Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2017

M. Kuehlewind
B. Trammell
ETH Zurich
D. Druta
AT&T

March 09, 2017

Manageability of the QUIC Transport Protocol
draft-kuehlewind-quic-manageability-00

Abstract

This document discusses manageability of the QUIC transport protocol, focusing on caveats impacting network operations involving QUIC traffic. Its intended audience is network operators, as well as content providers that rely on the use of QUIC-aware middleboxes, e.g. for load balancing.

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 http://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 September 10, 2017.

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]. Based on current deployment practices, QUIC is encapsulated in UDP and encrypted by default. The current version of QUIC integrates TLS [I-D.ietf-quic-tls] to encrypt all payload data and most header information. Given QUIC is an end-to-end transport protocol, all information in the protocol header, even that which can be inspected, is is not meant to be mutable by the network, and will therefore be integrity-protected to the extent possible.

This document provides guidance for network operation on the 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 assumptions that the QUIC protocol design takes toward the expected network treatment. It also discusses how common network management practices will be impacted by QUIC.

Of course, network management is not a one-size-fits-all endeavour: practices considered necessary or even mandatory within enterprise networks with certain compliance requirements, for example, would be impermissible on other networks without those requirements. This document therefore does not make any specific recommendations as to which practices should or should not be applied; for each practice, it describes what is and is not possible with the QUIC transport protocol as defined.

QUIC is at the moment very much a moving target. This document refers the state of the QUIC working group drafs as well as to changes under discussion, via issues and pull requests in GitHub current as of the time of writing.

1.1. Notational Conventions

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

2. Features of the QUIC Wire Image

In this section, we discusses those aspects of the QUIC transport protocol that have an impact on the design and operation of devices that forward QUIC packets. Here, we are 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. Since 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 fields used in the version negotiation process. This document is focused on the protocol as presently defined in [I-D.ietf-quic-transport] and [I-D.ietf-quic-tls], and will change to track those documents.

2.1. QUIC Packet Header Structure

The QUIC packet header is under active development; see section 5 of [I-D.ietf-quic-transport] for the present header structure, and https://github.com/quicwg/base-drafts/pull/361 for one current proposed redesign.

Currently the first bit of the QUIC header indicates the present of a long header that exposed more information than the short. The long header is typically used during connection start or for other control processes while the short header will be used on mostly packets to limited unnecessary header overhead. The following information may be exposed in the packet header:

- version number: The version number is present during version negotiation.

- connection ID: The connection ID identifies the connection associated with a QUIC packet, for load-balancing and NAT rebinding purposes; see Section 2.3.

- packet number: Every packet has an associated packet number; this packet number increases with each packet, and the least-significant bits of the packet number are present on each packet; see Section 2.4.

- public reset indication: Public reset packets expose the fact that a connection is being torn down to devices along the path. The applicability of public reset is currently under discussion; see https://github.com/quicwg/base-drafts/issues/353 and https://github.com/quicwg/base-drafts/pull/20.

- key phase: To support 0-RTT session establishment, QUIC uses two key phases; the key phase of each packet must be exposed to support efficient reception.

- additional flags: Additional flags for diagnostic use are also under consideration; see https://github.com/quicwg/base-drafts/issues/279.

[Editor's note: also further discuss which bits cannot change with versioning]

2.2. Integrity Protection of the Wire Image

As soon as the cryptograhic context is established, all information in the QUIC header, including that exposed in the packet header, is integrity protected. Therefore, devices on path MUST NOT change QUIC packet headers, as alteration of header information would cause packet drop due to a failed integrity check at the receiver.

2.3. Connection ID and Rebinding

The connection ID in the QUIC packer header is used to allow routing of QUIC packets at load balancers on other than five-tuple information, ensuring that related flows are appropriately balanced together; and to allow rebinding of a connection after one of the endpoint's addresses changes - usually the client's, in the case of the HTTP binding. The connection ID is proposed by the server during connection establishment. A flow might change one of its IP addresses but keep the same connection ID, as noted in Section 2.1, and the connection ID may change during a connection as well; see

section 6.3 of [I-D.ietf-quic-transport]. See also https://github.com/quicwg/base-drafts/issues/349 for ongoing discussion of the Connection ID.

