QUIC Working Group
Internet-Draft
Intended status: Experimental
Expires: 28 May 2026

W. Gage
Unaffiliated

24 November 2025

QUIC Over Reliable Transport
draft-gage-quic-qort-01

Abstract

This document defines QUIC operations when using an underlying reliable transport that, in contrast to UDP, can provide lossless in-order delivery of QUIC packets. The reliable transport may, for example, be TCP or SCTP or a reliable link such as that provided by the 5G radio link protocol.

Table of Contents

1. Introduction

The initial design for QUIC, as defined in [RFC9000], provides for the carriage of QUIC packets over UDP. Since UDP is an unreliable transport, [RFC9000] includes mechanisms for recovering lost packets and for ensuring in-order delivery of byte streams to upper-layer applications. And since UDP is a raw datagram transport, [RFC9000] also includes mechanisms for congestion control, flow control, integrity protection and privacy protection.

In some deployments, an upper-layer application may wish to exploit QUIC capabilities such as light-weight stream management or connection migration in an environment that includes a reliable underlying transport.

When QUIC packets are conveyed over a reliable transport such as [TCP] or [SCTP] or over a reliable link such as that provided by the 5G radio link protocol [NR-RLP], some aspects of [RFC9000] may be redundant or undesirable due to added overhead, to added latencies or to negative interactions with mechanisms of the underlying reliable transport.

This document defines _QUIC over a reliable transport (QORT)_, a set of procedures for conveying QUIC packets over a reliable transport where one or more of the reliability and protection mechanisms of [RFC9000] are replaced by those of the reliable transport.

1.1. Conventions

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.

1.2. Terminology

This document uses the following terminology:

QUIC: the protocol defined by the streams, connections, packets and frames specified in [RFC9000]. Within the context of this document, the term "QUIC" does not include the cryptographic protection described in [RFC9001] nor the loss detection and congestion control procedures of [RFC9002].

QORT: QUIC over a reliable transport (as defined in this document).

RPT: reliable and protected transport offering both cryptographic

protection and error-free, in-order application data delivery (e.g. TLS/TCP, TLS/SCTP).

RT: reliable transport offering error-free, in-order application data delivery but not offering cryptographic protection (e.g. SCTP).

session: a collection of QUIC connections and paths used to exchange QUIC packets between two endpoints.

1.3. Notation

This document uses the field notation defined in [RFC9000] and quoted below:

> Individual fields use the following notational conventions, with all lengths in bits:
>
> x (A): Indicates that x is A bits long
>
> x (i): Indicates that x holds an integer value using the variable length encoding described in [RFC9000], Section 16
>
> x (A..B): Indicates that x can be any length from A to B; A can be omitted to indicate a minimum of zero bits, and B can be omitted to indicate no set upper limit; values in this format always end on a byte boundary
>
> x (L) = C: Indicates that x has a fixed value of C; the length of x is described by L, which can use any of the length forms above
>
> x (L) = C..D: Indicates that x has a value in the range from C to D, inclusive, with the length described by L, as above
>
> x (L)...: Indicates that x is repeated zero or more times and that each instance has a length of L

2. Overview

The differences between the protocol stack of QORT and the protocol stack of [RFC9000] are illustrated in Figure 1:

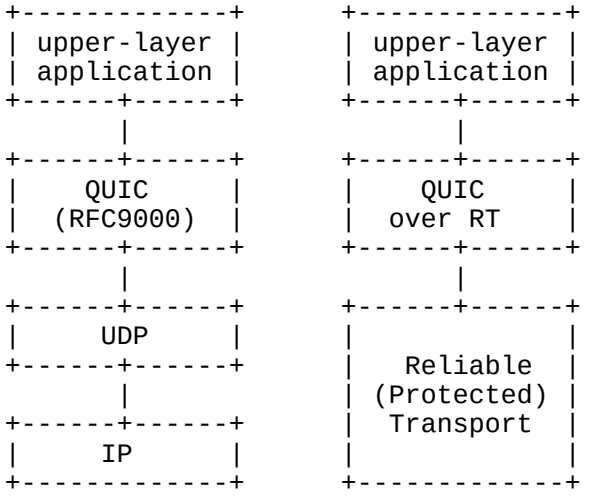

Figure 1: RFC9000 and QORT Protocol Stacks

[RFC9000] uses UDP/IP to convey QUIC packets between the endpoints of a QUIC session while QORT is designed to use a reliable transport between the endpoints. The reliable transport may include IP and may even include UDP (or it may not) so long as the requirements of Section 3 are met.

