

Internet Draft

David L. Black

Document: [draft-ietf-rddp-rdma-concerns-01.txt](#)

EMC

Expires: December 2004

Michael F. Speer

Sun

John Wroclawski

MIT

June 2004

## DDP and RDMA Concerns

### Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

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/lid-abstracts.txt>

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

### Abstract

This draft describes technical concerns that should be considered in the design of standardized RDMA and DDP protocols/mechanisms for use with Internet transport protocols. This draft was written to provide input to the proposed new Remote Direct Data Placement (rddp) WG, and is not intended for publication as an RFC.

This is an updated and resubmitted version of [draft-ietf-rddp-rdma-concerns-00.txt](#) to make it available for current discussions of mandatory-to-implement security in the RDDP WG. Sections [4.1](#), [4.2](#), and 5 are of particular relevance to that discussion.

Black

Expires - December 2004

[Page 1]

2004

DDP and RDMA Concerns

June

## Table of Contents

<a href="#">1. Overview.....</a>	<a href="#">2</a>
<a href="#">2. Conventions used in this document.....</a>	<a href="#">3</a>
<a href="#">3. Architectural Concerns.....</a>	<a href="#">3</a>
<a href="#">3.1 Buffer Management.....</a>	<a href="#">3</a>
<a href="#">3.2 Reliability.....</a>	<a href="#">4</a>
<a href="#">4. Memory is more general than Transport Buffers.....</a>	<a href="#">4</a>
<a href="#">4.1 Overwrites.....</a>	<a href="#">4</a>
<a href="#">4.2 Concurrent Operations to the Same Memory.....</a>	<a href="#">5</a>
<a href="#">4.3 Completions and Ordering.....</a>	<a href="#">5</a>
<a href="#">4.4 Transfer Granularity.....</a>	<a href="#">5</a>
<a href="#">5. Security Considerations.....</a>	<a href="#">6</a>
References.....	<a href="#">6</a>
Acknowledgements.....	<a href="#">7</a>
Author's Addresses.....	<a href="#">7</a>

## 1. Overview

A new effort to standardize RDMA (Remote Direct Memory Access) and DDP (Direct Data Placement) protocols/mechanisms for Internet transport protocols is going to take place in the proposed IETF Remote Direct Data Placement (rddp) WG. This draft describes technical concerns that should be addressed in the design and standardization of these protocols. A basic understanding of RDMA and DDP is assumed; while a basic introduction is included in this section; readers unfamiliar with these concepts may wish to refer to [[RDDP-arch](#), RDDP-ps] for more background.

Both Direct Data Placement (DDP) and Remote Direct Memory Access

(RDMA) have the goal of eliminating copies between the protocol stack and application buffers at the receiver. For example, when a

4-kilobyte file or disk block is retrieved, most operating systems expect the resulting block to be in 4kB of contiguous memory aligned to a 4kB boundary, but most networking interfaces do not behave in this fashion. The result is that a copy is required to produce an aligned 4kB block of data from the data delivered by the

network interface. This copy has undesirable performance impacts; the goal of DDP and RDMA is to enable elimination of this copy in an application- and protocol-independent fashion. The basic concept is that the sender identifies data to be placed directly into application buffers, and transmits that identification with the data so that the receiver can place the data directly into application buffers when it is received.

DDP is envisioned to share network transport buffers with applications, but to use application-specified tags and offsets to select buffers for use on receive. The primary purposes of this

Black

Expires - December 2004

[Page 2]

---

DDP and RDMA Concerns

June

2004

information are to separate application data from headers and deal with applications that return data in unpredictable orders (e.g., the results of concurrent file and disk operations may be returned to the invoker in arbitrary order). One way to view DDP on the wire is that it annotates (or "decorates") data that would have been sent anyway.

