Aalto University
School of Science

CS-E4260 Multimedia Services in Internet

Live streaming & Interactive multimedia

Matti Siekkinen & Gazi Karam Illahi

Outline

- Live video streaming
  - Live video delivery
  - Protocols
  - Latency in video streaming
- Cloud Gaming and Cloud VR
  - Cloud gaming
  - VR
- Conclusions

Live video delivery

- Live: Video created and streamed simultaneously

- Live streaming is popular
  - Viewers watch 3 billion hours of live video per month (2021)
  - Single Euro Cup stream (2021) = 17.5Tbps on average over Akamai
  - Facebook, Twitter, Snapchat, LinkedIn, Instagram all offer mobile live streaming-> also called social live streaming,
- Typical Applications
  - Sports/public event streaming: Broadcast events
  - Video game live streaming: Twitch, YouTube Gaming Live, Facebook Gaming.
  - Mobile Live Streaming: Facebook Live, Periscope, Instagram Live
  - Video Chat: FaceTime, Zoom, WhatsApp, Teams/Skype

Applications

- Live Event Streaming: Sports Events, Political Events, News
  - Video origin is, for example, a studio
    - Usually, high-capacity uplink available
    - Origin system is not resource constrained (usually)
    - Somewhat latency tolerant
    - Disparate viewing devices
- Video Game Play Streaming
  - Video origin is usually a gaming computer
    - Resource constrained (render game+ encode video, in real time)
  - Disparate viewing devices
- Mobile Live Streaming
  - Video origin is usually a mobile device
    - Very resource constrained
  - Disparate viewing devices
- Video Chat and Conferencing
  - Disparate video origin and destinations
  - Mix of resource constrained and well provisioned nodes

Live video delivery

- Extra challenges compared to video on-demand (VoD)
  - Limited caching
  - End to end latency
  - Workload variety (e.g., mega-events vs. Twitch channels vs. bi-directional video)

- Goals are similar to VoD
  - Users want good QoE
    - New dimension added to QoE: Latency
  - Service providers and CDNs want to meet user demand and minimize delivery cost

Outline

- Live video streaming
  - Live video delivery
  - Protocols
  - Latency in video streaming
- Interactive multimedia
  - Cloud gaming
  - VR and 360° video
- Conclusions

Protocols:

- Broadly two classes of protocols: Stateless & Stateful
  - Recall lectures on video streaming

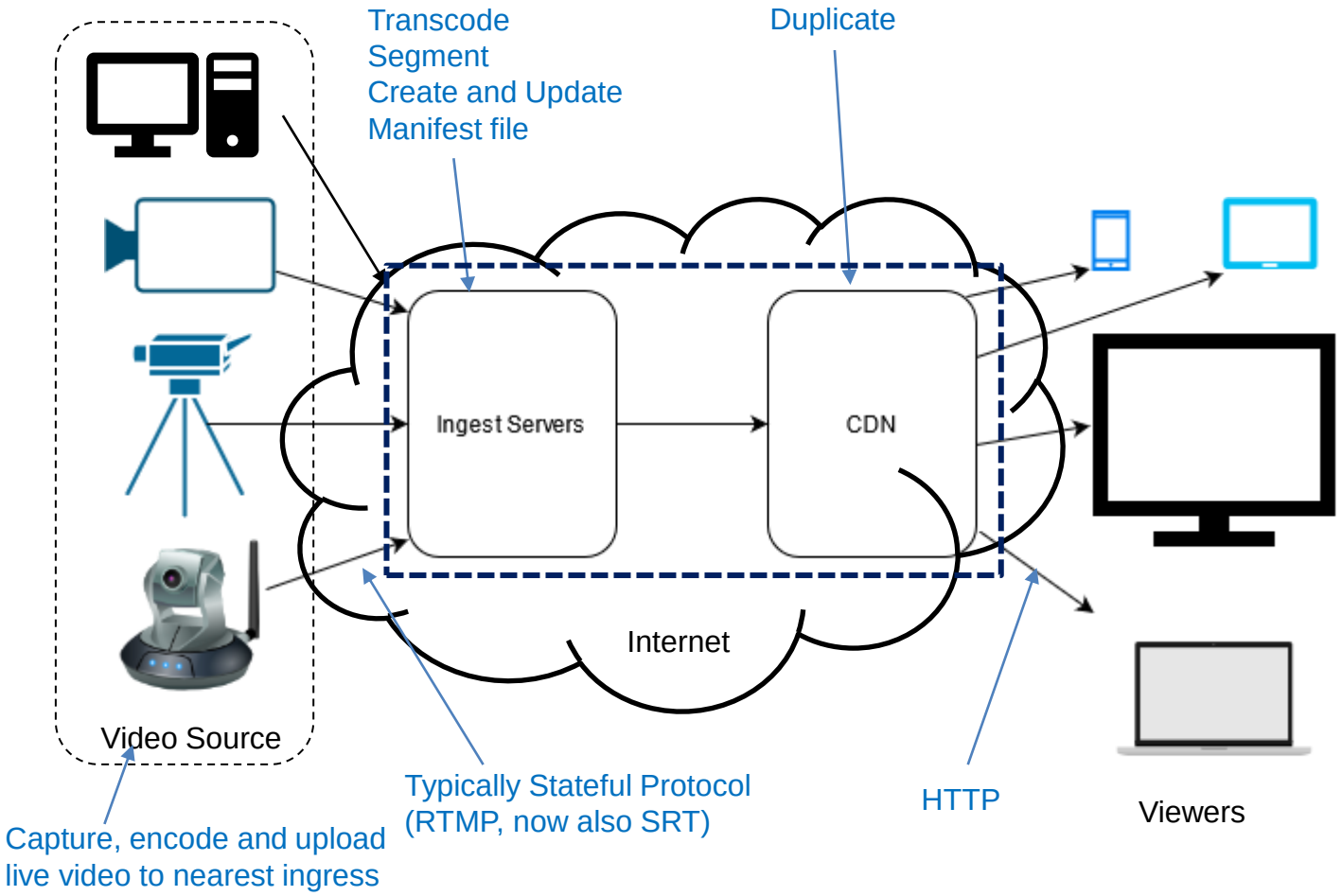
- Stateless Protocols are HTTP based, same as VoD
  - MPEG-DASH, HLS, MSS, HDS
  - Server does not store media session information of the client
  - Scalable: use the same infrastructure as VoD
  - Can be used in event live streaming, video game live streaming
  - Latency of the order of 10s of seconds
    - Low Latency versions of DASH and HLS reduce latency to the order of seconds