Depending on the characteristics of the reliable transport, one of two modes of QORT may be used by the upper-layer application:

* CRYPTO-QORT includes the cryptographic mechanisms described in Section 7 of [RFC9000] and is designed to operate over a reliable transport that does not provide integrity and privacy protection (i.e. an RT). For example, this mode may be used when operating QORT over SCTP [SCTP].

* BASIC-QORT is designed to operate over a reliable transport that also provides integrity and privacy protection (i.e. an RPT). For example, this mode may be used when operating QORT over TLS [TLS] (over TCP or SCTP) or over a 5G radio link [NR-RLP].

The mode selected by the client is indicated by the Version field in the long packet header of the QUIC INITIAL packet sent from the client to a server. Different QORT modes are distinguished by different version values (Section 4.1).

If CRYPTO-QORT mode is selected by the client, the QUIC INITIAL packet transmitted by an endpoint includes a QUIC CRYPTO frame for the initial cryptographic handshake and the transport parameters selected by the endpoint (Section 4.2).

If BASIC-QORT mode is selected by the client, the QUIC INITIAL packet transmitted by an endpoint includes a new QORT_CONFIGURATION frame that contains the transport parameters selected by the endpoint (Section 4.3).

Once the initial handshake has been completed, the client and server may exchange QUIC packets containing any of the [RFC9000] QUIC frames that are not subject to restrictions imposed by the selected QORT mode (Section 5).

| The differences between QORT and the proposal in [QMUX-DRAFT] are discussed in Appendix C.

3. Reliable Transport Services

QORT replaces the UDP/IP transport of [RFC9000] with a reliable transport that ensures the in-order exchange of error-free messages and may provide cryptographic protection.

3.1. Delimiting Messages

Some reliable transports provide message delimitation (aka. "framing"), allowing a sequence of bytes conveyed by the reliable transport to be identified and treated as a self-contained message. For example, in [SCTP] message delimitation is provided by the SCTP packet protocol and in [NR-RLP] message delimitation is provided by the packet data convergence protocol (PDCP). When message delimitation is provided by the reliable transport, each transmitted QUIC packet MUST be treated as a separate message.

Some reliable transports (e.g. TCP) only convey byte streams and do not (natively) provide message delimitation. When message delimitation is not provided by the reliable transport, each transmitted QUIC packet MUST be terminated by a QORT end-of-packet frame (Section 7).

3.2. In-Order Delivery

A reliable transport MUST provide in-order delivery of messages, ensuring that messages are presented to a receiving QORT entity in the same order that they sent by a transmitting QORT entity.

3.3. Guaranteed Delivery

A reliable transport MUST provide guaranteed delivery, ensuring that any bytes of a message lost in transmission between two endpoints are recovered, normally through retransmission. The reliable transport MUST buffer any bytes received out-of-order and MUST reorder any recovered bytes to ensure in-order delivery of the messages.

3.4. Congestion Control

A reliable transport MUST provide congestion control that is applied to the aggregate of packets transmitted by an endpoint.

3.5. Cryptographic Protection

A reliable transport MAY also provide integrity protection (e.g. a cryptographic hash of packet contents) and privacy protection (i.e. encryption). The ability of a reliable transport to provide cryptographic protection may influence the QORT mode of operation selected by a client (Section 4).

4. Modes of Operation

A client may choose one of two modes of operations for QUIC over a reliable transport -- CRYPTO-QORT mode or BASIC-QORT mode.

If a reliable transport provides integrity and privacy protection, a client SHOULD select BASIC-QORT mode, otherwise the client SHOULD select CRYPTO-QORT mode. The choice of QORT mode is, however, ultimately based on the requirements of the upper-layer application (see Appendix B).

4.1. Mode Selection

The QORT mode of operation selected by the client is indicated by the Version field in the long packet header of the QUIC INITIAL packet sent from the client to a server.