RDMA uses DDP or a DDP-like mechanism to implement remote read and write operations on memory regions explicitly exported by end systems. A tag is used to designate a memory region, and an offset

is used to indicate the address within that region. RDMA differs from DDP in that it provides a memory abstraction rather than a transport buffer abstraction. This raises concerns based on the ways in which transport buffers differ from memory in general. In addition, the system coupling over a potentially unreliable network

implied by DDP and RDMA raises several architectural concerns.

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#), although they are used here to describe requirements on protocol development and standardization rather than on protocol implementations.

### 3. Architectural Concerns

Both DDP and RDMA expose memory resources on the receiver to one or more potentially untrustworthy sender(s) over a potentially unreliable network. This has a number of architectural implications, particularly for resource management.

#### 3.1 Buffer Management

Traditional network stacks utilize a pool of interchangeable (aka anonymous) buffers to hold data received from the network. By using specific identifiable application buffers, DDP and RDMA make the memory used for specific receive operations identifiable and may cause protocols to devote more resources to the receive function than might otherwise be the case. In situations where effective use is being made of DDP and/or RDMA, the actual resource demand on the system may be lessened (e.g., because applications only expose memory that is in their working set), but it is necessary to anticipate applications that use DDP and RDMA in a way that increases resource demands and take appropriate precautions to limit system degradation.

#### 3.2 Reliability

RDMA is motivated by experiences with both local DMA and transfers over reliable channels; these experiences will not be completely applicable to RDMA over IP networks. Local DMA provides an extreme

example, in that a local DMA failure is usually caused by hardware problems that often result in the hardware being considered to have failed. In contrast, RDMA over IP must deal with a variety of "stupid IP network tricks" as part of its normal operation. Channel behavior is a less extreme example as channel controllers must expect occasional channel failures and be prepared to deal with the result; one example can be found in multipathing software for disk storage access.

This set of concerns is roughly analogous to the reliability difference between local and remote procedure calls and its impact on distributed system design [need to add a reference here]. The impact of the difference in reliability between local DMA and/or channels vs. RDMA needs to be considered as part of any specification effort, but may be best dealt with in applicability statements as opposed to making these considerations part of the core protocol specifications.

#### 4. Memory is more general than Transport Buffers

The following subsections describe concerns arising from the fact that memory that can be read and/or written is a more general and capable abstraction than a transport buffer.

##### 4.1 Overwrites

A transport buffer can be written exactly once when the data is received; in contrast memory can be written multiple times. This creates the opportunity for received DDP and RDMA data to overwrite other data, including previously received data (that may or may not have been transferred to the application(s)). DDP and RDMA specifications MUST contain mechanisms to prevent overwrites from impairing system integrity and to isolate the effect of overwrites so that interference among otherwise unrelated applications is prevented. In addition the specifications MUST contain mechanisms that allow applications to control the exposure of memory used for DDP and RDMA receives to subsequent overwrites; this is to enable an application to know that a check on received data (e.g., for integrity) is performed after changes to it can no longer be made by remote nodes via DDP or RDMA.

2004

## 4.2 Concurrent Operations to the Same Memory

If a remote (or local) write takes place concurrently with a read to the same memory, the read may return an arbitrary mix of the old and new contents of the memory. If a remote (or local) write takes place concurrently with another write, the resulting memory contents may be an arbitrary mix of the data from the two writes. These results are generally considered undesirable, and should be avoided. DDP and RDMA specifications must consider how these situations are to be avoided (e.g., application-level synchronization may be required), so that at worst they will occur only as the result of application errors in using DDP and RDMA.

## 4.3 Completions and Ordering

RDMA Read and Write operations are asynchronous with respect to the protocol layers above RDMA, hence completion mechanisms are necessary to enable applications to determine when RDMA operations have completed, although these mechanisms need not be invoked for every RDMA operation. In addition, an RDMA specification **MUST** include the assumptions that an application may and may not make about the state of "prior" RDMA operations based on observing the completion of a specific RDMA operation. The word "prior" is in quotes because an RDMA specification will need to define it as part of specifying permissible inference of completion of "prior" operations; the definition is likely to involve a partial order.

