Applicability of Remote Direct Memory Access Protocol (RDMA) and Direct
                       Data Placement (DDP)
                draft-ietf-rddp-applicability-08.txt

Status of this Memo

   By submitting this Internet-Draft, each author represents that any
   applicable patent or other IPR claims of which he or she is aware
   have been or will be disclosed, and any of which he or she becomes
   aware will be disclosed, in accordance with Section 6 of BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt.

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

   This Internet-Draft will expire on December 23, 2006.

Copyright Notice

Abstract

   This document describes the applicability of Remote Direct Memory
   Access Protocol (RDMAP) and the Direct Data Placement Protocol (DDP).
   It compares and contrasts the different transport options over IP
   that DDP can use, provides guidance to ULP developers on choosing
   between available transports and/or how to be indifferent to the
   specific transport layer used, compares use of DDP with direct use of

   the supporting transports, and compares DDP over IP transports with
   non-IP transports that support RDMA functionality.

Table of Contents

1. Introduction

   Remote Direct Memory Access Protocol (RDMAP) and Direct Data
   Placement (DDP) work together to provide application independent
   efficient placement of application payload directly into buffers
   specified by the Upper Layer Protocol (ULP).

   The DDP protocol is responsible for direct placement of received
   payload into ULP specified buffers. The RDMAP protocol provides
   completion notifications to the ULP and support for Data Sink
   initiated fetch of advertised buffers (RDMA Reads).

   DDP and RDMAP are both application independent protocols which allow
   the ULP to perform remote direct data placement. DDP can use
   multiple standard IP transports including SCTP and TCP.

   By clarifying the situations where the functionality of these
   protocols are applicable, this document can guide implementers,
   application and protocol designers in selecting which protocols to
   use.

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

   o 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
      application specific solutions built on top of direct use of the
      underlying transport.

   o DDP supports both untagged and tagged buffers. Tagged buffers
      allow the Data Sink ULP to be indifferent to what order (or in
      what messages) the Data Source sent the data, or what order
      packets are received in. Typically tagged data can be used for
      payload transfer, while untagged is best used for control
      messages. However each upper layer protocol can determine the
      optimal use of tagged and untagged messages for itself. This
      document will discuss when Data Source flexibility is of benefit
      to applications.

   o RDMAP consolidates ULP notifications, thereby minimizing the
      number of required ULP interactions.

   o 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.

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. Due to the fact that DDP is defined
cleanly as a layer over existing IP transports, DDP has simpler
ordering rules than some prior RDMA protocols. This may have some
implications for application designers.

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.

   Lower Layer Protocol (LLP) The transport protocol that provides
      services to DDP. This is an IP transport with any required
      adaptation layer. Adaptation layers are defined for SCTP and TCP.

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

   Tagged Message A DDP message that is directed to a ULP specified
      buffer based upon imbedded addressing information. In the
      immediate sense, the destination buffer is specified by the
      message sender. The message receiver is given no independent
      indication that a tagged message has been received.

   Untagged Message A DDP message that is directed to a ULP specified
      buffer based upon a Message Sequence Number being matched with a
      receiver supplied buffer. The destination buffer is specified by
      the message receiver. The message receiver is notified by some
      mechanism that an untagged message has been received.

   Upper Layer Protocol (ULP) The direct user of RDMAP/DDP services. In
      addition to protocols such as iSER [[7](#)] and NFSv4 over RDMA [[8](#)],
      the ULP may be embedded in an application, or a middleware layer
      as is often the case for the Sockets Direct Protocol (SDP) and
      Remote Procedure Call (RPC) protocols.

3. Direct Placement

   Direct Data Placement optimizes the placement of ULP payload into the
   correct destination buffers, typically eliminating intermediate
   copying. Placement is enabled without regard to order of arrival,
   order of transmission or requiring per-placement interaction with the
   ULP.

   RDMAP minimizes the required ULP interactions . This capability is
   most valuable for applications that require multiple transport layer
   packets for each required ULP interaction.

3.1. Direct Placement using only the LLP

   Direct data placement can be achieved without RDMA. Pre-posting of
   receive buffers could allow a non-RDMA network stack to place data
   directly to user buffers.

   The degree to which DDP optimizes depends on which transport is being
   compared with, and on the nature of the local interface. Without
   RDMAP/DDP 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.

