do not share Partial Mutual Trust do not share RDMA Read Request Queue resources.

8. If an RNIC provides the ability to share receive buffers across multiple Streams, the combination of the RNIC and the Privileged Resource Manager MUST be able to detect if the Remote Peer is attempting to consume more than its fair share of resources so that the Local Peer can apply countermeasures to detect and prevent the attack.

8.2. Security Services for RDMAP

RDMAP is using IP-based network services to control, read, and write data buffers over the network. Therefore, all exchanged control and data packets are vulnerable to spoofing, tampering, and information disclosure attacks.

RDMAP Streams that are subject to impersonation attacks or Stream hijacking attacks can be authenticated, have their integrity protected, and be protected from replay attacks. Furthermore, confidentiality protection can be used to protect from eavesdropping.

8.2.1. Available Security Services

The IPsec protocol suite [RFC2401] defines strong countermeasures to protect an IP stream from those attacks. Several levels of protection can guarantee session confidentiality, per-packet source authentication, per-packet integrity, and correct packet sequencing.

RDMAP security may also profit from SSL or TLS security services provided for TCP-based ULPs [RFC4346]. Used underneath RDMAP, these security services also provide for stream authentication, data integrity, and confidentiality. As discussed in [RDMASEC], limitations on the maximum packet length to be carried over the network and potentially inefficient out-of-order packet processing at the data sink make SSL and TLS less appropriate for RDMAP than IPsec.

If SSL is layered on top of RDMAP, SSL does not protect the RDMAP headers. Thus, a man-in-the-middle attack can still occur by modifying the RDMAP header to incorrectly place the data into the wrong buffer, thus effectively corrupting the data stream.

By remaining independent of ULP and LLP security protocols, RDMAP will benefit from continuing improvements at those layers. Users are provided flexibility to adapt to their specific security requirements and the ability to adapt to future security challenges. Given this,

the vulnerabilities of RDMAP to active third-party interference are no greater than any other protocol running over an LLP such as TCP or SCTP.

8.2.2. Requirements for IPsec Services for RDMAP

Because IPsec is designed to secure arbitrary IP packet streams, including streams where packets are lost, RDMAP can run on top of IPsec without any change. IPsec packets are processed (e.g., integrity checked and possibly decrypted) in the order they are received, and an RDMAP Data Sink will process the decrypted RDMA Messages contained in these packets in the same manner as RDMA Messages contained in unsecured IP packets.

The IP Storage working group has defined the normative IPsec requirements for IP Storage [RFC3723]. Portions of this specification are applicable to the RDMAP. In particular, a compliant implementation of IPsec services for RDMAP MUST meet the requirements as outlined in Section 2.3 of [RFC3723]. Without replicating the detailed discussion in [RFC3723], this includes the following requirements:

1. The implementation MUST support IPsec ESP [RFC2406], as well as the replay protection mechanisms of IPsec. When ESP is utilized, per-packet data origin authentication, integrity, and replay protection MUST be used.

2. It MUST support ESP in tunnel mode and MAY implement ESP in transport mode.

3. It MUST support IKE [RFC2409] for peer authentication, negotiation of security associations, and key management, using the IPsec DOI [RFC2407].

4. It MUST NOT interpret the receipt of a IKE Phase 2 delete message as a reason for tearing down the RDMAP stream. Since IPsec acceleration hardware may only be able to handle a limited number of active IKE Phase 2 SAs, idle SAs may be dynamically brought down, and a new SA be brought up again, if activity resumes.

5. It MUST support peer authentication using a pre-shared key, and MAY support certificate-based peer authentication using digital signatures. Peer authentication using the public key encryption methods [RFC2409] SHOULD NOT be used.

6. It MUST support IKE Main Mode and SHOULD support Aggressive Mode. IKE Main Mode with pre-shared key authentication SHOULD NOT be used when either of the peers uses a dynamically assigned IP address.

