Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2018

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

July 03, 2017

Manageability of the QUIC Transport Protocol
draft-ietf-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 January 4, 2018.

Table of Contents

1. Introduction

QUIC [QUIC] is a new transport protocol currently under development in the IETF quic working group, focusing on support of semantics as needed for HTTP/2 [QUIC-HTTP]. Based on current deployment practices, QUIC is encapsulated in UDP and encrypted by default. The current version of QUIC integrates TLS [QUIC-TLS] to encrypt all payload data and most control 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 drafts 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, also the wire image of the header format can change. However, at least the mechanism by which a receiver can determine which version is used and the meaning and location of fields used in the version negotiation process need to be fixed.

This document is focused on the protocol as presently defined in [QUIC] and [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 [QUIC] for the present header structure.

The first bit of the QUIC header indicates the present of a long header that exposes more information than the short header. The long header is typically used during connection start or for other control processes while the short header will be used on most data packets to

limited unnecessary header overhead. The fields and location of these fields as defined by the current version of QUIC for the long header are fixed for all future version as well. However, note that future versions of QUIC may provide additional fields. In the current version of quic the long header for all header types has a fixed length, containing, besides the Header Form bit, a 7-bit header Type, a 64-bit Connection ID, a 32-bit Packet Number, and a 32-bit Version. The short header is variable length where bits after the Header Form bit indicate the present on the Connection ID, and the length of the packet number.

The following information may be exposed in the packet header:

- header type: the long header has a 7-bit header type field following the Header Form bit. The current version of QUIC defines 7 header types, namely Version Negotiation, Client Initial, Server Stateless Retry, Server Cleartext, Client Cleartext, 0-RTT Protected, 1-RTT Protected (key phase 0), 1-RTT Protected (key phase 1), and Public Reset.

- connection ID: The connection ID is always present on the long and optionally present on the short header indicated by the Connection ID Flag. If present at the short header it at the same position then for the long header. The position and length pf the congestion ID itself as well as the Connection ID flag in the short header is fixed for all versions of QUIC. The connection ID identifies the connection associated with a QUIC packet, for load-balancing and NAT rebinding purposes; see Section 3.5 and Section 2.3. Therefore it is also expected that the Connection ID will either be present on all packets of a flow or none of the short header packets. However, this field is under endpoint control and there is protocol mechanism that hinders the sending endpoint to revise its decision about exposing the Connection ID at any time during the connection.

- packet number: Every packet has an associated packet number. The packet number increases with each packet, and the least-significant bits of the packet number are present on each packet. In the short header the length of the exposed packet number field is defined by the (short) header type and can either be 8, 16, or 32 bits. See Section 2.4.

- version number: The version number is present on the long headers and identifies the version used for that packet, expect for the Version negotiation packet. The version negotiation packet is fixed for all version of QUIC and contains a lit of versions that is supported by the sender. However the version in the version field of the header is the reflected version of the clients

initial packet and is therefore explicitly not supported by the sender.

- o key phase: The short header further has a Key Phase flag that is used by the endpoint identify the right key that was used to encrypt the packet. Different key phases are indicated with the use of the long header by using to different header types for protected long header packets.

2.2. Integrity Protection of the Wire Image

As soon as the cryptographic context is established, all information in the QUIC header, including those exposed in the packet header, is integrity protected. Further, information that were sent and exposed in previous packets when the cryptographic context was established yet, e.g. for the cryptographic initial handshake itself, will be validate later during the cryptographic handshake, such as the version number. Therefore, devices on path MUST NOT change any information or bits in QUIC packet headers. As alteration of header information would cause packet drop due to a failed integrity check at the receiver, or can even lead to connection termination.

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, and a server might provide additional connection IDs that can the used by the client at any time during the connection. Therefore if a flow changes one of its IP addresses it may keep the same connection ID, or the connection ID may also change together with the IP address migration, avoiding linkability; see Section 7.6 of [QUIC].

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. The packet number increases with every packet but they sender may skip packet numbers.

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. Initial Handshake and PMTUD

[Editor's note: text needed.]

2.6. Version Negotiation and Greasing

Version negotiation is not protected, given the used protection mechanism can change with the version. However, the choices provided in the list of version in the Version Negotiation packet will be validated as soon as the cryptographic context has been established. Therefore any manipulation of this list will be detected and will cause the endpoints to terminate the connection.