Protocols:

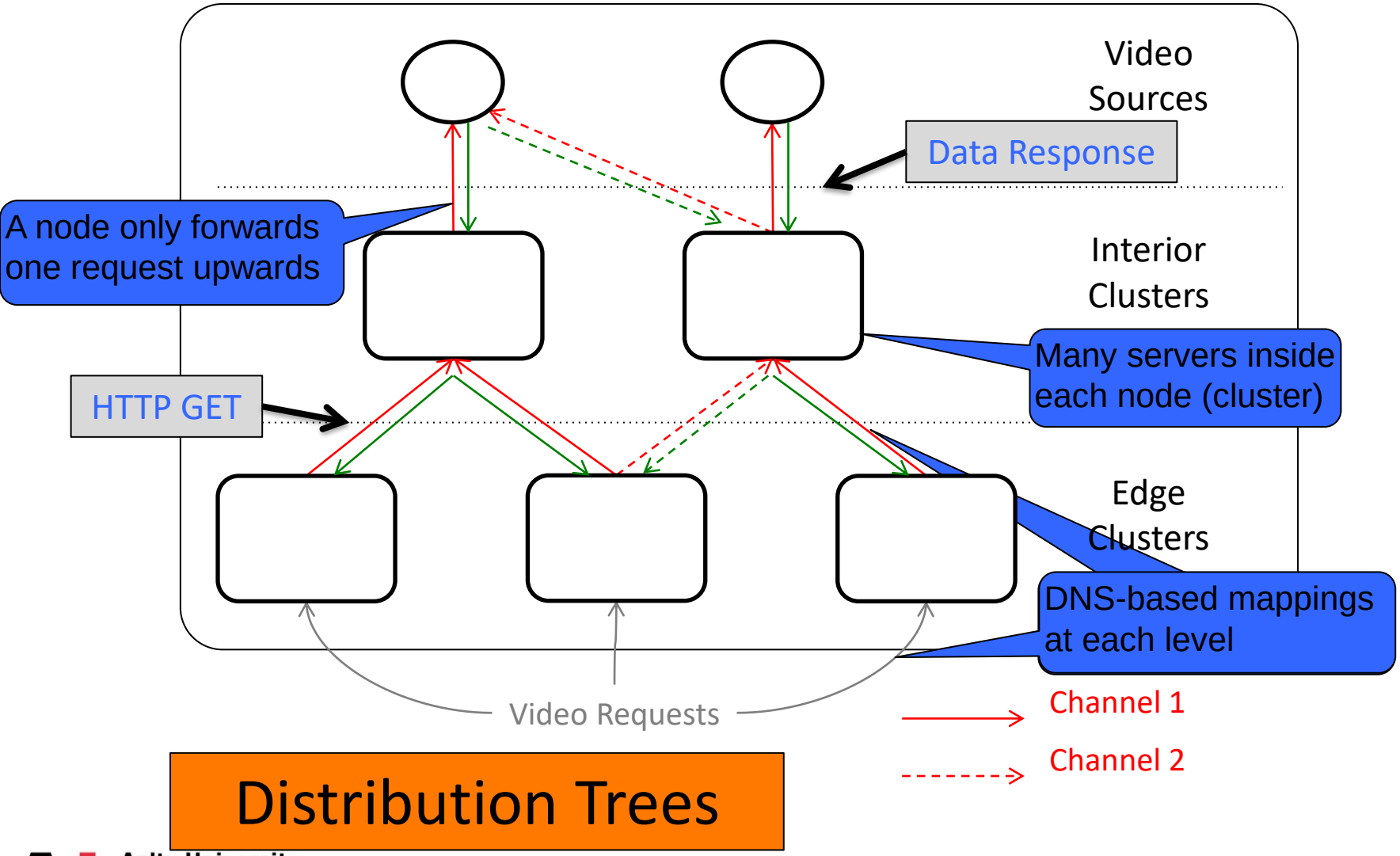
- Broadly two classes of protocols: Stateless & Stateful

- Stateful Protocols: RTMP, RTSP, WebRTC, SRT
  - Server and client both store “state” of the streaming session
  - Not easily scalable, saving and processing state of thousands or more sessions can be expensive for the server.
  - Can use either TCP or UDP as the transport protocol
  - Sub second latency possible
  - Useful for latency constrained applications like conversational video, cloud gaming etc.

HTTP Live Video:



HTTP live video with CDN (Akamai)



Distribution Trees

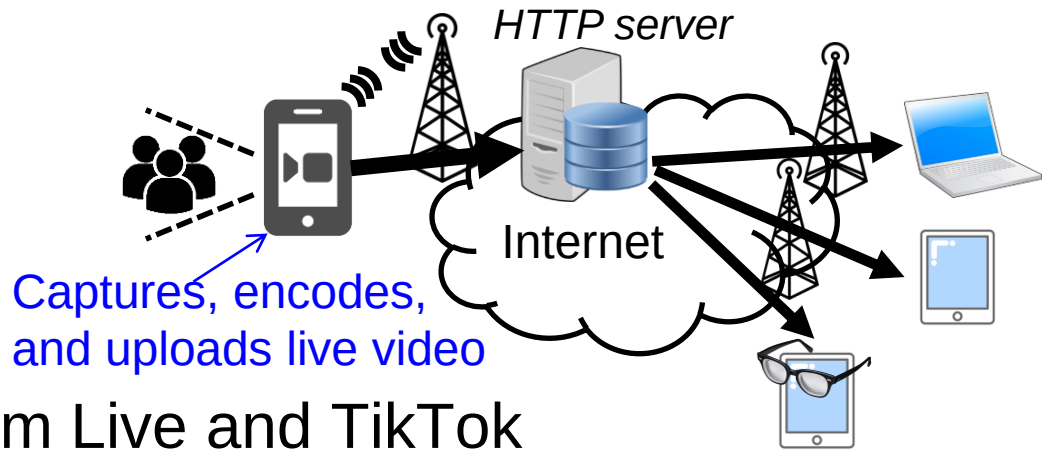
HTTP Live video vs. VoD with CDNs

- Both use HTTP
  - Otherwise, CDN is not used
  - Can use stateful streaming protocols (e.g., RTMP/SRT) for ingress
  - Client to edge cluster/server mapping can be similar
  - Recall, e.g., the Akamai mapping system (CDN lecture)

