QuicR
Media delivery protocol over QUIC

IETF 114

Cullen Jennings. Suhas Nandakumar, Christian Huitema
Unifying Multimedia Streaming and Interactive Communication

Streaming
- AV Sources
- Media Tx
- Media Encoders
- Stream Segmenters
- Media Servers
- Delivery
- HTTP Servers
- CDN
- Clients
- Media Rx
- HTTP

Interactive Communication
- Clients
- Media Tx/Rx
- Setup/Ctrl
- RTP
- Media Servers
- Control
  - Negotiate
- Media
  - Transcode
  - Switching
  - Mixing
- Clients
- Media Tx/Rx
- Setup/Ctrl
- RTP

New Experiences
- multi-users
- interactivity (lower latency)

large markets (scale, cost)
Use cases (Lower latency, Interactivity, Scale)

• Watching a soccer game live, but at the same time using your mobile phone to watch a feed of camera over the goal that is "real time"

• Large company wide meetings or conferences where lots of people are watching, but during the Q & A any of them can switch to become an active participant.

• E sports streaming where commentators and players can react real time to fans comments vs have a significant delay from when the fan makes a comment to when the fan hears or sees the commentator's reaction to it.

• < your favorite here >
QuicR

Publish/Subscribe based end-to-end encrypted media delivery protocol
Unifies streaming and interactive media flows
Caches and Relay Friendly
Common ingest and distribution protocol
Supports QUIC Streams and QUIC Datagrams (API controlled per media stream) - in end points and relays.
Control streams for configuring media streams and Media Streams for delivering media
Knobs to rate control/react to congestion at media senders (sources, relays)
QuicR - Pub/Sub Protocol

- Media Senders post media objects and segments as named resources
- Media Receivers request for the same by sending subscribes to the name
- QuicR Names ~= HTTP Resource Names
  - Unique and specific to a domain
  - Identify a cacheable resource
  - Authorized for use
    Names support prefixing/wildcarding, allowing clients make one request with prefix than multiple requests for each resource
  - Eg: Get me all video from channel-22 [ channel-22/video/*]
QuicR Ingest/Distribution Topology

Relay - Origin
Ingest Point

1 Subscription
For Alice's Media
Sent from Relay-B

N subscriptions
For Alice's media
Sent to Relay-B

Relay-A Cached
Media sent to R1, R2

Alice
Media sender

R1
Media receivers

Relay-B

Pub:alice-video
(1 copy)

R2
Pub:alice-video
(1 copy)

R3
Media receivers

R4

Rn
Media receivers

Subscriptions to Media, quicr://abc.com/channel-22/alice/*

Media publish flow

(1 copy)

Cached copies

Pub:alice-video
(1 copy)
QuicR - Relays

Store/Selective Forward Behavior
Handles name subscriptions
Distributes media objects matching name/name-prefix to subscribers
Has no access to media, but can read transport metadata (authenticated)
Metadata specifies - priority/best-before and drop/store encoded as 8 bit flags to handle congestion
Metadata includes group and object information to form full segments (and cache)
Fragmented media are sent as-is to keep latencies low (pipelining)
QuicR - Media objects

Media is divided into groups of objects
For video, each group represents “group of pictures”
  - [IDR frame, P1, P2, …. ]
For audio, each object belongs to its own group
Each object is uniquely identified
Client’s can ask media per group instead of per object
QuicR Early Prototype

Demo
- Relay deployed in Ohio AWS Cluster
- One media sender (opus 48khz, h264 720p)
- 3 media receivers
- Performance seems to be comparable to interactive experiences.

https://github.com/Quicr/qmedia/blob/main/README.md
Advantages

Support low latency media delivery comparable to RTP via
  efficient pipelining,
  Name prefixing,
  QUIC Datagram delivery

Support scalability comparable to streaming media
  RTP Scaling is expensive and may not be possible at this scale
  Effective use of relays/caches (akin to CDN)

Enables converged use-cases that needs seamless switching between interactive and streaming experiences

Unified Protocol across media contribution and distribution
<fin>