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

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 23, 2019.
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 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
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 |
|                |
| [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)
|                |
|      [64..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. While the QUIC Long Header packet type field retains its original allocations between 0x7C and 0x7F, as of draft -15, the first octet of the Short Header now appears as follows:

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

Where:

K = indicates the key phase.
R = reserved bits, set randomly by endpoints not actively using them.

This potentially produces values of the first octet in the ranges 48-55 which potentially conflicts with DTLS, and 80-87 which conflicts with TURN channels (not an issue).

3. Security Considerations

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

[I-D.ietf-quic-transport]


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