AVTCORE Working Group
INTERNET-DRAFT
Category: Informational
Expires: July 23, 2019

B. Aboba
Microsoft Corporation
P. Thatcher
Google
C. Perkins
University of Glasgow
23 January 2019

QUIC Multiplexing

draft-aboba-avtcore-quic-multiplexing-03.txt

Abstract

If QUIC is to be used in a peer-to-peer manner, with NAT traversal, then it is necessary to be able to demultiplex QUIC and other protocols used in WebRTC on a single UDP port. This memo discusses options for demultiplexing.

Status of This Memo

Table of Contents

1. Introduction 2
1.1. Terminology 2
2. Solution 3
2.1. Subsequent changes 4
3. Security Considerations 4
4. IANA Considerations 5
5. References 5
5.1. Informative references 5
Acknowledgments 7
Authors' Addresses 7

1. Introduction

QUIC [I-D.ietf-quic-transport] is a new network transport protocol. While it is initially intended as a replacement for TCP in order to better support HTTP/2 [RFC7540] it should eventually be useful as a general purpose transport. HTTP is an asymmetric client-server protocol, but other uses of QUIC might operate in a peer-to-peer manner and so will need effective NAT traversal using ICE [RFC5245], which which makes use of STUN [RFC5389] and TURN [RFC5766] to discover NAT bindings. Therefore for QUIC to be utilized for peer-to-peer data transport, QUIC and STUN must be able to multiplex on the same port.

In a WebRTC scenario where RTP is used to transport audio and video and QUIC is used for data exchange, SRTP [RFC3711] is keyed using DTLS-SRTP [RFC5764] and therefore SRTP/SRTCP [RFC3550], STUN, TURN, DTLS [RFC6347] and QUIC will need to be multiplexed on the same port.

Within the W3C, a Javascript API for the use of QUIC for peer-to-peer data exchange [WEBRTC-QUIC] is under development within the ORTC

Community Group.

As noted in [RFC7983] Figure 3, protocol demultiplexing currently relies upon differentiation based on the first octet, as follows:

```
                 +----------------+
                 |        [0..3] -+--> forward to STUN
                 |                |
                 |      [16..19] -+--> forward to ZRTP
                 |                |
     packet -->  |      [20..63] -+--> forward to DTLS
                 |                |
                 |      [64..79] -+--> forward to TURN Channel
                 |                |
                 |    [128..191] -+--> forward to RTP/RTCP
                 +----------------+
```

Figure 1: RFC 7983 packet demultiplexing algorithm.

As noted by Colin Perkins and Lars Eggert in [QUIC-Issue] this created a potential conflict with the design of the QUIC headers described in versions of [I-D.ietf-quic-transport] prior to -08.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Solution

At IETF 100, Colin Perkins presented a demultiplexing proposal [QUIC-MULTI]. The proposal which was subsequently proposed as a Pull Request to the QUIC Transport specification and merged in draft-ietf-quic-transport-08, involved renumbering of the QUIC long header packet type field as well as inverting the sense of the "C" bit in the short header packet.

The demultiplexing algorithm resulting from the changes appears as follows:

```
             +----------------+
             |        [0..3] -+--> forward to STUN
             |                |
             |      [16..19] -+--> forward to ZRTP
             |                |
 packet -->  |      [20..63] -+--> forward to DTLS
             |                |
             |      [64..79] -+--> forward to TURN Channel
             |     [64..127] -+--> forward to QUIC (Short Header)
             |                |
             |    [128..191] -+--> forward to RTP/RTCP
             |    [250..255]  +--> forward to QUIC (Long Header)
             +----------------+
```

Figure 2: Revised packet demultiplexing algorithm.

Note that while the above diagram has a potential conflict between packets sent in TURN Channels and the QUIC short header, this conflict is not considered serious for WebRTC where TURN Channels are rarely used.

2.1. Subsequent changes

Since then, additional changes have been made to the QUIC transport headers. As of draft 18, the QUIC Long Header packet type field defined in [I-D.ietf-quic-transport] Section 17.2 appears as follows:

```
+-+-+-+-+-+-+-+-+
|1|1|T T|X|X|X|X|
+-+-+-+-+-+-+-+-+
```

Where:

T = Long Packet Type (0x0 - 0x3)
X = Type-Specific Bits.