7. When digital signatures are used to achieve authentication, either IKE Main Mode or IKE Aggressive Mode MAY be used. In these cases, an IKE negotiator SHOULD use IKE Certificate Request Payload(s) to specify the certificate authority (or authorities) that are trusted in accordance with its local policy. IKE negotiators SHOULD check the pertinent Certificate Revocation List (CRL) before accepting a PKI certificate for use in IKE's authentication procedures.

8. Access to locally stored secret information (pre-shared or private key for digital signing) must be suitably restricted, since compromise of the secret information nullifies the security properties of the IKE/IPsec protocols.

9. It MUST follow the guidelines of Section 2.3.4 of [RFC3723] on the setting of IKE parameters to achieve a high level of interoperability without requiring extensive configuration.

Furthermore, implementation and deployment of the IPsec services for RDDP should follow the Security Considerations outlined in Section 5 of [RFC3723].

9. IANA Considerations

This document requests no direct action from IANA. The following consideration is listed here as commentary.

If RDMAP was enabled a priori for a ULP by connecting to a well-known port, this well-known port would be registered for the RDMAP with IANA. The registration of the well-known port will be the responsibility of the ULP specification.

10. References

10.1. Normative References

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

[iSER] 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.

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

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

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

[RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation of ISAKMP", RFC 2407, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC3723] Aboba, B., Tseng, J., Walker, J., Rangan, V., and F. Travostino, "Securing Block Storage Protocols over IP", RFC 3723, April 2004.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[SCTP] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, September 2007.

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

10.2. Informative References

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4346] Dierks, T. and E. Rescorla, "The TLS Protocol Version 1.1", RFC 4346, April 2006.

[RFC4835] Manral, V., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4835, April 2007.

Appendix A. DDP Segment Formats for RDMA Messages

This appendix is for information only and is NOT part of the standard. It simply depicts the DDP Segment format for the various RDMA Messages.

A.1. DDP Segment for RDMA Write

The following figure depicts an RDMA Write, DDP Segment:

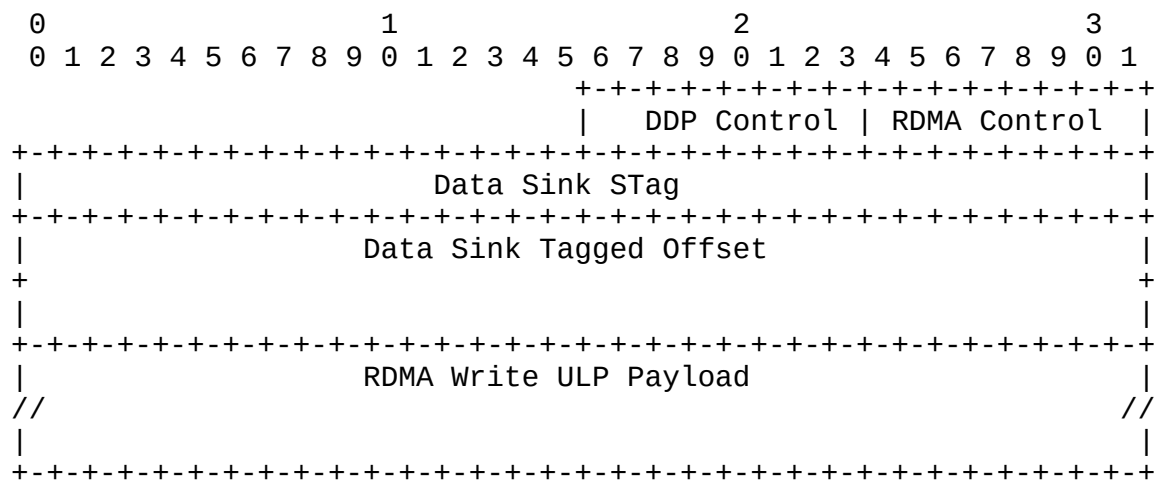

Figure 11: RDMA Write, DDP Segment Format

A.2. DDP Segment for RDMA Read Request

The following figure depicts an RDMA Read Request, DDP Segment:

