

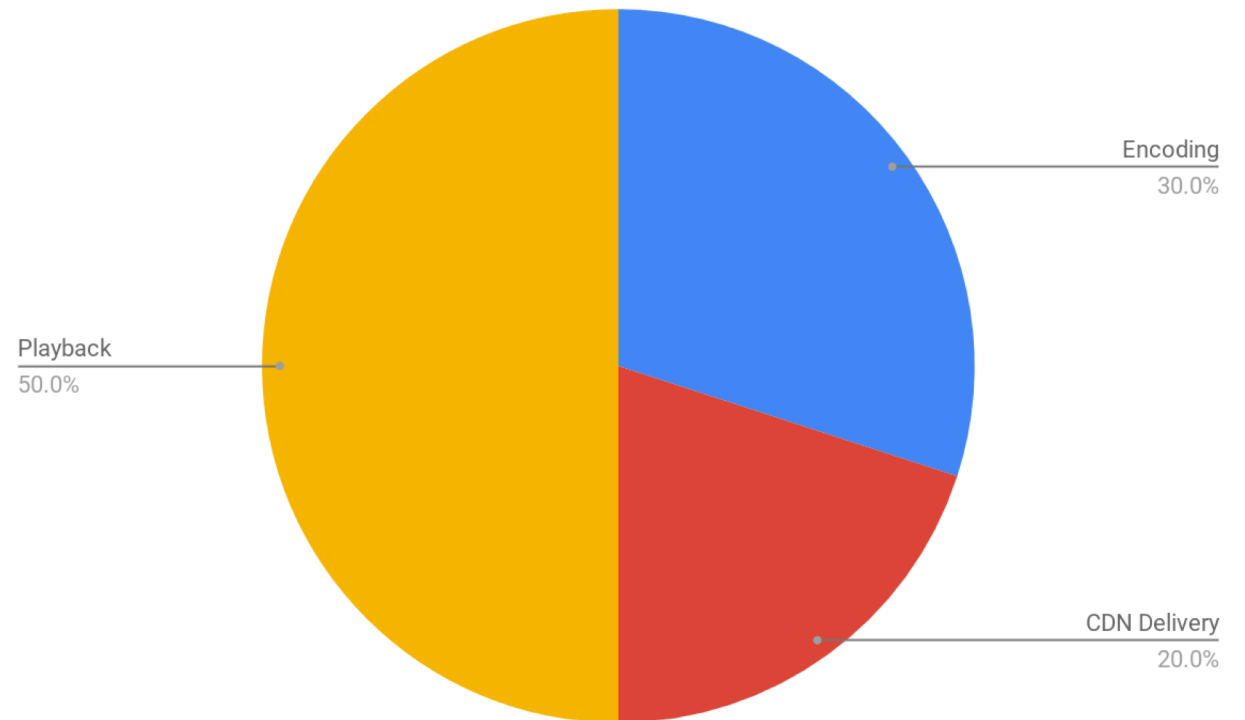


Delivering Ultra Low-Latency Live Streaming with DASH and CMAF

- Interactive Media
 - Game shows
 - Personal broadcast
 - Screen broadcast
- Time Sensitive Media
 - Gambling
 - Auctions

Latency Causes

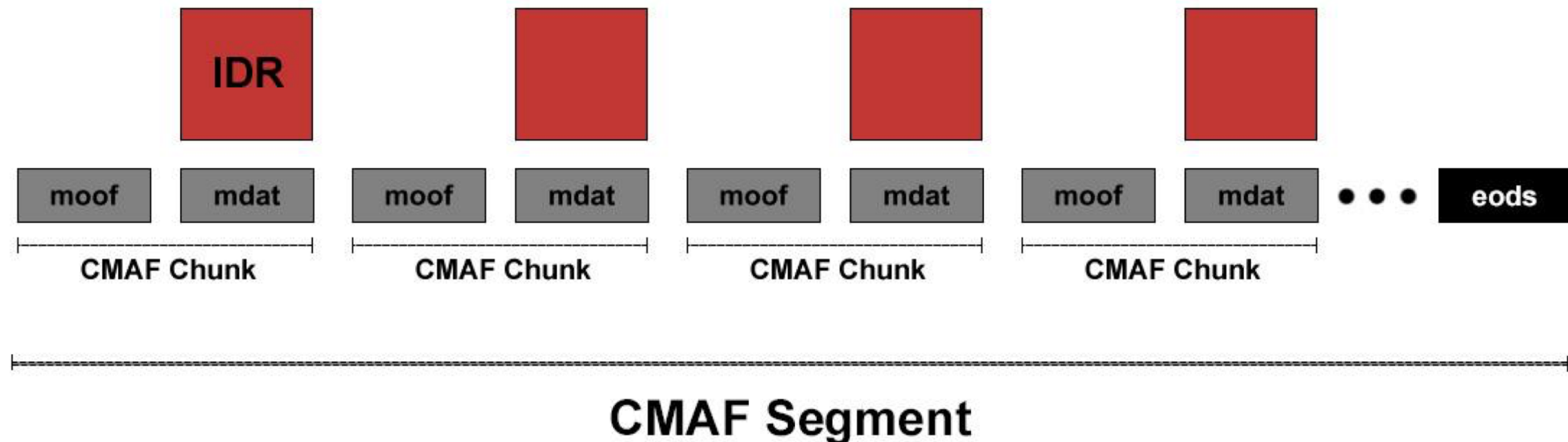
1. Encoding and Packaging
2. CDN Ingest and Delivery
3. Playback



- B-Frames increase encoding latency
- Small Closed GOP sizes
- Modify Encoder Buffer Size
- Bitrate selections based on output

Chunked CMAF Packaging

- Using closed GOP with fixed segment duration
- Start each segment with IDR frame
- Segments made up of CMAF chunks
 - CMAF chunks can be as short as 1 frame



Other Packaging Considerations

- Keep DVR window short
- Optimize segment length for your use case
- Understand platform you're targeting

- Map CMAF chunk to HTTP chunk at Origin
- Use HTTP chunked transfer to retrieve chunks from Origin as they are written
- Players can send HTTP chunked transfer requests to CDN and receive bytes as they are propagated to edges
- Sometimes players can request chunks early and CDN needs to gracefully handle 404s related to these requests

1. Playback drift, falling very behind the live edge
2. Knowing where the live edge is
3. Short buffer management
4. Quality of Experience

- Proposed DASH extension
- Player usually sends request when entire segment available
- availabilityTimeOffset indicates difference of availabilityStartTime of the segment and UTC time of server when it can start delivering segment

availabilityTimeOffset = segment duration - chunk duration -
UTC mismatch between server and client

- Set player internal timestamp with UTC time
- Possible to sync with manifest defined live window
- Still need to account for delivery time and buffer ahead

If player falls significantly behind live edge different choices available

1. Do nothing and player keeps falling behind
2. Implement playback speed-up logic to chase back to the live edge (e.g. if player falls behind 2 second chase threshold playback will start at 1.2x speed until back at chase threshold)
3. Implement seek to live edge

- Sometimes player will run out of buffer while waiting for chunks to be available on server
- Player can adapt to this by increasing buffer when under-run event occurs and retries succeed

- With knowledge of when chunks will be available, a defined live edge allows players to sync to a specific time after the live edge
- The further from the live edge the easier to control buffers and improve quality of experience

- Reliably handling adaptive bitrate logic when using chunked transfer to the players
- Managing latency around decoding pipelines on different devices
- Improving quality without increasing latency

Questions?