If the server is able to support the QORT mode selected by the client, the server MUST respond with a QUIC INITIAL packet where the Version field in the long packet header is the same as the version indicated by the client.

If the server is not able to support the QORT mode selected by the client, the server MUST close the connection with an Error_qortNotSupported (Section 10).

Because the reliable transport has been established between the client and server before the start of QORT operations, there is no mode of operation negotiation -- the server either accepts the mode selected by the client or it closes the connection. Similarly, QORT does not support negotiation of the QUIC protocol version using a VERSION NEGOTIATION packet.

In this document, the mode of operation Version dictates compatibility with Version 0x00000001 of QUIC; compatibility with other versions of QUIC are beyond the scope of this document.

| Note that the selected mode must be indicated in the Version
| field of all long QUIC packet headers.

4.2. CRYPTO-QORT mode of operation

The CRYPTO-QORT mode of operation is initiated using a cryptographic handshake sequence similar to that described in Section 7 of [RFC9000].

The CRYPTO-QORT mode of operation differs from [RFC9000] in the following ways:

* the Version field in a QUIC long header packet MUST be set to Version01_qortCrypto (Section 13.1).

* deprecated QUIC frames listed in Section 5 and in Section 5.1 MUST NOT be included in any QUIC packets.

In the CRYPTO-QORT mode of operation, a receiving endpoint must be able to detect the end of a QUIC packet before attempting to decrypt the packet. Therefore CRYPTO-QORT mode of operation can only be used with a reliable transport that provides message delimitation (Section 3.1).

An exemplary CRYPTO-QORT handshake sequence is illustrated in Figure 2.