   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 native SCTP or TCP
   (without RDMA). This is easier using native SCTP because the
   application would only have to predict the sequence of messages and
   the maximum size of each message, not the exact size of each message.

   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 always be made on a per-message basis (not
   per byte).

   The Lower Layer Protocol, LLP, can also be used directly if ULP
   specific knowledge is built into the protocol stack to allow "parse
   and place" handling of received packets. Such a solution either
   requires interaction with the ULP, or that the protocol stack have
   knowledge of ULP specific syntax rules.

   DDP achieves the benefits of directly placing incoming payload
   without requiring tight coupling between the ULP and the protocol
   stack. However, "parse and place" capabilities can certainly provide
   equivalent services to a limited number of ULPs.

3.2. Fewer Required ULP Interactions

   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 a high ratio of content transferred per
   required ULP interaction. 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.

[4](#). **Tagged Messages**

This section covers the major benefits from the use of Tagged Messages.

A more critical advantage of DDP is the ability of the Data Source to use tagged buffers. Tagging messages allows the Data Source to choose the ordering and packetization of its payload deliveries. With direct data placement based solely upon pre-posted receives, the packetization and delivery of payload must be agreed by the ULP peers in advance.

The Upper Layer Protocol can allocate content between untagged and/or tagged messages to maximize the potential optimizations. Placing content within an untagged message can deliver the content in the same packet that signals completion to the receiver. This can improve latency. It can even eliminate round trips. But it requires making larger anonymous buffers to be available.

Some examples of data that typically belongs in the untagged message would include:

   short fixed-size control data that is inherently part of the control message. This is especially true when the data is a required part of the control message.

   relatively short payload that is almost always needed, especially when its inclusion would eliminate a round-trip to fetch the data. Examples would include the initial data on a write request and advertisements of tagged buffers.

Tagged messages standardizes direct placement of data without per-packet interaction with the upper layers. Even if there is an upper layer protocol 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 ULP first. DDP provides a standardized 'packing list' which can be interpreted without requiring ULP interaction. Indeed, it is designed to be implementable in hardware.

[4.1](#). **Order Independent Reception**

Tagged messages are directed to a buffer based on an included Steering Tag. Additionally, no notice is provided to the ULP for each individual Tagged Message's arrival. Together these allow tagged messages received out-of-order to be processed without intermediate buffering or additional notifications to the ULP.

4.2. Reduced ULP Notifications

   RDMAP offers both tagged and untagged messages. No receiving side
   ULP interactions are required for tagged messages. By optimally
   dividing traffic between tagged and untagged messages the ULP can
   limit the number of events that must be dealt with at the ULP layer.
   This typically reduces the number of context switches required and
   improves performance.

   RDMAP further reduces required ULP interactions consolidating
   completion notifications of tagged messages with the completion
   notification of a trailing untagged message. For most ULPs this
   radically reduces the number of ULP required interactions even
   further.

   While RDMAP consolidation of notices is beneficial to most
   applications, it may be detrimental to some applications that benefit
   from streamed delivery to enable ULP processing of received data as
   promptly as possible. A ULP that uses RDMAP cannot begin processing
   any portion of an exchange until it receives notification that the
   entire exchange has been placed. An "exchange" here is a set of zero
   or more tagged messages and a single terminating untagged message.
   An application that would prefer to begin work on the received
   payload, no matter what order it arrived in, as soon as possible
   might prefer to work directly with the LLP. RDMAP is optimized for
   applications that are more concerned when the entire exchange is
   complete.

   An application that benefits from being able to begin processing of
   each received packet as quickly as possible may find RDMAP interferes
   with that goal.

   Such an application might be able to retain most of the benefits of
   RDMAP by using the DDP layer directly. However, in addition to
   taking on the responsibilities of the RDMAP layer, the application
   would likely have more difficulty finding support for a DDP-only API.
   Many hardware implementations may choose to tightly couple RDMAP and
   DDP, and might not provide an API directly to DDP services.

   These features minimize 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.

4.3. Simplified ULP Exchanges

   The notification rules for Tagged Messages allows ULPs to create
   multi-message "exchanges" consisting of zero or more tagged messages

that represent a single step in the ULP interaction. The receiving
ULP is notified that the untagged message has arrived, and implicitly
of any associated tagged messages.

