Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 15, 2017

M. Kuehlewind
B. Trammell
ETH Zurich

January 11, 2017

Applicability and Management of the QUIC Transport Protocol
draft-kuehlewind-quic-appman-00

Abstract

This document discusses the applicability and manageability of the QUIC transport protocol, focusing on caveats impacting application protocol development and deployment over QUIC, and network operations involving QUIC traffic.

Table of Contents

1. Introduction

QUIC [I-D.ietf-quic-transport] is a new transport protocol currently under development in the IETF quic working group, focusing on support of semantics as needed for HTTP/2 [I-D.ietf-quic-http] such as stream-multiplexing to avoid head-of-line blocking. Based on current deployment practices, QUIC is encapsulated in UDP and encrypted by default. This means the version of QUIC that is currently under development will integrate TLS 1.3 [I-D.ietf-quic-tls] to encrypt all payload data including all header information needed for for stream-multiplexing and most on the other header information. Given QUIC is an end-to-end transport protocol, all information in the protocol header is not meant to be mutable by the network, and will therefore be integrity- protected to the extent possible.

This document serves two purposes:

1. It provides guidance for application developers that want to use the QUIC protocol without implementing it on their own. This includes general guidance for application use of HTTP/2 over QUIC as well as the use of other application layer protocols over QUIC. For specific guidance on how to integrate HTTP/2 with QUIC, see [I-D.ietf-quic-http].

2. It provides guidance for network operation and management of QUIC traffic. This includes guidance on how to interpret and utilize information that is exposed by QUIC to the network as well as explaining requirement and assumption that the QUIC protocol design takes toward the expected network treatment.

1.1. Notational Conventions

The words "MUST", "MUST NOT", "SHOULD", and "MAY" are used in this document. It's not shouting; when they are capitalized, they have the special meaning defined in [RFC2119].

2. Applicability of QUIC

In the following subsections we discuss specific caveats to QUIC's applicability, and issues that application developers must consider when using QUIC as a transport for their application.

2.1. The Necessity of TCP Fallback

QUIC uses UDP as a substrate for userspace implementation and port numbers for NAT and middlebox traversal. While there is no evidence of widespread, systematic disadvantage of UDP traffic compared to TCP in the Internet [Edeline16], somewhere between three [Trammell16] and five [Swett16] percent of networks simply block UDP traffic. All applications running on top of QUIC must therefore either be prepared to accept connectivity failure on such networks, or be engineered to fall back to TLS, or TLS-equivalent crypto, over TCP. These applications must operate, perhaps with impaired functionality, in the absence of features provided by QUIC not present in TLS over TCP: The most obvious difference is that TCP does not stream multiplexing and there stream multiplex would need to be implemented in the application layer if needed. Further, TCP by default does not support 0-RTT session resumption. TCP Fast Open can be used but might no be supported by the far end or could be blocked on the network path. Note that there is at least evidence of middleboxes blocking SYN data even if TFO was successfully negotiated. Moreover, while encryption (in this case TLS) is inseparable integrated with QUIC, TLS negotiation over TCP can be blocked. In case it is RECOMMENDED to abort the connection.

We hope that the deployment of a proposed standard version of the QUIC protocol will provide an incentive for these networks to permit QUIC traffic. Indeed, the ability to treat QUIC traffic statefully as in Section 3.5 removes one network management incentive to block this traffic.

2.2. Zero RTT: Here There Be Dragons

QUIC provides for 0-RTT connection establishment (see section 3.2 of [I-D.ietf-quic-transport]). However, data in the frames contained in the first packet of a such a connection must be treated specially by the application layer. Since a retransmission of these frames resulting from a lost acknowledgment may cause the data to appear twice, either the application- layer protocol has to be designed such that all such data is treated as idempotent, or there must be some application-layer mechanism for recognizing spuriously retransmitted frame and dropping them.

[EDITOR'S NOTE: discuss defenses against replay attacks using 0-RTT data.]

2.3. Stream versus Flow Multiplexing

QUIC's stream multiplexing feature allows applications to run multiple streams over a single connection, without head-of-line blocking between streams, associated at a point in time with a single five-tuple. Streams are meaningful only to the application; since stream information is carried inside QUIC's encryption boundary, no information about the stream(s) whose frames are carried by a given packet is visible to the network.

Stream multiplexing is not intended to be used for differentiating streams in terms of network treatment. Application traffic requiring different network treatment should therefore be carried over different five-tuples (i.e. multiple QUIC connections). Given QUIC's ability to send application data on the first packet of a connection (if a previous connection to the same host has been successfully established to provide the respective credentials), the cost for establishing another connection are extremely low.

[EDITOR'S NOTE: For discussion: If establishing a new connection does not seem to be sufficient, the protocol's rebinding functionality (see section 3.7 of [I-D.ietf-quic-transport]) could be extended to allow multiple five-tuples to share a connection ID simultaneously, instead of sequentially.]

3. Manageability of QUIC

This section discusses those aspects of the QUIC transport protocol that have an impact on the design and operation of devices that forward QUIC packets. This section is concerned primarily with QUIC's unencrypted wire image, which we define as the information available in the packet header in each QUIC packet, and the dynamics of that information.

QUIC is a versioned protocol. Everything about the header format can change except the mechanism by which a receiver can determine whether and where a version number is present, and the meaning of the version number field itself.