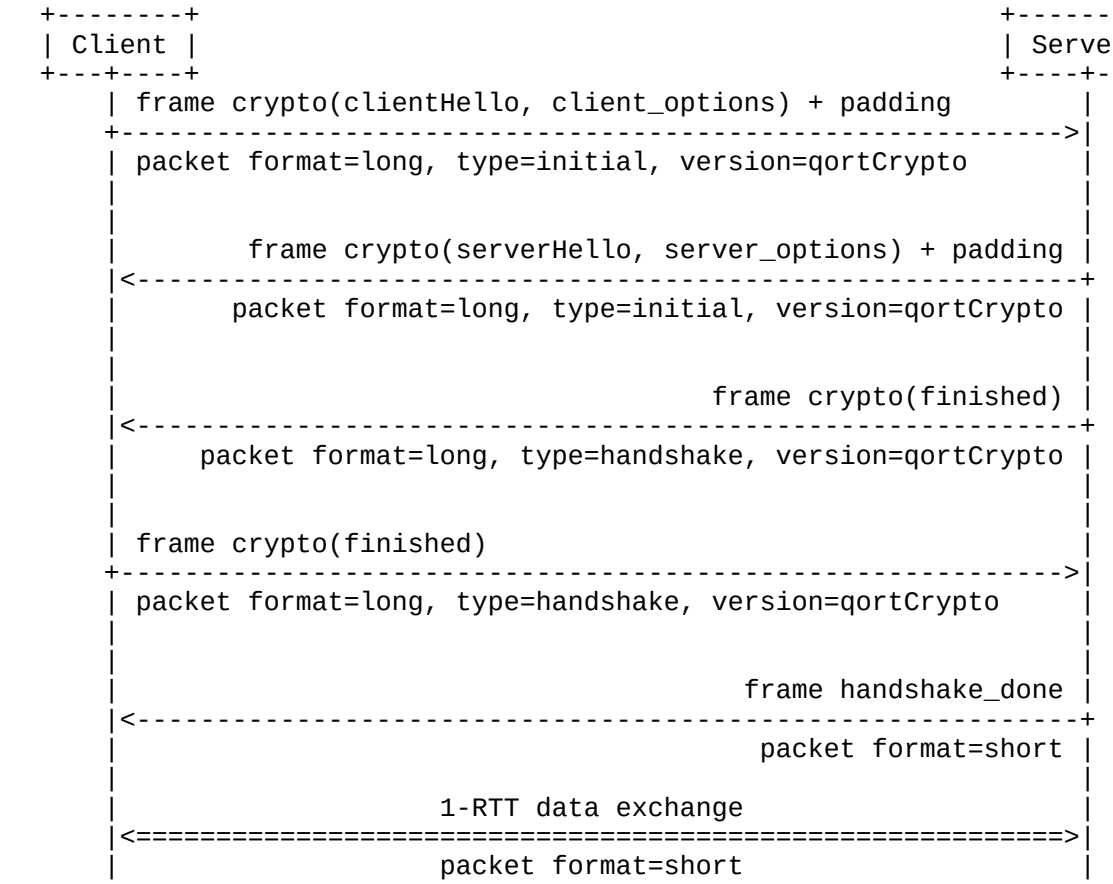

Figure 2: CRYPTO-QORT Handshake

4.3. BASIC-QORT mode of operation

The BASIC-QORT mode of operation assumes that cryptographic protection (if needed) has been established by the reliable transport before the QORT handshake between client and server begins. As a consequence, the BASIC-QORT mode of operation involves an initial exchange of transport parameters between the endpoints but does not include a cryptographic handshake.

The BASIC-QORT mode of operation differs from [RFC9000] in the following ways:

* the Version field in a QUIC long header packet MUST be set to Version01_qortBasic (Section 13.1).

* the QUIC INITIAL packets exchanged between endpoints MUST include a QORT_CONFIGURATION frame (Section 8.1).

* QUIC HANDSHAKE packets MUST NOT be exchanged between endpoints.

* deprecated QUIC frames listed in Section 5 and in Section 5.2 MUST NOT be included in any QUIC packets.

The BASIC-QORT handshake sequence is illustrated in Figure 3.

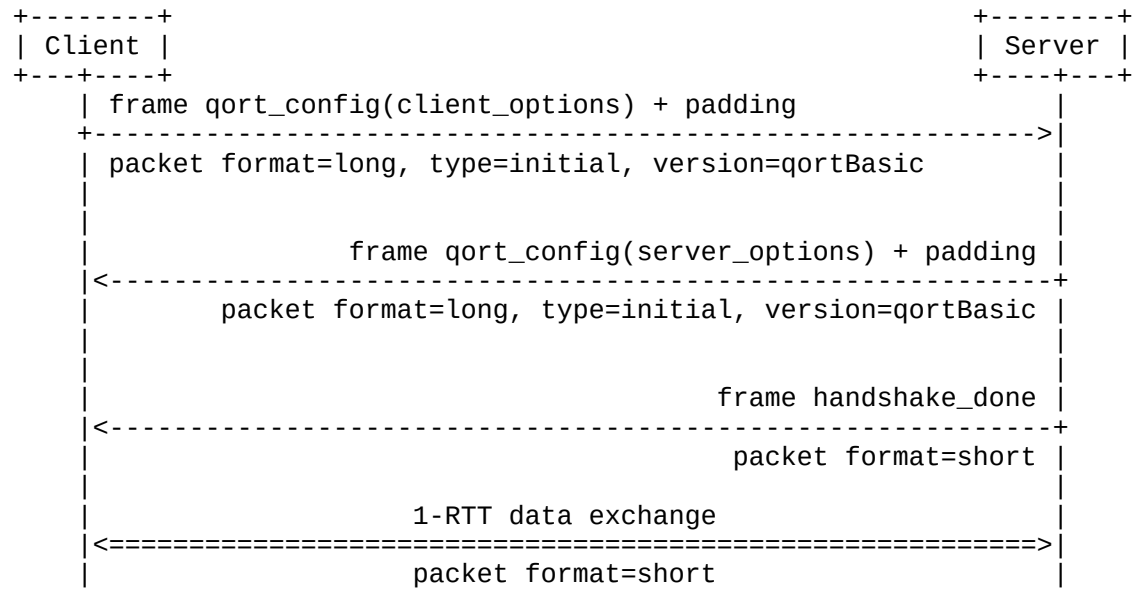

Figure 3: BASIC-QORT Handshake

5. Deprecated QUIC Frames

When either QORT mode is enabled, the following QUIC frame types defined by [RFC9000] MUST NOT be transmitted by either endpoint in a session:

* ACK_ECN frames are not needed because handling of ECN signals is a function of the reliable transport.

* MAX_DATA frames are not needed because restrictions (if any) on the amount of data to be exchanged must be negotiated and enforced by the reliable transport.

