Achieving Low-Latency Streaming At Scale

Introductions

Jamie Sherry

Senior Product Manager, Wowza Media Systems

Mike Talvensaari

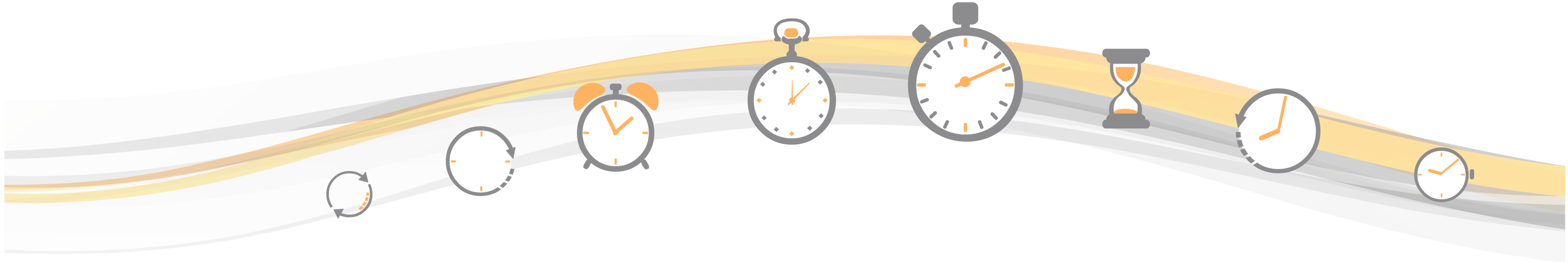
VP, Product & User Experience, Wowza Media Systems

WOWZA™
media systems

Founded in 2005, Wowza offers a complete portfolio to power today’s video streaming ecosystem from encoding to delivery. Wowza provides both software and managed streaming services for producers, developers and engineers to build unique streaming experiences for any device or audience size. Wowza Streaming Engine is Wowza’s flagship award-winning media server software. Wowza Streaming Cloud is an end-to-end live streaming service. Wowza also offers Wowza GoCoder a mobile SDK and free app for live streaming and Wowza Player, a modern HTML5 video player that is tightly integrated with other Wowza products.

What is Latency?

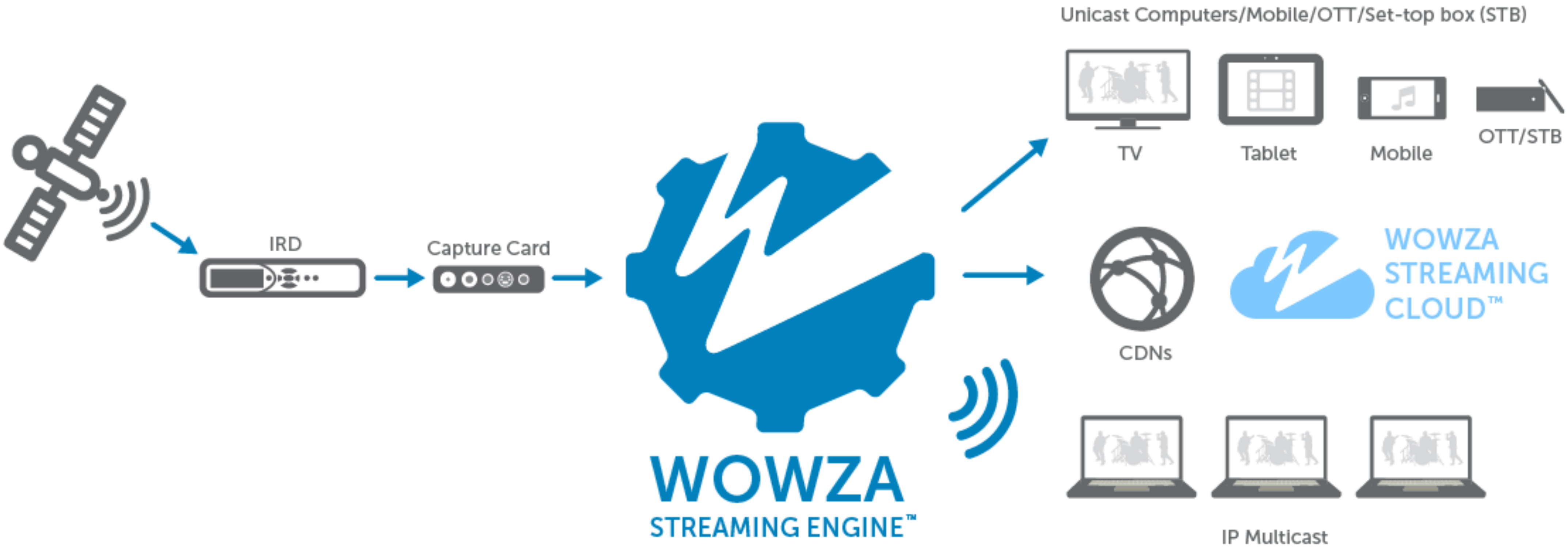
- **Latency** is a time interval between the stimulation and response, or, from a more general point of view, a time delay between the cause and the effect of some physical change in the system being observed.
- In terms of streaming, latency is the delay between the initial capture of the video and the viewer.