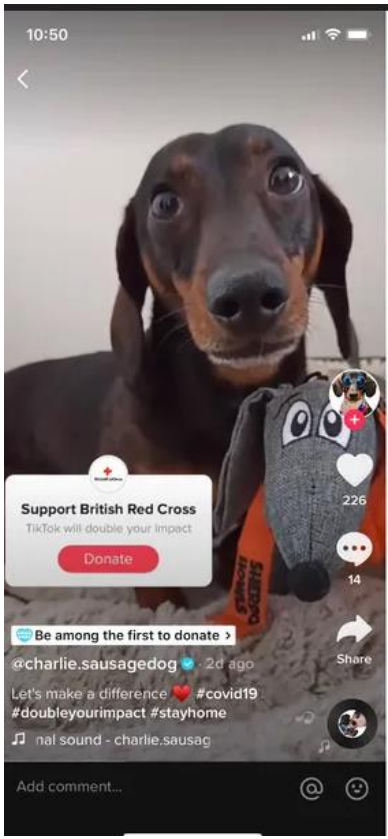
- Distribution trees do not exist with VoD as such
  - Content can be cached at the edge → not very useful with live video
  - Trees can be optimized [1]
    - How to map internal clusters to upstream clusters

[1] M. K. Mukerjee et al. *Practical, Real-time Centralized Control for CDN-based Live Video Delivery.* In ACM SIGCOMM 2015

Mobile live video streaming



- Facebook Live, Instragam Live and TikTok Live example services
  - Personal video broadcasting with mobile devices
- Special class of live streaming
  - Constrained device also the source of video
  - Bitrate adaptation requires transcoding in the upload server
- Latency even more important
  - Live feedback from viewers: text chat and “hearts”
  - Long latency between video and feedback hurts user experience



SOURCE: TikTok via The Verge

Mobile live video broadcasting

- **Stateful** protocol for ingest streams
  - Typically, RTMP(S)
  - Connection to e.g., geographically closest server to broadcaster
  - Provides low latency but does not scale to outbound streams

- **HTTP** (HLS/DASH) to broadcast to viewers
  - Delivery using a CDN
    - Ingest RTMP(S) stream repackaged into HTTP(S)
  - Low latency HLS/DASH used (chunked transfer encoding)

Conversational Video Streaming

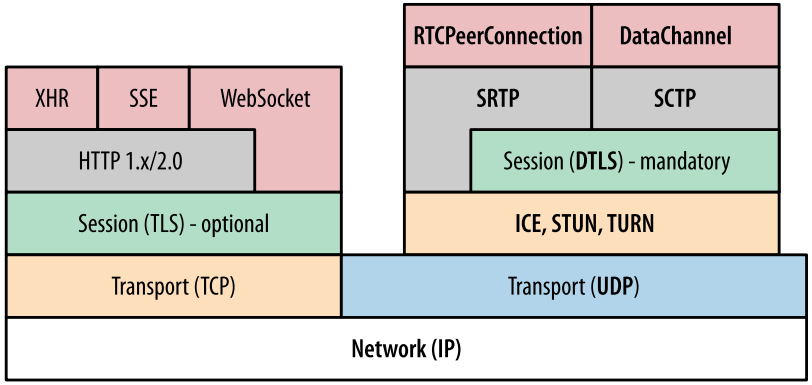
- Bidirectional Video Streaming
- 100ms -500ms acceptable delay reported [1][2]
- Constrained devices as end points
- May have intermediary transcoding and/or signalling server
- WebRTC is the de-facto standard open protocol
  - Proprietary protocols like Skype, Zoom exist.
- Special case in LTE/5G
  - Multimedia Telephony Service for IMS (MTSI) [3]
    - Possible support for WebRTC like data channels in the IMS

1. Baldi, M. and Ofek, Y.. End-to-end delay analysis of videoconferencing over packet-switched networks. IEEE/ACM Transactions on Networking (TON (2000) vol. 8 (4), pp. 479-492. ACM, New York, NY, USA. DOI=10.1109/90.865076
2. Jennifer Tam, Elizbeth Carter, Sara Kiesler, and Jessica Hodgins. 2012. Video increases the perception of naturalness during remote interactions with latency. In Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts (CHI EA '12). ACM, New York, NY, USA, 2045-2050. DOI=10.1145/2223656.2223750
3. 3GPP TS 26.114 V17.1.0

