QRT: QUIC RTP Tunnelling
draft-hurst-quic-rtp-tunnelling-00

Abstract

   QUIC is a UDP-based transport protocol for stream-orientated,
   congestion-controlled, secure, multiplexed data transfer. RTP
   carries real-time data between endpoints, and the accompanying
   control protocol RTCP allows monitoring and control of the transfer
   of such data. With RTP and RTCP being agnostic to the underlying
   transport protocol, it is possible to multiplex both the RTP and
   associated RTCP flows into a single QUIC connection to take advantage
   of QUIC features such as low-latency setup and strong TLS-based
   security.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] provides end-to-end network transport functions suitable for applications transmitting data, such as audio and video, over multicast or unicast network services for the purposes of telephony, video streaming, conferencing and other real-time applications.

The QUIC transport protocol is a UDP-based stream-orientated and encrypted transport protocol aimed at offering improvements over the common combination of TCP and TLS for web applications. Compared with TCP+TLS, QUIC offers much reduced connection set-up times,

improved stream multiplexing aware congestion control, and the
ability to perform connection migration. QUIC offers two modes of
data transfer:

*   Reliable transfer using STREAM frames, as specified in
    [QUIC-TRANSPORT], [QUIC-RECOVERY], etc.

*   Unreliable transfer using DATAGRAM extension frames, as specified
    in [QUIC-DATAGRAM].

RTP has traditionally been run over UDP or DTLS to achieve timely but
unreliable data transfer. For use cases such as real-time audio and
video transmission, the underlying media codecs can be considered in
part fault-tolerant to an unreliable transport mechanism, with
missing data from the stream resulting in glitches in the media
presentation, such as missing video frames or gaps in audio playback.
By purposely using an unreliable transport mechanism, applications
can minimise the added latency that would otherwise result from
managing the large packet reception buffers needed to account for
network reordering or transport protocol retransmission.

1.1. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in BCP
14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

Packet and frame diagrams in this document use the format described
in [QUIC-TRANSPORT].

1.2. Definitions

*   Endpoint: host capable of being a participant in a QRT session.

*   QRT session: A QUIC connection carrying one or more RTP sessions,
    each with or without an accompanying RTCP channel.

*   Client: The endpoint which initiates the QUIC connection.

*   Server: The endpoint which accepts the incoming QUIC connection.

2. Use Cases for an RTP Mapping over QUIC

   The following sections describe some possible use cases for an RTP
   and RTCP mapping over QUIC, hereafter QRT. The examples were chosen
   to illustrate some basic concepts, and are neither an exhaustive list
   of possible use cases nor a limitation on what QRT may be used for.

2.1. Live Event Contribution Feed

   A news organisation wishes to provide a two-way link to a live event
   for distribution as part of an item in a news programme hosted in a
   studio with a news anchor. The single camera remote production crew
   will include a camera operator, sound technician and the reporter.
   In order to deliver this experience, the following media flows are
   required:

   * A high-quality video feed from the remote camera to the news
      organisation's gallery;

   * One or more audio feeds for microphones at the event, including an
      ambient microphone attached to the camera, a lapel microphone for
      the reporter, and a handheld microphone to conduct interviews, all
      synchronized;

   * A video feed of the programme output from the gallery, after
      mixing for local monitoring and for use on a comfort monitor;

   * An audio feed from the anchor in the studio to the reporter;

   * A two-way audio feed from the gallery to the remote production
      crew for talkback communication;

   * A tally light feed for the remote camera.

   These media flows may be realised as a group of RTP sessions, some of
   which must be synchronised together. The talkback streams do not
   require any tight synchronisation with other streams in the group,
   whereas the camera video feed and various microphone feeds need to be
   tightly synchronised together.

   At the event, a production machine running a software package that
   includes a QRT client has two connections to the Internet; a high-
   speed fibre link and a bonded cellular network link for backup.

   In order to prevent a bad actor on the network path being able to
   tamper with the contribution, all communication between the news
   organisation's gallery and the remote production need to be
   encrypted. Because all the data is flowing between the same two

endpoints, only a single QRT session is required, and the various RTP
sessions that are encapsulated by the QRT session are (de)multiplexed
at each end.