Other Related Terms

- Time to first frame - Time delay from when a person clicks the play button and when video appears
- Broadcast Delay – The practice of intentionally delaying the broadcast of live material to prevent profanity, bloopers, or violence (wikipedia)
- Quality - Higher quality = higher resolution = more data to send
- Scale - How many inputs/participants, how far, how many viewers
- Bandwidth - How much traffic can the infrastructure ideally handle
- Throughput - How much traffic is the infrastructure really delivering
- Bitrate - How many bits of data are being processed over time

Why is there latency?



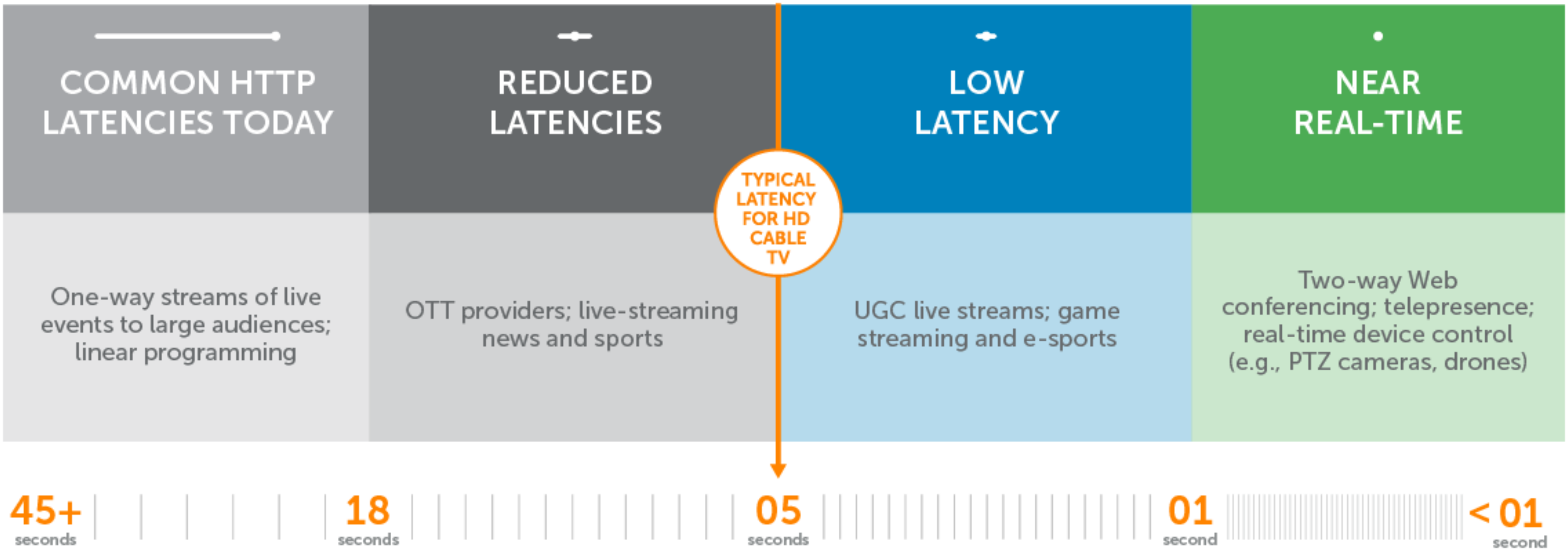
Q: Why is latency a problem?

