QUIC Reliable Stream Resets

draft-seemann-quic-reliable-stream-reset-00

Marten Seemann, QUIC WG, IETF 115
QUIC Stream Resets

- don’t retransmit STREAM frame(s)
- don’t deliver STREAM data to application
- instead: surface reset error immediately

STREAM frame → RESET_STREAM frame
What if the STREAM data contained a stream identifier?

- Common pattern: Applications prefix stream data with an application-layer identifier
  - for example: WebTransport Session ID
- Application-layer message may be composed of a header + body
RELIABLE_RESET_STREAM frame

Looks like a RESET_STREAM frame, with one additional field:

RELIABLE_RESET_STREAM Frame {
  Type (i) = 0x72,
  Stream ID (i),
  Application Protocol Error Code (i),
  Final Size (i),
  Reliable Size (i),
}
RELIABLE_RESET_STREAM

- Reliable Size
- Final Size
RESET_STREAM is just a degenerate REliable_RESET_STREAM
Endpoint behavior

- Retransmit STREAM data up to Reliable Size
- Drop retransmissions for everything after that
- Deliver STREAM data up to Reliable Size to the application
- Then: surface the stream reset error
Implementation Complexity

Changing the QUIC stream state machine sounds scary!

```bash
> git diff master --stat -- '!*_test.go'
internal/wire/frame_parser.go    |    2 +
internal/wire/reset_stream_frame.go |  47 +++++++++++++
internal/wire/stream_frame.go    |    2 +
receive_stream.go                |   21 +++++
send_stream.go                   | 114 ++++++++++++++++++++++++---
5 files changed, 136 insertions(+), 50 deletions(−)
```
Thoughts?
WebTransport Stream Architecture

HTTP Handler -> HEADERS

WebTransport session #1 -> WEBTRANSPORT_STREAM with Session ID #1

WebTransport session #2 -> WEBTRANSPORT_STREAM with Session ID #2

HTTP/3

QUIC

streams