Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 13, 2020

D. Schinazi
Google LLC

March 12, 2020

Using QUIC Datagrams with HTTP/3
draft-schinazi-quic-h3-datagram-03

Abstract

The QUIC DATAGRAM extension provides application protocols running over QUIC with a mechanism to send unreliable data while leveraging the security and congestion-control properties of QUIC. However, QUIC DATAGRAM frames do not provide a means to demultiplex application contexts. This document defines how to use QUIC DATAGRAM frames when the application protocol running over QUIC is HTTP/3 by adding an identifier at the start of the frame payload.

Discussion of this work is encouraged to happen on the QUIC IETF mailing list quic@ietf.org [1] or on the GitHub repository which contains the draft: https://github.com/DavidSchinazi/draft-h3-datagram [2].

Table of Contents

1. Introduction

The QUIC DATAGRAM extension [DGRAM] provides application protocols running over QUIC [QUIC] with a mechanism to send unreliable data while leveraging the security and congestion-control properties of QUIC. However, QUIC DATAGRAM frames do not provide a means to demultiplex application contexts. This document defines how to use QUIC DATAGRAM frames when the application protocol running over QUIC is HTTP/3 [H3] by adding an identifier at the start of the frame payload.

This design mimics the use of Stream Types in HTTP/3, which provide a demultiplexing identifier at the start of each unidirectional stream.

Discussion of this work is encouraged to happen on the QUIC IETF mailing list quic@ietf.org [3] or on the GitHub repository which contains the draft: https://github.com/DavidSchinazi/draft-h3-datagram [4].

1.1. Conventions and Definitions

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. HTTP/3 DATAGRAM Frame Format

When used with HTTP/3, the Datagram Data field of QUIC DATAGRAM frames uses the following format:

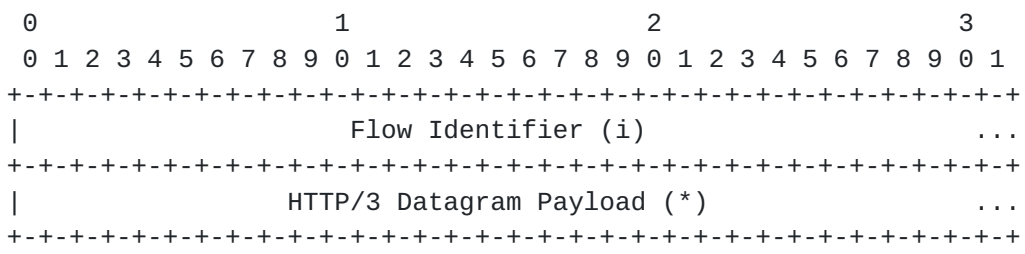

Figure 1: HTTP/3 DATAGRAM Frame Format

Flow Identifier: A variable-length integer indicating the Flow Identifier of the datagram (see Section 2.1).

HTTP/3 Datagram Payload: The payload of the datagram, whose semantics are defined by individual applications.

2.1. Flow Identifiers

Flow identifiers represent bidirectional flows of datagrams within a single QUIC connection. These are conceptually similar to UDP ports and allow basic demultiplexing of application data. The primary role of flow identifiers is to provide a standard mechanism for demultiplexing application data flows, which may be destined for different processing threads in the application, akin to UDP sockets.

Beyond this, a sender SHOULD ensure that DATAGRAM frames within a single flow are transmitted in order relative to one another. If multiple DATAGRAM frames can be packed into a single QUIC packet, the sender SHOULD group them by flow identifier to promote fate-sharing within a specific flow and improve the ability to process batches of datagram messages efficiently on the receiver.

3. Flow Identifier Allocation

Implementations of HTTP/3 that support the DATAGRAM extension MUST provide a flow identifier allocation service. That service will allow applications co-located with HTTP/3 to request a unique flow identifier that they can subsequently use for their own purposes. The HTTP/3 implementation will then parse the flow identifier of

incoming DATAGRAM frames and use it to deliver the frame to the appropriate application.

Even flow identifiers are client-initiated, while odd flow identifiers are server-initiated. This means that an HTTP/3 client implementation of the flow identifier allocation service MUST only provide even identifiers, while a server implementation MUST only provide odd identifiers. Note that, once allocated, any flow identifier can be used by both client and server - only allocation carries separate namespaces to avoid requiring synchronization.

4. Security Considerations

This document currently does not have additional security considerations beyond those defined in [QUIC] and [DGRAM].

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. Normative References

[DGRAM] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", draft-ietf-quic-datagram-00 (work in progress), February 2020.

[H3] Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", draft-ietf-quic-http-27 (work in progress), February 2020.

[QUIC] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-27 (work in progress), February 2020.

[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>.

[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>.

6.2. URIs

[1] mailto:quic@ietf.org

[2] https://github.com/DavidSchinazi/draft-h3-datagram

[3] mailto:quic@ietf.org

[4] https://github.com/DavidSchinazi/draft-h3-datagram

Acknowledgments

The DATAGRAM frame identifier was previously part of the DATAGRAM frame definition itself, the author would like to acknowledge the authors of that document and the members of the IETF QUIC working group for their suggestions.

Author's Address

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

Email: dschinazi.ietf@gmail.com