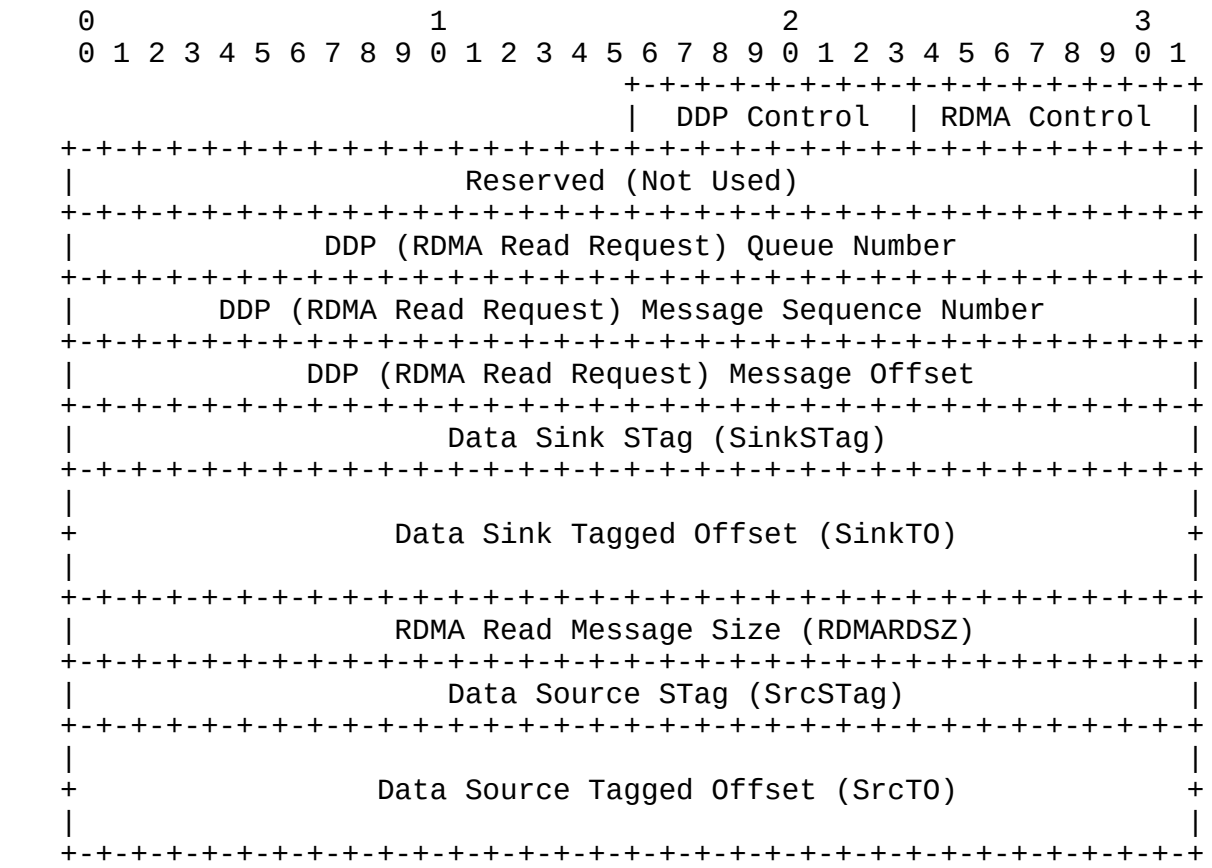

Figure 12: RDMA Read Request, DDP Segment format

A.3. DDP Segment for RDMA Read Response

The following figure depicts an RDMA Read Response, DDP Segment:

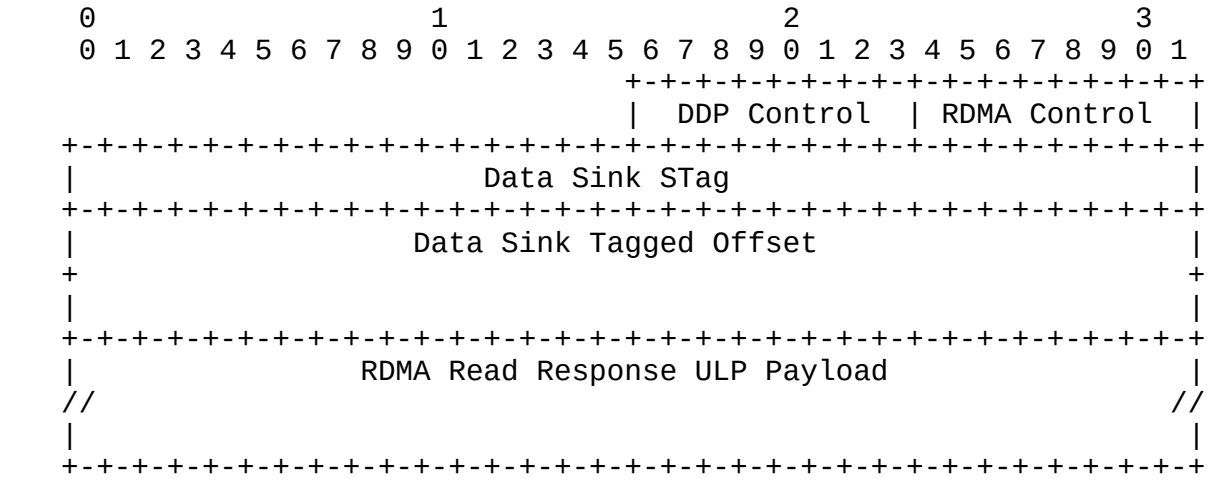

Figure 13: RDMA Read Response, DDP Segment Format

A.4. DDP Segment for Send and Send with Solicited Event

The following figure depicts a Send and Send with Solicited Request, DDP Segment:

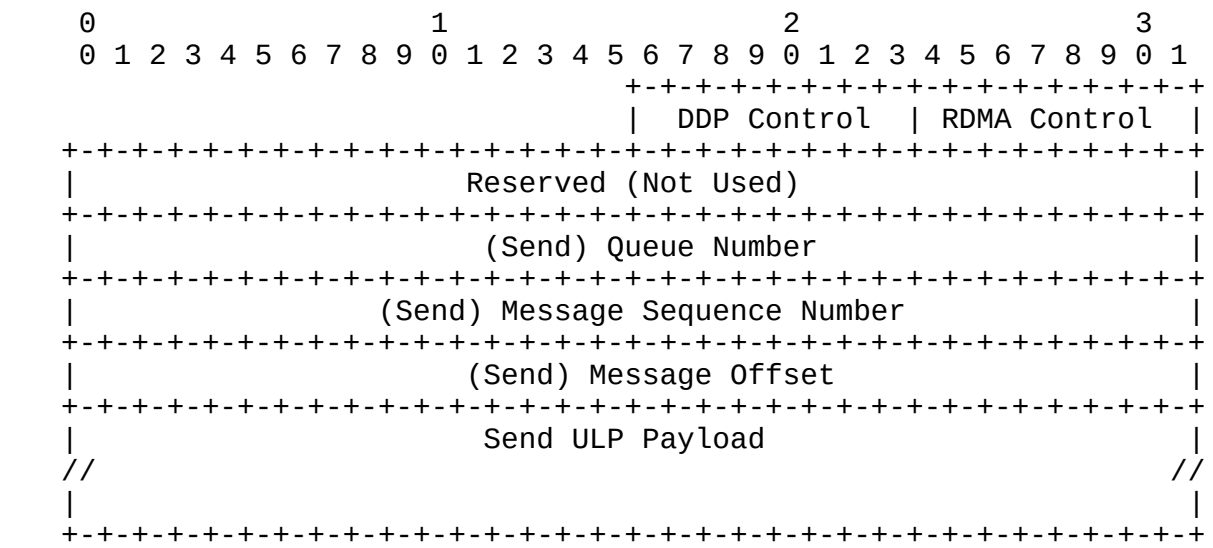

Figure 14: Send and Send with Solicited Event, DDP Segment Format

A.5. DDP Segment for Send with Invalidate and Send with SE and Invalidate

The following figure depicts a Send with Invalidate and Send with Solicited and Invalidate Request, DDP Segment:

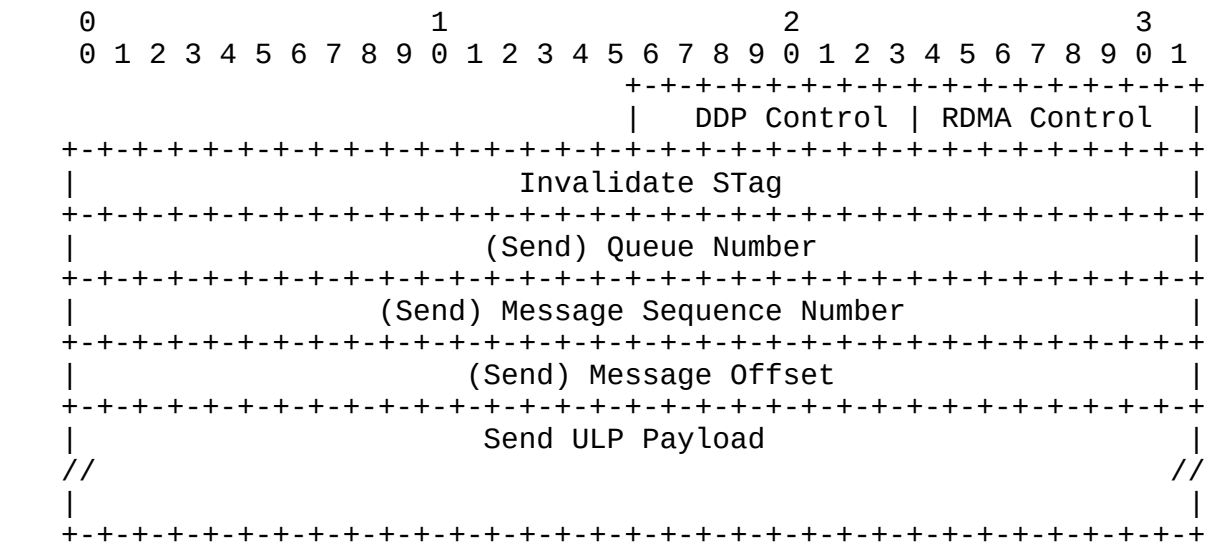

Figure 15: Send with Invalidate and Send with SE and Invalidate, DDP Segment Format

A.6. DDP Segment for Terminate

The following figure depicts a Terminate, DDP Segment:

