Latency

A Joint SLC/RealEyes Production

www.realeyes.com
www.streaminglearningcenter.com

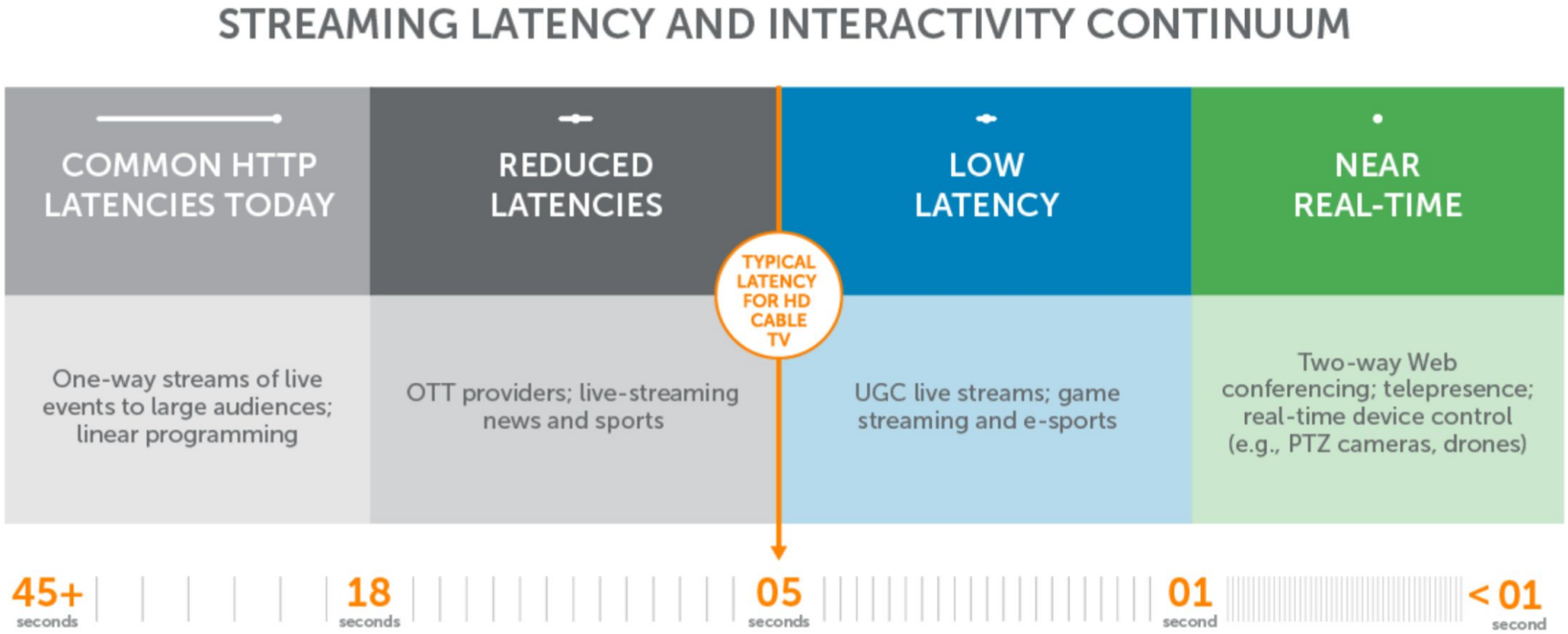
Agenda

- 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

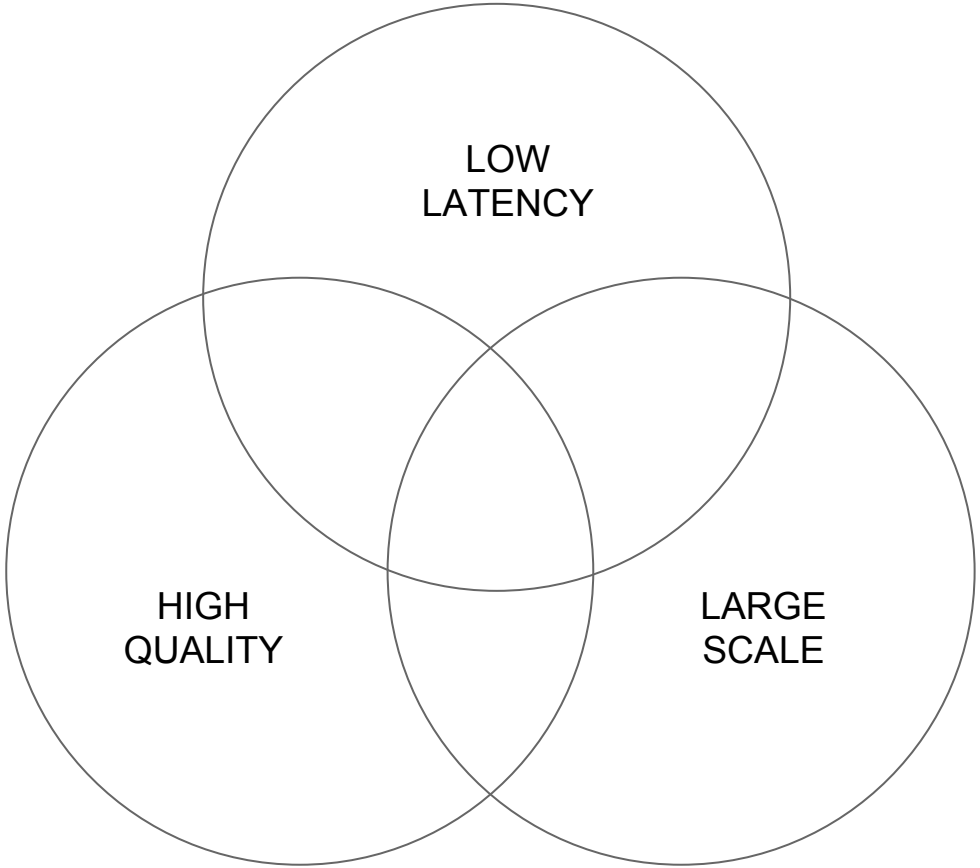
Latency


STREAMING LATENCY AND INTERACTIVITY CONTINUUM
COMMON HTTP LATENCIES TODAY
REDUCED LATENCIES
LOW LATENCY
NEAR REAL-TIME
TYPICAL LATENCY FOR HD CABLE TV
One-way streams of live events to large audiences; linear programming
OTT providers; live-streaming news and sports
UGC live streams; game streaming and e-sports
