Agenda

- Understanding the problem
- Reducing latency
  - Delivery
  - Player
  - Content
  - Up and Coming
  - Some test results
Latency

- Time to video play
  - Important to all viewers

- Latency behind live stream
  - Important to some live events
  - Critical to some events involving betting or auctioning
Latency

STREAMING LATENCY AND INTERACTIVITY CONTINUUM

COMMON HTTP LATENCIES TODAY
One-way streams of live events to large audiences; linear programming

REDUCED LATENCIES
OTT providers; live-streaming news and sports

LOW LATENCY
UGC live streams; game streaming and e-sports

NEAR REAL-TIME
Two-way Web conferencing; telepresence; real-time device control (e.g., PTZ cameras, drones)

45+ seconds 18 seconds 05 seconds 01 second < 01 second
Latency

- Time to video play
  - Important to all viewers

- Latency behind live stream
  - Important to some live events
  - Critical to some events involving betting or auctioning

- Standards
  - Traditional: 30+s
  - Low Latency: 10s or less (1-2s diff from TV)
  - Ultra Low-Latency: 3s or less
Latency

LOW LATENCY

HIGH QUALITY

LARGE SCALE
Key Factors

- **PLAYER**
  - Initial Buffer
  - Bitrate Selection
  - Manifest Size
  - Bandwidth

- **CONTENT**
  - Segment Size
  - Bitrate
  - Encoding
  - Packaging

- **DELIVERY**
  - Protocol
  - Caching
  - Scale
Encoding/Packaging Delay

- Content generation & notification delay
Key Factors

- **Content**
  - Segment Size
  - KFI

- **Delivery: Persistent vs Non-Persistent**
  - Protocol
  - CDN: (massive impact at scale)
    - Encoder > Ingest > Transcoder > Mid Tier > Origin Shield > Edge Cache > Client | Buffer
    - Encoder > Packager > Ingest > Origin Ingest/Cache > Edge Cache > Client | Buffer

- **Player**
  - Pushback from live
  - Initial buffer
Generic Solution

- 1-2 sec Segment
- Rolling DVR
- Start playback after 1-6 segments
Dangers

- Buffer Starve

- More Overhead:
  - Network
  - CPU/GPU

- Encoding/Packaging
  - Segment & KFI

- More Caching Overhead

- Delivery Race Conditions
TESTING

1, 2, 3, Testing...Testing....is this thing on?
How We Tested

- Enable configuration based test library
  - What to play
  - Initial bitrate
  - Start time

- Applications implement:
  - QOS display
  - Unified remote logging for test aggregation


- Made Native apps for Roku, Android, & FireTV

- Reporting Data: [https://docs.google.com/spreadsheets/d/1iBTGgRcMvh0nCRsP9MpwYRo5tdH65ZmjZCzeKqsUGAc/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1iBTGgRcMvh0nCRsP9MpwYRo5tdH65ZmjZCzeKqsUGAc/edit?usp=sharing)

Baseline Startup - All at 6 Second Segment/2 Keyframe

<table>
<thead>
<tr>
<th>Segment/KFI</th>
<th>7800</th>
<th>6000</th>
<th>4500</th>
<th>3000</th>
<th>2000</th>
<th>730</th>
<th>365</th>
</tr>
</thead>
<tbody>
<tr>
<td>Startup Latency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Browser</td>
<td>1403ms</td>
<td>1752ms</td>
<td>869ms</td>
<td>557ms</td>
<td>616ms</td>
<td>637ms</td>
<td>646ms</td>
</tr>
<tr>
<td>- iOS</td>
<td>753ms</td>
<td>472ms</td>
<td>460ms</td>
<td>474ms</td>
<td>444ms</td>
<td>491ms</td>
<td>448ms</td>
</tr>
<tr>
<td>- Android</td>
<td>2486ms</td>
<td>1916ms</td>
<td>1746ms</td>
<td>3207ms</td>
<td>1670ms</td>
<td>2567ms</td>
<td>2152ms</td>
</tr>
<tr>
<td>- Roku</td>
<td>2935ms</td>
<td>2058ms</td>
<td>2227ms</td>
<td>1495ms</td>
<td>1672ms</td>
<td>1801ms</td>
<td>1658ms</td>
</tr>
</tbody>
</table>
Screen Shots
DELIVERY