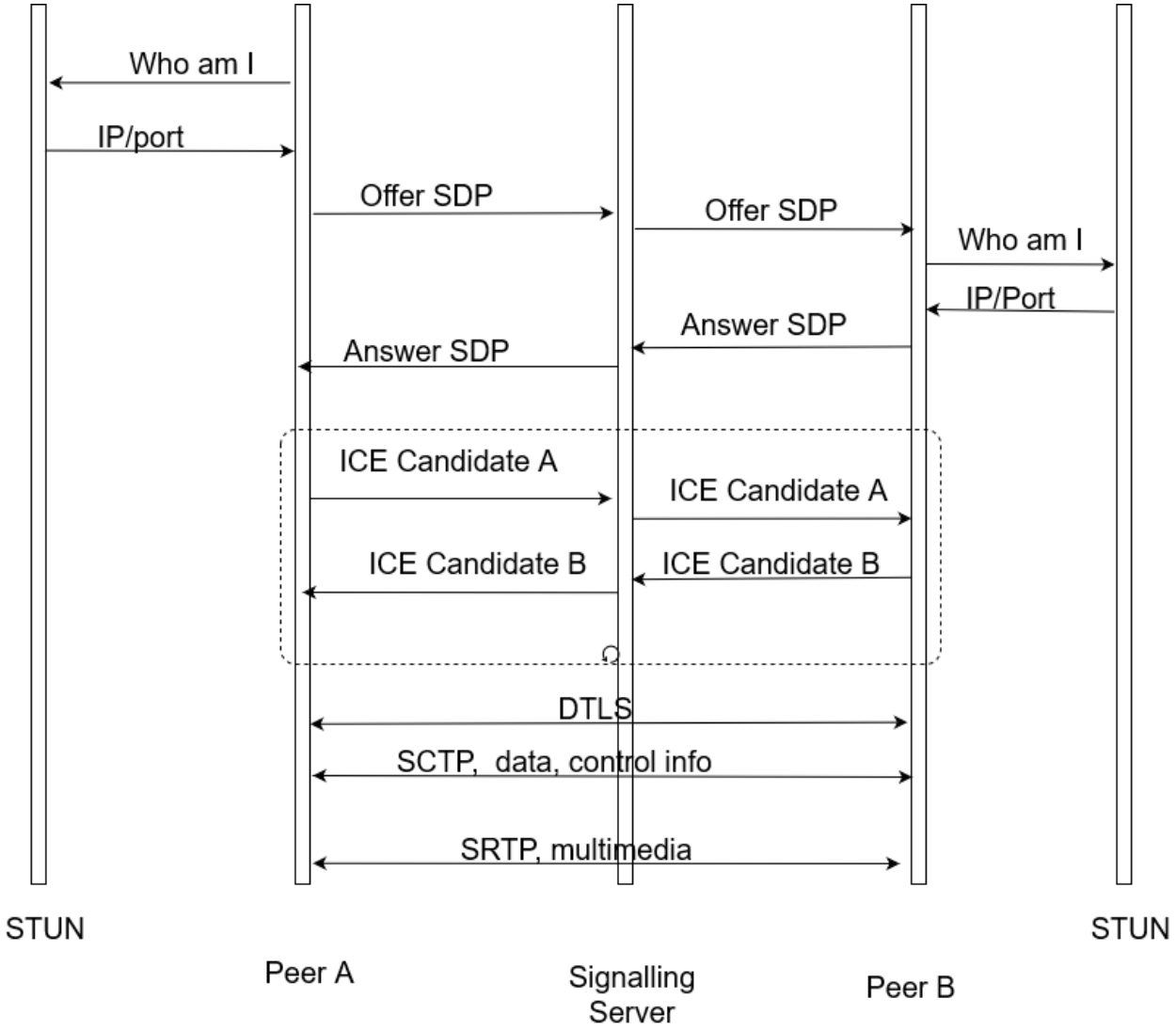
WebRTC

- WebRTC-> A suite of protocols for Peer-to-Peer RTC
  - Signal, Connection, Security, Communication
  - Signal: Exchange Control & Session Information
    - WebRTC defines what to exchange and in which format, but not what communication protocol to use.
    - SDP: out of band
  - Connect: ICE, NAT Traversal with STUN/TURN
  - Security: DTLS and SRTP
  - Communication: RTP (secured with SRTP) and SCTP (secured with DTLS)
    - RTP typically uses UDP for transport->Low latency
    - Possibility of real time bitrate adaptation->react to network and end device conditions
- Widespread support, Chrome, Firefox, Safari, Edge, all support WebRTC -> Enabling browser to browser RTC
- Rate/Congestion control in built
- Also becoming protocol of choice