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

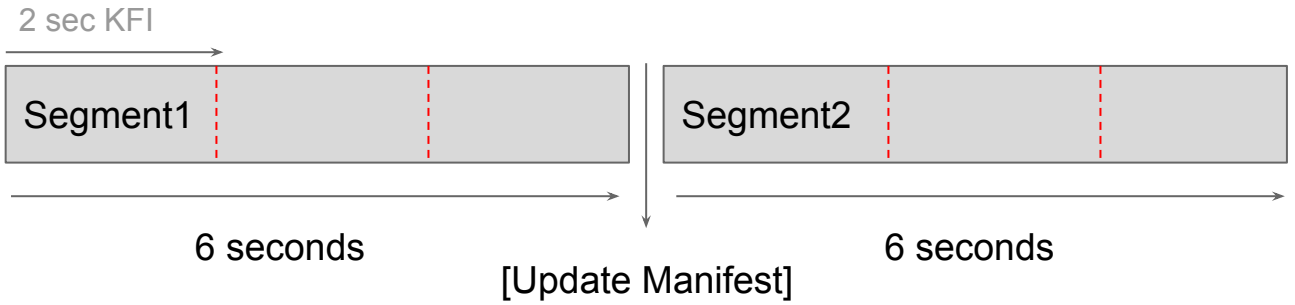


Key Factors

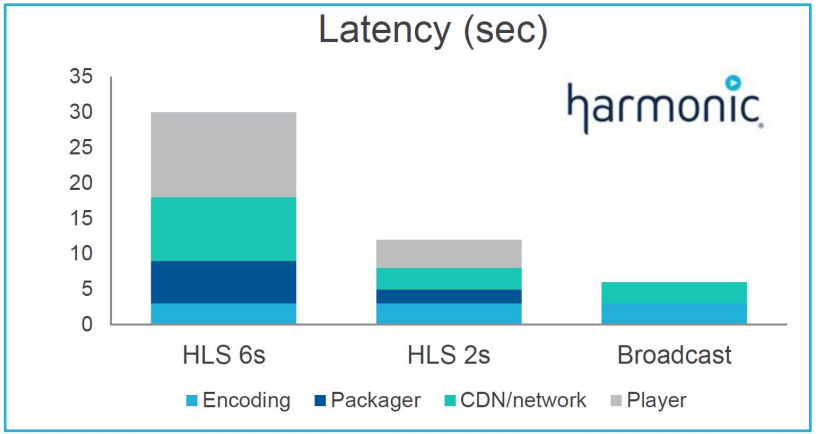
- **PLAYER**
  - Initial Buffer
  - Bitrate Selection
  - Manifest Size
  - Bandwidth
- **CONTENT**
  - Segment Size
  - Bitrate
  - Encoding
  - Packaging
- **DELIVERY**
  - Protocol
  - Caching
  - Scale

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
  - Segmement & KFI
- More Caching Overhead
- Delivery Race Conditions

TESTING

1, 2, 3, Testing...Testing....is this thing on?