30 Minutes or less or its FREE!
Delivery Standards

- Non-Persistent

- Persistent
Delivery Standards

- Non-Persistent
  - Generally TCP
  - ByteRange & Chunking
    - HTTP 1.1

- Persistent
Delivery Standards

- Non-Persistent
  - Generally TCP
  - ByteRange & Chunking
    - HTTP 1.1

- Persistent
  - Web Sockets
  - WRTC
  - UDP
  - HTTP 2
  - SRT
  - QUIC
# Delivery Standards

## Non-Persistent
- Generally TCP
- ByteRange & Chunking
  - HTTP 1.1
- CMAF Chunks
- LHLS

## Persistent
- Web Sockets
- WRTC
- UDP
- HTTP 2
- SRT
- QUIC
- All about the infrastructure
- Akamai Advanced Media Solutions: Media Services Live 4.0
<table>
<thead>
<tr>
<th>KEY DIFFERENTIATORS</th>
<th>BENEFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media Ingest Acceleration</td>
<td>Allows for improved ingestion performance over the open internet to match broadcast quality</td>
</tr>
<tr>
<td>Self Healing</td>
<td>Brings the reliability and availability required for live 24/7 streaming content &amp; large events</td>
</tr>
<tr>
<td>Low-Latency</td>
<td>Live content 1-2 seconds behind broadcast</td>
</tr>
<tr>
<td>Enhanced Monitoring &amp; Alerting</td>
<td>Allowing customers to quickly identify and mitigate first mile issues</td>
</tr>
<tr>
<td>DVR and Archive</td>
<td>Provide end users ways to match the TV experience online</td>
</tr>
</tbody>
</table>

Generic storage systems do not offer any of these features and benefits. They are NOT purpose-built like Media Services Live.
Key Choices

● HTTP/TCP - Same old stuff
  ○ Reliable
  ○ Slow

● Socket
  ○ Faster
  ○ More complex
PLAYER

Don’t hate the player, hate the game
Segment Size and Key Frame Interval

● Why important
  ○ Players download a number of segments before they start playback
  ○ Longer segments take longer to download

● Apple’s recommendations
  ○ 6 second segment size/2 second keyframe

● Buffer
  ○ May need to increase number of segments received before playback starts to avoid buffering
  ○ Recommend 3-6 seconds of chunks (more on this later)
## Our Tests - Impact on Startup Latency

<table>
<thead>
<tr>
<th>Segment/KFI</th>
<th>.5/.5</th>
<th>1/1</th>
<th>2/2</th>
<th>3/3</th>
<th>4/2</th>
<th>5/1</th>
<th>6/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>Avg: 40.44 Max: 84.44</td>
<td>Avg: 41.11 Max: 84.82</td>
<td>Avg: 41.42 Max: 85.48</td>
<td>Avg: 41.53 Max: 85.87</td>
<td>Avg: 41.42 Max: 85.48</td>
<td>Avg: 41.11 Max: 84.82</td>
<td>Avg: 41.42 Max: 85.48</td>
</tr>
<tr>
<td>Startup Latency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Browser</td>
<td>321ms</td>
<td>257ms</td>
<td>355ms</td>
<td>416ms</td>
<td>440ms</td>
<td>349ms</td>
<td>333ms</td>
</tr>
<tr>
<td>- iOS</td>
<td>777ms</td>
<td>503ms</td>
<td>435ms</td>
<td>497ms</td>
<td>445ms</td>
<td>447ms</td>
<td>460ms</td>
</tr>
<tr>
<td>- FireTV</td>
<td>3188ms</td>
<td>2626ms</td>
<td>2487ms</td>
<td>1870ms</td>
<td>1764ms</td>
<td>1962ms</td>
<td>1499ms</td>
</tr>
<tr>
<td>- Roku</td>
<td>2935ms</td>
<td>2149ms</td>
<td>1800ms</td>
<td>2051ms</td>
<td>1984ms</td>
<td>1713ms</td>
<td>1954ms</td>
</tr>
</tbody>
</table>
Player Side Adjustments