A: Latency is not usually a problem – but people think it is.

- For many (or possibly most) live streams, latency does not matter
- HTTP streaming intentionally introduces latency for improved reliability.

- **For some live streams, latency is critical.**

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

TYPICAL LATENCY FOR HD CABLE TV

LOW LATENCY

UGC live streams; game streaming and e-sports

NEAR REAL-TIME

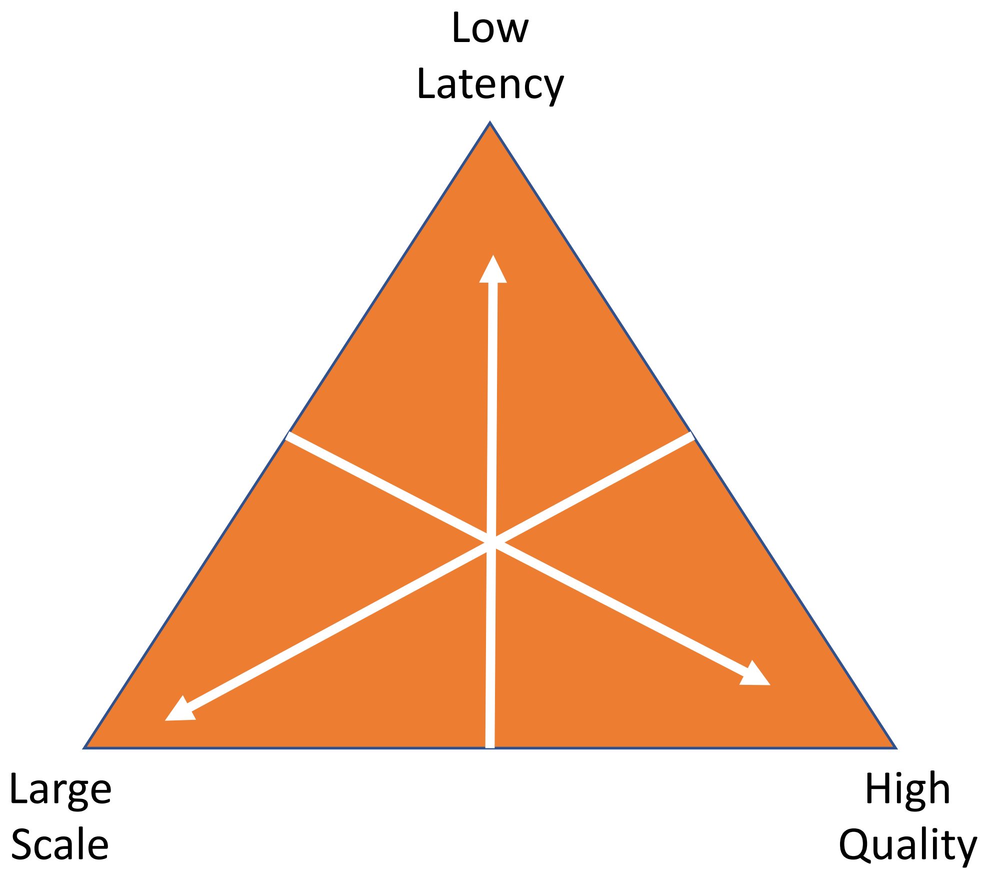
Two-way Web conferencing; telepresence; real-time device control (e.g., PTZ cameras, drones)

Demo

What Factors Impact Latency

Main factors affecting latency are:

1. Quality
2. Scale



As either (or both) increase latency increases

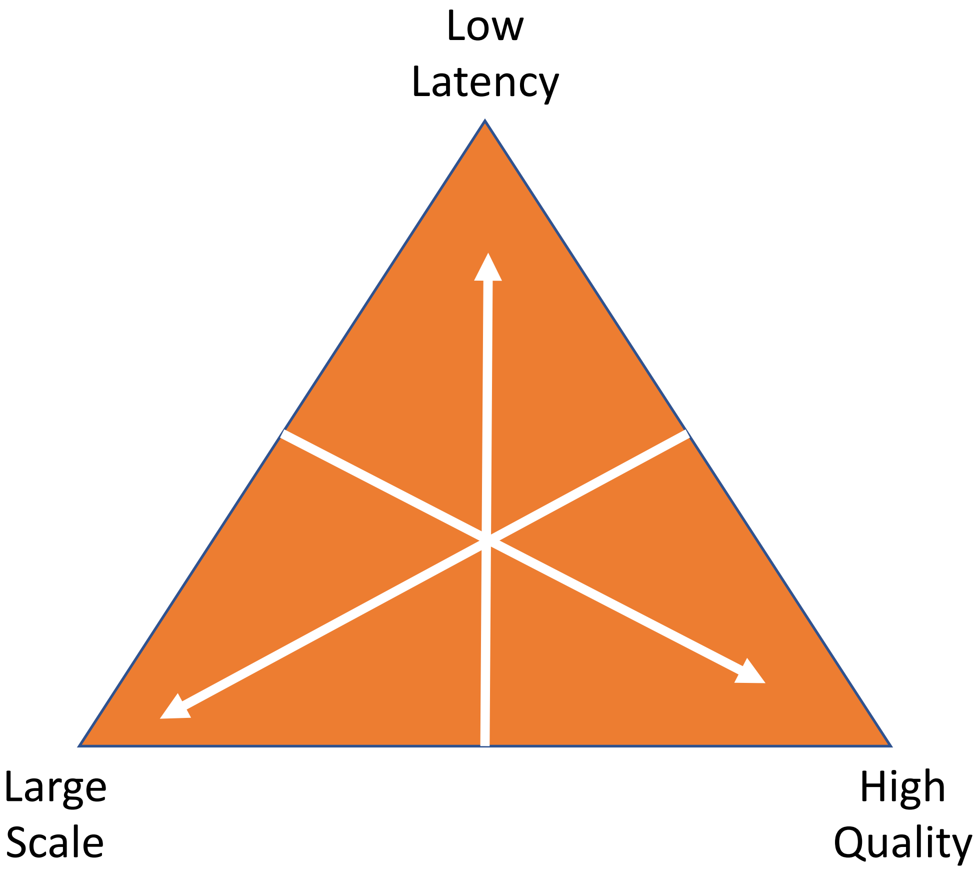
What Factors Impact Latency

Quality

- Resolution
- Two-way
- Multi-thread
- Frame rate
- Smooth playback
- Chunk/Block Size
- Buffering

Scale

- Distance
- Participants
- Viewers
- Streams
- Complexity
- Variability
- Diverse endpoints



How Quality Injects Latency

Higher quality = higher resolution = more data to send

- Resolution increases the amount of data in any frame, block, chunk or time slice. If we hold infrastructure constant as resolution goes up the window and latency to deliver a segment increases.
- Trade-offs: Some mitigation techniques

Reducing Buffer

Reducing the buffer requirement will reduce latency but it also makes network fluctuations more visible to the viewer – e.g. a 1ms buffer ideally means content can be viewed as little as 1ms after capture but a 0.1-second network interruption will result in 100 buffer segments being un-recoverable in a live stream.

Reducing Framerate

Reducing the (frame rate) number of frames per second reduces the smoothness of the viewing experience – e.g. 15 fps is generally a good viewing experience for low movement broadcasts like presentation screen capture/sharing or many church services, it will produce a choppy experience for viewing sports

Increasing Throughput

Increasing throughput with more bandwidth(network) and bitrate(compute) = More infrastructure, more overhead, more cost

How Scale Injects Latency

Scale increases distances, complexity and variability

- As distance, the number of streams, and viewers increase, network imperfections, variability, and degradation increase and can amplify latency.
- Trade-offs: Some mitigation techniques

Increasing Buffer

Increasing buffer size adds ability for both encoders, transcoders and players to deal with variability smoothly but increasing buffer inherently increases latency. By definition, buffer size is the minimum amount of data for which a process can be initiated. If more data must be accumulated, then processing will take longer.