for interactive multimedia



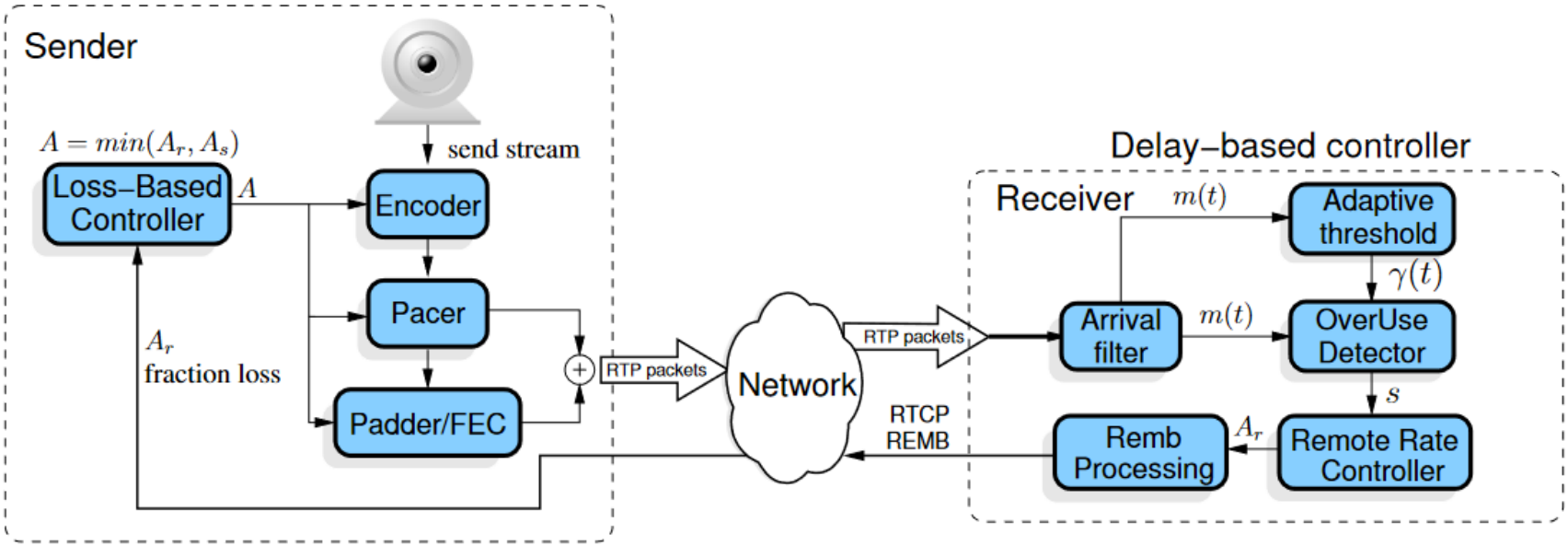
WebRTC: Session



WebRTC

- Jitter Buffer
  - Out of order packets/frames
    - Can increase latency
- Rate Control
  - Use RTCP: TMMBR, TMMBN and REMB Network Status messages
    - Temporary Maximum Media Stream Bit Rate Request - A requested bitrate for a single SSRC (synchronization source).
    - Temporary Maximum Media Stream Bit Rate Notification - A message to notify that a TMMBR has been received.
    - Receiver Estimated Maximum Bitrate - A requested bitrate for the entire session.
  - Transport Wide Congestion Control (TWCC)
    - Receiver sends timing information of received packets
    - Sender compares the received timing info with its own records of transmission timing
    - Sender estimates network conditions-> estimate bitrate
- Congestion Controller: Plug in algo
  - IETF RTP Media Congestion Avoidance Techniques (RMCAT)
    - Google Congestion Control (most deployed)
    - NADA: Network Assisted Dynamic Adaptation
    - SCReAM - Self-Clocked Rate Adaptation for Multimedia

WebRTC: GCC


Sender
send stream
$A = min(A_r, A_s)$
Loss–Based Controller
A
Encoder
Pacer
Padder/FEC
A_r
fraction loss
RTP packets
Network
RTP packets
RTCP
REMB
Delay–based controller
Receiver
$m(t)$
Adaptive threshold
$\gamma(t)$
Arrival filter
$m(t)$
OverUse Detector
s
Remb Processing
A_r
Remote Rate Controller

What is 360° video?

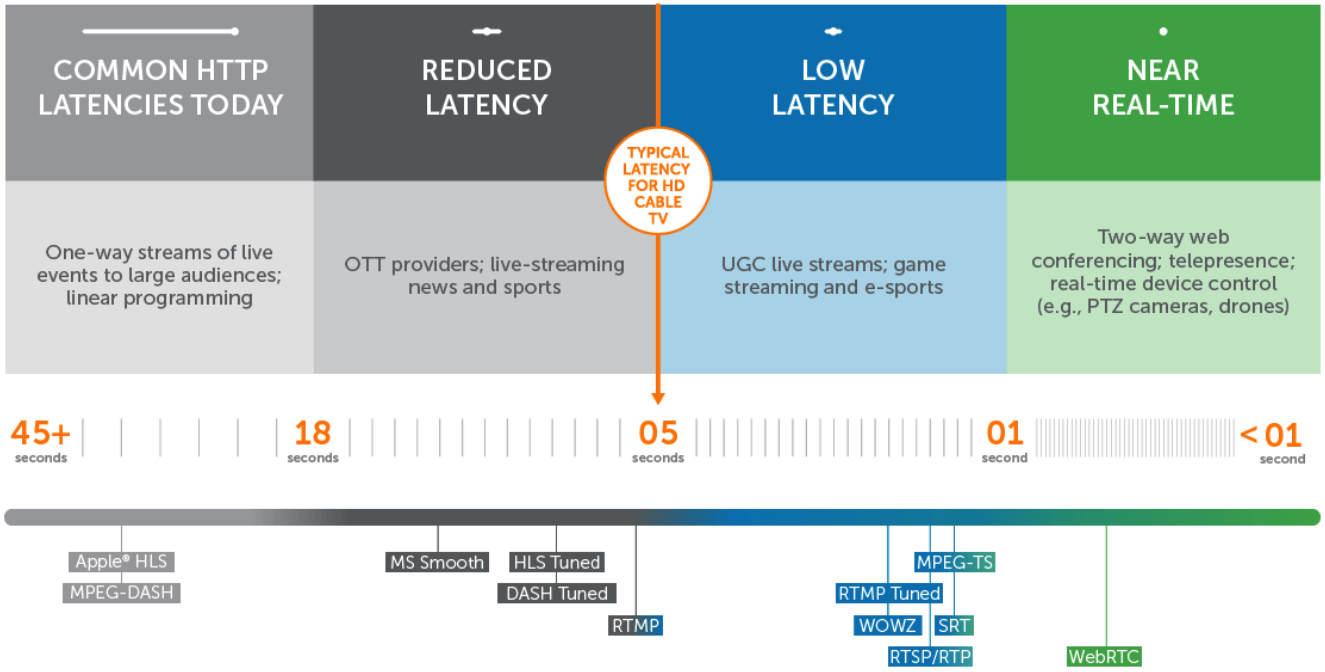
- Video with possibility to look into every direction
  - Recorded with dedicated 360 camera → ready 360 video content
  - Use multiple cameras with overlapping views → stitching to create 360 video
  - Monoscopic or stereoscopic (requires more than one lens/camera)
- Pre-recorded and encoded content transmitted to HMD
  - No rendering, only video decoding → smartphone with headset suffices
    - Motion sensors delay must be short when tracking head movements
  - User navigates the video with help of head tracking
- Static content, no interaction
  - User input does not cause new scene/objects to be rendered
  - Only FoV navigation within the video
- Can be streamed over HTTP e.g., using MPEG-OMAF representation

