Network File System Version 4 C. Lever
Internet-Draft Oracle
Intended status: Informational November 6, 2018
Expires: May 10, 2019

RDMA Connection Manager Private Data For RPC-Over-RDMA Version 1
draft-ietf-nfsv4-rpcrdma-cm-pvt-data-01

Abstract

This document specifies the format of RDMA-CM Private Data exchanged between RPC-over-RDMA version 1 peers as a transport connection is established. Such private data is used to indicate peer support for remote invalidation and larger-than-default inline thresholds.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on May 10, 2019.

Table of Contents

1. Introduction

The RPC-over-RDMA version 1 transport protocol enables the use of RDMA data transfer for upper layer protocols based on RPC [RFC8166]. The terms "Remote Direct Memory Access" (RDMA) and "Direct Data Placement" (DDP) are introduced in [RFC5040].

The two most immediate shortcomings of RPC-over-RDMA version 1 are:

- Setting up an RDMA data transfer (via RDMA Read or Write) can be costly. The small default size of messages transmitted using RDMA Send forces the use of RDMA Read or Write operations even for relatively small messages and data payloads.

  The original specification of RPC-over-RDMA version 1 provided an out-of-band protocol for passing inline threshold values between connected peers [RFC5666]. However, [RFC8166] eliminated support for this protocol making it unavailable for this purpose.

- Unlike most other contemporary RDMA-enabled storage protocols, there is no facility in RPC-over-RDMA version 1 that enables the use of remote invalidation [RFC5042].

RPC-over-RDMA version 1 has no means of extending its XDR definition in such a way that interoperability with existing implementations is preserved. As a result, an out-of-band mechanism is needed to help relieve these constraints for existing RPC-over-RDMA version 1 implementations.

This document specifies a simple, non-XDR-based message format designed to be passed between RPC-over-RDMA version 1 peers at the time each RDMA transport connection is first established. The purpose of this message format is two-fold:

- To provide immediate relief from certain performance constraints inherent in RPC-over-RDMA version 1
- To enable experimentation with parameters of the base RDMA transport over which RPC-over-RDMA runs

The message format may be extended as needed. In addition, interoperation between implementations of RPC-over-RDMA version 1 that present this message format to peers and those that do not recognize this message format is guaranteed.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Advertised Transport Properties

3.1. Inline Threshold Size

Section 3.3.2 of [RFC8166] defines the term "inline threshold." An inline threshold is the maximum number of bytes that can be transmitted using one RDMA Send and one RDMA Receive. There are a pair of inline thresholds for a connection: a client-to-server threshold and a server-to-client threshold.

If an incoming message exceeds the size of a receiver's inline threshold, the receive operation fails and the connection is typically terminated. To convey a message larger than a receiver's inline threshold, an NFS client uses explicit RDMA data transfer operations, which are more expensive to use than RDMA Send.

The default value of inline thresholds for RPC-over-RDMA version 1 connections is 1024 bytes (see Section 3.3.3 of [RFC8166]). This value is adequate for nearly all NFS version 3 procedures.

NFS version 4 COMPOUND operations [RFC7530] are larger on average than NFS version 3 procedures [RFC1813], forcing clients to use explicit RDMA operations for frequently-issued requests such as LOOKUP and GETATTR. The use of RPCSEC_GSS security also increases

the average size of RPC messages, due to the larger size of RPCSEC_GSS credential material included in RPC headers [RFC7861].

If a sender and receiver could somehow agree on larger inline thresholds, frequently-used RPC transactions avoid the cost of explicit RDMA operations.

3.2. Remote Invalidation

After an RDMA data transfer operation completes, an RDMA peer can use remote invalidation to request that the remote peer RNIC invalidate an STag associated with the data transfer [RFC5042].