A ULP where all exchanges would naturally be untagged messages would
derive virtually no benefit from the use of RDMAP/DDP as opposed to
SCTP directly. But while tagged buffers are the justification for
RDMAP/DDP, untagged buffers are still necessary. Without untagged
buffers the only method to exchange buffer advertisements would
require out-of-band communications. Most RDMA-aware ULPs use
untagged buffers for requests and responses. Buffer advertisements
are typically done within these untagged messages.

More importantly there would be no reliable method for the upper
layer peers to synchronize. The absence of any guarantees about
ordering within or between tagged messages is fundamental to allowing
the DDP layer to optimize transfer of tagged payload.

So no ULP can be defined entirely in terms of tagged messages.
Eventually a notification that confirms delivery must be generated
from the RDMAP/DDP layer.

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 RDMA enabled interaction would
consist of the following:

   Client sends transaction request to server's as an untagged
   message.

   This message includes buffer advertisements for the buffers where
   the results are to be placed.

   The Server sends multiple tagged messages to the advertised
   buffers.

   The Server sends transaction reply as an untagged message to the
   client.

   Client receives single notification, indicating completion of the
   interaction.

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.

4.4. Order Independent Sending

   Use of tagged messages 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 RDMA 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 Sink's 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, a file server may 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.

Generally, the best use of an untagged message is to synchronize and
to deliver data that is naturally tied to the same message as the
synchronization. For initial data transfers this has the additional
benefit of avoiding the need to advertise specific tagged buffers for
indefinite time periods. Instead anonymous buffers can be used for
initial data reception. Because anonymous buffers do not need to be
tied to specific messages in advance this can be a major benefit.

4.5. Untagged Messages and 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, a ULP may specify an initial untagged
buffer size to be used immediately after session establishment, and
then optionally specify mechanisms for negotiating changes.

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.

5. **RDMA Read**

   RDMA Reads are a further service provided by RDMAP. RDMA Reads allow
   the Data Sink to fetch exactly the portion of the peer ULP buffer
   required on a "just in time" basis. This can be done without
   requiring per-fetch support from the Data Source ULP.

   Storage servers may wish to limit the maximum write buffer allocated
   to any single session. The storage server may be a very minimal
   layer between the client and the disk storage media, or the server
   may merely wish to limit the total resources that would be required
   if all clients could push the entire payload they wished written at
   their own convenience.

   In either case, 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. RDMA Reads allow the Storage Server to
   fetch the payload on a "just in time" basis. 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, or a
   Storage Server with numerous clients can fetch buffers from the
   individual clients in the order that is most convenient to the
   server.

   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
   operation. 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 consistency.

6. LLP Comparisons

   Normally the choice of underlying IP transport is irrelevant to the
   ULP. RDMAP and DDP provides the same services over either. There
   may be performance impacts of the choice, however. 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 (Marker PDU Alignment) adaptation layer.

6.1. Multistreaming Implications

   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, congestion control and acknowledgements.

   Completions are controlled by the DDP Source Sequence Number (DDP-
   SSN) on a per stream basis. Therefore combining multiple DDP Streams
   into a single SCTP association cannot result in a dropped packet
   carrying data for one stream delaying completions on others.

6.2. Out of Order Reception Implications

   The use of unordered Data Chunks with SCTP guarantees that the DDP
   layer will be able to perform placements when IP datagrams are
   received out of order.

   Placement of out-of-order DDP Segments carried over MPA/TCP is not
   guaranteed, but certainly allowed. The ability of the MPA receiver
   to process out-of-order DDP Segments may be impaired when alignment
   of TCP segments and MPA FPDUs is lost. Using SCTP, each DDP Segment
   is encoded in a single Data Chunk and never spread over multiple IP
   datagrams.

6.3. Header and Marker Overhead

   MPA and TCP headers together are smaller than the headers used by
   SCTP and its adaptation layer. However, this advantage can be
   reduced by the insertion of MPA markers. The different in ULP
   payload per IP Datagram is not likely to be a signifigant factor.