During the live contribution, an accident cuts the fibre connection
to the remote production crew. Using the QUIC connection migration
mechanism presented in Section 9 of [QUIC-TRANSPORT], the QRT session
migrates from the fibre link onto the backup cellular link. This
preserves the state of the RTP sessions across a network migration
event, and all sessions continue.

2.2. Audio and Video Conference via a Central Server

A teleconference is taking place across multiple sites using a
centralised server. All participants connect to this single server,
and the server acts as an RTP mixer to reduce the number of RTP
sessions being sent to all participants, as well as re-encoding the
streams for efficiency reasons.

One participant of this conference has connected via mobile phone.
However, when the participant enters the range of a previously-
associated WiFi network, the mobile phone switches its network
connection across to this new network. The QRT session can then
migrate across, and the participant is able to continue the call with
minimal interruption.

3. QRT Sessions

A QRT session is defined as a QUIC connection which carries one or
more RTP sessions (including any associated RTCP flows) using
"DATAGRAM" frames, as specified in Section 4. Those RTP sessions may
be part of one or more RTP multimedia sessions, and a multimedia
session may be comprised of RTP sessions carried in one or more QRT
sessions.

A QRT session inherits the standard QUIC handshake as specified in
[QUIC-TRANSPORT], and all communications between endpoints are
secured as specified in [QUIC-TLS].

4. RTP Sessions

QRT allows multiple RTP sessions to be carried in a single QRT
session. Each RTP session is operated independently of all the
others, and individually discriminated by an QRT Flow Identifier, as
described below in Section 4.1.

RTP and RTCP packets are carried in QUIC "DATAGRAM" frames, as
described in [QUIC-DATAGRAM]. QUIC allows multiple QUIC frames to be
carried within a single QUIC packet, so multiple RTP/RTCP packets for
one (or more) RTP sessions may therefore be carried in a single QUIC
packet, subject to the network path MTU. If multiple RTP packets are
to be carried within a single QUIC packet, then all but the final
"DATAGRAM" frame must specify the length of the datagram, since the
RTP packet header does not provide its own length field.
[QUIC-DATAGRAM] specifies that if a "DATAGRAM" frame is received
without a Length field, then this "DATAGRAM" frame extends to the end
of the QUIC packet.

4.1. QRT Flow Identifier

[RFC3550] specifies that RTP sessions are distinguished by pairs of
transport addresses. However, since QUIC allows for connections to
migrate between transport address associations, and because we wish
to multiplex multiple RTP session flows over a single QRT session,
this profile of RTP amends this statement and instead introduces a
flow identifier to distinguish between RTP sessions. The QRT Flow
Identifier is a 62-bit unsigned integer between 0 and 2^62 - 1.

This specification does not mandate a means by which QRT Flow
Identifiers are allocated for use within QRT sessions. An example
mapping for this is discussed in Section 6 below. Implementations
SHOULD allocate flow identifiers that make the most efficient use of
the variable length integer packing mechanism, by not using flow
identifiers greater than can be expressed in the smallest variable
length integer field until all available flow identifiers have been
used.

The flow of packets belonging to an RTP session is identified using
an RTP Session Flow Identifier header carried in the "DATAGRAM" frame
payload before each RTP/RTCP packet. This flow identifier is encoded
as a variable-length integer, as defined in [QUIC-TRANSPORT].

```
QRT Datagram Payload {
  QRT Flow Identifier (i),
  RTP/RTCP Packet (..)
}
```

                    Figure 1: QRT Datagram Payload

Similar to QUIC stream IDs, the least significant bit (0x1) of the
QRT Flow Identifier distinguishes between an RTP and an RTCP packet
flow. "DATAGRAM" frames which carry RTP packet flows set this bit to
0, and "DATAGRAM" frames which carry RTCP packet flows set this bit
to 1. As a consequence, RTP packet flows have even numbered QRT Flow

Identifiers, and RTCP packet flows have odd-numbered QRT Flow
Identifiers. Carriage of RTCP packets is discussed further in
Section 4.2.

```
+======================+====================================+
| Least significant bit | Flow identifier category          |
+======================+====================================+
| 0x0                   | RTP packet flow for an RTP session  |
+----------------------+------------------------------------+
| 0x1                   | RTCP packet flow for an RTP session |
+----------------------+------------------------------------+
```

Table 1: RTP session flow identifer categories

Author's Note: The author welcomes comments on whether a state
model of RTP session flows would be beneficial. Currently, once
an RTP session has been used by an endpoint, it is then considered
an extant RTP session and implementations would have to keep any
resources allocated to that RTP session until the QRT session is
complete. In addition, how should endpoints react to receiving
packets for unknown QRT flow identifiers?

