QUIC
Internet-Draft
Intended status: Standards Track
Expires: 17 December 2025

M. Seemann
奥一穂 (K. Oku)
Fastly

15 June 2025

QUIC Stream Resets with Partial Delivery
draft-ietf-quic-reliable-stream-reset-07

Abstract

QUIC defines a RESET_STREAM frame to abort sending on a stream. When a sender resets a stream, it also stops retransmitting STREAM frames for this stream in the event of packet loss. On the receiving side, there is no guarantee that any data sent on that stream is delivered.

This document defines a new QUIC frame, the RESET_STREAM_AT frame, that allows resetting a stream, while guaranteeing delivery of stream data up to a certain byte offset.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://quicwg.github.io/reliable-stream-reset/draft-ietf-quic-reliable-stream-reset.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-quic-reliable-stream-reset/.

Discussion of this document takes place on the QUIC Working Group mailing list (mailto:quic@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/quic/. Subscribe at https://www.ietf.org/mailman/listinfo/quic/.

Source for this draft and an issue tracker can be found at https://github.com/quicwg/reliable-stream-reset.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

QUIC version 1 ([RFC9000]) allows streams to be reset. When a stream is reset, the sender doesn't retransmit stream data for the respective stream. On the receiving side, the QUIC stack is free to surface the stream reset to the application immediately, without providing any stream data it has received for that stream.

Some applications running on top of QUIC use bytes at the beginning of the stream to communicate critical information related to that stream. For example, WebTransport ([WEBTRANSPORT]) uses a variable-length encoded integer to associate a stream with a particular WebTransport session.

Since QUIC does not provide guaranteed delivery of steam data for reset streams, it is possible that a receiver is unable to read critical information. In the example above, a reset stream can cause the receiver to fail to associate incoming streams with their respective subcomponent of the application. Therefore, it is desirable that the receiver can rely on the delivery of critical information to applications, even if the QUIC stream is reset before data is read by the application.

Another use case is relaying data from an external data source. When a relay is sending data being read from an external source and encounters an error, it might want to use a stream reset to signal that error, while at the same time guaranteeing that all data received from the source is delivered to the peer.

This document extends QUIC with a variant of stream resets that reliably delivers the beginning of a stream up to a sender-specified offset, communicated using the RESET_STREAM_AT frame. It can be considered a form of range-based partial reliability. As a variant of reset, application protocols continue to treat this stream function as an abrupt termination; see Section 2.4 of [RFC9000].

2. 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.

3. Transport Parameter

Support for receiving RESET_STREAM_AT frames is advertised by sending the reset_stream_at (0x17f7586d2cb571) transport parameter (Section 7.4 of [RFC9000]) with an empty value. An implementation that understands this transport parameter MUST treat the receipt of a non-empty value as a connection error of type TRANSPORT_PARAMETER_ERROR.

When using 0-RTT, both endpoints MUST remember the value of this transport parameter. This allows use of this extension in 0-RTT packets. When the server accepts 0-RTT data, the server MUST NOT disable this extension on the resumed connection.

4. RESET_STREAM_AT Frame

Conceptually, the RESET_STREAM_AT frame is a RESET_STREAM frame with an added Reliable Size field.

```
RESET_STREAM_AT Frame {
  Type (i) = 0x24,
  Stream ID (i),
  Application Protocol Error Code (i),
  Final Size (i),
  Reliable Size (i),
}
```

Figure 1: RESET_STREAM_AT Frame Format

The RESET_STREAM_AT frame contains the following fields:

Stream ID: A variable-length integer encoding of the stream ID of the stream being terminated.

Application Protocol Error Code: A variable-length integer containing the application protocol error code (Section 20.2 of [RFC9000]) that indicates why the stream is being closed.

Final Size: A variable-length integer indicating the final size of the stream by the sender, in units of bytes; see Section 4.5 of [RFC9000].

Reliable Size: A variable-length integer indicating the amount of data that needs to be delivered to the application even though the stream is reset.

If the Reliable Size is larger than the Final Size, the receiver MUST close the connection with a connection error of type FRAME_ENCODING_ERROR.

RESET_STREAM_AT frames are ack-eliciting, and MUST only be sent in the application data packet number space. When lost, they MUST be retransmitted, unless the stream state has transitioned to "Data Recvd" or "Reset Recvd" due to transmission and acknowledgement of other frames (see Section 5.2).

5. Resetting Streams

A sender that wants to reset a stream but also deliver some bytes to the receiver sends a RESET_STREAM_AT frame with the Reliable Size field specifying the amount of data to be delivered.

When using a RESET_STREAM_AT frame, the initiator MUST guarantee reliable delivery of stream data of at least Reliable Size bytes. If STREAM frames containing data up to that byte offset are lost, the initiator MUST retransmit this data, as described in Section 13.3 of [RFC9000]. Data sent beyond that byte offset SHOULD NOT be retransmitted.

As described in Section 3.2 of [RFC9000], a stream reset signal might be suppressed or withheld, and the same applies to a stream reset signal carried in a RESET_STREAM_AT frame. Similarly, the Reliable Size of the RESET_STREAM_AT frame does not prevent a QUIC stack from delivering data beyond the specified offset to the receiving application.

Note that a Reliable Size value of zero is valid. A RESET_STREAM_AT frame with this value is logically equivalent to a RESET_STREAM frame (Section 3.2 of [RFC9000]). When resetting a stream without the intent to deliver any data to the receiver, the sender MAY use either RESET_STREAM or RESET_STREAM_AT with a Reliable Size of zero.

As stated in Section 4.5 of [RFC9000], the final size for a stream cannot change once it is known. If a frame is received indicating a change in the final size for the stream, an endpoint SHOULD respond with an error of type FINAL_SIZE_ERROR.

