CMAF Low Latency

May 7th, 2019

Paul MacDougall
Principal Sales Engineer
Why is latency higher today?

History and Background - Encoder, CDN, Player

What’s changing?

Latency Definition

Background and Problem Statement

Low Latency Solution Architecture

Demo

Limitations and Challenges

Closing Statements

Questions
CMAF Workflow

Note: HTTP Chunks ≠ CMAF Chunks
Latency available at scale

Via Mainstream CDNs Streaming Technologies

High Latency

Typical HLS Latency
For non-HLS HTTP adaptive streaming format

Typical Latency
Using traditional streaming and tuned HTTP adaptive streaming

Reduced Latency
Live sports events; Second-screen experiences; Breaking news

Low Latency
Video surveillance; Gambling; Betting; Online auctions

Ultra-Low Latency
Voice chat; Video gaming; Web Conferencing

Real Time

Streaming Latency Continuum

Typical Latency for HD Cable TV

HLS
Smooth Streaming, HDS, DASH
RTMP
RTP/RTSP, WebRTC

60+ seconds
45 seconds
30 seconds
15 seconds
5 seconds
1 second
500 milliseconds
200 milliseconds

BitmovinPlayer
Architecture: Why the latency was higher?

Fundamental assumptions/tradeoffs:

- Segment duration and latency are correlated
- Robust playback experience is prioritized over latency
Background: Encoding

Where does latency occur in the encoding pipeline?

Video ingest
For an encoder (SDI inputs): 0.
For a transcoder (compressed inputs): decoding latency depends on how the input stream was encoded

Video pre-processing
Denoising, deinterlacing, resizing... may add latency
- E.g. MCTF: Motion-Compensated Temporal Filtering
- E.g. Motion Adaptive De-Interlacing

Video compression
Latency depends on the video compression pipeline's architecture & algorithms:
- "GOP pattern": the use of out-of-order pictures ("B pictures): increases latency
- Look-ahead: increases latency
- In the old days (DVB, ATSC...): "VBV compliance" rate control added latency. Not required for OTT TV.

OTT packager
Builds OTT segments to publish. 1 segment = 1 MP4 (or TS) file
To build 6-second segments, the packager must aggregate 6 seconds worth of content before releasing it as one file.
With the player, buffer management is critical to ensure stability

- Waits for about 2-3 segments (12-18s) to be available before starting playback
- Maintains 20-40s buffer duration for HLS/DASH streams

Thereby introducing an inherent 20 second latency.
CMAF Low Latency Chunking

Traditional segment (mp4 file)

CMAF low-latency segment (mp4 file)
End-to-end latency should not necessarily impact the playback experience.

Use HTTP/1.1 Chunked Transfer protocol

Latency < N * segment chunk duration

Low Latency

Encoder-Packager still creating segment 5 chunks

Encoder-Packager still creating segment 5 chunks

Player would request starting from segment 5

Player would request starting from segment 5

End-to-end latency should not necessarily impact the playback experience
Architecture Breakdown: Player

1. Segment Loader
   - Load HTTP chunks (as opposed to fully formed segments)
   - Requires underlying browser to support HTTP chunked transfer encoding (Firefox*)
     - Fetch API
     - ReadableStream of Streams API

2. Latency Controller
   - Extract timing data (set latency vs. current time)
   - Measure playback latency continuously
   - Observe buffer levels
   - Manipulate playback speed or seek behavior to achieve target latency

3. Decoder / Renderer
   - Render chunks without waiting on fully formed segments
Limitations and Challenges

The reality
CMAF and CMAF LL adoption/production readiness is still quite early across the ecosystem.

Major encoders are still doing MPEG TS, which does not support CMAF LL
Predict mid 2019 for everyone to catch up

Challenges
- CMAF LL proven on DASH enabled devices
- CMAF LL for Apple devices is still unsolved (unless you have a fully custom non AVPlayer)
- Low latency comes at the cost of overall stream stability
  - Twitch use case (had to rollback)
  - Video quality selection might not be optimal (cannot accurately predict bandwidth)
  - Lower the latency, higher the risk in terms of stream robustness
  - Drifting can occur if audio and video chunks are not aligned

Positives
- Start up time actually improves*
- Seek time improves and playback resumes quicker*

Takeaways
- Really up to content owner to decide what latency they want to target
- Recommend 5 seconds (still better than today’s 30s)
- Enable selectively, apply safeguards in your application
- Allow users to choose the bitrate of their choice
Questions?

go.bitmovin.com/lower-latency