6.4. Middlebox Support

   Even with the MPA adaptation layer, DDP traffic carried over MPA/TCP

will appear to all network middleboxes as a normal TCP connection.
In many environments there may be a requirement to use only TCP
connections to satisfy existing network elements and/or to facilitate
monitoring and control of connections. While SCTP is certainly just
as monitorable and controllable as TCP, there is no guarantee that
the network management infrastructure has the required support for
both.

6.5. Processing Overhead

A DDP stream delivered via MPA/TCP will require more processing
effort that one delivered over SCTP. However this extra work may be
justified for many deployments where full SCTP support is unavailable
in the endpoints of the network, or where middleboxes impair the
usability of SCTP.

6.6. Data Integrity Implications

Both the SCTP and MPA/TCP adaptation provide end-to-end CRC32c
protection against data accidental corruption, or its equivalent.

A ULP that requires a greater degree of protection may add it own.
However, DDP and RDMAP headers will only be guaranteed to have the
equivalent of end-to-end CRC32c protection. A ULP that requires data
integrity checking more thorough than an end-to-end CRC32c should
first invalidate all STags that reference a buffer before applying
their own integrity check.

CRC32c only provides protection against random corruption. To
protect against unauthorized alteration or forging of data packets
security methods must be applied. The security draft [RDMA-Security]
[6] specifies usage of RFC2406 [1] for both adaptation layers.

6.6.1. MPA/TCP Specifics

It is mandatory for MPA/TCP implementations to implement CRC32c, but
it is NOT mandatory to use the CRC32c during an RDMA connection. The
activating or deactivating of the CRC in MPA/TCP is an administrative
configuration operation at the local and remote end. The
administration of the CRC(ON/OFF) is invisible to the ULP.

Applications SHOULD trust that this administrative option will only
be used when the end-to-end protection is at least as effective as a
transport layer CRC32c. Applications SHOULD NOT apply additional
protection as a guard against this administrative option being turned
on inadvertently.

Administrators MUST NOT enable CRC32c suppression unless the end-to-

end protection is truly equivalent.

If the CRC is active/used for one direction/end , then the use of the CRC is mandatory in both directions/ends.

If both ends have been configured NOT to use the CRC, then this is allowed as long as an equivalent protection(comparable or better than/to CRC) from undetected errors on the connection is provided.

6.6.2. SCTP Specifics

SCTP provides CRC32c protection automatically. The adaptation to SCTP provides for no option to suppress SCTP CRC32c protection.

6.7. Non-IP 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.

6.7.1. No RDMA Layer Ack

DDP does not provide for its own acknowledgements. The only form of ack provided at the RDMAP layer is an RDMA Read Response. DDP and RDMAP rely almost entirely upon other layers for flow control and pacing. The LLP is relied upon to guarantee delivery and avoid network congestion, and ULP level acking is relied upon for ULP pacing and to avoid ULP buffer overruns.

Previous RDMA protocols, such as InfiniBand, have been able to use their integration with the transport layer to provide stronger ordering guarantees. It is important that application designers that require such guarantees to provide them through ULP interaction.

Specifically:

There is no ability for a local interface to "fence" outbound
messages to guarantee that prior tagged messages have been placed
prior to sending a tagged message. The only guarantees available
from the other side would be an RDMA Read Response (coming from
the RDMAP layer) or a response from the ULP layer. Remember that
the normal ordering rules only guarantee when the Data Sink ULP
will be notified of untagged messages, it does not control when
data is placed into receive buffers.

Re-use of tagged buffers must be done with extreme care. The fact
that an untagged message indicates that all prior tagged messages
have been placed does not guarantee that no later tagged message
have. The best strategy is to only change the state of any given
advertised buffers with with untagged messages.

As covered elsewhere in this document, flow control of untagged
messages MUST be provided by the ULP itself.

6.8. Other IP Transports

Both TCP and SCTP provide DDP with reliable transport with TCP
friendly rate control. As currently DDP is defined to work over
reliable transports and implicitly relies upon some form of rate
control.

DDP is fully compatible with a non-reliable protocol. Out-of-order
placement is obviously not dependent on whether the other DDP
Segments ever actually arrive.

However, RDMAP requires the LLP to provide reliable service. An
alternate completion handling protocol would be required if DDP were
to be deployed over an unreliable IP transport.