* NEW_TOKEN frames are not needed because priori validation of

the client address (if supported) is a function of the reliable transport.

If an endpoint receives one of these frame types, it MUST close the connection with an error of type Error_qortInvalidFrame (Section 13).

5.1. Deprecated QUIC Frames in CRYPTO-QORT Mode

When CRYPTO-QORT mode is selected, the following additional QUIC frame types defined by [RFC9000] MUST NOT be transmitted by either endpoint in a connection:

* (TBD)

If an endpoint receives one of these frame types when operating in CRYPTO-QORT mode, it MUST close the connection with an error of type Error_qortInvalidFrame (Section 13).

5.2. Deprecated QUIC Frames in BASIC-QORT Mode

When BASIC-QORT mode is selected, the following additional QUIC frame types defined by [RFC9000] MUST NOT be transmitted by either endpoint in a connection:

* CRYPTO frames are not needed because cryptographic protection is provided by the reliable transport.

If an endpoint receives one of these frame types when operating in BASIC-QORT mode, it MUST close the connection with an error of type Error_qortInvalidFrame (Section 13).

6. Use of QUIC ACK Frames

QUIC ACK frames are not needed for error recovery nor are they needed for congestion control because the reliable transport provides both of these services (Section 3).

However, in some instances, an ACK frame may be used as an indication that information previously transmitted in a packet by one QUIC endpoint has been processed by the peer endpoint in a QUIC connection. For this reason, QORT retains the use and processing of ACK frames (but not of ACK_ECN frames, Section 5).

7. Delimiting QUIC Packets

When a reliable transport does not provide message delimitation (Section 3.1), a transmitting QORT endpoint MUST identify the end of a QUIC packet by including a QORT_ENDOFPACKET frame (Section 8.2) as the last QUIC frame in the packet. See example in Appendix A.1.

When a reliable transport does provide message delimitation, QORT_ENDOFPACKET frames MAY be used to identify the end of a QUIC packet even though this adds unnecessary overhead. It is RECOMMENDED that QORT_ENDOFPACKET frames not be used when the reliable transport provides message delimitation.

When a receiving endpoint detects a QORT_ENDOFPACKET frame, it MUST stop processing the current QUIC packet. Any bytes remaining in the receive buffer of the endpoint MUST be treated as the start of a new QUIC packet.

8. New QORT Frame Types

QORT procedures utilise one new QUIC frame type -- QORT_ENDOFPACKET -- when operating over a reliable transport that does not provide message delimitation (Section 7).

In addition, QORT procedures utilise one additional new QUIC frame type -- QORT_CONFIGURATION -- when operating in BASIC-QORT mode. The QORT_CONFIGURATION frame type MUST NOT be used when operating in CRYPTO-QORT mode.

The QORT_CONFIGURATION frame type is ack-eliciting; the QORT_ENDOFPACKET frame type is not ack-eliciting.

Both QORT_CONFIGURATION and QORT_ENDOFPACKET are non-probing frames.

8.1. QORT_CONFIGURATION frame

A QORT_CONFIGURATION frame contains transport parameters defined by the sending endpoint. As such, a QORT_CONFIGURATION frame is a direct replacement for the TLS quic_transport_parameters extension described in Section 8.2 of [RFC9001].

When a client initiates the BASIC-QORT mode of operation, the client MUST include a QORT_CONFIGURATION frame in its QUIC INITIAL packet. See example in Appendix A.2.

When a server agrees to use the BASIC-QORT mode of operation, the server MUST include a QORT_CONFIGURATION frame in its responding QUIC INITIAL packet.

A QORT_CONFIGURATION frame (Figure 4) includes the following fields:

```
QORT_CONFIGURATION Frame {
  Type (i) = Type_qortConfiguration,
  Transport_Parameter_List_Length (i)
  Transport_Parameter_List (..) ...,
}
```

Figure 4: QORT_CONFIGURATION Frame Fields

* Type is set to Type_qortConfiguration (Section 13.3).

* Transport_Parameter_List_Length is set to the length of the Transport_Parameter_List, in number of octets.

* Transport_Parameter_List is a list of transport parameters.