How We Tested

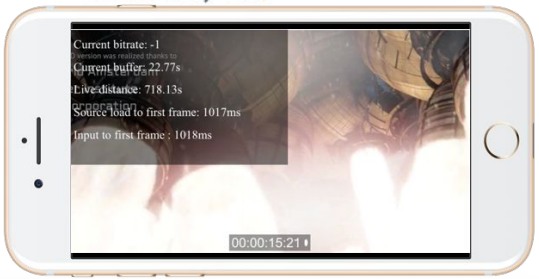
- Enable configuration based test library
  - What to play
  - Initial bitrate
  - Start time
- Applications implement:
  - QOS display
  - Unified remote logging for test aggregation
- Let’s look at the JSON file: http://office.realeyes.com/demos/smw-2017-data/data.json
- Made Native apps for Roku, Android, & FireTV
- Reporting Data: https://docs.google.com/spreadsheets/d/1iBTGgRcMvh0nCRsP9MpwYRo5tdH65ZmjZCzeKqsUGAc/edit?usp=sharing
- Browser application for desktop: http://smw-demo.realeyes.com/player/

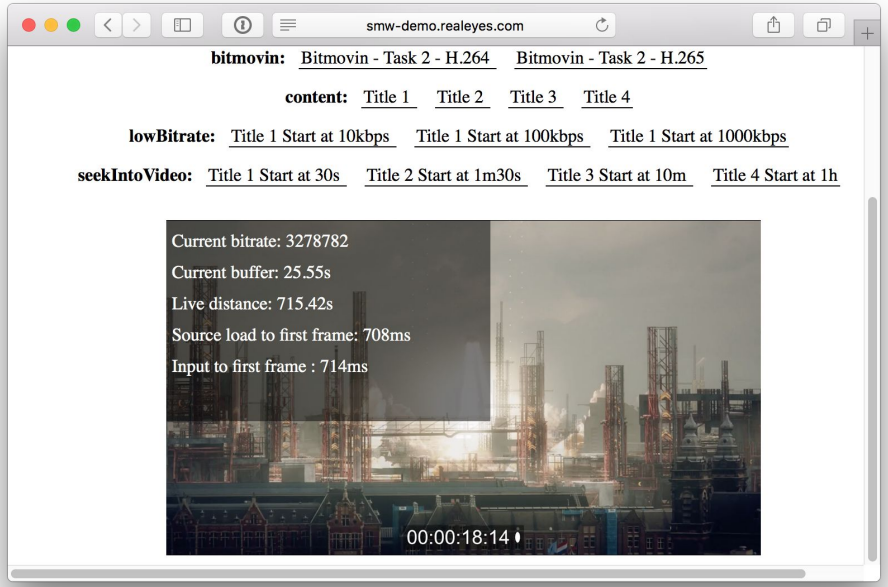
Baseline Startup -
All at 6 Second Segment/2 Keyframe

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

Screen Shots

(realeyes)