As noted in the prior section on tagged buffers as ULP credits,
neither RDMAP or DDP provide any flow control for tagged messages.
If no transport layer flow control is provided, an RDMAP/DDP
application would be only limited by the link layer rate, almost
inevitably resulting in severe network congestion.

RDMAP encourages applications to be ignorant of the underlying
transport PMTU. The ULP is only notified when all messages ending in
a single untagged message have completed. The ULP is not aware of
the granularity or ordering of the underlying message. This approach
assumes that the ULP is only interested in the complete set of
messages, and has no use for a subset of them.

6.9. **LLP Independent Session Establishment**

   For an RDMAP/DDP application, the transport services provided by a
   pair of SCTP Streams and by a TCP connection both provide the same
   service (reliable delivery of DDP Segments between two connected
   RDMAP/DDP endpoints).

6.9.1. **RDMA-only Session Establishment**

   It is also possible to allow for transport neutral establishment of
   RDMAP/DDP sessions between endpoints. Combined, these two features
   would allow most applications to be unconcerned as to which LLP was
   actually in use.

   Specifically, the procedures for DDP Stream Session establishment
   discussed in section 3 of the SCTP mapping, and section 13.3 of the
   MPA/TCP mapping, both allow for the exchange of ULP specific data
   ("Private Data") before enabling the exchange of DDP Segments. This
   delay can allow for proper selection and/or configuration of the
   endpoints based upon the exchanged data. For example, each DDP
   Stream Session associated with a single client session might be
   assigned to the same DDP Protection Domain.

   To be transport neutral, the applications should exchange Private
   Data as part of session establishment messages to determine how the
   RDMA endpoints are to be configured. One side must be the Initiator,
   and the other the Responder.

   With SCTP, a pair of SCTP streams can be used for successive sessions
   while the SCTP association remains open. With MPA/TCP each
   connection can be used for at most one session. However, the same
   source/destination pair of ports can be re-used sequentially subject
   to normal TCP rules.

   Both SCTP and MPA limit the private data size to a maximum of 512
   bytes.

   MPA/TCP requires the end of the TCP connection that initiated the
   conversion to MPA mode to send the first DDP Segment. SCTP does not
   have this requirement. ULPs which wish to be transport neutral
   should require the initiating end to send the first message. A zero-
   length RDMA Write can be used for this purpose if the ULP logic
   itself does naturally support this restriction.

6.9.2. **RDMA-Conditional Session Establishment**

   It is sometimes desirable for the active side of a session to connect
   with the passive side before knowing whether the passive side

supports RDMA.

This style of session establishment can be supported with either TCP
or SCTP, but not as transparently as for RDMA-only sessions. Pre-
existing non-RDMA servers are also far more likely to be using TCP
than SCTP.

With TCP. a normal TCP connection is established. It is then used by
the ULP to determine whether or not to convert to MPA mode and use
RDMA. This will typically be integral with other session
establishment negotiations.

With SCTP, the establishment of an association tests whether RDMA is
supported. If not supported, the application simply requests the
association without the RDMA adaptation indication.

One key difference is that with SCTP the determination as to whether
the peer can support RDMA is made before the transport layer
association/connection is established while with TCP the established
connection itself is used to determine whether RDMA is supported.

7. 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.

8. IANA Considerations

There are no IANA considerations in this document.

9. Security considerations

   RDMA security considerations are discussed in [RDMA-SEC] [6]. This
   document will only deal with the more usage oriented aspects, and
   where there are implications in the choice of underlying transport.

9.1. Connection/Association Setup

   Both the SCTP and TCP adaptations allow for existing procedures to be
   followed for the establishment of the SCTP association or TCP
   connection. Use of DDP does not impair the use of any security
   measures to filter, validate and/or log the remote end of an
   association/connection.

9.2. Tagged Buffer Exposure

   DDP only exposes ULP memory to the extent explicitly allowed by ULP
   actions. These include posting of receive operations and enabling of
   Steering Tags.

   Neither RDMAP or DDP place requirements on how ULP's advertise
   buffers. A ULP may use a single Steering Tag for multiple buffer
   advertisements. However, the ULP should be aware that enforcement on
   STag usage is likely limited to the overall range that is enabled.
   If the remote peer writes into the 'wrong' advertised buffer, neither
   the DDP or RDMAP layer will be aware of this. Nor is there any
   report to the ULP on how the remote peer specifically used tagged
   buffers.