This potentially produces values of the first octet in the ranges 192-255.

The QUIC Short Header packet type field defined in [I-D.ietf-quic-transport] Section 17.3 appears as follows:

```
+-+-+-+-+-+-+-+-+
|0|1|S|R|R|K|P P|
+-+-+-+-+-+-+-+-+
```

Where:

S = Spin Bit
R = Reserved bits
K = Key Phase bit
P = Packet Number Length.

This potentially produces values of the first octet in the ranges 64-127 (assuming that the reserved bits may not always be set to zero).

As a result, the multiplexing scheme supported in -18 operates as follows:

```
             +----------------+
             |        [0..3] -+--> forward to STUN
             |                |
             |      [16..19] -+--> forward to ZRTP
             |                |
 packet -->  |      [20..63] -+--> forward to DTLS
             |                |
             |      [64..79] -+--> forward to TURN Channel
             |     [64..127] -+--> forward to QUIC (Short Header)
             |                |
             |    [128..191] -+--> forward to RTP/RTCP
             |    [192..255]  +--> forward to QUIC (Long Header)
             +----------------+
```

Figure 3: Packet demultiplexing algorithm in Draft 18.

3. Security Considerations

The solutions discussed in this document could potentially introduce some additional security considerations beyond those detailed in [RFC7983].

Due to the additional logic required, if mis-implemented, heuristics have the potential to mis-classify packets.

When QUIC is used for only for data exchange, the TLS-within-QUIC exchange [I-D.ietf-quic-tls] derives keys used solely to protect the QUIC data packets. If properly implemented, this should not affect the transport of SRTP nor the derivation of SRTP keys via DTLS-SRTP, but if badly implemented, both transport and key derivation could be adversely impacted.

4. IANA Considerations

This document does not require actions by IANA.

5. References

5.1. Informative References

[I-D.ietf-quic-tls]
Thomson, M. and S. Turner, "Using Transport Layer Security (TLS) to Secure QUIC", draft-ietf-quic-tls-18 (work in progress), January 23, 2019.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-18 (work in progress), January 23, 2019.

[QUIC-Issue] Perkins, C., "QUIC header format/demultiplexing", https://github.com/quicwg/base-drafts/issues/426, March, 2017.

[QUIC-MULTI] Perkins, C., "QUIC Multiplexing and Peer-to-Peer", presentation to IETF AVTCORE WG at IETF 100, <https://datatracker.ietf.org/meeting/100/materials/slides-100-avtcore-quic-multiplexing-with-rtp-03>, November 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>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <http://www.rfc-editor.org/info/rfc3711>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <http://www.rfc-editor.org/info/rfc5389>.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010, <http://www.rfc-editor.org/info/rfc5764>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <http://www.rfc-editor.org/info/rfc5766>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540>.

[RFC7983] Petit-Huguenin, M. and G. Salgueiro, "Multiplexing Scheme Updates for Secure Real-time Transport Protocol (SRTP) Extension for Datagram Transport Layer Security (DTLS)", RFC 7983, DOI 10.17487/RFC7983, September 2016, <https://www.rfc-editor.org/info/rfc7983>.

[WEBRTC-QUIC] Thatcher, P. and B. Aboba, "QUIC API For WebRTC", W3C Editor's Draft (work in progress), October 2018, <https://w3c.github.io/webrtc-quic>

Acknowledgments

We would like to thank Martin Thomson, Roni Even and other participants in the IETF QUIC and AVTCORE working groups for their discussion of the QUIC multiplexing issue, and their input relating to potential solutions.

Authors' Addresses

Bernard Aboba
Microsoft Corporation

One Microsoft Way
Redmond, WA 98052
USA

Email: bernard.aboba@gmail.com

Peter Thatcher
Google
747 6th St S
Kirkland, WA 98033
USA

Email: pthatcher@google.com

Colin Perkins
School of Computing Science
University of Glasgow
Glasgow G12 8QQ
United Kingdom

Email: csp@csperkins.org