

An Unreliable Datagram Extension to QUIC

draft-ietf-quic-datagram

Tommy Pauly, Eric Kinnear, David Schinazi

QUIC

IETF 110, March 2021, Virtual

Status

Down to four substantive open issues

Basis for work in MASQUE and WEBTRANS

Milestone to send draft-ietf-quic-datagram to the IESG in July 2021

Let's get this shipped!

No Flow IDs

<https://github.com/quicwg/datagram/issues/6>

Section 5 explains that applications are responsible for defining and managing demux flow identifiers

Question keeps coming up, but consensus seems clear

Highlight this in a new subsection (5.1)?

Clarify Loss Recovery

<https://github.com/quicwg/datagram/issues/8>

Text doesn't specify what to do if there is a suspected loss or PTO of only DATAGRAM frames

Consensus seems to be to treat these like any other packets, and send a PING frame if nothing else is enqueued

Transport Parameter

<https://github.com/quicwg/datagram/issues/3>

Currently is `max_datagram_frame_size`

Proposal (and PR) to use `max_datagram_payload_size`

- Payloads of size 0 should be allowed, but currently 0 in the TP means “not allowed”
- Do we need any limit?
- If we have a max size, should we have a frame to update it?

Application Events

<https://github.com/quicwg/datagram/issues/15>

Text currently says that QUIC MAY notify the application about ack'ed DATAGRAMs

Issue is about what happens if a receiver drops acked frames due to running out of space

Who owns the buffer? QUIC or the application?

Suggestion to say that if QUIC ever drops a received packet, it must inform the application

