Network Working Group
Internet-Draft
Expires: August 22, 2003

C. Bestler

February 21, 2003

# Applicability of Remote Direct Memory Access Protocol (RDMA) and Direct Data Placement (DDP)

**draft-bestler-rddp-applicability-00.txt**

## Status of this Memo



## Copyright Notice



## Abstract

This document describes the applicability of Remote Direct Memory Access Protocol (RDMAP) and the Direct Data Placement Protocol (DDP).

Table of Contents

## 1. Introduction

They provide for application independent efficient placement of application payload into Upper Layer Protocol (ULP) specified buffers. DDP can use multiple standard IP transports including SCTP and TCP. RDMAP provides RDMA services, on top of DDP. This document contrasts the applicability of RDMAP/DDP versus direct use of the underlying IP transports, and versus non-IP transports designed specifically with RDMA capabilities.

The applicability of RDMAP/DDP is driven by their unique capabilities:

- The existence of an application independent protocol allows common solutions to be implemented in hardware and/or the kernel. This document will discuss when common data placement procedures are of the greatest benefit to applications as contrasted with direct use of the underlying transport.

- DDP supports both untagged and tagged buffers. Tagged buffers allow the Data Sink ULP to be indifferent to what order (or in what packets) the Data Source delivered the data. This document will discuss when Data Source flexibility is of benefit to applications.

DDP works over standard unmodified IP transports, such as SCTP. Some non-IP transports, such as InfiniBand, directly integrate RDMA features. This document will review the applicability of providing RDMA services over ubiquitous IP transports as opposed to the use of customized transport protocols.

RDMAP defines RDMA Reads, which allow remote access to advertised buffers. This document will review the advantages of using RDMA Reads as contrasted to alternate solutions.

The full capabilities of DDP and RDMAP can only be fully realized by applications that are designed to exploit them. The co-existence of RDMAP/DDP aware local interfaces with traditional socket interfaces will also be explored.

Finally, DDP support is defined for at least two IP transports: SCTP and TCP. The rationale for supporting both transports is reviewed, as well as when each would be the appropriate selection.

## 2. Definitions

Advertisement - 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.

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.

## 3. Conventions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC2119 [1].

## 4. Direct Placement

Direct Data Placement optimizes the placement of ULP payload into the correct destination buffers while minimizing the required ULP interactions and typically eliminating intermediate copying. This capability is most valuable for applications that require multiple transport layer packets for each required ULP interaction.

While reducing the number of required ULP interactions is in itself desirable, it is critical for high speed connections. The burst packet rate for a high speed interface could easily exceed the host systems ability to switch ULP contexts.

Content access applications are primary examples of applications with both high bandwidth and high content to required ULP interaction ratios. These applications include file access protocols (NAS), storage access (SAN), database access and other application specific forms of content access such as HTTP, XML and email.

The degree to which this is an optimization depends on which transport is being compared with, and on the nature of the local interface. Pre-posting of receive buffers allows direct placement of incoming data. However pre-posting buffers requires the receiving side to accurately predict the required buffers and their sizes. This is not feasible for all ULPs. By contrast, DDP only requires the ULP to predict the sequence and size of incoming untagged messages.

Direct Data Placement can be achieved without RDMA. Pre-posting of receive buffers allows any network stack to place data directly to user buffers.

An application that could predict incoming messages and required nothing more than direct placement into buffers might be able to do so with a properly designed local interface to SCTP or TCP. Doing so for TCP requires making predictions at a byte level rather than a message level.

The main benefit of DDP for such an application would be that pre-posting of receive buffers is a mandated local interface capability, and that predictions can be made on a per-message basis (not per byte).

## 5. Tagged Buffers

A more critical advantage of DDP is the ability of the Data Source to use tagged buffers. Tagging transfers allows the Data Source to choose the ordering and packetization of its payload deliveries. With direct data placement, the packetization and delivery of payload must be agreed by the ULP peers. Even if there is an encoding of what is being transferred, as is common with middleware solutions, this information is not understood at the application independent layers. The directions on where to place the incoming data cannot be accessed without switching to the application layer first. DDP provides a standardized 'packing list' which can be interpreted without application layer involvement. Indeed, it is designed to be implementable in hardware.

The use of a standardized 'packing list' minimizes the required interactions with the ULP. This can be extremely beneficial for applications that use multiple transport layer packets to accomplish what is a single ULP interaction.

There are other benefits of tagged buffers covered in later sections.