An RDMA consumer requests remote invalidation by posting an RDMA Send With Invalidate Work Request in place of an RDMA Send Work Request. Each RDMA Send With Invalidate carries one STag to invalidate. The receiver of an RDMA Send With Invalidate performs the requested invalidation and then reports that invalidation as part of the completion of a waiting Receive Work Request.

An RPC-over-RDMA responder can use remote invalidation when replying to an RPC request that provided Read or Write chunks. The requester thus avoids dispatching an extra Work Request, the resulting context switch, and the invalidation completion interrupt as part of completing an RPC transaction that uses chunks. The upshot is faster completion of RPC transactions that involve RDMA data transfer.

There are some important caveats which contraindicate the blanket use of remote invalidation:

- Remote invalidation is not supported by all RNICs.
- Not all RPC-over-RDMA responder implementations can generate RDMA Send With Invalidate Work Requests.
- Not all RPC-over-RDMA requester implementations can recognize when remote invalidation has occurred.
- On one connection in different RPC-over-RDMA transactions, or in a single RPC-over-RDMA transaction, an RPC-over-RDMA requester can expose a mixture of STags that may be invalidated remotely and some that must not be. No indication is provided at the RDMA layer as to which is which.

A responder therefore must not employ remote invalidation unless it is aware of support for it in its own RDMA stack, and on the requester. And, without altering the XDR structure of RPC-over-RDMA version 1 messages, it is not possible to support remote invalidation

with requesters that mix STags that may and must not be invalidated remotely in a single RPC or on the same connection.

However, it is possible to provide a simple signaling mechanism for a requester to indicate it can deal with remote invalidation of any STag it has presented to a responder. There are some NFS/RDMA client implementations that can successfully make use of such a signaling mechanism.

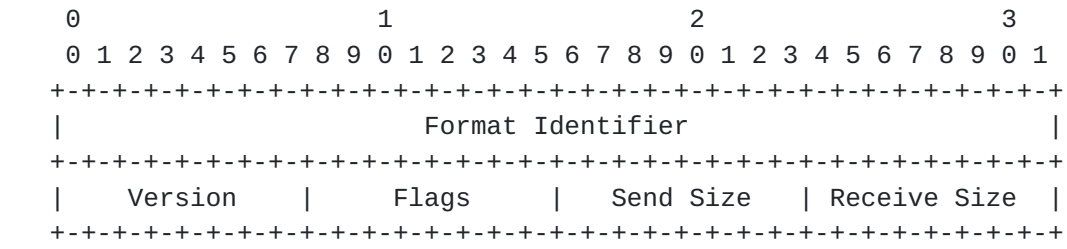
4. Private Data Message Format

With an InfiniBand lower layer, for example, RDMA connection setup uses a Connection Manager when establishing a Reliable Connection [IBARCH]. When an RPC-over-RDMA version 1 transport connection is established, the client (which actively establishes connections) and the server (which passively accepts connections) populate the CM Private Data field exchanged as part of CM connection establishment.

The transport properties exchanged via this mechanism are fixed for the life of the connection. Each new connection presents an opportunity for a fresh exchange.

For RPC-over-RDMA version 1, the CM Private Data field is formatted as described in the following subsection. RPC clients and servers use the same format. If the capacity of the Private Data field is too small to contain this message format, the underlying RDMA transport is not managed by a Connection Manager, or the underlying RDMA transport uses Private Data for its own purposes, the CM Private Data field cannot be used on behalf of RPC-over-RDMA version 1.

The first 8 octets of the CM Private Data field is to be formatted as follows:



Format Identifier: This field contains a fixed 32-bit value that identifies the content of the Private Data field as an RPC-over-RDMA version 1 CM Private Data message. The value of this field is always 0xf6ab0e18, in network byte order. The use of this field is further expanded upon in Section 4.1.

Version: This 8-bit field contains a message format version number. The value "1" in this field indicates that exactly eight octets are present, that they appear in the order described in this section, and that each has the meaning defined in this section. Further considerations about the use of this field are discussed in Section 5.