Outline

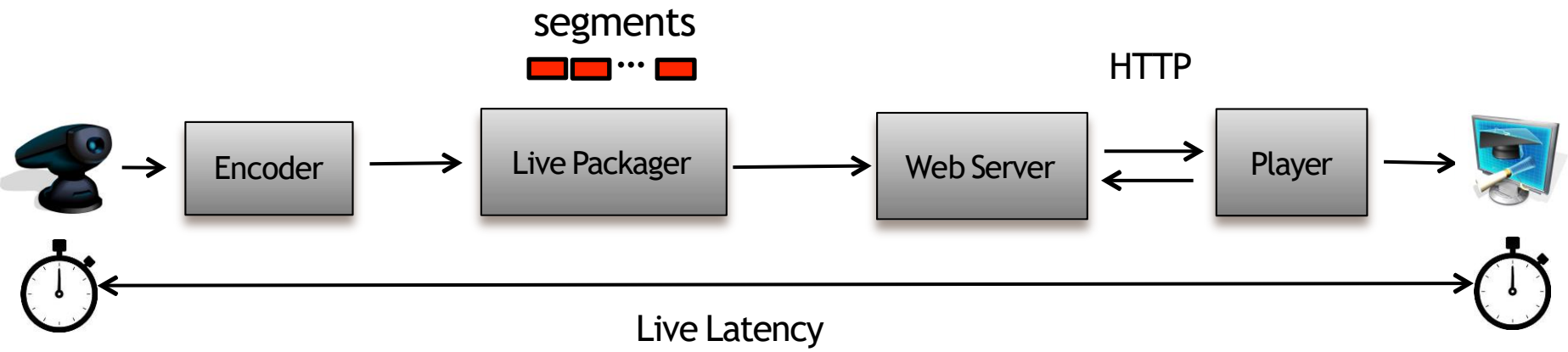
- Live video streaming
  - Live video delivery
  - Protocols
  - Latency in video streaming
- Cloud Gaming and Cloud VR
  - Cloud gaming
  - VR
- Conclusions

Latency

- Protocol has massive effect on latency
  - Different resource requirements
- Latency needs depend on applications
- Trade-off between resource requirements vs latency

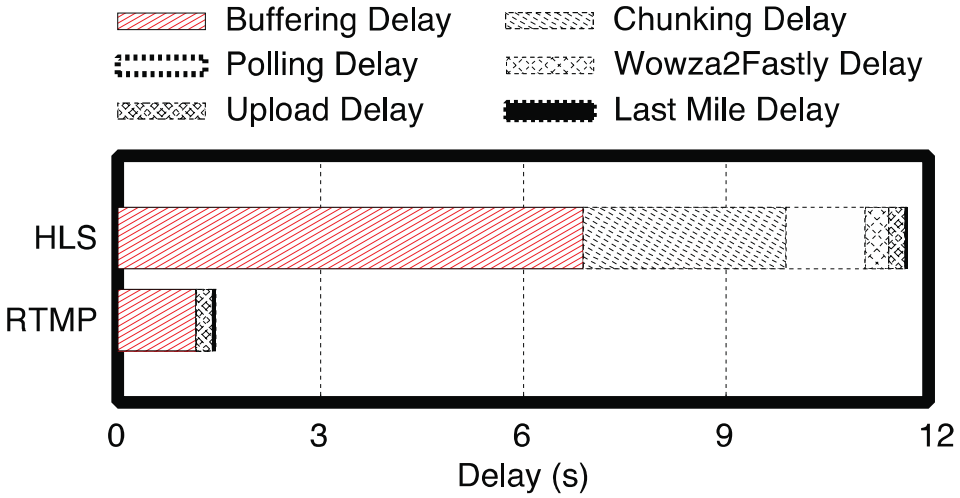


HTTP streaming latency (1/2)



- Sources of latency
  - Encoding/Decoding: usually not so much
  - Network transmission: depends on client, source locations
  - Playback buffering: usually desired to avoid stalls
  - Segmenting: wait until whole segment is complete

HTTP streaming latency (2/2)



- HLS/DASH segment duration is several seconds
  - Buffering is also a function of segments: min. buffer 1 segment → add 1/2s
  - RTMP operates on individual frames
- HLS/DASH is pull, RTMP is push
  - HLS needs to periodically check for new chunks

[1] B. Wang et al: *Anatomy of a Personalized Livestreaming System.* ACM Internet Measurement Conference (IMC). 2016.

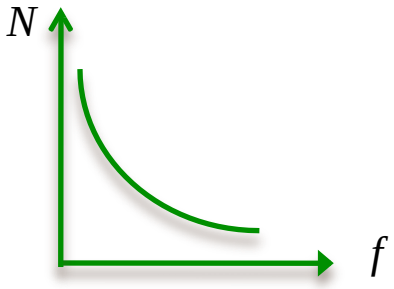
Can't we simply reduce segment size with HTTP streaming?

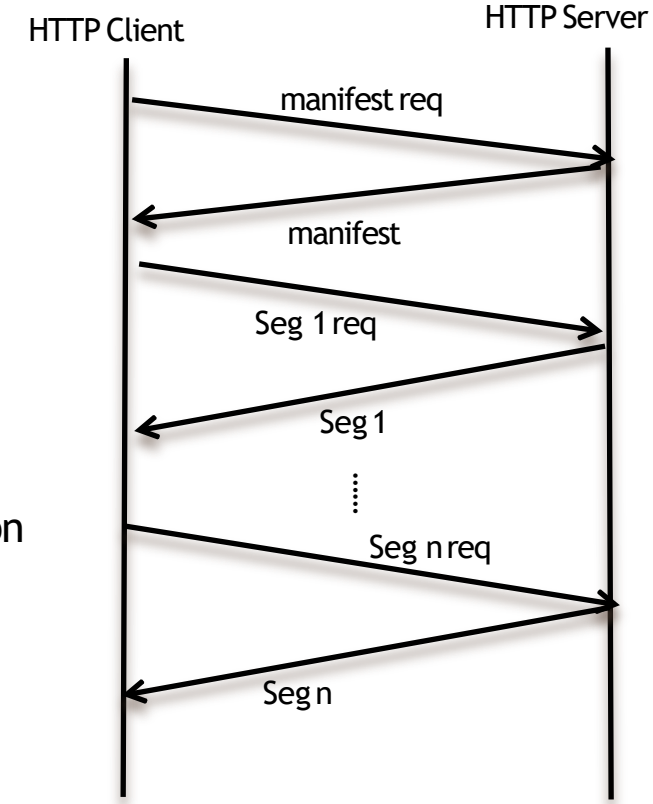
- Small chunk encoding overhead
  - Normally all chunks begin with a key frame
  - Smaller chunk → higher overhead
- Request explosion problem

