Multicast Extensions for QUIC

IETF 114, QUIC wg

draft-jholland-quic-multicast

Jake Holland
Max Franke
Lucas Pardue

Outline

- Basic Operation
- Points to Highlight
- Implementation Status
- Discussion/Next Steps

“Why?” elided for time. Please see:

- IETF 111 Web Multicast Bar Bof (slides)
- IETF 112 secdispatch (slides)

Basic Operation

- Source-Specific IP Multicast for some server --> client data
- Anchored on the unicast connection
    - Frames from multicast channels could equally have been sent unicast
    - No special restrictions on unicast connection
- Server-driven (with client consent)
    - Server MAY ask client to join channels (via extension frames in the draft)
    - Client MAY join as requested
- Client provides limits (for congestion control as in RFC 8085)
    - Aggregate Max Rate
    - Max Channel Count
- Client ACKs over unicast
    - per-channel packet number space
    - similar to multipath (w/multiple packet number spaces)

Points to Highlight

- “Connection” is still single server <--> single client
- Multicast channels carry ONLY server --> client packets
- All packets are interpreted in context of a connection
  - ChannelID = layer of indirection for a Connection, in client receive
    - Not client-chosen since same packet is delivered to many clients
  - Like multipath with one more layer of direction
- Covers security goals from draft-krose-multicast-security
  - Encrypted, but with keys shared across multiple clients
  - Integrity-guaranteed by Merkle tree w/ secure unicast anchor
  - secdispatch 112 feedback: needed a specific proposal before eval

Implementation Status/Maturity

- Demo (and later, reference) implementation in progress https://github.com/GrumpyOldTroll/quiche (fork of https://github.com/google/quiche)
  - W3C Multicast Community Group working sessions since April 2022
  - Informative to spec, several insights & iterations
- New security issue we noticed last week:
  - Webtransport traffic may need extra enforcement mechanism for origin policy
    - (Perhaps add an “MC_ORIGIN” frame to be sent in channel packets)

Protocol Extensions

- Transport Parameters
  - declare multicast support + client initial limits
- New Extension Frames
  - Server -> Client
    - Channel lifetime & static properties: **MC_ANNOUNCE**, **MC_RETIRE**
    - Key rotation for encryption: **MC_KEY**
    - Requests of client's channel state: **MC_JOIN**, **MC_LEAVE**
    - Integrity guarantees: **MC_INTEGRITY**
  - Client -> Server
    - Report channel join status: **MC_STATE**
    - Report packets received: **MC_ACK**
    - Congestion control limits: **MC_LIMITS**

Next Steps

- Interest in potential adoption?
- If there is interest: more work needed before an adoption call?
    - Spec maturity?
    - Some deployment?
- (PS: Come discuss further next session in MBONED)