Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 20, 2019

T. Pauly
E. Kinnear
Apple Inc.
D. Schinazi
Google LLC

December 17, 2018

An Unreliable Datagram Extension to QUIC
draft-pauly-quic-datagram-01

Abstract

This document defines an extension to the QUIC transport protocol to add support for sending and receiving unreliable datagrams over a QUIC connection.

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 June 20, 2019.

Table of Contents

1. Introduction 2
1.1. Specification of Requirements 2
2. Motivation 2
3. Transport Parameter 3
4. Datagram Frame Type 3
5. Behavior and Usage 4
5.1. Datagram Identifiers 5
5.2. Flow Control and Acknowledgements 5
6. Security Considerations 5
7. IANA Considerations 5
8. Acknowledgments 6
9. Informative References 6
Authors' Addresses 6

1. Introduction

The QUIC Transport Protocol [I-D.ietf-quic-transport] provides a secure, multiplexed connection for transmitting reliable streams of application data. Reliability within QUIC is performed on a per-stream basis, so some frame types are not eligible for retransmission.

Some applications, particularly those that need to transmit real-time data, prefer to transmit data unreliably. These applications can build directly upon UDP [RFC0768] as a transport, and can add security with DTLS [RFC6347]. Extending QUIC to support transmitting unreliable application data would provide another option for secure datagrams, with the added benefit of sharing a cryptographic and authentication context used for reliable streams.

This document defines four new DATAGRAM QUIC frame types, which carry application data without requiring retransmissions.

1.1. Specification of Requirements

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.

2. Motivation

Transmitting unreliable data over QUIC provides benefits over existing solutions:

- Applications that open both a reliable TLS stream and an unreliable DTLS flow to the same peer can benefit by sharing a single handshake and authentication context between a reliable QUIC stream and flow of unreliable QUIC datagrams. This can reduce the latency required for handshakes.

- QUIC uses a more nuanced loss recovery mechanism than the DTLS handshake, which has a basic packet loss retransmission timer. This may allow loss recovery to occur more quickly for QUIC data.

- QUIC datagrams, while unreliable, can support acknowledgements, allowing applications to be aware of whether a datagram was successfully received.

These reductions in connection latency, and application insight into the delivery of datagrams, can be useful for optimizing audio/video streaming applications, gaming applications, and other real-time network applications.

Unreliable QUIC datagrams can also be used to implement an IP packet tunnel over QUIC, such as for a Virtual Private Network (VPN). Internet-layer tunneling protocols generally require a reliable and authenticated handshake, followed by unreliable secure transmission of IP packets. This can, for example, require a TLS connection for the control data, and DTLS for tunneling IP packets. A single QUIC connection could support both parts with the use of unreliable datagrams.

3. Transport Parameter

Support for receiving the DATAGRAM frame types is advertised by means of a QUIC Transport Parameter (name=accepts_datagrams, value=12). An endpoint that includes this parameter supports the DATAGRAM frame types and is willing to receive such frames on this connection. Endpoints MUST NOT send DATAGRAM frames until they have sent and received the accepts_datagrams transport parameter. An endpoint that receives a DATAGRAM frame when it has not sent the accepts_datagrams transport parameter MUST terminate the connection with error PROTOCOL_VIOLATION.

4. Datagram Frame Type

DATAGRAM frames are used to transmit application data in an unreliable manner. The DATAGRAM frame type takes the form 0b001000XX (or the set of values from 0x20 to 0x23). The least significant bit of the DATAGRAM frame type is the LEN bit (0x01). It indicates that there is a Length field present. If this bit is set to 0, the Length field is absent and the Datagram Data field extends to the end of the

packet. If this bit is set to 1, the Length field is present. The second least significant bit of the DATAGRAM frame type is the DATAGRAM_ID bit (0x02). It indicates that there is a Datagram ID field present. If this bit is set to 0, the Datagram ID field is absent and the Datagram ID is assumed to be zero. If this bit is set to 1, the Datagram ID field is present.

A DATAGRAM frame is shown below.