DELIVERY

30 Minutes or less or its FREE!

Delivery Standards

- Non-Persistent

- Persistent

Delivery Standards

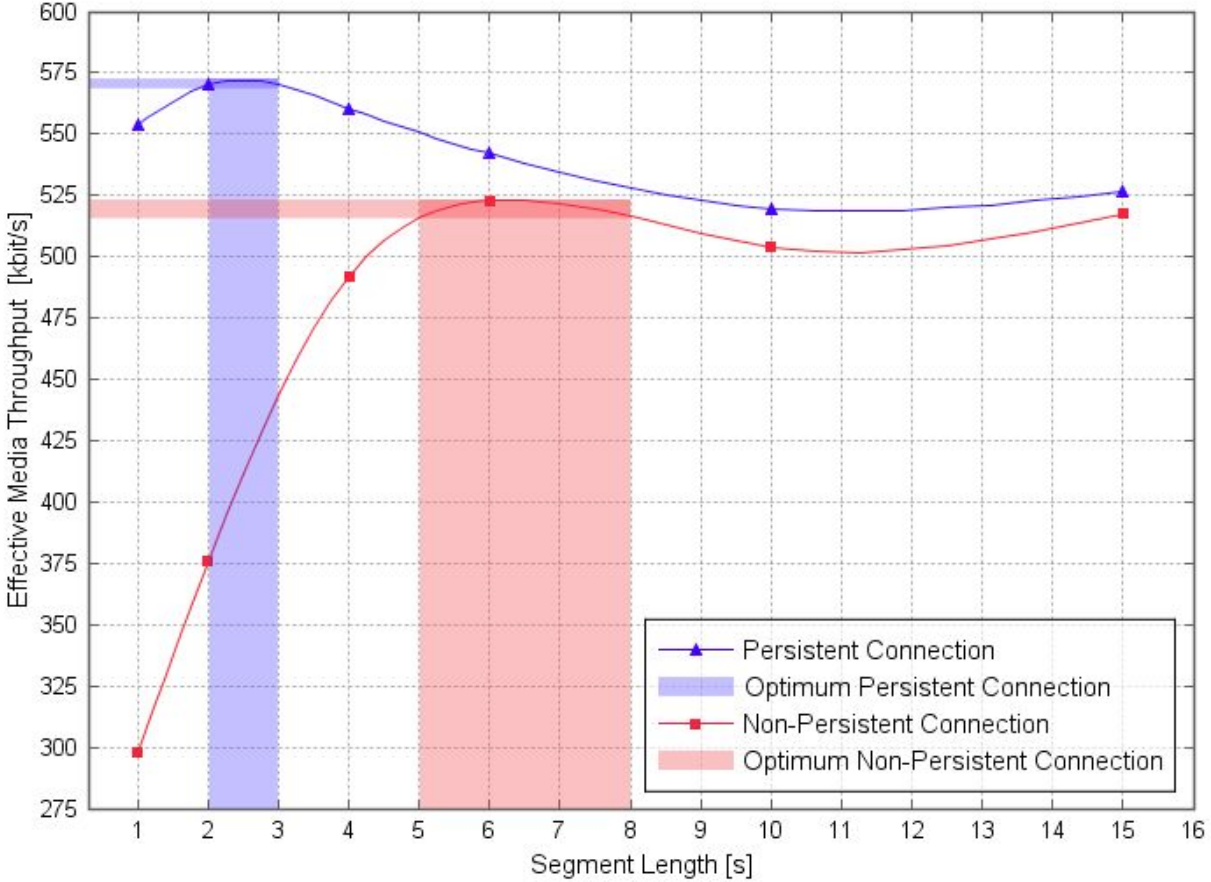
- Non-Persistent
  - Generally TCP
  - ByteRange & Chunking
    - HTTP 1.1

- Persistent

Delivery Standards

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

- Persistent
  - Web Sockets
  - WRTC
  - UDP
  - HTTP 2
  - SRT
  - QUIC


Effective Media Throughput [kbit/s]
600
575
550
525
500
475
450
425
400
375
350
325
300
275
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Segment Length [s]
Persistent Connection
Optimum Persistent Connection
Non-Persistent Connection
Optimum Non-Persistent Connection

(realeyes)