Also note that the list of versions in the Version Negotiation packet may contain reserved versions. This mechanism is used to avoid ossification in the implementation on the selection mechanism. Further, a client may send a Initial Client packet with a reserved version number to trigger version negotiation. In the Version Negotiation packet the connection ID and packet number of the Client Initial packet are reflected to provide a proof of return-routability. Therefore changing these information will also cause the connection to fail.

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. [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.

Both, 1-RTT and O-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 may additional also send data packets, right after the client hello. These data may be reorder in the network, therefore it may be possible that 0-RTT Protected data packet are seen before the Client Initial packet.

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 all 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, by observing the Client Initial packet and the Server's reply to this packet which maybe a Server Cleartext, Version Negotiation, or Server Stateless Retry packet.

Note that adding a 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 Initial packet. This can be used to identify the first packet of the connection.

If the QUIC handshake was not observed by the defense system, the connection ID can be used as a confirmation signal as per [STATEFULNESS] if present in both directions.

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. If the connection migration is not visible to the network that performs the DDoS detection, an active, migrated QUIC connection may be blocked by such a system under attack. However, 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

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 using the Connection ID

The Connection ID is used in part to support load balancing in content distribution networks (CDNs), which operate complex, geographically distributed pools of back-end servers, fronted by load balancing systems. These load balancers are responsible for identifying the most appropriate server for each connection and for routing all packets belonging to that connection to the chosen server.

Load balancers are often deployed in pools for redundancy and load sharing. For high availability, it is important that when packets belonging to a flow start to arrive at a different load balancer in the load balancer pool, the packets continue to be forwarded to the original server in the server pool.

Support for seamless connection migration is an important design goal of QUIC - a necessity due to the proliferation of mobile connected devices. This connection persistence provides an additional challenge for multi-homed anycast-based services often employed by large content owners and CDNs. The challenge is that a migration to a different network in the middle of the connection greatly increases the chances of the packets routed to a different anycast point of presence (POP) due to the new network's different connectivity and Internet peering arrangements. The load balancer in the new POP, potentially thousands of miles away, will not have information about the established flow and would not be able to route it back to the original POP.

Load balancers may cooperate with servers or server pools behind them to use a server-generated Connection ID value, in order to support stateless load balancing, even across NAT rebinding or other address change events (see Section 2.3). See section 5.7 of [QUIC].

Server-generated Connection IDs must not encode any information other that that needed to route packets to the appropriate backend server(s): typically the identity of the backend server or pool of

servers, if the data-center's load balancing system keeps "local" state of all flows itself. Care must be exercised to ensure that the information encoded in the Connection ID is not sufficient to identify unique end users. Note that by encoding routing information in the Connection ID, load balancers open up a new attack vector that allows bad actors to direct traffic at a specific backend server or pool. It is therefore recommended that Server-Generated Connection ID includes a cryptographic MAC that the load balancer pool server are able to identify and discard packets featuring an invalid MAC.

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. Contributors

Igor Lubashev contributed text to Section 3.5 on the use of the connection ID for load balancing.

7. 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.

8. References

8.1. Normative References

[QUIC] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-04 (work in progress), June 2017.

[QUIC-HTTP] Bishop, M., "Hypertext Transfer Protocol (HTTP) over QUIC", draft-ietf-quic-http-04 (work in progress), June 2017.

[QUIC-TLS] Thomson, M. and S. Turner, "Using Transport Layer Security (TLS) to Secure QUIC", draft-ietf-quic-tls-04 (work in progress), June 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>.

8.2. 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, <http://www.sigcomm.org/sites/default/files/ccr/papers/2015/July/0000000-0000002.pdf>.

[IPIM] Allman, M., Beverly, R., and B. Trammell, "In-Protocol Internet Measurement (arXiv preprint 1612.02902)", December 2016, <https://arxiv.org/abs/1612.02902>.

[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>.

[STATEFULNESS] Kuehlewind, M., Trammell, B., and J. Hildebrand, "Transport-Independent Path Layer State Management", draft-trammell-plus-statefulness-03 (work in progress), March 2017.

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