4.2. RTCP Mapping

An RTP session may have RTCP packet flows associated with it. These
flows are carried with different QRT Flow Identifiers, as described
in Section 4.1. The QRT Flow Identifier of the RTCP packet flow is
always the value of the RTP packet flow QRT Flow Identifier + 1. For
example, for an RTP packet flow using flow identifier 18, the RTCP
packet flow would use flow identifier 19.

Since RTCP packets contain a length field in their header,
implementations MAY combine several RTCP packets pertaining to the
same RTP session into a single "DATAGRAM" frame. Alternatively,
implementations MAY choose to carry these RTCP packets each in their
own "DATAGRAM" frame.

4.2.1. Restricted RTCP Packet Types

Author's Note: I have specifically avoided calling this section
"Prohibited RTCP packet types" for the time being, so as to not
unnecessarily exclude the carriage of these packet types for the
purposes of experimentation. Similarly, most statements below use
SHOULD NOT instead of MUST NOT. The author welcomes comments on
whether the document should prohibit the sending of some or all of
these packet types.

In order to reduce duplication, the following RTCP packet types
SHOULD NOT be sent in a QRT session:

* The "Generic NACK" packet. [RFC4585] states that Generic NACK
   feedback SHOULD NOT be used if the underlying transport protocol
   is capable of providing similar feedback information to the
   sender. Since all "DATAGRAM" frames are ACK-eliciting, QUIC
   already fulfils this requirement.

* The "Loss RLE" Extended Report (XR) packet defined in [RFC3611]
   contains information that should already be known to both ends of
   the QUIC connection by means of the loss detection mechanism
   specified in [QUIC-RECOVERY].

* The "Port Mapping" packet type defined in [RFC6284] is used to
   negotiate UDP port pairs for the carriage of RTP and RTCP packets
   to peers. This does not apply in a QRT session, because the QUIC
   endpoints manage the UDP port association(s) for the QUIC
   connection as a whole.

5. Loss Recovery and Retransmission

   Author's Note: Do we want to mandate (make a MUST) doing
   session-multiplexing instead of SSRC-multiplexing for RTP
   retransmission?

[RFC4588] specifies two schemes to support retransmission in the case
of RTP packet loss. Since QRT natively supports RTP session
multiplexing on a single QUIC connection, endpoints choosing to
implement retransmission SHOULD do so using the session-multiplexing
scheme.

The selection of a new QRT Flow Identifier to use for the
retransmission RTP session is implementation-specific. Section 6.1
specifies how the mapping between original and retransmission RTP
sessions is expressed using the Session Description Protocol (SDP).

6. Using the Session Description Protocol to Advertise QRT Sessions

   [RFC4566] describes a format for advertising multimedia sessions,
   which is used by protocols such as [RFC3261].

   This specification introduces a new SDP value attribute ""qrtflow""
   as a means of assigning QRT Flow Identifiers to RTP and RTCP packet
   flows. Its formatting in SDP is described by the following ABNF
   [RFC5234]:

```
qrtflow-attribute = "a=qrtflow:" qrt-flow-id
qrt-flow-id       = 1*DIGIT ; unsigned 62-bit integer
```

Per Section 4.1 the value of the "qrt-flow-id" is required to be an
even number. (The odd-numbered RTCP flow associated with the RTP
session is not explicitly signalled in the SDP object.)

