Overview of Proposed Solutions Space

March 2022

Nermeen Ismail, Cullen Jennings, Suhas Nandakumar, Mo Zanaty

V3

Use Case: Interacting with Broadcast & Sport Betting

- Users at a soccer game (or esports) watching the game IRL and also viewing a feed of camera in the goal net or a different angle
- Remote users watching same game and betting
  - desire to not have a feed with less latency than zoom call to same game
- Remote users watching game and want to see the goal before they hear the neighbors cheer
- Remote user becomes an interactive participant in the broadcast for commentary or game show

Needs:

Latency comparable with WebRTC across viewers of all types (remote, live ..). Latency compatible with IRL for local users.

Scaling comparable to DASH

Quality comparable to DASH

Ability for any user to become a live media contributor

Fast adaptation to changing bandwidth/delay

Side note: Dash IF presented great set of uses cases in MOPs this week. Report at https://dashif.org/webRTC/report

Use-case: Large Company Meeting

- Enterprise conference call with 100,000 people spread around the world
- At a given time, moderator has authorized a small set to speak but any of participant can be authorized with no loss of media continuity
- Similar uses cases to online course to many people any of which can ask a question, or fans watching esports where a fan can join the commentator streams

Needs:

Latency comparable to WebRTC

Scaling comparable to DASH

Quality comparable to TelePresence

Ability for any user to become a media contributor

Allows optimization of shared bandwidth for users at same location

Fast adaptation to changing bandwidth/delay

Use-case: Less than perfect WiFi or 5G

- Improve media quality for people on poor WiFi
- Person streaming over WiFi with intermittent loss trying to stream a high quality, low latency 4k stream to cloud
- A local relay in AP or on WAN can have low RTT retransmissions over WiFi
- Similar use cases on distribution side and with relay in 5G edge compute

Needs:

Latency comparable to WebRTC

Scaling comparable to DASH

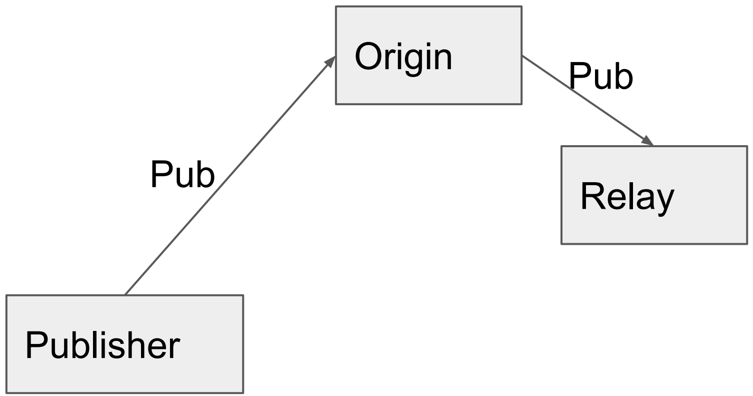
Better Quality than WebRTC or RMTP or Dash

Fast adaptation to changing bandwidth/delay

RUSH

Client pub video to server over QUIC

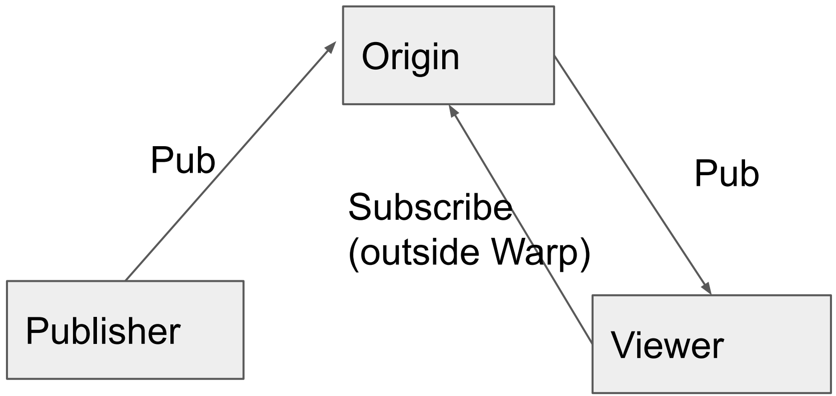
- **1-stream mode** where this is like TCP
- **Multi-stream mode** where 1 stream per audio packet and 1 per video frame
- Way for server to tell clients to find another server (goaway)
- **64 bit ID** to identify the content
- Fully reliable media over stream