An application that had no opportunity to use tagged buffers would derive virtually no benefit from the use of DDP as opposed to SCTP. But while tagged buffers are the justification for DDP, DDP still relies on untagged buffers. Without them the only method to exchange buffer advertisements would involve out-of-band communications and/or sharing of compile time constants. However, most ULP protocols built upon RDMA transports continue to use untagged buffers for requests and responses.

Limiting use of untagged buffers to requests and responses by moving all bulk data using tagged transfers can greatly simplify the amount of prediction that the Data Sink must perform in pre-posting receive buffers. For example, a typical DDP enabled exchange would consist of the following operations:

Client sends transaction request to server's untagged buffer.

This request includes buffer advertisements for the buffers where it wants the results to be placed.

Server performs multiple tagged puts to the advertised buffers.

Server sends transaction reply to client's untagged buffer.

With this type of exchange the pacing and required size of untagged buffers is highly predictable. The variability of response sizes is

absorbed by tagged transfers.

Use of tagged transfers is especially applicable when the Data Sink does not know the actual size, structure or location of the content it is requesting (or updating).

For example, suppose the Data Sink ULP needs to fetch four related pieces of data into a four separate buffers. With SCTP the Data Sink ULP could receive four messages into four separate buffers, only having to predict the maximum size of each. However it would have to dictate the order in which the Data Source supplied the separate pieces. If the Data Source found it advantageous to fetch them in a different order it would have to use intermediate buffering to re-order the pieces into the expected order even though the application only required that all four be delivered and did not truly have an ordering requirement.

Techniques such as RAID striping and mirroring represent this same problem, but one step further. What appears to be a single resource to the Data Sink is actually stored in separate locations by the Data Source. Non-DDP protocols would either require the Data Source to fetch the material in the desired order or force the Data Source to use its own holding buffers to assemble an image of the destination buffer.

While sometimes referred to as a "buffer-to-buffer" solution, RDMA more fundamentally enables remote buffer access. The ULP is free to work with larger remote buffers than it has locally. This reduces buffering requirements and the number of times the data must be copied in an end-to-end transfer.

There are numerous reasons why the Data Sink would not know the true order or location of the requested data. It could be different for each client, different records selected and/or different sort orders, RAID striping, file fragmentation, volume fragmentation, volume mirroring and server-side dynamic compositing of content (such as server side includes for HTTP).

In all of these cases the Data Source is free to assemble the desired data in the Data Sinks buffer in whatever order the component data becomes available to it. It is not constrained on ordering. It does not have to assemble an image in its own memory before creating it in the Data Sink's buffers.

Note that while DDP enables use of tagged messages for bulk transfer, there are some application scenarios where untagged messages would still be used for bulk transfer. For example, under the Direct Access File Server (DAFS) protocol the file server does not expose

its own memory to its clients. A client wishing to write may advertise a buffer which the server will issue RDMA Reads upon. However, when performing a small write it may be preferable to include the data in the untagged message rather than incurring an additional round trip with the RDMA Read and its response.

## 6. Tagged Buffers as ULP Credits

The handling of end-to-end buffer credits differs considerably with DDP than when the ULP directly uses either TCP or SCTP.

With both TCP and SCTP buffer credits are based upon the receiver granting transmit permission based on the total number of bytes. These credits reflect system buffering resources and/or simple flow control. They do not represent ULP resources.

DDP defines no standard flow control, but presumes the existince of a ULP mechanism. The presumed mechanism is that the Data Sink ULP has issued credits to the Data Source allowing the Data Source to send a specific number of untagged messages.

The ULP peers must ensure that the sender is aware of the maximum size that can be sent to any specific target buffer. One method of doing so is to use a standard size for all untagged buffers within a given connection. For example, DAFS specifies an initial size requirement for session establishment, during which the untagged buffer size for the remainder of the session is negotiated.

Tagged buffers are ULP resources advertised directly from ULP to ULP. A DDP put to a known tagged buffer is constrained only by transport level flow control, not by available system buffering.

Either tagged or untagged buffers allows bypassing of system buffer resources. Use of tagged buffers additionally allows the Data Source to choose what order to exercise the credits in.

To the extent allowed by the ULP, tagged buffers are also divisible resources. The Data Sink can advertise a single 100 KB buffer, and then receive notifications from its peer that it had written 50 KB, 20 KB and 30 KB to that buffer in three successive transactions.