The example in Figure 2 below shows a hypothetical QRT server
advertising an endpoint to use for live contribution. It instructs a
prospective client to send a VC2-encoded video stream and a Vorbis-
encoded audio stream on two separate RTP sessions. In addition, it
uses the SDP grouping framework described in [RFC5888] to ensure lip
synchronisation between both of those RTP sessions.

```
v=0
o=gfreeman 1594130940 1594135167 IN IP6 qrt.example.org
s=Live Event Contribution
c=IN IP6 2001:db8::7361:6d68
t=1594130980 1594388466
a=group:LS 1 2
m=video 443 RTP/QRT 96
a=qrtflow:0
a=rtpmap:96 vc2
a=mid:1
a=sendonly
m=audio 443 RTP/QRT 97
a=qrtflow:2
a=rtpmap:97 vorbis
a=mid:2
a=sendonly
```

                Figure 2: SDP object describing a QRT session

Since the value of a QRT Flow Identifier for an associated RTCP flow
is specified in Section 4.2, SDP advertisements containing the
"a=qrtflow:" attribute MUST NOT contain an instance of the "a=rtcp:"
attribute as defined in [RFC3605].

6.1. Using the Session Description Protocol to Advertise QRT Sessions using RTP Retransmission

The example in Figure 3 below shows a hypothetical QRT session
advertisement for a bidirectional RTP session carrying an MPEG-2
Transport Stream in each direction on QRT Flow Identifier 0, and a
corresponding pair of retransmission flows on QRT Flow Identifier 2.

```
v=0
o=gfreeman 1594130940 1594135167 IN IP6 qrt.example.org
s=Live Event Contribution
c=IN IP6 2001:db8::4242:4351:5254
t=1594130980 1594388466
m=video 443 RTP/QRT 33
a=qrtflow:0
m=video 443 RTP/QRT 96
a=rtpmap:97 rtx/90000
a=fmtp:96 apt=33;rtx-time=4000
a=qrtflow:2
```

Figure 3: SDP object describing a QRT session with RTP retransmission

7. Exposing Round-Trip Time to RTP applications

Section 5 of [QUIC-RECOVERY] specifies a mechanism for QUIC endpoints to estimate the rount-trip time (RTT) of a connection. QRT implementations SHOULD expose the values of "min_rtt", "smoothed_rtt" and "rttvar" for each network path to the RTP layer, and they MAY use these values either alone or in combination with RTCP messages to discern the round-trip time of the QRT session.

   Author's Note: The author welcomes comments on how appropriate these QUIC RTT measurements are to the RTP layer.

8. Protocol Identifier

The QRT protocol specified in this document is identified by the Application-Layer Protocol Negotiation (ALPN) [RFC7301] identifier "qrt".

8.1. Draft Version Identification

   RFC Editor's Note: Please remove this section prior to publication of a final version of this document.

Only implementations of the final, published RFC can identify themselves as "qrt". Until such an RFC exists, implementations MUST NOT identify themselves using this string. Implementations of draft versions of the protocol MUST add the string "-h" and the corresponding draft number to the identifier. For example, draft-hurst-quic-rtp-tunnelling-00 is identified using the string "qrt-h00".

Non-compatible experiments that are based on these draft versions MUST append the string "-" and an experiment name to the identifier. For example, an experimental implementation based on draft-hurst-

quic-rtp-tunnelling-00 which uses extension features not registered
with the appropriate IANA registry might identify itself as "qrt-h00-
extension-foo". Note that any label MUST conform to the "token"
syntax defined in Section 5.7.2 of [HTTP-SEMANTICS]. Experimenters
are encouraged to coordinate their experiments.

9. Security Considerations

Implementations of the protocol defined in this specification are
subject to the security considerations discussed in [QUIC-TRANSPORT]
and [QUIC-TLS].

10. IANA Considerations

10.1. Registration of Protocol Identification String

This document creates a new registration for the identification of
the QUIC RTP Tunnelling protocol in the "Application-Layer Protocol
Negotiation (ALPN) Protocol IDs" registry established by [RFC7301].

The "qrt" string identifies RTP sessions multiplexed and carried over
a QUIC transport layer:

Protocol: QUIC RTP Tunnelling

Identification Sequence: 0x71 0x72 0x74 ("qrt")

Specification: This document, Section 8

10.2. Registration of SDP Protocol Identifier

This document creates a new registration for the SDP Protocol
Identifier ("proto") "RTP/QRT" in the SDP Protocol Identifiers
("proto") registry established by [RFC4566].