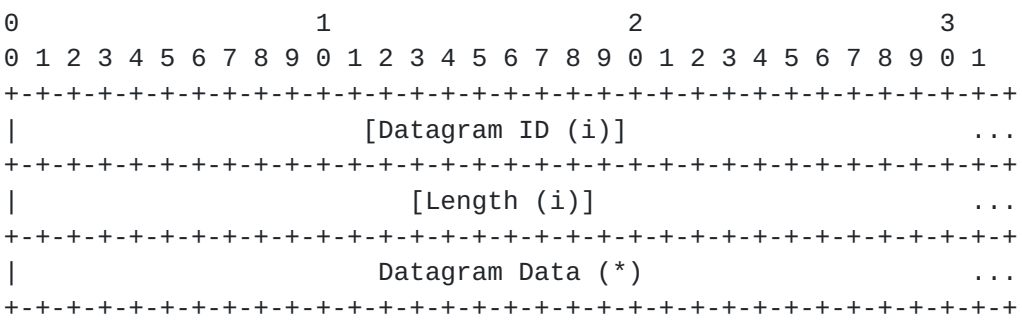

Figure 1: DATAGRAM Frame Format

The fields of a DATAGRAM frame are as follows:

Datagram ID: A variable-length integer indicating the datagram ID of the datagram (see Section 5.1).

Length: A variable-length integer specifying the length of the datagram in bytes. If the length is zero, the data extends to the end of the QUIC packet.

Datagram Data: The bytes of the datagram to be delivered.

5. Behavior and Usage

When an application sends an unreliable datagram over a QUIC connection, QUIC will generate a new DATAGRAM frame and send it in the first available packet. This frame SHOULD NOT be delayed, but MAY be coalesced with other STREAM or DATAGRAM frames.

When a QUIC endpoint receives a valid DATAGRAM frame, it SHOULD deliver the data to the application immediately.

DATAGRAM frames MUST be protected with either 0-RTT or 1-RTT keys.

5.1. Datagram Identifiers

Since several applications relying on datagrams have the need to identify which application-level flow a given datagram is a part of, DATAGRAM frames carry a datagram identifier. Applications that do not have a need for the identifier can use the value zero on their DATAGRAM frames and use the DATAGRAM_ID bit to omit sending the identifier over the wire. If an application uses a mixture of DATAGRAM frames with and without the DATAGRAM_ID bit set, the frames without it are assumed to be part of the application-level flow with Datagram ID zero.

5.2. Flow Control and Acknowledgements

Although the DATAGRAM frame is not retransmitted upon loss detection, it does contribute to the maximum data for the overall connection. Packets that contain only DATAGRAM frames do need to be acknowledged, but implementations SHOULD defer and batch acknowledgements since the timing of these acknowledgements is not used for loss recovery.

The DATAGRAM frame does not provide any explicit flow control signaling apart from the connection-level flow control. DATAGRAM frames are flow controlled only when the maximum data for the connection is hit, at which point the BLOCKED frame is sent.

In cases in which a DATAGRAM frame is blocked due to connection-level flow control or congestion control, an implementation MAY drop the frame without sending it.

6. Security Considerations

The DATAGRAM frame shares the same security properties as the rest of the data transmitted within a QUIC connection. All application data transmitted with the DATAGRAM frame, like the STREAM frame, MUST be protected either by 0-RTT or 1-RTT keys.

7. IANA Considerations

This document registers a new value in the QUIC Transport Parameters:

Value: 12 (if this document is approved)

Parameter Name: accepts_datagrams

Specification: Indicates that the connection should enable support for unreliable DATAGRAM frames. An endpoint that advertises this transport parameter can receive datagrams frames from the other endpoint.

This document also registers a new value in the QUIC Frame Type registry:

Value: 0x1c - 0x1d (if this document is approved)

Frame Name: DATAGRAM

Specification: Unreliable application data

8. Acknowledgments

Thanks to Ian Swett, who inspired this proposal.

9. Informative References

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-16 (work in progress), October 2018.

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

[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>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[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>.

Authors' Addresses

Tommy Pauly
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America

Email: tpauly@apple.com

Eric Kinnear
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America

Email: ekinnear@apple.com

David Schinazi
Google LLC
1600 Amphitheatre Parkway
Mountain View, California 94043
United States of America

Email: dschinazi.ietf@gmail.com