- Number of segments before playback starts
  - General practice
    - Browser - 1-3 segments
    - iOS - 3 segments
    - Android - 3 segments
    - Roku - 3 segments
    - With 6 second segments, that’s 18 seconds of video
  - Our tests
    - Segment size to 1 second
    - Varied number of segments
CONTENT

Content is king...unless the queen says otherwise
What’s at Stake?

- If transmuxing, greater number of keyframes and segments may impact CPU requirements
- Modify chunks in real time HLS playlist
  - Less is better but all depends on segment size - min 10 segments
  - Wowza recommends 12 seconds of data in each playlist
- May increase caching needs
  - Need to cache more segments to ease access
  - Wowza recommends 50 seconds of segments
Which Stream First

- Perspective - Apple recommends:
  - 2 Mbps stream retrieved first for Wi-Fi/Ethernet
  - 730 kbps variant for cellular

- Many encoding tools don’t implement this
  - Generally done as first rendition in master

- Also can be tied to player logic (recommended)

- Can make significant difference on startup latency
HTTP 1.1 Chunked Transfer Coding

- **What is it?** Streaming data transfer mechanism that enables transfers of chunks within a segment before the complete segment is retrieved
  - Player can start to receive portions of a segment before it’s delivered from the encoder

- **How to implement?**
  - HTTP 1.1 spec
    - Takes the concept byte range mixed with segments
Encoding/Packaging Delay

- Content generation & notification delay
CMAF Chunks

Without CMAF Chunk:

```plaintext
data
segment 1.mp4
```

DRAWBACK: must wait, at least, the encoding of the full segment before transferring/decoding/displaying

With CMAF Chunks:

```plaintext
data
data
data
data
segment 1.mp4
```

BENEFIT: can start transferring/decoding/display video before the end of the segment encoding

[ MEDIA LAYER ]
Pre-Announce Streams in Manifest

- Add segment URLs to the playlist before actually produced
- When combined with chunked transfer coding, ensures that all segments are retrieved as quickly as possible
Caveats of Pre-Announcing

- Using predictive tags impacts ability for discontinuities & seg duration
- ABR calculation limitations
- Need an Origin Shield!!
PERISCOPE: Low-Latency HLS (LHLS)

- Delivered using HTTP/1.1 Chunked Transfer Coding
- Pre-Announce Segments - roughly 2-3 in the future
- Connections stay open until bits are received
  - First receives MPEG Transport Stream (TS) segment header for the next currently unavailable segment
  - Then bits are streamed in as they are created
  - When HTTP2 becomes broader spectrum will reduce socket overhead
- CDN Vendor Needs to Support Chunked Transfer Coding
- Solid CDN Origin Shield Needed
- Side benefit of pre-warms cache for replay content

https://medium.com/@periscopecode/introducing-lhls-media-streaming-eb6212948bef
Low-Latency HLS (LHLS)

https://medium.com/@periscopocode/introducing-lhls-media-streaming-eb6212948bef
THE FUTURE

Is it tomorrow...or yesterday? Depends on your perspective.
Third Party Implementations

- HTTP 2
- SRT
- WOWZ
- QUIC
- Aspera: FASP
QUESTIONS?

- David Hassoun  
  david@realeyes.com
- Jun Heider  
  jun@realeyes.com
- Jan Ozer  
  jozer@mindspring.com