Signaling That a QUIC Receiver Has Enough Stream Data

Abstract

Sending on QUIC streams can only be aborted early by the sender with a RESET_STREAM frame. This document describes how a receiver can indicate when the data they have received is enough, allowing the sender to reliably deliver some data, but abort sending for anything more than the indicated amount.

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://martinthomson.github.io/quic-enough/draft-thomson-quic-enough.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-thomson-quic-enough/.

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/martinthomson/quic-enough.

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 https://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."
1. Introduction

RFC 9000 [QUIC] describes how streams can be completed or reset. A completed stream is delivered in its entirety, with the sender sending new STREAM frames until all data has been received and acknowledged by a receiver. On the other hand, data on a reset stream might never be sent, or - if it is sent - it might not be retransmitted in the case of loss.

The RESET_AT frame [RESET-AT] provides a way for a sender to split a stream into two parts: an early part that is transmitted reliably and any remainder, which is effectively reset. This combines aspects of both completed and reset streams, even though the entire content of a stream is not delivered.

This document describes the ENOUGH frame in Section 3, which provides an analogue of the STOP_SENDING QUIC frame (Section 19.11 of [QUIC]). This allows a stream receiver to indicate that it has received enough data and that any data beyond the indicated offset is not needed. A sender that receives this ENOUGH frame can cease sending and emit a RESET_AT frame at or beyond the indicated offset.
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. QUIC ENOUGH Frame

A QUIC ENOUGH frame (0xTBD) is shown in Figure 1.

ENOUGH Frame {
  Type (i) = 0xTBD,
  Stream ID (i),
  Application Protocol Error Code (i),
  Offset (i),
}

Figure 1: The QUIC ENOUGH Frame Format

Like STOP_SENDING, an ENOUGH frame can be sent for streams in the "Recv" or "Size Known" states; see Section 3.2 of [QUIC]. Receiving a STOP_SENDING frame for a locally initiated stream that has not yet been created MUST be treated as a connection error of type STREAM_STATE_ERROR; see Section 20.1 of [QUIC]. An endpoint that receives an ENOUGH frame for a receive-only stream MUST terminate the connection with error STREAM_STATE_ERROR.

ENOUGH frames contain the following fields:

Stream ID: A variable-length integer carrying the stream ID of the stream being ignored.

Application Protocol Error Code: A variable-length integer containing the application-specified reason the sender is ignoring the stream; see Section 20.2 of [QUIC].

Offset: A variable-length offset into the stream. Note that a value of 0 makes this frame equivalent to STOP_SENDING. This value MAY be larger than the final size of the frame; see Section 4.5 of [QUIC], in which case this frame has no effect.

ENOUGH frames have no direct effect on the stream state machine.

4. Negotiation

Endpoints advertise their support of the extension described in this document by sending the enough (0xTBD) transport parameter
(Section 7.4 of [QUIC]) 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; see Section 20.1 of [QUIC].

In order to allow reliable stream resets in 0-RTT packets, the client MUST remember the value of this transport parameter. If 0-RTT data is accepted by the server, the server MUST not disable this extension on the resumed connection.

This extension MUST NOT be advertised unless support for the RESET_AT frame is also advertised; see Section 3 of [RESET-AT]. An endpoint MUST treat receipt of this transport parameter without the reliable_reset_stream transport parameter as a connection error of type TRANSPORT_PARAMETER_ERROR.

5. Security Considerations

TODO Security

6. IANA Considerations

This document has no IANA actions.

7. Normative References


Acknowledgments

Martin Duke asked if this was possible. This is the result.
Author's Address

Martin Thomson
Mozilla

Email: mt@lowentropy.net