Flags: This 8-bit field contains bit flags that indicate the support status of optional features, such as remote invalidation. The meaning of these flags is defined in Section 5.1.

Send Size: This 8-bit field contains an encoded value corresponding to the maximum number of bytes this peer is prepared to transmit in a single RDMA Send on this connection. The value is encoded as described in Section 5.2.

Receive Size: This 8-bit field contains an encoded value corresponding to the maximum number of bytes this peer is prepared to receive with a single RDMA Receive on this connection. The value is encoded as described in Section 5.2.

4.1. Interoperability Considerations

The extension described in this document is designed to allow RPC-over-RDMA version implementations that use this extension to interoperate fully with RPC-over-RDMA version 1 implementations that do not exchange this information. Realizing this goal requires that implementations of this extension follow the practices described in the rest of this section.

RPC-over-RDMA version 1 implementations that support the extension described in this document are intended to interoperate fully with RPC-over-RDMA version 1 implementations that do not recognize the exchange of CM Private Data. When a peer does not receive a CM Private Data message which conforms to Section 4, it needs to act as if the remote peer supports only the default RPC-over-RDMA version 1 settings, as defined in [RFC8166]. In other words, the peer is to behave as if a Private Data message was received in which bit 8 of the Flags field is zero, and both Size fields contain the value zero.

The Format Identifier field is provided in order to distinguish RPC-over-RDMA version 1 Private Data from private data inserted by layers below or above RPC-over RDMA version 1. During connection establishment, RPC-over-RDMA version 1 implementations check for this protocol number before decoding subsequent fields. If this protocol number is not present as the first 4 octets, an RPC-over-RDMA receiver needs to ignore the CM-Private Data (ie., behave as if no RPC-over-RDMA version 1 Private Data has been provided).

5. Updating the Message Format

Although the message format described in this document provides the ability for the client and server to exchange particular information about the local RPC-over-RDMA implementation, it is possible that there will be a future need to exchange additional properties. This would make it necessary to extend or otherwise modify the format described in this document.

Any modification faces the problem of interoperating properly with implementations of RPC-over-RDMA version 1 that are unaware of this existence of the new format. These include implementations that that do not recognize the exchange of CM Private Data as well as those that recognize only the format described in this document.

Given the message format described in this document, these interoperability constraints could be met by the following sorts of new message formats:

- A format which uses a different value for the first four bytes of the format, as provided for in the registry described in Section 6.

- A format which uses the same value for the Format Identifier field and a value other than one (1) in the Version field.

Although it is possible to reorganize the last three of the eight bytes in the existing format, extended formats are unlikely to do so. New formats would take the form of extensions of the format described in this document with added fields starting at byte eight of the format and changes to the definition of previously reserved flags.

5.1. Feature Support Flags

The bits in the Flags field are labeled from bit 8 to bit 15, as shown in the diagram above. When the Version field contains the value "1", the bits in the Flags field are to be set as follows:

Bit 15: When both connection peers have set this flag in their CM Private Data, the responder MAY use RDMA Send With Invalidate when transmitting RPC Replies. Each RDMA Send With Invalidate MUST invalidate an STag associated only with the XID in the rdma_xid field of the RPC-over-RDMA Transport Header it carries.

When either peer on a connection clears this flag, the responder MUST use only RDMA Send when transmitting RPC Replies.

Bits 14 - 8: These bits are reserved and are always zero.

5.2. Inline Threshold Values

Inline threshold sizes from 1KB to 256KB can be represented in the Send Size and Receive Size fields. A sender computes the encoded value by dividing the actual value by 1024 and subtracting one from the result. A receiver decodes this value by performing a complementary set of operations.

The client uses the smaller of its own send size and the server's reported receive size as the client-to-server inline threshold. The server uses the smaller of its own send size and the clients's reported receive size as the server-to-client inline threshold.