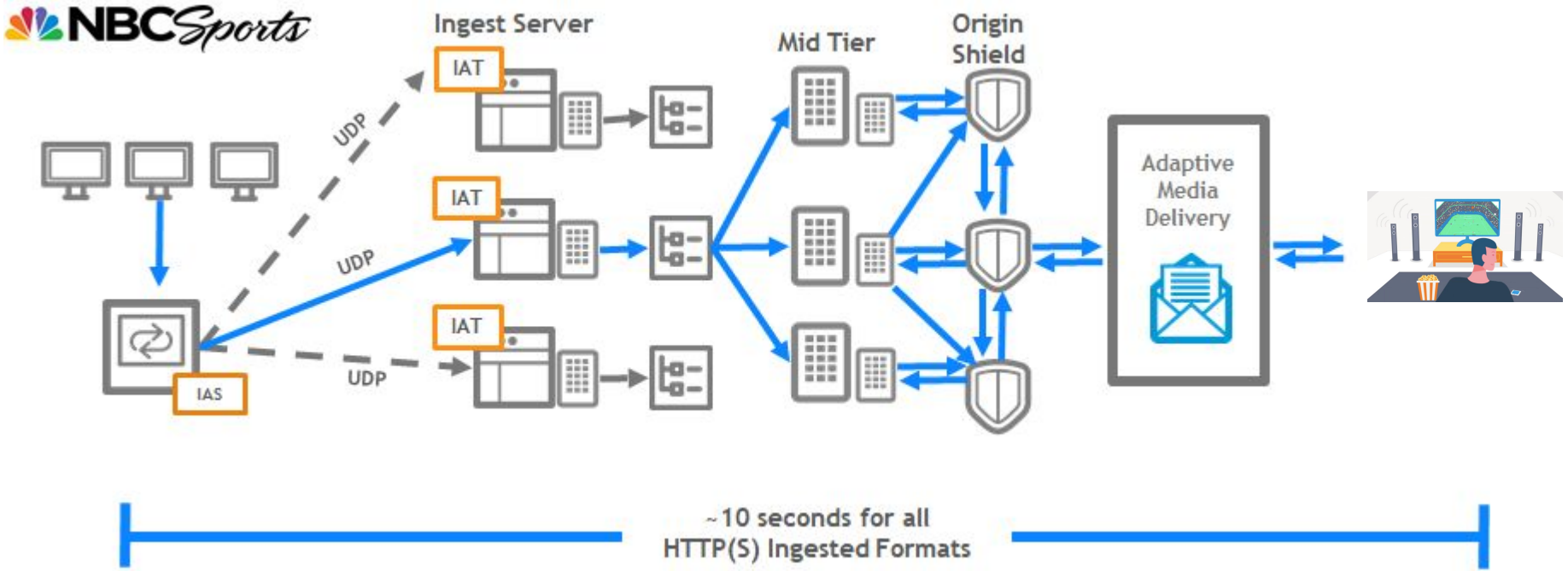
Delivery Standards

- 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**

Akamai
FASTER FORWARD

(realeyes)



The Solution – Media Services Live with **liveOrigin™** Capabilities

KEY DIFFERENTIATORS	BENEFITS
Media Ingest Acceleration	Allows for improved ingestion performance over the open internet to match broadcast quality
Self Healing	Brings the reliability and availability required for live 24/7 streaming content & large events
Low-Latency	Live content 1-2 seconds behind broadcast
Enhanced Monitoring & Alerting	Allowing customers to quickly identify and mitigate first mile issues
DVR and Archive	Provide end users ways to match the TV experience online

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

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

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

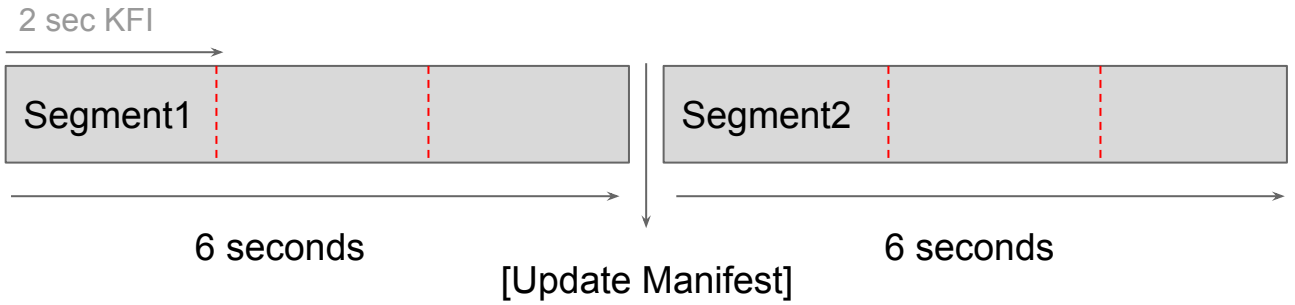
HTTP 1.1 Chunked Transfer Coding

- **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

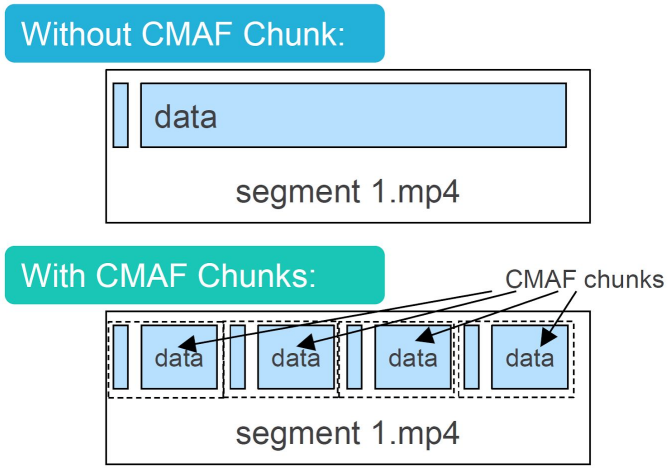
[TRANSPORT LAYER]