ULP-management of tagged buffer resources, independent of transport and DDP layer credits, is an additional benefit of RDMA protocols. Large bulk transfers cannot be blocked by limited general purpose buffering capacity. Applications can flow control based upon higher level abstractions, such as number of outstanding requests, independent of the amount of data that must be transferred.

However, use of system buffering, as offered by direct use of the underlying transports, can be preferable under certain circumstances.

One example would be when the number of target ULP buffers is sufficiently large, and the rate at which any writes arrive is sufficiently low, that pinning all the target ULP buffers in memory

would be undesirable. The maximum transfer rate, and hence the maximum amount of system buffering required, may be more stable and predictable than the total ULP buffer exposure.

Another would be the Data Sink wishes to receive a stream of data at a predictable rate, but does not know in advance what the size of each data packet will be. This is common from streaming media that has been encoded with a variable bit rate. With DDP the Data Sink would either have to use untagged buffers large enough for the largest packet, or advertise a circular buffer. If for security or other reasons the Data Sink did not want the size of its buffer to be publicly known, using the underlying SCTP transport directly may be preferable because of their byte-oriented credits.

## 7. RDMA Read

RDMAP/DDP is more than just a "buffer-to-buffer" solution. A simple buffer transfer protocol could have been designed to efficiently transfer buffers. RDMA Reads allow the Data Sink to fetch exactly the portion of the peer ULP buffer required on a "just in time" basis. Further, this can be done without requiring per-fetch support from the Data Source ULP.

Storage servers typically have a maximum write buffer. There is little benefit in transferring data from the Data Source far in advance of when it will be written to the persistent storage media. In this fashion a relatively small number of block sized buffers can be used to execute a single transaction that specified writing a large file.

This same capability can be used when the desired portion of the advertised buffer is not known in advance. For example the advertised buffer could contain performance statistics. The data sink could request the portions of the data it required, without requiring an interaction with the Data Source ULP.

This is applicable for many applications that publish semi-volatile data that does not require transactional validity checking (i.e., authorized users have read access to the entire set of data). It is less applicable when there are ULP consistency checks that must be performed upon the data. Such applications would be better served by having the client send a request, and having the server use RDMA Writes to publish the requested data. Neither RDMAP or DDP provide mechanisms for bundling multiple disjoint updates into an atomic transaction. Therefore use of an advertised buffer as a data resource is subject to the same caveats as any randomly updated data resource, such as flat files, that do not enforce their own referential integrity.

## 8. Specialized Transports

DDP is defined to operate over ubiquitous IP transports such as SCTP and TCP. This enabled a new DDP-enabled node to be added anywhere to an IP network. No DDP-specific support from middle-boxes is required.

There are non-IP transport fabric offering RDMA capabilities. Because these capabilities are integrated with the transport protocol they have some technical advantages when compared to RDMA over IP. For example fencing of RDMA operations can be based upon transport level acks. Because DDP is cleanly layered over an IP transport, any explicit RDMA layer ack must be separate from the transport layer ack.

There may be deployments where the benefits of RDMA/transport integration outweigh the benefits of being on an IP network.

## 9. Local Interface Implications

Full utilization of DDP and RDMAP capabilities requires a local interface that explicitly requests these services. Protocols such as Sockets Direct Protocol (SDP) can allow applications to keep their traditional byte-stream or message-stream interface and still enjoy many of the benefits of the optimized wire level protocols.

## 10. Comparison of IP Transports

It is the responsibility of the ULP to determine which IP transport is best suited to its needs.

SCTP provides for preservation of message boundaries. Each DDP segment will be delivered within a single SCTP packet. The equivalent services are only available with TCP through the use of the MPA adaptation layer.

SCTP also provides multi-streaming. When the same pair of hosts have need for multiple DDP streams this can be a major advantage. A single SCTP association carries multiple DDP streams, consolidating connection setup and flow control.

Even with the MPA adaptation layer, DDP traffic will appear to all network traffic as normal TCP connection. In many environmenets there may be a requirement to use only TCP connections to satisfy existing network elements and/or to facilitate monitoring and control of connections.

A DDP stream delivered via MPA/TCP will require more processing effort than one delivered over SCTP. However this extra work may be justified for many deployments where full SCTP support is unavailable in the intermediate network.

## References

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

## Author's Address

Caitlin Bestler
1241 W. North Shore
# 2G
Chicago, IL 60626
USA

Phone: +1-773-743-1594
EMail: cait@asomi.com

## Full Copyright Statement



## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.