6. IANA Considerations

In accordance with [RFC8126], the author requests that IANA create a new registry in the "Remote Direct Data Placement" Protocol Category Group. The new registry is to be called the "RDMA-CM Private Data Identifier Registry". This is a registry of 32-bit numbers that identify the Upper Layer protocol associated with data that appears in the RDMA-CM Private Data area.

The information that must be provided to add an entry to this registry will be an IESG-approved Standards Track specification defining the semantics and interoperability requirements of the proposed new value and the fields to be recorded in the registry. The fields in this registry include: Field Identifier, Format Description, and Reference.

The initial contents of this registry are a single entry:

Field Identifier	Format Description	Reference
0xf6ab0e18	RPC-over-RDMA version 1 CM Private Data	[RFC-TBD]

Table 1: RDMA-CM Private Data Identifier Registry

The Expert Review policy, as defined in Section 4.5 of [RFC8126] is to be used to handle requests to add new entries to the "File Provenance Information Registry". New protocol numbers can be assigned at random as long as they do not conflict with existing entries in this registry.

7. Security Considerations

RDMA-CM Private Data typically traverses the link layer in the clear. A man-in-the-middle attack could alter the settings exchanged at connect time such that one or both peers might perform operations that result in premature termination of the connection.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5040] Recio, R., Metzler, B., Culley, P., Hilland, J., and D. Garcia, "A Remote Direct Memory Access Protocol Specification", RFC 5040, DOI 10.17487/RFC5040, October 2007, <https://www.rfc-editor.org/info/rfc5040>.

[RFC5042] Pinkerton, J. and E. Deleganes, "Direct Data Placement Protocol (DDP) / Remote Direct Memory Access Protocol (RDMAP) Security", RFC 5042, DOI 10.17487/RFC5042, October 2007, <https://www.rfc-editor.org/info/rfc5042>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8166] Lever, C., Ed., Simpson, W., and T. Talpey, "Remote Direct Memory Access Transport for Remote Procedure Call Version 1", RFC 8166, DOI 10.17487/RFC8166, June 2017, <https://www.rfc-editor.org/info/rfc8166>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[IBARCH] InfiniBand Trade Association, "InfiniBand Architecture Specification Volume 1", Release 1.3, March 2015, <http://www.infinibandta.org/content/pages.php?pg=technology_download>.

[RFC1813] Callaghan, B., Pawlowski, B., and P. Staubach, "NFS Version 3 Protocol Specification", RFC 1813, DOI 10.17487/RFC1813, June 1995, <https://www.rfc-editor.org/info/rfc1813>.

[RFC5666] Talpey, T. and B. Callaghan, "Remote Direct Memory Access Transport for Remote Procedure Call", RFC 5666, DOI 10.17487/RFC5666, January 2010, <https://www.rfc-editor.org/info/rfc5666>.

[RFC7530] Haynes, T., Ed. and D. Noveck, Ed., "Network File System (NFS) Version 4 Protocol", RFC 7530, DOI 10.17487/RFC7530, March 2015, <https://www.rfc-editor.org/info/rfc7530>.

[RFC7861] Adamson, A. and N. Williams, "Remote Procedure Call (RPC) Security Version 3", RFC 7861, DOI 10.17487/RFC7861, November 2016, <https://www.rfc-editor.org/info/rfc7861>.

Acknowledgments

Thanks to Christoph Hellwig and Devesh Sharma for suggesting this approach, and to Tom Talpey and Dave Noveck for their expert comments and review. The author also wishes to thank Bill Baker and Greg Marsden for their support of this work.

Special thanks go to Transport Area Director Spencer Dawkins, NFSV4 Working Group Chairs Spencer Shepler and Brian Pawlowski, and NFSV4 Working Group Secretary Thomas Haynes.

Author's Address

Charles Lever
Oracle Corporation
1015 Granger Avenue
Ann Arbor, MI 48104
United States of America

Email: chuck.lever@oracle.com