Network Working Group M. Kuehlewind
Internet-Draft B. Trammell
Intended status: Informational ETH Zurich
Expires: September 9, 2017 March 08, 2017

Applicability of the QUIC Transport Protocol
draft-kuehlewind-quic-applicability-00

Abstract

This document discusses the applicability of the QUIC transport protocol, focusing on caveats impacting application protocol development and deployment over QUIC. Its intended audience is designers of application protocol mappings to QUIC, and implementors of these application protocols.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Notational Conventions 2
2. The Necessity of Fallback 3
3. Zero RTT: Here There Be Dragons 3
4. Stream versus Flow Multiplexing 4
5. Prioritization 4
6. Graceful connection closure 5
7. Information exposure and the Connection ID 5
8. Use of Versions and Cryptographic Handshake 5
9. IANA Considerations 5
10. Security Considerations 5
11. Acknowledgments 6
12. References 6
12.1. Normative References 6
12.2. Informative References 6
Authors' Addresses 7

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 and most header information.

This document 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].

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

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. The Necessity of 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 some other transport protocol. This fallback SHOULD provide TLS 1.3 or equivalent cryptographic protection, if available, in order to keep fallback from being exploited as a downgrade attack. In the case of HTTP, this fallback is TLS 1.3 over TCP.

These applications must operate, perhaps with impaired functionality, in the absence of features provided by QUIC not present in the fallback protocol. For fallback to TLS over TCP, the most obvious difference is that TCP does not provide stream multiplexing and therefore stream multiplexing 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 could be used, but might no be supported by the far end or could be blocked on the network path. Note that there is some evidence of middleboxes blocking SYN data even if TFO was successfully negotiated (see [PaaschNanog]). 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, allowing the application to present a suitable prompt to the user that secure communication is unavailable.

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 discussed in section 3.1 of [draft-kuehlewind-quic-manageability] would remove one network management incentive to block this traffic.

3. 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 frames and dropping them.

Applications that cannot treat data that may appear in a 0-RTT connection establishment as idempotent MUST NOT use 0-RTT establishment. For this reason the QUIC transport SHOULD provide an interface for the application to indicate if 0-RTT support is in general desired or a way to indicate if data is idempotent.

4. 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.]

5. Prioritization

Stream prioritization is not exposed to the network, nor to the receiver. Prioritization can be realized by the sender and the QUIC transport should provide and interface for applications to prioritize streams [I-D.ietf-quic-transport].

Priority handling of retransmissions may be implemented in the transport layer and [I-D.ietf-quic-transport] does not specify a specific way how this must be handled. Currently QUIC only provides fully reliable stream transmission, and as such prioritization of retransmission is likely beneficial. For not fully reliable streams priority scheduling of retransmissions over data of higher-priority streams might not be desired. In this case QUIC could also provide an interface or derive the prioritization decision from the reliability level of the stream.

6. Graceful connection closure

[EDITOR'S NOTE: give some guidance here about the steps an application should take; however this is still work in progress]

7. Information exposure and the Connection ID

QUIC exposed some information to the network in the unencrypted part of the header. This is either because there is no encryption context established yet or because this information is intended to be consumed by the network. Some of these information can be optionally exposed (still under discussion). Given that exposing these information can have privacy implications, an application may indicate to not support exposure of certain information.

In case of the connection ID this can be the case if the application has additional information that the client is not behind a NAT and the server is not behind a load balancer, and therefore it is unlikely that the addresses will be re-binded.

8. Use of Versions and Cryptographic Handshake

Versioning in QUIC may change the whole protocol behavior, beside some header fields that have been declared to be fixed. As such a new or higher version of QUIC does not necessarily provide a better service but just a very different service, an application needs to be able to select which versions of QUIC it wants to use.

The use of a different encryption scheme than TLS1.3 or higher needs a new version of QUIC. [I-D.ietf-quic-transport] specifies requirements for the cryptographic handshake as currently realized by TLS1.3 and described in a separate specification [I-D.ietf-quic-tls]. This split is performed to enable light-weight versioning with different cryptographic handshakes.

9. IANA Considerations

This document has no actions for IANA.

10. Security Considerations

See the security considerations in [I-D.ietf-quic-transport] and [I-D.ietf-quic-tls]; the security considerations for the underlying transport protocol are relevant for applications using QUIC, as well.

Application developers should note that any fallback they use when QUIC cannot be used due to network blocking of UDP SHOULD guarantee the same security properties as QUIC; if this is not possible, the

connection SHOULD fail to allow the application to explicitly handle fallback to a less-secure alternative. See Section 2.

11. 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.

12. References

12.1. 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>.

12.2. Informative References

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

[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-01 (work in progress), January 2017.

[PaaschNanog]
Paasch, C., "Network Ssupport for TCP Fast Open (NANOG 67 presentation)", June 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