Warp

Origin pushes video to client over QUIC

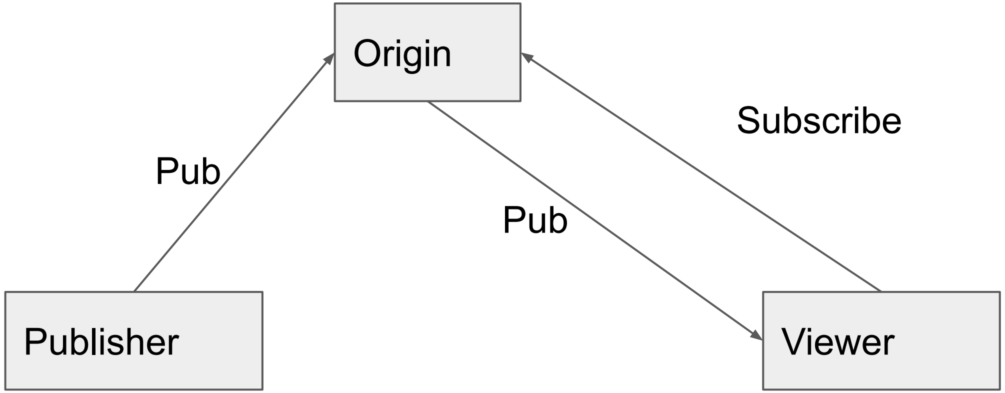
- Send **I frame + dependent video frames** in 1 stream
- **Priority** with audio > new video > older video
- Client cancels streams that are starving
- Uses **CMAF format** headers
- Set up connection and how to identify content is out of scope (but required)
- Fully reliable media over stream



QuicR (without relays)

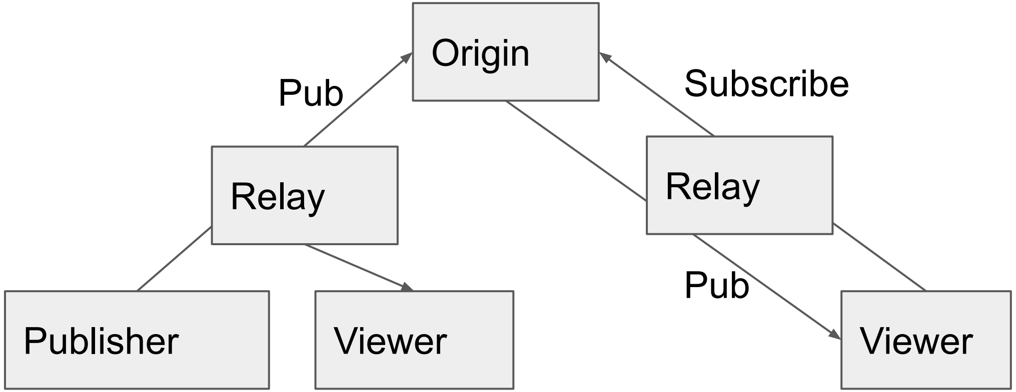
Client pub video to server/ Client ask for video and origin sends it to client over QUIC or WebTransport

- Option to send **I frame + dependent video** frames in 1 QUIC stream
- Option to send data in a non reliable way **over QUIC Datagram**
- Send data **fragments of data as they arrive** downstream
- Higher priorities for lower bitrate versions of same media
- Media received by Origin **can be same format** as what Origin sends



QuicR (with relays)

- Like previous slide but with explicitly design for CDN like relays
- Relays can be discovered with **DNS, Origin Redirect, or local network**
- **Simple metadata** to enable Relays to make caching decisions
- Option for **“Relay Short-circuiting”** for locally connected receivers



Summary of proposed design space

“North star” view of the eventual solution that solves problems today solutions can’t solve

Shorter term work on how the “pub” arrows send media

- Do “Group of Pictures” over 1 stream
- Do audio over streams or Datagrams
- Latency largely determined by QUIC congestion controller (changes out of scope of this WG)
- Content Naming

Design ingress and distribution to work together

Design allows option of “relays”

Easier to build, develop, and deploy than WebRTC