Fence and stream abstractions to enforce and prevent ordering (respectively) **MAY** be included in RDMA and DDP specifications, but are **NOT REQUIRED**.

## 4.4 Transfer Granularity

IP transports include the functionality to bundle data so that a set of small user transfers is accomplished via a single larger transfer across the network and through the relevant portions of the protocol stacks. By defining specific remote operations that an application may reasonably expect to complete in a timely fashion, RDMA may disrupt this behavior by requiring smaller transfers to be done promptly. The potential inefficiencies of the resulting behavior for protocol stacks and networks have been known

for a long time; see the discussion of the small-packet problem in [[RFC 896](#)]. Any RDMA specification MUST consider the ability to bundle operations and the potential performance impact of performing multiple smaller transfers in place of a single larger one. This may also apply to DDP, but the first priority is that DDP SHOULD NOT cause major changes to the transmission behavior of any transport protocol to which it is applied by comparison to the same stream without the DDP annotations (some degree of minor

change is unavoidable due to the space consumed by the DDP annotations).

## 5. Security Considerations

With the possible exception of the Completion and Ordering concerns described in [Section 4.3](#), all of these concerns have security implications in that failing to deal with them adequately may expose attacks on system resources, correct operation and/or integrity.

When memory is accessible via the network, such access must be controlled, as allowing arbitrary access by untrusted entities discloses the contents of the memory (read access) and/or allows it to be corrupted (write access). Specifically, it is necessary to provide mechanisms that enable applications to control RDMA and DDP access to their exported memory by both identity (RDMA and DDP) and type of access (read vs. write - RDMA only); this inherently involves authentication of the principals granted access in order to distinguish authorized from unauthorized access. Such authentication MAY be implemented outside the DDP and/or RDMA protocols (e.g., in the application or a separate security protocol such as TLS [[RFC 2246](#)] or IPsec [[RFC 2401](#)]) provided that means are specified to securely couple the authorization of DDP and RDMA operations to the corresponding authentications.

## References

- [RDDP-arch] Bailey, S. and T. Talpey, "The Architecture of Direct Data Placement (DDP) And Remote Direct Memory Access (RDMA) On Internet Protocols", Internet-Draft [draft-ietf-rddp-arch-04.txt](#), Work in Progress, January 2004.
- [RDDP-ps] Romanow, A., J. Mogul, T. Talpey, and S. Bailey, "RDMA over IP Problem Statement", Internet-Draft [draft-ietf-rddp-problem-statement-03.txt](#), Work in Progress, January 2004.
- [RFC 896] Nagle, J., "Congestion Control in IP/TCP Internetworks", [RFC 896](#), January 1984.
- [RFC 2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), [BCP 14](#), March 1997.
- [RFC 2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", [RFC 2246](#), January 1999.
- [RFC 2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.

Black

Expires - December 2004

[Page 6]

---

	DDP and RDMA Concerns	June
2004		

## Acknowledgements

This draft is based in part on a presentation and discussion at an end2end research group meeting at MIT in May 2002 - the authors thank the end2end RG for providing the opportunity and gratefully acknowledge the comments and suggestions of participants.

## Author's Addresses

David L. Black	
EMC Corporation	
176 South Street	Phone: +1 (508) 293-7953
Hopkinton, MA, 01748, USA	Email: <a href="mailto:black_david@emc.com">black_david@emc.com</a>

Michael F. Speer  
Sun Microsystems, Inc.



4150 Network Circle UMPK17-103 Phone: +1 (650) 786-6445  
Santa Clara, CA 95054 Email: michael.speer@sun.com

John Wroclawski  
MIT Lab for Computer Science  
200 Technology Square Phone: +1 (617) 253-7885  
Cambridge, MA 02139 Email: jtw@lcs.mit.edu