   Unless the ULP peers have an adequate basis for mutual trust, the
   receiving ULP might be well advised to use a distinct STag for each
   interaction, and to invalidate it after each use or to require its
   peer to use the RDMAP option to invalidate the STag with its
   responding untagged message.

9.3. Impact of Encrypted Transports

   While DDP is cleanly layered over the LLP, its maximum benefit may be
   limited when the LLP Stream is secured with a streaming cypher, such
   as Transport Layer Security (TLS) RFC4346 [9]. If the LLP must
   decrypt in order, it cannot provide out-of-order DDP Segments to the
   DDP layer for placement purposes. IPsec RFC2406 [1]. tunnel mode
   encrypts entire IP Datagrams. IPsec transport mode encrypts TCP
   Segments or SCTP packets, as does use of DTLS RFC4347 [10] over UDP
   beneath TCP or SCTP. Neither IPsec nor this use of DTLS preclude
   providing out-of-order DDP Segments to the DDP layer for placement.

   Note that end-to-end use of cryptographic integrity protection may

allow suppression of MPA CRC generation and checking under certain
circumstances. This is one example where the LLP may be judged to
have "or equivalent" protection to an end-to-end CRC32c.

10. References

10.1. Normative references

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

[2]   Recio, R., "An RDMA Protocol Specification",
      draft-ietf-rddp-rdmap-05 (work in progress), July 2005.

[3]   Shah, H., "Direct Data Placement over Reliable Transports",
      draft-ietf-rddp-ddp-05 (work in progress), July 2005.

[4]   Stewart, R., "Stream Control Transmission Protocol (SCTP) Remote
      Direct Memory Access (RDMA) Direct Data Placement (DDP)
      Adaptation", draft-ietf-rddp-sctp-03 (work in progress),
      June 2006.

[5]   Culley, P., "Marker PDU Aligned Framing for TCP Specification",
      draft-ietf-rddp-mpa-04 (work in progress), May 2006.

[6]   Pinkerton, J., "DDP/RDMAP Security", draft-ietf-rddp-security-09
      (work in progress), May 2006.

10.2. Informative References

[7]   Ko, M., "iSCSI Extensions for RDMA Specification",
       draft-ietf-ips-iser-05 (work in progress), October 2005.

[8]   Callaghan, B. and T. Talpey, "NFS Direct Data Placement",
       draft-ietf-nfsv4-nfsdirect-02 (work in progress), October 2005.

[9]   Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS)
       Protocol Version 1.1", RFC 4346, April 2006.

[10]   Rescorla, E. and N. Modadugu, "Datagram Transport Layer
       Security", RFC 4347, April 2006.

Authors' Addresses

    Caitlin Bestler
    Broadcom Corporation
    16215 Alton Parkway
    P.O. Box 57013
    Irvine, CA 92619-7013
    USA

    Phone: 949-926-6383
    Email: caitlinb@broadcom.com

    Lode Coene
    Siemens
    Atealaan 26
    Herentals,   2200
    Belgium

    Phone: +32-14-252081
    Email: lode.coene@siemens.com

Intellectual Property Statement

   The IETF takes no position regarding the validity or scope of any
   Intellectual Property Rights or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; nor does it represent that it has
   made any independent effort to identify any such rights. Information
   on the procedures with respect to rights in RFC documents can be
   found in BCP 78 and BCP 79.

   Copies of IPR disclosures made to the IETF Secretariat and any
   assurances of licenses to be made available, or the result of an
   attempt made to obtain a general license or permission for the use of
   such proprietary rights by implementers or users of this
   specification can be obtained from the IETF on-line IPR repository at
   http://www.ietf.org/ipr.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights that may cover technology that may be required to implement
   this standard. Please address the information to the IETF at
   ietf-ipr@ietf.org.

Disclaimer of Validity

   This document and the information contained herein are provided on an
   "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS
   OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET
   ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED,
   INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
   INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
   WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

   Copyright (C) The Internet Society (2006). This document is subject
   to the rights, licenses and restrictions contained in BCP 78, and
   except as set forth therein, the authors retain all their rights.

Acknowledgment

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