The "RTP/QRT" string identifies a profile of RTP where sessions are
multiplexed and carried over a QUIC transport layer:

SDP Protocol Name: RTP/QRT

Reference: This document, Section 6

10.3. Registration of SDP Attribute Field

This document creates a new registration for the SDP Attribute Field
("att-field") "qrtflow" in the SDP Attribute Field registry
established by [RFC4566].

   SDP Attribute Field: "qrtflow"

   Reference: This document, Section 6

11. References

11.1. Normative References

   [HTTP-SEMANTICS]
           Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke,
           Ed., "HTTP Semantics", Work in Progress, Internet-Draft,
           draft-ietf-httpbis-semantics-12,
           <https://tools.ietf.org/html/draft-ietf-httpbis-semantics-
           12>.

   [QUIC-DATAGRAM]
           Pauly, T., Ed., Kinnear, E., Ed., and D. Schinazi, Ed.,
           "An Unreliable Datagram Extension to QUIC", Work in
           Progress, Internet-Draft, draft-ietf-quic-datagram-01,
           <https://tools.ietf.org/html/draft-ietf-quic-datagram-01>.

   [QUIC-RECOVERY]
           Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection
           and Congestion Control", Work in Progress, Internet-Draft,
           draft-ietf-quic-recovery-32,
           <https://tools.ietf.org/html/draft-ietf-quic-recovery-32>.

   [QUIC-TRANSPORT]
           Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based
           Multiplexed and Secure Transport", Work in Progress,
           Internet-Draft, draft-ietf-quic-transport-32,
           <https://tools.ietf.org/html/draft-ietf-quic-transport-
           32>.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119,
           DOI 10.17487/RFC2119, March 1997,
           <https://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, <https://www.rfc-editor.org/info/rfc3550>.

   [RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute
           in Session Description Protocol (SDP)", RFC 3605,
           DOI 10.17487/RFC3605, October 2003,
           <https://www.rfc-editor.org/info/rfc3605>.

   [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
              Description Protocol", RFC 4566, DOI 10.17487/RFC4566,
              July 2006, <https://www.rfc-editor.org/info/rfc4566>.

   [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R.
              Hakenberg, "RTP Retransmission Payload Format", RFC 4588,
              DOI 10.17487/RFC4588, July 2006,
              <https://www.rfc-editor.org/info/rfc4588>.

   [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax
              Specifications: ABNF", STD 68, RFC 5234,
              DOI 10.17487/RFC5234, January 2008,
              <https://www.rfc-editor.org/info/rfc5234>.

   [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description
              Protocol (SDP) Grouping Framework", RFC 5888,
              DOI 10.17487/RFC5888, June 2010,
              <https://www.rfc-editor.org/info/rfc5888>.

   [RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan,
              "Transport Layer Security (TLS) Application-Layer Protocol
              Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301,
              July 2014, <https://www.rfc-editor.org/info/rfc7301>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

11.2. Informative References

   [QUIC-TLS] Thomson, M., Ed. and S. Turner, Ed., "Using Transport
              Layer Security (TLS) to Secure QUIC", Work in Progress,
              Internet-Draft, draft-ietf-quic-tls-32,
              <https://tools.ietf.org/html/draft-ietf-quic-tls-32>.

   [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
              A., Peterson, J., Sparks, R., Handley, M., and E.
              Schooler, "SIP: Session Initiation Protocol", RFC 3261,
              DOI 10.17487/RFC3261, June 2002,
              <https://www.rfc-editor.org/info/rfc3261>.

   [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed.,
              "RTP Control Protocol Extended Reports (RTCP XR)",
              RFC 3611, DOI 10.17487/RFC3611, November 2003,
              <https://www.rfc-editor.org/info/rfc3611>.

   [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey,
              "Extended RTP Profile for Real-time Transport Control
              Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585,
              DOI 10.17487/RFC4585, July 2006,
              <https://www.rfc-editor.org/info/rfc4585>.

   [RFC6284] Begen, A., Wing, D., and T. Van Caenegem, "Port Mapping
              between Unicast and Multicast RTP Sessions", RFC 6284,
              DOI 10.17487/RFC6284, June 2011,
              <https://www.rfc-editor.org/info/rfc6284>.

Acknowledgments

Author's Address

   Sam Hurst
   BBC Research & Development

   Email: sam.hurst@bbc.co.uk