As defined in Section 18 of [RFC9000], each transport parameter in the list of transport parameters is encoded as an (identifier, length, value) tuple -- i.e.

```
Transport Parameter {
  Transport Parameter ID (i),
  Transport Parameter Length (i),
  Transport Parameter Value (..),
}
```

Figure 5: RFC9000 Transport Parameter Fields

The set of valid QORC transport parameters is described in Section 9.

8.2. QORT_ENDOFPACKET frame

A QORT_ENDOFPACKET frame is used to identify the end of a QUIC packet (Section 7). A QORT_ENDOFPACKET frame (Figure 6) includes the following fields:

```
QORT_ENDOFPACKET Frame {
  Type (i) = Type_qortEndOfPacket,
}
```

Figure 6: QORT_ENDOFPACKET Frame Fields

* Type is set to Type_qortEndOfPacket (Section 13.3).

9. Transport Parameters

The different modes of QORT operation use different mechanisms for the exchange of transport parameters by the endpoints:

* in CRYPTO-QORT mode of operation, transport parameters are exchanged as a TLS extension in a CRYPTO frame conveyed in an INITIAL packet (Section 4.2);

* in BASIC-QORT mode of operation, transport parameters are exchanged as in a QORT_CONFIGRATION frame conveyed in an INITIAL packet (Section 4.3).

9.1. RFC9000 Transport Parameters

Not all transport parameters defined in [RFC9000] are valid for use with QUIC over a reliable transport. When either QORT mode is enabled, a QUIC transport parameter with an ’x’ in the "not allowed?" column of Table 1 MUST NOT be transmitted by either endpoint.

transport parameter	allowed?	not allowed?
original_destination_connection_id	x	
max_idle_timeout	x	
stateless_reset_token	x	
max_udp_payload_size	x	
initial_max_data		x
initial_max_stream_data_bidi_local	x	
initial_max_stream_data_bidi_remote	x	
initial_max_stream_data_uni	x	
initial_max_streams_bidi	x	
initial_max_streams_uni	x	
ack_delay_exponent	x	
max_ack_delay	x	
disable_active_migration	x	
preferred_address		x
active_connection_id_limit	x	
initial_source_connection_id	x	
retry_source_connection_id		x

Table 1: Allowed/Not-allowed QUIC Transport Parameters

If an endpoint receives one of the "not allowed?" transport parameters, it MUST close the connection with an error of type Error_qortInvalidParameter (Section 13).

10. QORT Connection Errors

This document extends the QUIC Transport Error Codes of [RFC9000], Section 22.5 with the following values (Section 13.2):

* Error_qortInvalidFrame indicates that an unknown frame type was received or that the QUIC frame type is not allowed in the QORT mode of operation (Section 5).

* Error_qortInvalidParameter indicates that a received transport parameter was invalid (e.g. was badly formatted) or is not allowed in a QORT mode of operation (Section 9) .

* Error_qortNotSupported indicates that the QORT mode of operation selected by a client is not supported by the server.

* Error_qortProtocolViolation indicates an error with protocol compliance that is not covered by a more specific error code -- e.g. an endpoint received a QORT_CONFIGURATION frame when BASIC-QORT mode is not selected.

QORT connection errors MUST be processed according to [RFC9000], Section 11.1.

11. QUIC extensions

In general, any extension to Version 0x00000001 of QUIC that does not rely on the deprecated QUIC frames of Section 5 may be used with QORT (e.g. [DATAGRAM], [PMQUIC]). However this must be be verified by the extension designers and is beyond the scope of this document.

12. Security Considerations

QORT does not change the operating principles of [RFC9000] and, as such, is subject to the same security considerations as [RFC9000], Section 21.

Specific security considerations associated with the use of QORT procedures include:

* packet protection. When using BASIC-QORT mode. the reliable transport is also expected to provide cryptographic protection of its payload (i.e. QUIC packets). As a consequence, all bits of a QUIC packet are cryptographically protected. Therefore, in BASIC-QORT mode, it should not be possible to mount an attack that relies on unfettered visibility of bits in a QUIC packet header (Section 3 of [RFC9312]).

13. IANA Considerations

If approved, the following _provisional_ entries will be added to the IANA QUIC Protocol Registry [IANA].

> | The values in this section of the Internet Draft are
> | preliminary, for testing purposes only.

