Workgroup: Network File System Version 4
Internet-Draft:
draft-cel-nfsv4-rpc-over-quicv1-01
Published: 5 July 2022
Intended Status: Standards Track
Expires: 6 January 2023
Authors: B. Coddington S. Mayhew C. Lever, Ed.
Red Hat Red Hat Oracle

Remote Procedure Call over QUIC Version 1

Abstract

This document specifies a protocol for conveying Remote Procedure (RPC) messages on QUIC version 1 connections. It requires no revision to application RPC protocols or the RPC protocol itself.

Note

This note is to be removed before publishing as an RFC.

Discussion of this draft occurs on the NFSv4 working group mailing list, archived at https://mailarchive.ietf.org/arch/browse/nfsv4/. Working Group information is available at https://datatracker.ietf.org/wg/nfsv4/about/.

Submit suggestions and changes as pull requests at https://github.com/chucklever/i-d-rpc-over-quicv1. Instructions are on that page.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

QUIC is a reliable, connection-oriented network transport protocol that is designed to be general-purpose and secure [RFC9000]. Its features include integrated transport layer security, multiple streams over each connection, fast reconnect, and robust recovery from packet loss and network congestion.

Open Network Computing Remote Procedure Call (often shortened to "RPC") is a Remote Procedure Call protocol that runs over a variety of network transports [RFC5531]. RPC implementations so far use UDP [RFC0768] or TCP [RFC0793]. This document specifies how to transport RPC messages over QUIC version 1.

Explain motivations:

*TLS

*Multiple streams--though applications are speculative at this point. (Maybe they will allow more sophisticated prioritization of traffic without the overhead of multiple TCP connections?)

*Lower-latency connection setup--though NFS connections are typically long-lived.

*Likely SMB adoption of QUIC should make QUIC implementations widely available.

In addition, this section needs to document and demonstrate specific use cases that cannot be addressed using existing RPC transports and security mechanisms such as RPC-over-TCP, RPC-with-TLS, or RPC-over-RDMA.

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. RPC-over-QUIC Framework

RPC is first and foremost a message-passing protocol. This section covers the implementaion details of exchanging RPC messages over QUICv1. Readers should already be familiar with ONC RPC [RFC5531].

3.1. Transport Layer Security

RPC-over-QUIC provides peer authentication and encryption services using a framework based on Transport Layer Security (TLS). Ergo, RPC-over-QUIC inherently fulfills many of the requirements of [I-D.ietf-nfsv4-rpc-tls]. The details of QUIC's use of TLS are specified in [RFC9001]. In particular:

*With QUICv1, security at the transport layer is always enabled. Thus, there is no need or use for the STARTTLS mechanism described in Section 4 of [I-D.ietf-nfsv4-rpc-tls].

*The discussion in [I-D.ietf-nfsv4-rpc-tls] about the opportunistic use of TLS does not apply to RPC-over-QUIC.

*The peer authentication requirements in Section 5.2 of [I-D.ietf-nfsv4-rpc-tls] do apply to RPC-over-QUIC.

*The PKIX Extended Key Usage values defined in [I-D.ietf-nfsv4-rpc-tls] are also valid for use with RPC-over-QUIC.

*The ALPN defined in Section 8.2 of [I-D.ietf-nfsv4-rpc-tls] is also used for RPC-over-QUIC.

3.2. RPC Message Framing

Record marking on QUIC is exactly as in TCP. See Section 11 of [RFC5531].

Discussion: This is the simplest thing to do.

bfields: Open question whether we should do something more complicated that adds RDMA-like features or at least provides some minimal help with data alignment. Possibilities might include a single additional integer giving the offset of a payload, serving only as a hint; or reference additional streams in the same connection for payloads; or even looking into full RDMA--long-term there may be interest in supporting RDMA over QUIC, and we may be able to piggyback on that effort.

cel: Direct data placement over TCP can already be accomplished today using MPA/DDP protocols (formerly known as iWARP). Using a software iWARP implementation means no special hardware is necessary. Likewise, if MPA/DDP can be made to support QUIC, much of the need for a separate RPC-over-QUIC is moot. In addition, it would bring automatically transport layer security to other RDMA-enabled protocols (such as RPC-over-RDMA).

lars: If changes to the RPC-over-QUIC binding might be desired in the future, how would they be negotiated/expressed? Should a versioned ALPN be used instead of the one from [I-D.ietf-nfsv4-rpc-tls]?

3.3. Connections and Streams

QUIC provides a "stream" abstraction, described in Section 2 of [RFC9000]. Each QUIC stream can be unidirectional or bidirectional. QUIC supports a nearly unlimited number of concurrent streams per connection.

