AVTCORE Working Group
INTERNET-DRAFT
Category: Informational
Expires: April 24, 2018

B. Aboba
Microsoft Corporation
P. Thatcher
Google
C. Perkins
University of Glasgow
23 October 2017

QUIC Multiplexing
draft-aboba-avtcore-quic-multiplexing-00.txt

Abstract

This document describes potential approaches to multiplexing of QUIC along with RTP, RTCP, DTLS, STUN, TURN and ZRTP in WebRTC peer-to-peer data exchange.

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 April 24, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

There are a number of ways in which communication between WebRTC peers may utilize QUIC. One of these is transport of RTP over QUIC, described in [I-D.rtpfolks-quic-rtp-over-quic]. Another is use of QUIC [I-D.ietf-quic-transport] for data exchange. A Javascript API for use of QUIC in WebRTC data exchange has been incorporated into the ORTC API [ORTC], under development within the W3C ORTC Community Group.

In a WebRTC scenario where ICE [RFC5245] is utilized for NAT traversal, SRTP [RFC3711] is keyed using DTLS-SRTP [RFC5764] and QUIC is used for data exchange, RTP/RTCP [RFC3550] STUN [RFC5389], TURN [RFC5766], DTLS [RFC6347], ZRTP [RFC6189] and QUIC may all need to be multiplexed over a single ICE transport.

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: DTLS-SRTP receiver's packet demultiplexing algorithm.

As noted by Colin Perkins and Lars Eggert in [QUIC-Issue] this creates a potential conflict with the current design of the QUIC headers described in [I-D.ietf-quic-transport], since the first octet of the QUIC header is either:

```
 +-+-+-+-+-+-+-+-+
 |1|   Type (7)  |  Long header packet
 +-+-+-+-+-+-+-+-+
```

which potentially produces values of the first octet in the range 129-134, conflicting with RTP/RTCP, or

```
 +-+-+-+-+-+-+-+-+
 |0|C|K| Type (5)|  Short header packet
 +-+-+-+-+-+-+-+-+
```

which produces values for the first octet in the ranges 1-3, 33-35, 65-67 or 97-99, potentially conflicting with STUN, DTLS and TURN.

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. Solutions

This section presents potential solutions to the QUIC multiplexing problem, including changes to the QUIC headers, addition of a multiplexing octet and use of heuristics.

2.1. QUIC Header Changes

As noted in [QUIC-Issue], one potential solution involves changes to the QUIC headers, such as setting the top two bits of the first octet of a QUIC packet to 1. This would imply a reduction in the size of the type fields:

```
+-+-+-+-+-+-+-+-+
|1|1|1|Type (5) |  Long header packet
+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+
|1|1|0|C|K|Type3|  Short header packet
+-+-+-+-+-+-+-+-+
```

Note: [QUIC-Spin] proposes to add a spin bit to the type octet within the QUIC header, in order to allow for RTT calculation. This would leave 4 bits for the type field in the long header packet and 2 bits for the type field in the short header, which would accomodate the type field values allocated in [I-D.ietf-quic-transport].

2.1.1. Pros and Cons

The advantage to this approach is that it adds no additional overhead on-the-wire. However it does require a reduction in the size of the QUIC Type fields and could potentially require allocation of the following initial octet code points for QUIC: For the Long header, 225-230 (241-246 when the spin bit is set) and for the Short header, 193-195 (209-11 with spin bit set), 209-211 (225-227 with spin bit set) and 217-219 (233-235 with the spin bit set). Utilizing all of these code points for QUIC would leave limited code points available for future allocations.