13.1. QUIC version numbers

This document defines two new (preliminary) QUIC version numbers that are bound to Version 0x00000001 of QUIC (Section 4):

* Version01_qortBasic (0x54e51e9e)

* Version01_qortCrypto (0x147214bd)

13.2. QORT transport error codes

This document extends the QUIC Transport Error Codes of [RFC9000], Section 22.5 with the following (preliminary) values:

* Error_qortInvalidFrame (0x2fc2ed12e1295ab1)

* Error_qortInvalidParameter (0x0b244a578d7dc6c4)

* Error_qortNotSupported (0x341b50c67a845ff4)

* Error_qortProtocolViolation (0x0ed79e3b4c52d445)

13.3. New QUIC frame types

This document defines two new (preliminary) QUIC frame types:

* Type_qortConfiguration (0x27f72376051c5308) -- Section 8.1

* Type_qortEndOfPacket (0x3b5690c1243618cd) -- Section 8.2

14. References

14.1. Normative References

[IANA] Internet Assigned Numbers Authority, "QUIC Protocol Registry", <https://www.iana.org/assignments/quic/>.

[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/rfc/rfc2119>.

[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/rfc/rfc8174>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/rfc/rfc9000>.

[RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/rfc/rfc9001>.

[RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <https://www.rfc-editor.org/rfc/rfc9002>.

14.2. Informative References

[DATAGRAM] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", RFC 9221, DOI 10.17487/RFC9221, March 2022, <https://www.rfc-editor.org/rfc/rfc9221>.

[NR-RLP] 3rd Generation Partnership Project (3GPP), "Technical Specification 38.300; NR and NG-RAN Overall Description; Stage 2", <https://www.3gpp.org/DynaReport/38300.htm>.

[PMQUIC] Gage, B., "QUIC Path Management for Multi-Path Configurations", Work in Progress, Internet-Draft, draft-gage-quic-pathmgmt-04, 3 August 2025, <https://datatracker.ietf.org/doc/html/draft-gage-quic-pathmgmt-04>.

[QMUX-DRAFT] Oku, K., Pardue, L., Iyengar, J., and E. Kinnear, "QMux", Work in Progress, Internet-Draft, draft-opik-quic-qmux-01, 13 November 2025, <https://datatracker.ietf.org/doc/html/draft-opik-quic-qmux-01>.

[RFC9312] Khlewind, M. and B. Trammell, "Manageability of the QUIC Transport Protocol", RFC 9312, DOI 10.17487/RFC9312, September 2022, <https://www.rfc-editor.org/rfc/rfc9312>.

[SCTP] Stewart, R., Txen, M., and K. Nielsen, "Stream Control Transmission Protocol", RFC 9260, DOI 10.17487/RFC9260, June 2022, <https://www.rfc-editor.org/rfc/rfc9260>.

[TCP] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/rfc/rfc9293>.

[TLS] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

Appendix A. QORT Packet Examples

This section contains examples of QUIC packets when using QORT procedures.

A.1. QUIC Packet with QORT_ENDOFPACKET Frame

When message delimitation is not provided by the reliable transport, each transmitted QUIC packet must be terminated by a QORT_ENDOFPACKET frame. This is illustrated in Figure 7.


```
{
  "name": "transport:packet_received",
  "data": {
    "header": {
      "packet_number": 1,
      "packet_type": "1RTT",
      "dcid": "faee8572bad88622",
      "spin_bit": 0
    },
    "frames": [
      {
        "frame_type": "ack",
        "ack_delay": 2.888,
        "acked_ranges": [
          [
            0,
            0
          ]
        ]
      },
      {
        "frame_type": "stream",
        "fin": false,
        "length": 21,
        "offset": 0,
        "stream_id": 3
      },
      {
        "frame_type": "stream",
        "fin": false,
        "length": 4,
        "offset": 0,
        "stream_id": 7
```

```
        },
        {
          "frame_type": "stream",
          "fin": false,
          "length": 1,
          "offset": 0,
          "stream_id": 11
        },
        {
          "frame_type": "rt end_of_packet"
        }
      ],
      "raw": {
        "length": 63
      }
    }
  }
```