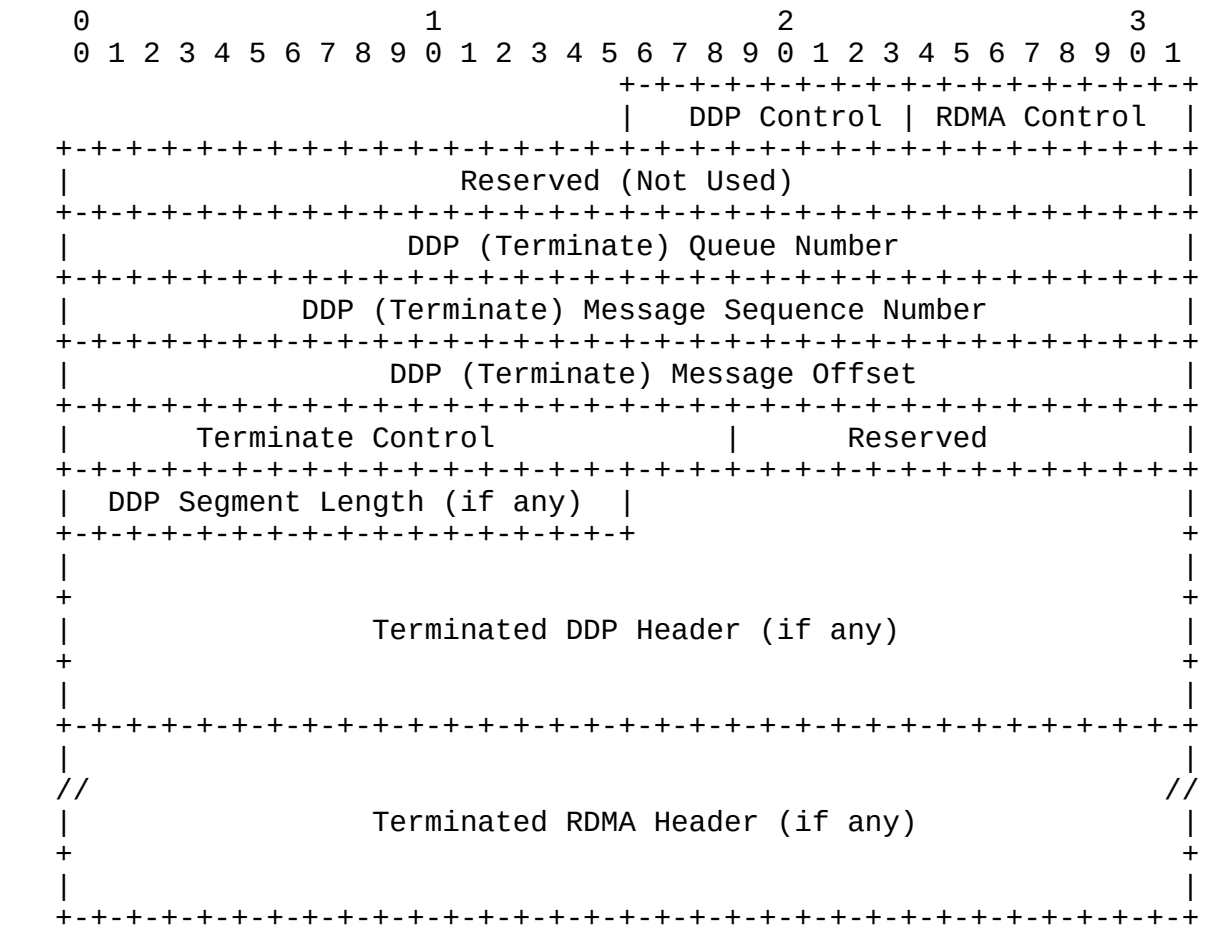

Figure 16: Terminate, DDP Segment Format

Appendix B. Ordering and Completion Table

The following table summarizes the ordering relationships that are defined in Section 5.5, "Ordering and Completions", from the standpoint of the local peer issuing the two Operations. Note that in the table that follows, Send includes Send, Send with Invalidate, Send with Solicited Event, and Send with Solicited Event and Invalidate.

First Op	Later Op	Placement guarantee at Remote Peer	Placement guarantee at Local Peer	Ordering guarantee at Remote Peer
Send	Send	No placement guarantee. If guarantee is necessary, see footnote 1.	Not applicable	Completed in order.
Send	RDMA Write	No placement guarantee. If guarantee is necessary, see footnote 1.	Not applicable	Not applicable
Send	RDMA Read	No placement guarantee between Send Payload and RDMA Read Request Header	RDMA Read Response Payload will not be placed at the local peer until the Send Payload is placed at the Remote Peer	RDMA Read Response Message will not be generated until Send has been Completed
RDMA Write	Send	No placement guarantee. If guarantee is necessary, see footnote 1.	Not applicable	Not applicable

RDMA Write	RDMA Write	No placement guarantee. If guarantee is necessary, see footnote 1.	Not applicable	Not applicable
RDMA Write	RDMA Read	No placement guarantee between RDMA Write Payload and RDMA Read Request Header	RDMA Read Response Payload will not be placed at the local peer until the RDMA Write Payload is placed at the Remote Peer	Not applicable
RDMA Read	Send	No placement guarantee between RDMA Read Request Header and Send payload	Send Payload may be placed at the remote peer before the RDMA Read Response is generated. If guarantee is necessary, see footnote 2.	Not applicable
RDMA Read	RDMA Write	No placement guarantee between RDMA Read Request Header and RDMA Write payload	RDMA Write Payload may be placed at the Remote Peer before the RDMA Read Response is generated. If guarantee is necessary, see footnote 2.	Not applicable

RDMA Read	RDMA Read	No placement guarantee of the two RDMA Read Request Headers Additionally, there is no guarantee that the Tagged Buffers referenced in the RDMA Read will be read in order	No placement guarantee of the two RDMA Read Response Payloads.	Second RDMA Read Response will not be generated until first RDMA Read Response is generated.

