Multicast Extensions for QUIC

IETF 114, MBONED

draft-jholland-quic-multicast

Jake Holland
Max Franke
Lucas Pardue

Outline

- Basic Idea
- Problems this approach solves
- Current state of the draft
- Reference implementation
- Next steps

Basic idea

- Use QUIC unicast connections as anchor for multicast
  - Client declares multicast support and limits (e.g. max rate)
  - Server picks fitting SSM channel(s) and tells client to join them
  - Client attempts join and receives data packets over multicast
- Unicast connection is used for integrity/ crypto keys
  - Similar to AMBI
  - Each packet sent over multicast has integrity frame (i.e. checksum) associated with it
  - Multicast packets are encrypted
- Client ACKs each packet over unicast
  - Server can choose to retransmit lost packets over unicast or (if many clients lost it) over multicast again
  - Guarantees reliability
- Multicast only works from server to client
- Multicast packets are also part of the QUIC connection, client does not need to differentiate

Problems this approach solves

- Feasible way to get multicast into browsers via QUIC
- Encryption for packets (though weak as all receivers can decrypt)
- Integrity prevents injection of packets by attackers
- Reliability means application layer does not have to worry about FEC etc.
- Since it is still just a QUIC connection to the application, support only required enabling a flag
- Pushes large data packets to multicast while only keeping integrity frames over unicast, though those can potentially be pushed via multicast as well

Current state of the draft

- Version 3, had several rounds of improvements and refinements already
- Overall structure seems clear, no obvious reasons that make it impossible
- Got feedback at QUIC WG…

Reference implementation

- Implementing in Chromium
- Got new frames and transport parameter added
- Server can push data, create channels etc.
    - The exact operation is highly dependent on what type of content is being pushed
- Still to figure out:
    - We will need WebTransport to push data eventually, should we continue with our current client or look for something else?
    - How to inject packets received over multicast into Chromiums event loop?
    - How does the server manage ACKing? How long should packets that haven’t been ACKed yet stick around?
- Many thanks to Kyle Rose, Enric Perpinyà and Léo Sarrazin for their help with the implementation

Next steps

- Continue working on the implementation
  - Hope to get a presentable Demo by IETF115
- Take feedback from QUIC into consideration