Figure 7: Packet with QORT_ENDOFPACKET Frame

A.2. QUIC Initial Packet with QORT_CONFIGURATION Frame

In BASIC-QORT mode of operation, the INITIAL QUIC packet must include a QORT_CONFIGURATION frame. This is illustrated in Figure 8.

```
{
  "name": "transport:packet_sent",
  "data": {
    "header": {
      "packet_number": 0,
      "packet_type": "initial",
      "dcid": "48dc009224099b2c",
      "scid": "18657c1acd3e0c5e"
    },
    "frames": [
      {
        "frame_type": "rt configuration",
        "transport_parameters": {
          "max_idle_timeout": 10000,
          "max_udp_payload_size": 1200,
          "initial_max_data": 1048576,
          "initial_max_stream_data_bidi_local": 1048576,
          "initial_max_stream_data_bidi_remote": 1048576,
          "initial_max_stream_data_uni": 1048576,
          "initial_max_streams_bidi": 128,
          "initial_max_streams_uni": 128,
          "ack_delay_exponent": 3,
          "max_ack_delay": 25,
          "disable_active_migration": false,
          "active_connection_id_limit": 8,
          "initial_source_connection_id": "18657c1acd3e0c5e",
          "max_datagram_frame_size": 65536
        }
      },
      {
        "frame_type": "padding",
        "length": 1089
      },
      {
        "frame_type": "rt end_of_packet"
      }
    ],
    "raw": {
      "length": 1200
    }
  }
}
```


Figure 8: INITIAL Packet with QORT_CONFIGURATION Frame

Appendix B. QORT use with reliable transports

Table 2 indicates how QORT is expected to be used with several reliable transports. The "Mode" indicates the QORT mode of operation and whether a QORT_ENDOFPACKET frame ("eop") is used for packet delimitation.

Mode	TLS/TCP	TLS/SCTP	NR-RLP	TCP	SCTP
BASIC with eop	(1)	(2)	(2)	(3)	(3)
BASIC without eop	(x)	(1)	(1)	(x)	(3)
CRYPTO with eop	(x)	(2,4)	(2,4)	(x)	(2)
CRYPTO without eop	(x)	(4)	(4)	(x)	(1)

Table 2: QORT modes over various transports

(1) recommended mode of operation
(2) possible, but not recommended - redundant end-of-packet frames
(3) possible, but not recommended - no cryptographic protection
(4) possible, but not recommended - redundant cryptographic protection
(x) not possible - message delimitation required

Appendix C. Comparison to [QMUX-DRAFT]

| This section is provided for information only.

[QMUX-DRAFT] proposes to convey QUIC frames over a reliable, bi-directional, byte-oriented stream. The following sections highlight key differences between [QMUX-DRAFT] and QORT.

C.1. Use of QUIC packets

[QMUX-DRAFT] does not use QUIC packets for encapsulating QUIC frames. Instead, [QMUX-DRAFT] sends QUIC frames directly over the reliable transport.

By contrast, QORT retains the use of QUIC packets to support QUIC operations that are dependent on information in the QUIC packet header as described in Section 17 of [RFC9000] -- e.g. packet numbering, coalescence, cryptographic protection, connection identification and migration.

C.2. Connections and connection identifiers

[QMUX-DRAFT] does not include mechanisms for identifying QUIC connections, for managing QUIC connection identifiers, or for path migration. Essentially, all QUIC frames are associated with a zero-length connection identifier.

By contrast, QORT retains the concept of QUIC connections and associated connection identifiers to support connection management, path migration and multipath operations.

C.3. Cryptographic protection

[QMUX-DRAFT] requires an underlying transport that provides cryptographic protection. As a consequence, [QMUX-DRAFT] offers no cryptographic capabilities.

By contrast, QORT offers a CRYPTO-QORT mode of operation that reuses [RFC9000] cryptographic protection mechanisms such as a cryptographic handshake using QUIC CRYPTO frames and key rotation using the key phase bit of a QUIC packet header.

C.4. Stream processing

While [QMUX-DRAFT] reuses QUIC STREAM frames for exchanging application data, Section 4.1 of [QMUX-DRAFT] introduces some restrictions on STREAM frame use.

By contrast, QORT does not change any [RFC9000] procedures related to stream processing.

Author’s Address

Bill Gage
Unaffiliated
Ottawa
Canada
Email: billgage.ietf@gmail.com