Figure 17: Operation Ordering

Footnote 1: If the guarantee is necessary, a ULP may insert an RDMA Read operation and wait for it to complete to act as a Fence.

Footnote 2: If the guarantee is necessary, a ULP may wait for the RDMA Read operation to complete before performing the Send.

Appendix C. Contributors

Dwight Barron
Hewlett-Packard Company
20555 SH 249
Houston, TX 77070-2698 USA
Phone: 281-514-2769
EMail: dwight.barron@hp.com

Caitlin Bestler
Broadcom Corporation
16215 Alton Parkway
Irvine, CA 92619-7013 USA
Phone: 949-926-6383
EMail: caitlinb@broadcom.com

John Carrier
Cray, Inc.
411 First Avenue S, Suite 600
Seattle, WA 98104-2860 USA
Phone: 206-701-2090
EMail: carrier@cray.com

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

Uri Elzur
Broadcom Corporation
16215 Alton Parkway
Irvine, California 92619-7013 USA
Phone: +1 (949) 585-6432
EMail: Uri@Broadcom.com

Hari Ghadia
Gen10 Technology, Inc.
1501 W Shady Grove Road
Grand Prairie, TX 75050
Phone: (972) 301 3630
EMail: hghadia@gen10technology.com

Howard C. Herbert
Intel Corporation
MS CH7-404
5000 West Chandler Blvd.
Chandler, Arizona 85226
Phone: 480-554-3116
EMail: howard.c.herbert@intel.com

Mike Ko
IBM
650 Harry Rd.
San Jose, CA 95120
Phone: (408) 927-2085
EMail: mako@us.ibm.com

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

Dave Minturn
Intel Corporation
MS JF1-210
5200 North East Elam Young Parkway
Hillsboro, Oregon 97124
Phone: 503-712-4106
EMail: dave.b.minturn@intel.com

Mike Penna
Broadcom Corporation
16215 Alton Parkway
Irvine, California 92619-7013 USA
Phone: +1 (949) 926-7149
EMail: MPenna@Broadcom.com

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

Hemal Shah
Broadcom Corporation
5300 California Avenue
Irvine, CA 92617 USA
Phone: +1 (949) 926-6941
EMail: hemal@broadcom.com

Allyn Romanow
Cisco Systems
170 W Tasman Drive
San Jose, CA 95134 USA
Phone: +1 408 525 8836
EMail: allyn@cisco.com

Tom Talpey
Network Appliance
1601 Trapelo Road #16
Waltham, MA 02451 USA
Phone: +1 (781) 768-5329
EMail: thomas.talpey@netapp.com

Patricia Thaler
Broadcom Corporation
16215 Alton Parkway
Irvine, CA 92619-7013 USA
Phone: +1-916-570-2707
EMail: pthaler@broadcom.com

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

Madeline Vega
IBM
11400 Burnet Rd. Bld.45-2L-007
Austin, TX 78758 USA
Phone: 512-838-7739
EMail: mvega1@us.ibm.com

Claudia Salzberg
IBM
11501 Burnet Rd. Bld.902-5B-014
Austin, TX 78758 USA
Phone: 512-838-5156
EMail: salzberg@us.ibm.com

Authors' Addresses

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

Bernard Metzler
IBM Research GmbH
Zurich Research Laboratory
Saeumerstrasse 4
CH-8803 Rueschlikon, Switzerland
Phone: +41 44 724 8605
EMail: bmt@zurich.ibm.com

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

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

Dave Garcia
24100 Hutchinson Rd.
Los Gatos, CA 95033 USA
Phone: +1 (831) 247-4464
Email: Dave.Garcia@StanfordAlumni.org

Full Copyright Statement

Intellectual Property