TCP vs UDP

Because it is a lighter weight protocol without error checking, monitoring, order of messaging and headers roughly 1/3 the size of TCP, UDP is inherently faster but there is no guarantee data is received.

Increasing Throughput

Increasing throughput with more bandwidth(network) and bitrate(compute) = More infrastructure, more complexity, more overhead, more cost

Where streaming latency has been

- Windows Media
  - MMS, RTSP, HTTP (using UDP or TCP)
  - Encoder, Server, and Player buffer size management
  - “Low delay” audio codecs
  - Fast Start, Advanced Fast Start, Fast Recovery, Fast Reconnect, FEC
  - 2-3 seconds (on a good network)

- Real Time Streaming Protocol (RTSP)
  - Developed and by RealNetworks
  - Relies on RTP and RTCP
  - Frames can be sent one at a time in real time
  - Can leverage UDP
  - Potential latency could be as low as ~125 ms with minimal buffering (2-3 frames behind)

Where streaming latency has been

- Real Time Messaging Protocol (RTMP)
  - Developed and open sourced by Adobe
  - 1-3 seconds and sometimes below 1 second

- HTTP
  - Apple HLS
    - 30+ seconds on iOS (using 10 second chunks)
  - MPEG-DASH
    - 10-20 seconds (variable)

Where latency is going

- WebRTC
  - Designed for real-time audio, video and data delivery over less-reliable connections
  - Leverages TCP or UDP
  - Multiple protocols related to RTSP/RTP
  - 1 second or less (as low as 200 ms)

- WebSocket
  - Designed to provide a standardized, two-way, reliable communications channel between a browser and a server
  - Works with TCP
  - Can be used with other streaming protocols including RTMP, WebRTC, Haivision SRT, Wowza WOWZ and Aspera FASP
  - As low as 200 ms

Where latency is going

- HTTP
  - Reducing to 4 seconds (using 1 second chunks) for DASH and below 8 seconds (using 2 second chunks) for HLS (includes GOP adjustments as well)
  - CMAF + HTTP/1.1 (using HTTP Chunked Transfer Coding) enables video transfer, decode, and display before the end of the chunk encoding (“chunks of chunks”)
  - Optimizing DASH MPD attributes (availabilityStartTime and minBufferTime)
  - Video encoding enhancements that do not impact decoding (H.264 GDR)

Where latency is going

- Quick UDP Internet Connections (QUIC)
  - Uses UDP (with TCP fallback)
  - Focuses on security and reliability
  - Tries to be like TCP while reducing connect and round trip times, packet loss, congestion control, and more using intelligent retransmissions and storage and delivery of information
  - Built with HTTP/2 in mind

- Secure Reliable Transport (SRT)
  - A video transport protocol that enables the delivery of high-quality and secure, low-latency video across the public Internet
  - Designed to deliver the best quality live video over the worst networks
  - Accounts for packet loss, jitter, and fluctuating bandwidth while maximizing quality

Where latency is going

- Wowza Streaming Cloud low latency API-only preview
  - Using Wowza WOWZ + WebSocket in an origin-edge architecture
  - Sub 3 seconds end-to-end (origin to player)
  - Highly scalable

Reducing workflow component latency

- You need to think about it at every step
  - Content creation (codec, bitrate, resolution)
  - Streaming workflow and devices
  - Buffer management (encoder, player)
  - Network considerations
    - How you optimize delivery (protocol, transport layer)
    - How you reach your audience (size, location, and quality)
- WebRTC and WebSocket can scale using traditional origin/mid-tier/edge configurations (still requires “aware” clients)
- CDNs are investigating options to enable low latency through their networks
  - “Tuned” HTTP is a front runner

Measuring latency

- This is not easy (especially at scale)
- Latency is commonly measured in seconds (or milliseconds) though some prefer frames
- A visual test is easy but requires accurate clocks (synchronized using “reliable” NTP servers)
- For “traditional” streaming protocols, markers in the stream can be used

Options for low latency at scale

- Build your own
  - Encoder
  - Server (origin/edge)
  - Player
- Buy
  - Akamai
  - Agora.io
  - Wowza Streaming Cloud (preview)

Questions?

Thank you!