Encoding/Packaging Delay

- Content generation & notification delay



CMAF Chunks



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

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

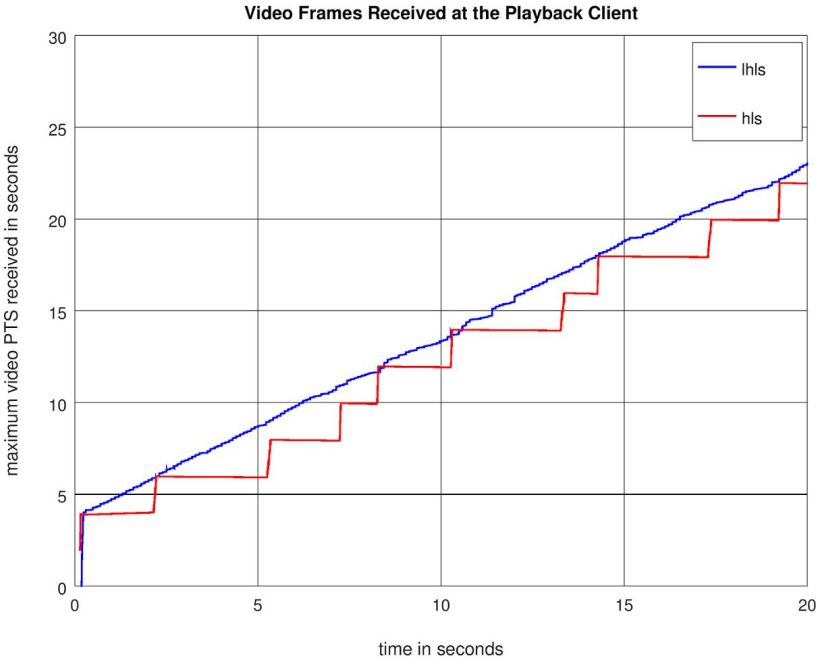
- Using predictive tags impacts ability for discontinuities & seg duration
- ABR calculation limitations
- Need an Origin Shield!!

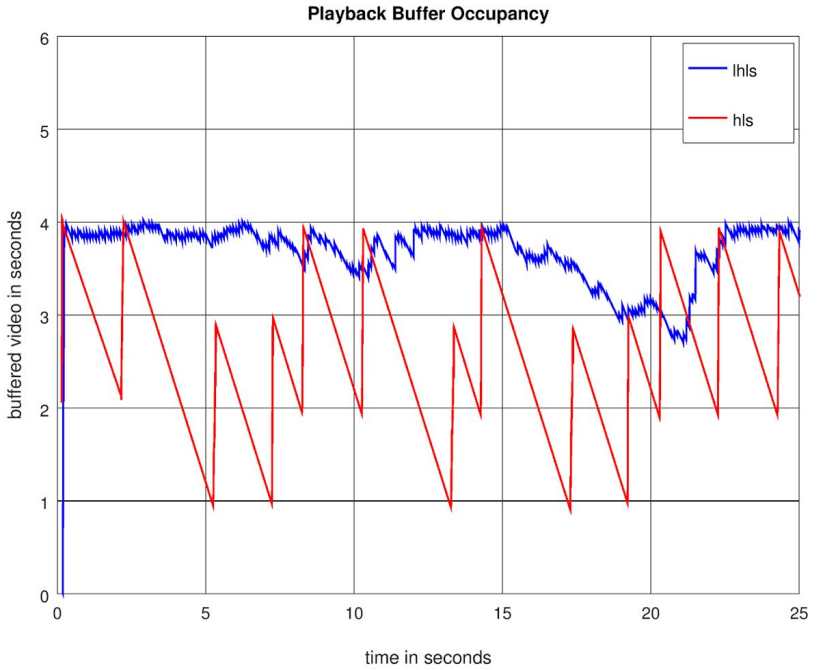
PERISCOPE: Low-Latency HLS (LHLS)

- 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/@periscopecode/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