The rest of this document is concerned with the public header structure of the version of the QUIC transport document that is current as of this writing.

3.1. QUIC Public Header Structure

In the current version of the QUIC protocol, the following information are optionally exposed in the QUIC header:

- flags: All QUIC packets have one byte of flags at the beginning of their header. The definition of these flags can change with the version of QUIC, expect for the version flag that indicated that the version number is present in the QUIC header. Other bits of the flag field in the current version of QUIC are the connection ID flag, the packet number size field, the public reset flag, and the key phase flag.

- version number: The version number is present if the version bit in the flags field is set. The version flag is always set in the first packet of a connection but could also be set in other packets.

- connection ID: The connection ID is present if the connection ID bit in the flag field is set. The connection ID flag is always set on the first packet of a connection and can be set on others. Further the connection ID flag is always set when the public reset bit is set as well. QUIC connections are resistant to IP address changes. Therefore if exposed, the same connection ID can occur in QUIC packet with different 5-tuples, indicating that this QUIC packet belongs to the same connection.

- packet number: The packet number is variable length as indicated by packet number size field. If the length is indicated as zero the packet number is not present. If the public reset flag is set, the packet number cannot be present.

- diversification nonce [EDITOR'S NOTE: talk about this once it's clear what it will be...]

3.2. Integrity Protection of the Wire Image

All information in the QUIC header, even if exposed to the network, is integrity protected, therefore a device on the network path MUST not change these information. Altering of header information would fail any integrity check, leading to packet drop at the receiver.

3.3. Connection ID and Rebinding

A flow might change one of its IP addresses but keep the same connection ID, as discussed in Section 3.1. [EDITOR'S NOTE: What does that mean for the network, if anything (given the connection ID is only rarely present)?]

3.4. Packet Numbers

Packet numbers are monotonically increasing. Packets containing retransmissions as well as packets containing only control information, such as acknowledgments, will get a new packet numbers. Therefore pure control and retransmission packets are impossible to distinguish on the wire.

While loss detection in QUIC is still based on packet numbers, congestion control by default provides richer information than vanilla TCP does. Especially, QUIC does not rely on duplicated ACKs, making it more tolerant of packet re-ordering.

3.5. Stateful Treatment of QUIC Traffic

Stateful network devices such as firewalls use exposed header information to support state setup and tear-down. In-line with [I-D.trammell-plus-statefulness] (which provides a general model for in- network state management), the presence of a Connection ID on QUIC traffic can be used as an association/confirmation signal; QUIC's public reset may be used as a partial one-way stop signal.

[EDITOR'S NOTE: note public reset changes for state management may be desirable: two-way stop as in [I-D.trammell-plus-statefulness] has nice properties.]

3.6. Measuring QUIC Traffic

Given packet numbers can be expected to be exposed on most packets (expect public reset but that terminates the connection anyway), packet numbers can be used by the network to measure loss that occurred between the sender and the measurement point in the network. Similarly, out-of-order packets indicate upstream reordering. Unlike

with TCP, there is no way to measure downstream loss and RTT passively.

[EDITOR'S NOTE: the addition of a simple packet number echo would allow passive RTT measurement and partial passive downstream loss/ reordering measurement. Packet number echo can be sampled at the echo-side (i.e. one-in-N packets or 1/p packets can carry an echo) for efficiency tradeoff, if necessary.]

[EDITOR'S NOTE: in-network devices can inspect and correlate connection IDs for partial tracking of mobility events.]

4. IANA Considerations

This document has no actions for IANA.

5. Security Considerations

Especially security- and privacy-relevant applicability and manageability considerations are given in Section 2.2, Section 3.2, and Section 3.3.

6. Acknowledgments

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI), and by the Swiss State Secretariat for Education, Research, and Innovation under contract no. 15.0268. This support does not imply endorsement.

7. References

7.1. Normative References

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-00 (work in progress), November 2016.

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

7.2. Informative References

[Edeline16]
Edeline, K., Kuehlewind, M., Trammell, B., Aben, E., and B. Donnet, "Using UDP for Internet Transport Evolution (arXiv preprint 1612.07816)", December 2016.

[I-D.ietf-quic-http]
Bishop, M., "Hypertext Transfer Protocol (HTTP) over QUIC", draft-ietf-quic-http-00 (work in progress), November 2016.

[I-D.ietf-quic-tls]
Thomson, M. and (. (Unknown), "Using Transport Layer Security (TLS) to Secure QUIC", draft-ietf-quic-tls-00 (work in progress), November 2016.

[I-D.trammell-plus-statefulness]
Kuehlewind, M., Trammell, B., and J. Hildebrand, "Transport-Independent Path Layer State Management", draft-trammell-plus-statefulness-02 (work in progress), December 2016.

[Swett16] Swett, I., "QUIC Deployment Experience at Google (IETF96 QUIC BoF presentation)", July 2016.

[Trammell16]
Trammell, B. and M. Kuehlewind, "Internet Path Transparency Measurements using RIPE Atlas (RIPE72 MAT presentation)", May 2016.

Authors' Addresses

Mirja Kuehlewind
ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Email: mirja.kuehlewind@tik.ee.ethz.ch

Brian Trammell
ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Email: ietf@trammell.ch