2.2. Multiplexing Shim

In this approach, an initial octet not allocated within [RFC7983] would be prepended to each QUIC packet, allowing QUIC packets to be differentiated from RTP, RTCP, DTLS, STUN, TURN and ZRTP based on the first octet alone. As an example, an octet with decimal value 192 could be used:

```
+-+-+-+-+-+-+-+-+
|1|1|0|0|0|0|0|0|
+-+-+-+-+-+-+-+-+
```

2.2.1. Pros and Cons

Advantages of this approach include simplicity and the consumption of only a single initial octet code point for demultiplexing of QUIC. The disadvantage is the addition of a single octet of overhead to every QUIC packet, which could impact performance where small payloads are exchanged, such as in peer-to-peer gaming.

2.3. Heuristics

During the QUIC WG interim in Seattle, Martin Thomson suggested the following heuristics for differentiation of QUIC packets from RTP/RTCP/DTLS/STUN/TURN/ZRTP:

1. Demultiplex differently during the "QUIC handshake" and "steady state".
2. During handshake, we only need to worry about the QUIC Long header, which simplifies the logic.
   a. Force all handshake packets to utilize the QUIC Long header.
   b. The QUIC Long header (0x1XXXXXXX) (or 0x11XXXXXX with the spin bit set) does not conflict with STUN (0x000000XX), DTLS (0x000XXXXX), or TURN Channel (0x01XXXXXX).
   c. The QUIC Long header does conflict with RTP/RTCP (0x10XXXXXX), but those packets typically aren’t sent until the QUIC handshake is completed. Corner case: an application starts off with audio and video keyed with DTLS-SRTP without QUIC, then the application wishes to add QUIC data (e.g. the user clicks on the "white-board" icon).
      i. Alternative: force the RTP padding bit to 1 using a one-byte pad if there isn’t already padding (pad == 0x01). Then force QUIC to have a type < 64 (the current max is 8).
      ii. Alternative: Disallow QUIC in this case, use SCTP data exchange instead.
3. During "steady state", we only need to worry about the QUIC Short header.

a. QUIC doesn't need the Long header after the handshake.
b. The QUIC Short header (0x0XXXXXXX or 0x01XXXXXX with the spin bit set) does not conflict with RTP/RTCP (0x10XXXXXX), so we only need to worry about conflicts with STUN/TURN/DTLS/ZRTP.
c. Disallow simultaneous use of DTLS and QUIC Short header packets.
   i. Alternative: when using DTLS and QUIC at the same time, only use the QUIC Long header. Not optimal, but isn't really needed.
d. ICE can be demuxed using the magic cookie and checksum.
   i. Alternative: STUN can only conflict with 3 QUIC packet types: Version Negotiation, Client Initial, and Server Stateless Retry. Out of those, none should be needed during the steady state.
e. You shouldn't need to demultiplex QUIC with TURN channel data or other STUN traffic. But what about consent packets?

2.3.1. Pros and Cons

This approach has the advantage that it requires no changes to QUIC headers, nor does it add any overhead to QUIC packets. Disadvantages include additional complexity within the multiplexing algorithm, the consumption of additional multiplexing code points, and potential future difficulties in adapting the algorithm to support changes to the QUIC protocol or additional protocols to be multiplexed.

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 misclassify 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-07 (work in progress), October 2017.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-07 (work in progress), October 2017.

[I-D.rtpfolks-quic-rtp-over-quic]
Ott, J., Even, R., Perkins, C. and V. Singh, "RTP over QUIC", Internet draft (work in progress), draft-rtpfolks-quic-rtp-over-quic-01, September 1, 2017.

[ORTC] Raymond, R., Aboba, B. and J. Uberti, "Object RTC (ORTC) API for WebRTC", W3C, http://draft.ortc.org/, October 2017.

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

[QUIC-Spin] Huitema, C., "QUIC Latency Spin Bit", https://github.com/quicwg/base-drafts/issues/609, 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>.

[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>.

[RFC6189] Zimmermann, P., Johnston, A., Ed., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", RFC 6189, DOI 10.17487/RFC6189, April 2011, <http://www.rfc-editor.org/info/rfc6189>.

[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>.

[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, September 2016.

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

Email: csp@csperkins.org