2.4. Packet Numbers

The packet number field is always present in the QUIC packet header. The packet number exposes the least significant 32, 16, or 8 bits of an internal packet counter per flow direction that increments with each packet sent. This packet counter is initialized with a random 31-bit initial value at the start of a connection.

Unlike TCP sequence numbers, this packet number increases with every packet, including those containing only acknowledgment or other control information. Indeed, whether a packet contains user data or only control information is intentionally left unexposed to the network.

While loss detection in QUIC is 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.

2.5. Greasing

[Editor's note: say something about greasing if added to the transport draft]

3. Specific Network Management Tasks

In this section, we address specific network management and measurement techniques and how QUIC's design impacts them.

3.1. Stateful Treatment of QUIC Traffic

Stateful network devices such as firewalls use exposed header information to support state setup and tear-down. [I-D.trammell-plus-statefulness] provides a general model for in-network state management on these devices, independent of transport protocol. Features already present in QUIC may be used for state maintenance in this model. Here, there are two important goals: distinguishing valid QUIC connection establishment from other traffic, in order to establish state; and determining the end of a QUIC connection, in order to tear that state down.

1-RTT connection establishment, using a TLS handshake on stream 0, is detectable using heuristics similar to those used to detect TLS over TCP. 0-RTT connection establishment, however, provides no particular

heuristic for differentiation from random background traffic at this time.

Exposure of connection shutdown is currently under discussion; see https://github.com/quicwg/base-drafts/issues/353 and https://github.com/quicwg/base-drafts/pull/20.

3.2. Measurement of QUIC Traffic

Passive measurement of TCP performance parameters is commonly deployed in access and enterprise networks to aid troubleshooting and performance monitoring without requiring the generation of active measurement traffic.

The presence of packet numbers on most QUIC packets allows the trivial one-sided estimation of packet loss and reordering between the sender and a given observation point. However, since retransmissions are not identifiable as such, loss between an observation point and the receiver cannot be reliably estimated.

The lack of any acknowledgement information or timestamping information in the QUIC packet header makes running passive estimation of latency via round trip time (RTT) impossible. RTT can only be measured at connection establishment time, and only when 1-RTT establishment is used.

Note that adding packet number echo (as in https://github.com/quicwg/base-drafts/pull/367 or https://github.com/quicwg/base-drafts/pull/368) to the public header would allow passive RTT measurement at on-path observation points. For efficiency purposes, this packet number echo need not be carried on every packet, and could be made optional, allowing endpoints to make a measurability/efficiency tradeoff; see section 4 of [IPIM]. Note further that this facility would have significantly better measurability characteristics than sequence-acknowledgement-based RTT measurement currently available in TCP on typical asymmetric flows, as adequate samples will be available in both directions, and packet number echo would be decoupled from the underlying acknowledgment machinery; see e.g. [Ding2015]

Note in-network devices can inspect and correlate connection IDs for partial tracking of mobility events.

3.3. DDoS Detection and Mitigation

For enterprises and network operators one of the biggest management challenges is dealing with Distributed Denial of Service (DDoS) attacks. Some network operators offer Security as a Service (SaaS)

solutions that detect attacks by monitoring, analyzing and filtering traffic. These approaches generally utilize network flow data [RFC7011]. If any flows pose a threat, usually they are routed to a "scrubbing environment" where the traffic is filtered, allowing the remaining "good" traffic to continue to the customer environment.

This type of DDoS mitigation is fundamentally based on tracking state for flows (see Section 3.1) that have receiver confirmation and a proof of return-routability, and classifying flows as legitimate or DoS traffic. The QUIC packet header currently does not support an explicit mechanism to easily distinguish legitimate QUIC traffic from other UDP traffic. However, the first packet in a QUIC connection will usually be a client cleartext packet with a version field and a connection ID. This can be used to identify the first packet of the connection (also see https://github.com/quicwg/base-drafts/issues/185).

