Live Media Ingest Challenges

Ying Yin
YouTube
Live Media Ingest High Level Requirements

- Support high visual quality: 4K HDR
  - Broadcast level events: sports, gaming tournaments, concerts

- Codec Agility
  - HEVC, VP9, AV1

- Support low and ultra low latency (< 1 sec)
  - Sports, gaming with interaction via chats

- Ease of adoption among encoders
  - Software, hardware, mobile, browsers

- Large scale deployment
  - Load balancing, release update
Live Media Ingest Challenges

- **RTMP**
  - Doesn’t have an official way to add new codecs, 4-bit enum for codecs
  - Latency: directly on top of TCP, has head-of-line blocking issue

- **WebRTC**
  - Adapts visual quality down quickly for conversational latency

- **HLS/DASH**
  - Higher latency as it is segment-based

- **Low Latency HLS/DASH**
  - More frequent playlist/manifest requests have overhead
  - Low adoption among encoders

- **SRT**
  - Has limitations for large scale deployment

- No perfect solution for ingesting high-end content at ultra low latency and at large scale.
Proposed MoQ Solutions

Focusing on Live Media Ingest specifically

- RUSH (draft-kpuqin-rush)
- SRT over QUIC (draft-sharabayko-srt-over-quic)
Questions?
Backup Slides
Most commonly used for media ingest to live streaming platforms.

**Visual Quality**
- Can maintain good visual quality at high bitrate.

**Codec Agility**
- Doesn’t have an official way to add new codecs.
- 4-bit enum for codecs

**Latency**
- Frame-based, can support ultra-low and low latency live streams.
- TCP based which has the head-of-line blocking issue and can’t easily use newer congestion control algorithms.

**Adoption**
- Widest adoption among software and hardware encoders.
- Not supported by browsers.

**Large scale deployment**
- TCP based, no issues
Designed for video conferencing which has much stricter latency requirements in order to maintain conversational interactivity.

**Visual Quality**
- Sacrifices too much quality for latency, not suitable for premium content.

**Codec Agility**
+ Can support new codecs such as HEVC, VP9.

**Latency**
+ UDP-based, can support ultra-low and low latency live streams.

**Adoption**
+ Supported by browsers
- No adoption by hardware encoders
- No much adoption by software encoders

**Large scale deployment**
- UDP based, load balancing is not a given for most off-the-shelf load balancers.
HTTP based protocols which are originally designed for distribution.

**Visual Quality**
+ Can maintain good visual quality at high bitrate.

**Codec Agility**
+ Can support new codecs such as HEVC, VP9.

**Latency**
- Segment-based, additional latency which is at least the duration of the segment.

**Adoption**
- Mostly supported by high-end hardware encoders.

**Large scale deployment**
- TCP based, no issues
Current Protocols - Low Latency HLS/DASH

Uses partial media segments to lower the latency.

**Visual Quality**
+ Can maintain good visual quality at high bitrate.

**Codec Agility**
+ Can support new codecs such as HEVC, VP9.

**Latency**
+ Lower latency than HLS/DASH
- More frequent playlist requests add more overhead

**Adoption**
- Even less adoption than HLS/DASH for ingest

**Large scale deployment**
- TCP based, no issues