5.1. Sending RESET_STREAM_AT after FIN

Similar to how it is possible to send a RESET_STREAM frame after a STREAM frame carrying the FIN bit, it is possible to send a RESET_STREAM_AT frame after a STREAM frame carrying the FIN bit.

Due to packet reordering, it is possible for a receiver to receive the RESET_STREAM_AT frame before receiving the STREAM frame carrying the FIN bit.

5.2. Multiple RESET_STREAM_AT / RESET_STREAM frames

The initiator MAY send multiple RESET_STREAM_AT frames for the same stream in order to reduce the Reliable Size. It MAY also send a RESET_STREAM frame, which is equivalent to sending a RESET_STREAM_AT frame with a Reliable Size of 0. When reducing the Reliable Size, the sender MUST retransmit the RESET_STREAM_AT frame carrying the smallest Reliable Size as well as stream data up to that size, until all acknowledgements for the stream data and the RESET_STREAM_AT frame are received.

When sending multiple RESET_STREAM_AT or RESET_STREAM frames for the same stream, the initiator MUST NOT increase the Reliable Size.

When receiving a RESET_STREAM_AT frame with a lower Reliable Size, the receiver only needs to provide data up the lower Reliable Size to the application. It MUST NOT expect the sender to deliver any data beyond that byte offset.

Reordering of packets might lead to a RESET_STREAM_AT frame with a higher Reliable Size being received after a RESET_STREAM_AT frame with a lower Reliable Size. The receiver MUST ignore any RESET_STREAM_AT frame that increases the Reliable Size.

When sending another RESET_STREAM_AT, RESET_STREAM or STREAM frame carrying a FIN bit for the same stream, the initiator MUST NOT change the Application Error Code or the Final Size. If the receiver detects a change in those fields, it MUST close the connection with a connection error of type STREAM_STATE_ERROR.

While multiple RESET_STREAM_AT frames can reduce Reliable Size, some applications might need to ensure that a minimum amount of data is always delivered on a stream. Application protocols can establish rules for streams that ensure that Reliable Size is not reduced below a certain threshold if that is necessary to ensure correct operation of the protocol.

5.3. Stream States

In terms of stream state transitions (Section 3 of [RFC9000]), the effect of a RESET_STREAM_AT frame is equivalent to that of the FIN bit. Both the RESET_STREAM_AT frame and the FIN bit on a STREAM frame serve the same role: signaling the amount of data to be delivered.

On the sending side, when the first RESET_STREAM_AT frame is sent, the sending part of the stream enters the "Data Sent" state. Once the RESET_STREAM_AT frame carrying the smallest Reliable Size and all stream data up to that byte offset have been acknowledged, the sending part of the stream enters the "Data Recvd" state. The transition from "Data Sent" to "Data Recvd" happens immediately if the application resets a stream and all bytes up to the specified Reliable Size have already been sent and acknowledged. Conversely, the transition might take multiple network roundtrips or require additional flow control credit issued by the receiver.

On the receiving side, when a RESET_STREAM_AT frame is received, the receiving part of the stream enters the "Size Known" state. Once all data up to the smallest Reliable Size have been received, it enters the "Data Recvd" state. Similarly to the sending side, transition from "Size Known" to "Data Recvd" might happen immediately or involve issuance of additional flow control credit.

6. Implementation Guidance

In terms of transport machinery, the RESET_STREAM_AT frame is more akin to the FIN bit than to the RESET_STREAM frame (see Section 5.3). By sending a RESET_STREAM_AT frame, the sender commits to delivering all bytes up to the Reliable Size.

To the endpoints, the main differences from closing a stream by using the FIN bit are:

* the offset up to which the sender commits to sending might be smaller than Final Size,

* this offset might get reduced by subsequent RESET_STREAM_AT frames, and

* the closure is accompanied by an error code.

7. Security Considerations

As the RESET_STREAM_AT frame is an extension to the stream machinery defined in QUIC version 1, the security considerations of [RFC9000] apply accordingly. Specifically, given that RESET_STREAM_AT frames indicate the offset up to which data is reliably transmitted, endpoints SHOULD remain vigilant against resource commitment and exhaustion attacks even after sending or receiving RESET_STREAM_AT frames, until the stream reaches the terminal state.

8. IANA Considerations

8.1. QUIC Transport Parameter

This document registers the reset_stream_at transport parameter in the "QUIC Transport Parameters" registry established in Section 22.3 of [RFC9000]. The following fields are registered:

Value: 0x17f7586d2cb571

Parameter Name: reset_stream_at

Status: Provisional (note that, prior to publication, the value will be replaced by a new value lower than 64)

Specification: This document

Change Controller: IETF (iesg@ietf.org)

Contact: QUIC Working Group (quic@ietf.org)

8.2. QUIC Frame Types

This document registers one new value in the "QUIC Frame Types" registry established in Section 22.4 of [RFC9000]. The following fields are registered:

Value: 0x24

Frame Type Name: RESET_STREAM_AT

Status: Provisional (will become Permanent once this document is approved)

Specification: This document

Change Controller: IETF (iesg@ietf.org)

Contact: QUIC Working Group (quic@ietf.org)

9. References

9.1. Normative References

[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>.

9.2. Informative References

[WEBTRANSPORT]
Frindell, A., Kinnear, E., and V. Vasiliev, "WebTransport over HTTP/3", Work in Progress, Internet-Draft, draft-ietf-webtrans-http3-12, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-webtrans-http3-12>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Marten Seemann
Email: martenseemann@gmail.com

Kazuho Oku
Fastly
Email: kazuhooku@gmail.com

Additional contact information:

奥一穂
Fastly