If the QUIC handshake was not observed by the defense system, the connection ID can be used as a confirmation signal as per [I-D.trammell-plus-statefulness]. In this case, similar as for all in-network functions that rely on the connection ID, a defense system can only rely on this signal for known QUIC's versions and if the connection ID is present (also see https://github.com/quicwg/base-drafts/issues/293).

Further, the use of a connection ID to support connection migration renders 5-tuple based filtering insufficient, and requires more state to be maintained by DDoS defense systems. However, it is questionable if connection migrations needs to be supported in a DDOS attack or if a defense system might simply rely on the fast resumption mechanism provided by QUIC. This problem is also related to these issues under discussion: https://github.com/quicwg/base-drafts/issues/203 and https://github.com/quicwg/base-drafts/issues/349

3.4. QoS support and ECMP

QUIC does not provide any additional information on requirements on Quality of Service (QoS) provided from the network. QUIC assumes that all packets with the same 5-tuple {dest addr, source addr, protocol, dest port, source port} will receive similar network treatment. That means all stream that are multiplexed over the same QUIC connection require the same network treatment and are handled by the same congestion controller. If differential network treatment is desired, multiple QUIC connection to the same server might be used, given that establishing a new connection using 0-RTT support is cheap and fast.

QoS mechanisms in the network MAY also use the connection ID for service differentiation as usually a change of connection ID is bind to a change of address which anyway is likely to lead to a re-route on a different path with different network characteristics.

Given that QUIC is more tolerant of packet re-ordering than TCP (see Section 2.4), Equal-cost multi-path routing (ECMP) does not necessarily need to be flow based. However, 5-tuple (plus eventually connection ID if present) matching is still beneficial for QoS given all packets are handled by the same congestion controller.

3.5. Load balancing

[Editor's note: explain how this works as soon as we have decided who chooses the connection ID and when to set it. Related to https://github.com/quicwg/base-drafts/issues/349]

4. IANA Considerations

This document has no actions for IANA.

5. Security Considerations

Supporting manageability of QUIC traffic inherently involves tradeoffs with the confidentiality of QUIC's control information; this entire document is therefore security-relevant.

Some of the properties of the QUIC header used in network management are irrelevant to application-layer protocol operation and/or user privacy. For example, packet number exposure (and echo, as proposed in this document), as well as connection establishment exposure for 1-RTT establishment, make no additional information about user traffic available to devices on path.

At the other extreme, supporting current traffic classification methods that operate through the deep packet inspection (DPI) of application-layer headers are directly antithetical to QUIC's goal to provide confidentiality to its application-layer protocol(s); in these cases, alternatives must be found.

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-tls]
Thomson, M. and S. Turner, "Using Transport Layer Security (TLS) to Secure QUIC", draft-ietf-quic-tls-01 (work in progress), January 2017.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-01 (work in progress), January 2017.

[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

[Ding2015]
Ding, H. and M. Rabinovich, "TCP Stretch Acknowledgments and Timestamps - Findings and Impliciations for Passive RTT Measurement (ACM Computer Communication Review)", July 2015.

[draft-kuehlewind-quic-applicability]
Kuehlewind, M. and B. Trammell, "Applicability of the QUIC Transport Protocol", March 2017.

[I-D.ietf-quic-http]
Bishop, M., "Hypertext Transfer Protocol (HTTP) over QUIC", draft-ietf-quic-http-01 (work in progress), January 2017.

[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.

[IPIM] Allman, M., Beverly, R., and B. Trammell, "In-Protocol Internet Measurement (arXiv preprint 1612.02902)", December 2016.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <http://www.rfc-editor.org/info/rfc7011>.

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

Dan Druta
AT&T

Email: dd5826@att.com