Abstract

This document defines how QUIC, Datagram Transport Layer Security (DTLS), Real-time Transport Protocol (RTP), RTP Control Protocol (RTCP), Session Traversal Utilities for NAT (STUN), Traversal Using Relays around NAT (TURN), and ZRTP packets are multiplexed on a single receiving socket.

This document updates RFC 7983 and RFC 5764.

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 May 21, 2021.
Copyright Notice

Copyright (c) 2020 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 .................................................. 3
   1.1. Terminology ............................................... 3
2. Multiplexing of TURN Channels ............................... 3
3. Updates to RFC 7983 ......................................... 4
4. Security Considerations ................................. 5
5. IANA Considerations .................................. 6
6. References .................................................... 6
   6.1. Normative References ................................. 6
   6.2. Informative References ............................. 7
Acknowledgements ............................................... 7
Authors’ Addresses ........................................... 8
1. Introduction

"Multiplexing Scheme Updates for Secure Real-time Transport Protocol (SRTP) Extension for Datagram Transport Layer Security (DTLS)"

[RFC7983] defines a scheme for a Real-time Transport Protocol (RTP) [RFC3550] receiver to demultiplex DTLS [RFC6347], Session Traversal Utilities for NAT (STUN) [RFC5389], Secure Real-time Transport Protocol (SRTP) / Secure Real-time Transport Control Protocol (SRTCP) [RFC3711], ZRTP [RFC6189] and TURN Channel packets arriving on a single port.

This document updates [RFC7983] and [RFC5764] to also allow QUIC [I-D.ietf-quic-transport] to be multiplexed on the same port. For peer-to-peer operation in WebRTC scenarios as described in [WEBRTC-QUIC][WEBRTC-QUIC-TRIAL], 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 need to be multiplexed on the same port.

Since new versions of QUIC are allowed to change aspects of the wire image, there is no guarantee that future versions of QUIC beyond version 1 will adhere to the multiplexing scheme described in this document.

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. Multiplexing of TURN Channels

TURN channels are an optimization where data packets are exchanged with a 4-byte prefix instead of the standard 36-byte STUN overhead (see Section 2.5 of [RFC5766]). [RFC7983] allocated the values from 64 to 79 in order to allow TURN channels to be demultiplexed when the TURN Client does the channel binding request in combination with the demultiplexing scheme described in [RFC7983].

As noted in [I-D.aboba-avtcore-quic-multiplexing], the first octet of a QUIC short header packet falls in the range 64 to 127, thereby overlapping with the allocated range for TURN channels of 64 to 79.

The first octet of QUIC long header packets fall in the range 192 to 255. Since QUIC long header packets preceed QUIC short header packets, if no packets with a first octet in the range of 192 to 255 have been received, a packet whose first octet is in the range of 64 to 79 can be demultiplexed unambiguously as TURN Channel traffic.
Since WebRTC implementations supporting QUIC data exchange do not utilize TURN Channels, once packets with a first octet in the range of 192 to 255 have been received, a packet whose first octet is in the range of 64 to 127 can be demultiplexed as QUIC traffic.

3. Updates to RFC 7983

This document updates the text in Section 7 of [RFC7983] (which in turn updates [RFC5764]) as follows:

OLD TEXT

The process for demultiplexing a packet is as follows. The receiver looks at the first byte of the packet. If the value of this byte is in between 0 and 3 (inclusive), then the packet is STUN. If the value is between 16 and 19 (inclusive), then the packet is ZRTP. If the value is between 20 and 63 (inclusive), then the packet is DTLS. If the value is between 64 and 79 (inclusive), then the packet is TURN Channel. If the value is in between 128 and 191 (inclusive), then the packet is RTP (or RTCP, if both RTCP and RTP are being multiplexed over the same destination port). If the value does not match any known range, then the packet MUST be dropped and an alert MAY be logged. This process is summarized in Figure 3.

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

Figure 3: The DTLS-SRTP receiver’s packet demultiplexing algorithm.

END OLD TEXT

NEW TEXT

The process for demultiplexing a packet is as follows. The receiver looks at the first byte of the packet. If the value of this byte is in between 0 and 3 (inclusive), then the packet is STUN. If the value is between 16 and 19 (inclusive), then the packet is ZRTP. If the value is between 20 and 63 (inclusive), then the packet is DTLS. If the value is in between 128 and 191 (inclusive), then the packet is RTP (or RTCP, if both RTCP and RTP are being multiplexed over the same destination port). If the value does not match any known range, then the packet MUST be dropped and an alert MAY be logged. This process is summarized in Figure 3.
RTP (or RTCP, if both RTCP and RTP are being multiplexed over the same destination port). If the value is between 80 and 127 or between 192 and 255 (inclusive) then the packet is QUIC. If the value is between 64 and 79 inclusive, then if a packet has been previously forwarded that is in the range of 192 and 255, then the packet is QUIC, otherwise it is TURN Channel.

If the value does not match any known range, then the packet MUST be dropped and an alert MAY be logged. This process is summarized in Figure 3.

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

Figure 3: The receiver’s packet demultiplexing algorithm.

END NEW TEXT

4. Security Considerations

The solution 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.
5. IANA Considerations

This document does not require actions by IANA.

6. References

6.1. Normative References

[I-D.ietf-quic-tls]

[I-D.ietf-quic-transport]


6.2. Informative References

[I-D.aboba-avtcore-quic-multiplexing]


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

Gonzalo Salgueiro
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, NC  27709
United States of America

Email:  gsalguei@cisco.com

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

Email:  csp@csperkins.org