Disclaimer

- Early stages Proof of concept (POC)
  - Just A way (NOT the way) of implementing MOQ

- Goal:
  - Learn about proposed technologies
  - Provide as much information / metrics possible

- Future:
  - Hope could help us to have a better discussion about MOQ options
  - Perhaps give some light over some trade-offs to make
Use cases

- Ingest
  - ULL live
- Egress
  - ULL live (live edge)
  - Live rewind
  - Highlights / VOD
- Not yet
  - Priorities (sendOrder)
  - ABR
Test Ultra low latency with Webcodecs + WebTransport: PLAYER
server -> Demux -> Decode -> Play

(Encoder audio sampling frequency should be the same that audioContext (player) sampling frequency, this is almost guaranteed if you use same freq)

Data needed

<table>
<thead>
<tr>
<th>Data type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>WT server:</td>
<td></td>
</tr>
<tr>
<td>StreamID:</td>
<td></td>
</tr>
<tr>
<td>Package type:</td>
<td></td>
</tr>
<tr>
<td>Player buffer (ms):</td>
<td>100</td>
</tr>
<tr>
<td>Audio jitter buffer (ms)</td>
<td>200</td>
</tr>
<tr>
<td>Video jitter buffer (ms)</td>
<td>100</td>
</tr>
</tbody>
</table>

Test Ultra low latency with Webcodecs: ENCODER
WebCam(v+a) -> Encode -> Mux -> Send -> Server

Data needed

<table>
<thead>
<tr>
<th>Data type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>WT server:</td>
<td></td>
</tr>
<tr>
<td>StreamID:</td>
<td></td>
</tr>
<tr>
<td>Package type:</td>
<td></td>
</tr>
<tr>
<td>Max audio sending buffer allowed (ms):</td>
<td>300</td>
</tr>
<tr>
<td>Max video sending buffer allowed (ms):</td>
<td>150</td>
</tr>
<tr>
<td>Max inflight audio requests:</td>
<td>100</td>
</tr>
<tr>
<td>Max inflight video requests:</td>
<td>60</td>
</tr>
<tr>
<td>Expiration time for media chunks (except init) (in sec):</td>
<td>10s</td>
</tr>
</tbody>
</table>
Main features

- Uses QUIC streams over WebTransport (WT)
- QUIC stream per frame

- Encoder
  - Opens WT session against relay
  - Pushes data to relay

- Player
  - Opens WT session against relay
  - Relay pushes data to player
Demo setup

~8000Km (4970 miles)

RTT ~ 140ms (ping)
DEMO Results

- H264 500Kbps + AAC 32Kbps
- Jitter buffer: 200ms
- E2E latency: ~ 400ms
- QOE: VC+ (IMHO)
Encoder
Player
Media Packager

VERY EXPERIMENTAL!!!!

Efficiency: 92%+
- Efficiency definition:
  Total payload bytes vs packager bytes sent to transport
- Test conditions:
  Video at 1Mbps, and Audio at 32Kbps
Relay

Cache key: streamID/media-type/seqId

Examples:
- mystreamabc/video/123
- mystreamabc/video/-1 (init)
- Player & encoder:
  https://github.com/facebookexperimental/webcodecs-capture-play

- Relay
  https://github.com/facebookexperimental/go-media-webtransport-server