Unless explicitly specified, when RPC protocol specifications refer to a "connection", this applies to a QUIC connection, not to a stream. As an example, in the case of NFSv4.1 [RFC8881], a BIND_CONN_TO_SESSION operation binds a QUIC connection and does not need to be repeated for each stream on the connection.

An RPC Reply **MUST** be sent over the same connection and stream as the Call message with a matching XID. Forward-direction RPC messages **MUST** be sent over a client-initiated bidirectional stream (stream type 0x00). Reverse-direction RPC messages **MUST** be sent over a server-initiated bidirectional stream (stream type 0x01). Otherwise,

unless otherwise explicitly specified, RPC callers are free to use streams as they wish, and responders have to accommodate callers that do so.

NFS requirement on resends: QUIC allows reconnecting using the same connection ID, so isn't breaking/reconnection somewhat ambiguous? When can a server drop or a client resend? Any advice needed for server-side DRC implementations?

lars: I'm not sure I understand what is meant by "reconnecting" above. Is this referring to connection migration? Or a 0-RTT repeated connection instance? Something else?

lars: Also, I'm not sure if the use of streams is fully specified by the above. Is the intent here to leave it to callers to decide if they want to use a fresh stream for each RPC, or reuse an existing stream for a series of RPCs?

4. Implementation Status

This section is to be removed before publishing as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs.

Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

There are no known implementations of RPC-over-QUICv1 as described in this document.

5. Security Considerations

Readers should refer to the discussion of QUIC's transport layer security in Section 21 of [RFC9000].

6. IANA Considerations

RFC Editor: In the following subsections, please replace RFC-TBD with the RFC number assigned to this document. Furthermore, please remove this Editor's Note before this document is published.

6.1. Netids for RPC-over-QUIC

Each new RPC transport is assigned one or more RPC "netid" strings. These strings are an rpcbind [RFC1833] string naming the underlying transport protocol, appropriate message framing, and the format of service addresses and ports, among other things.

This document requests that IANA allocate the following "Netid" registry strings in the "ONC RPC Netid" registry, as defined in [RFC5665]:

```
NC_QUIC    "quic"
NC_QUIC6   "quic6"
```

These netids **MUST** be used for any transport satisfying the requirements described in this document. The "quic" netid is to be used when IPv4 addressing is employed by the underlying transport, and "quic6" for IPv6 addressing. IANA should use this document (RFC-TBD) as the reference for the new entries.

lars: Why one per IP address family? This seems common practice with netids, but also seems to be a layering violation?

7. References

7.1. Normative References

[I-D.ietf-nfsv4-rpc-tls] Myklebust, T. and C. Lever, "Towards Remote Procedure Call Encryption By Default", Work in Progress, Internet-Draft, draft-ietf-nfsv4-rpc-tls-11, 23 November 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-nfsv4-rpc-tls-11>.

[RFC1833] Srinivasan, R., "Binding Protocols for ONC RPC Version 2", RFC 1833, DOI 10.17487/RFC1833, August 1995, <https://www.rfc-editor.org/rfc/rfc1833>.

[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/rfc/rfc2119>.

[RFC5531] Thurlow, R., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 5531, DOI 10.17487/RFC5531, May 2009, <https://www.rfc-editor.org/rfc/rfc5531>.

[RFC5665] Eisler, M., "IANA Considerations for Remote Procedure Call (RPC) Network Identifiers and Universal Address Formats", RFC 5665, DOI 10.17487/RFC5665, January 2010, <https://www.rfc-editor.org/rfc/rfc5665>.

[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/rfc/rfc8174>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/rfc/rfc9000>.

[RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/rfc/rfc9001>.

7.2. Informative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/rfc/rfc768>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/rfc/rfc793>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/rfc/rfc7942>.

[RFC8881] Noveck, D., Ed. and C. Lever, "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 8881, DOI 10.17487/RFC8881, August 2020, <https://www.rfc-editor.org/rfc/rfc8881>.

Acknowledgments

The authors express their deepest appreciation for the contributions of J. Bruce Fields who was an original author of this document. In addition, we are indebted to Lars Eggert and the QUIC working group for the creation of the QUIC transport protocol.

The editor is grateful to Bill Baker, Greg Marsden, and Martin Thomson for their input and support.

Special thanks to Transport Area Directors Martin Duke and Zaheduzzaman Sarker, NFSV4 Working Group Chairs David Noveck and Brian Pawlowski, and NFSV4 Working Group Secretary Thomas Haynes for their guidance and oversight.

Authors' Addresses

Benjamin Coddington
Red Hat
United States of America

Email: bcodding@redhat.com

Scott Mayhew
Red Hat
United States of America

Email: smayhew@redhat.com

Charles Lever (editor)
Oracle Corporation
United States of America

Email: chuck.lever@oracle.com