D - duration of the video f - segment duration

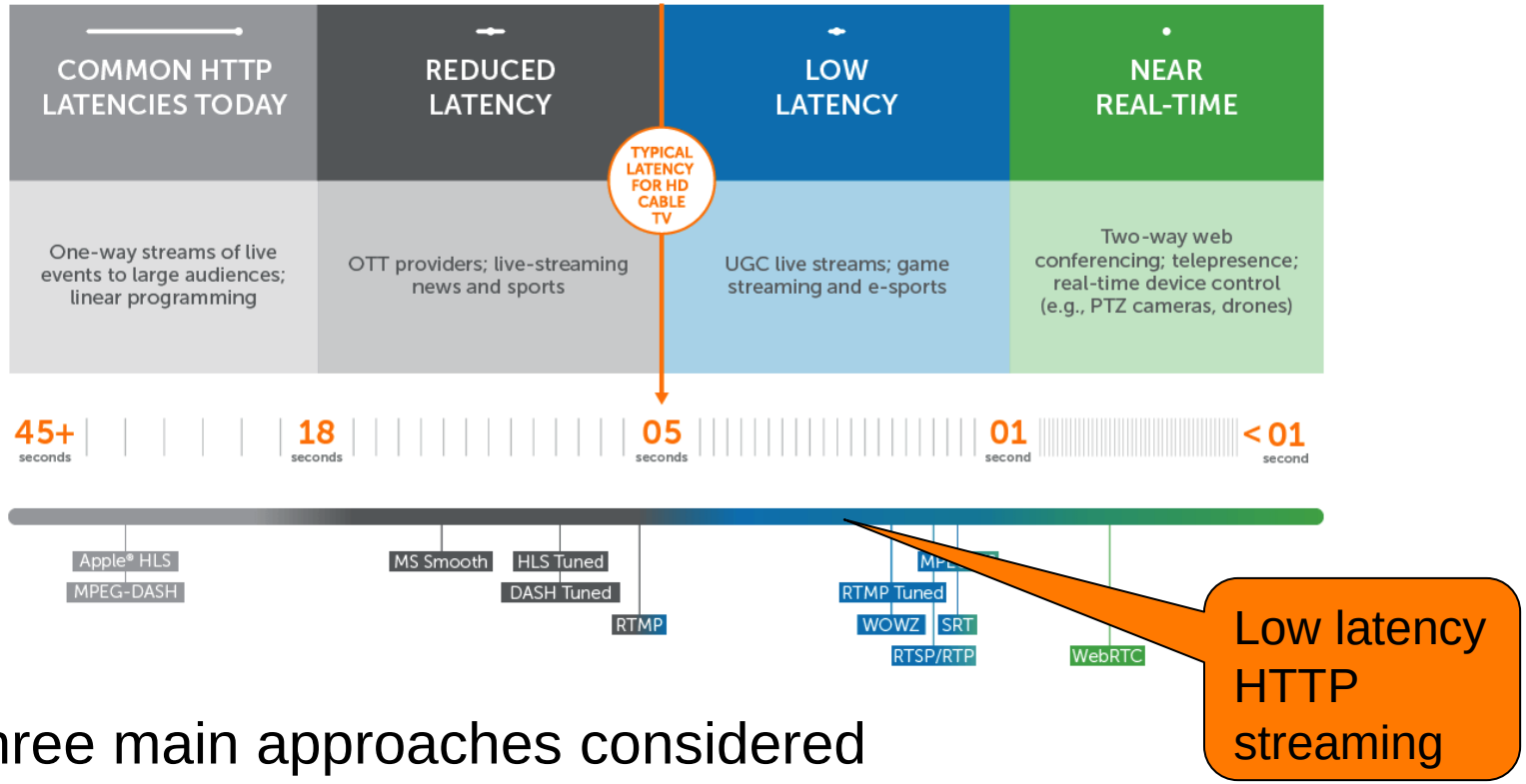
N - Number of requests:

$$N = D / f \;\; \alpha \;\; 1/f$$





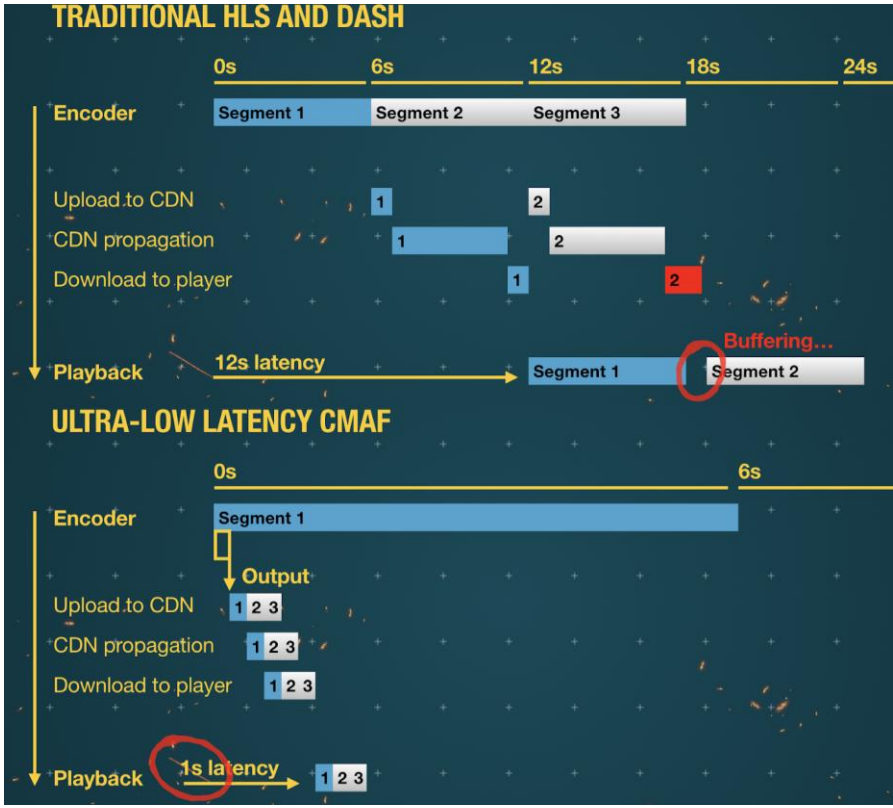
How to achieve low latency in HTTP

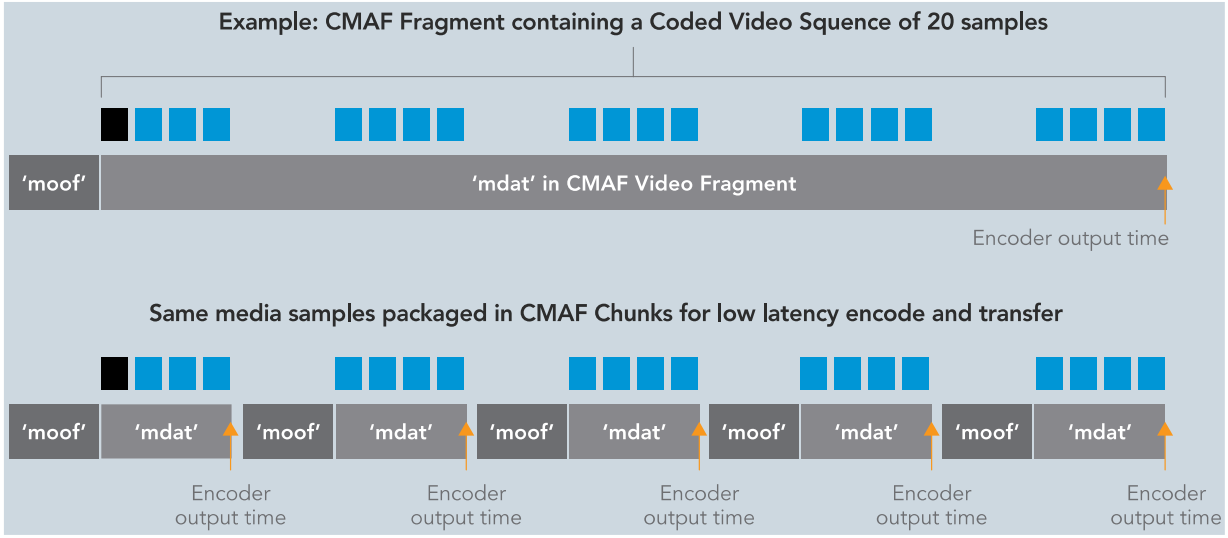


- Three main approaches considered
  - CMAF and HTTP 1.1 chunked transfer encoding
  - HTTP/2 server push
  - HTTP/3 (HTTP over QUIC)

CMAF and chunked transfer encoding

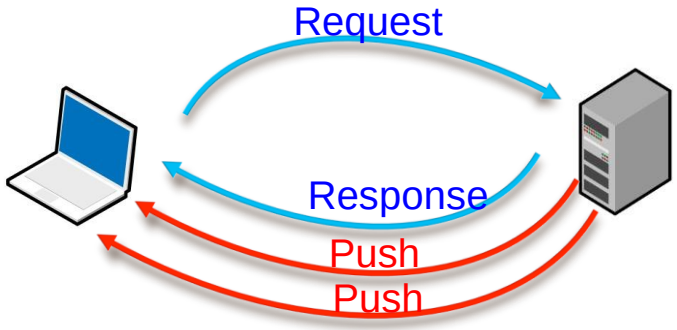
- Common Media Application Format (CMAF) standard
  - Enables efficient chunking of segments
- Chunked transfer encoding
  - Enables transferring parts (chunks) of segment
  - Introduced already in HTTP 1.1





HTTP/2 server push

- HTTP/2.0
  - First major revision to HTTP 1.1 (-97)
  - Support: most browsers, many CDNs, over 10% of websites
  - Relevant new feature: Server push



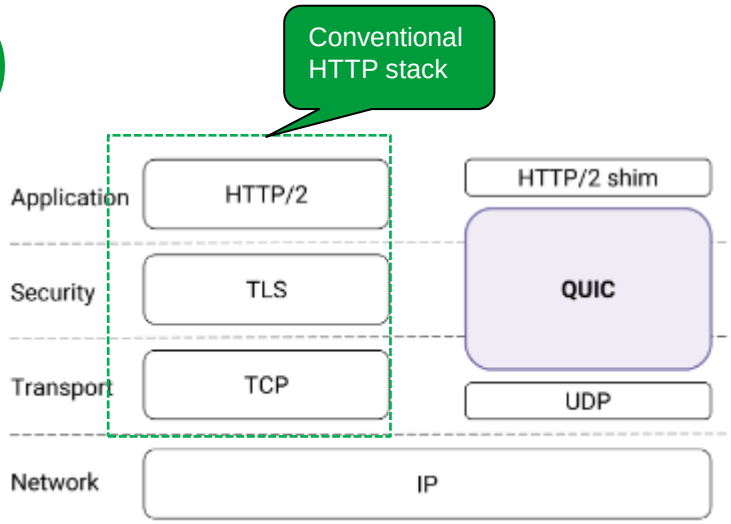
- Server allowed to push objects without request per object, unlike HTTP 1.1
- Apple's Low Latency HLS took this approach (WWDC19)
  - Partial segments ("Parts") → new syntax to HLS playlist format
  - Now supports non push based LL as well

HTTP/3 (HTTP over QUIC)

- QUIC: Transport layer protocol
  - Initially developed by Google, now a standard [1]
- Uses UDP instead of TCP to create multiplexed “streams”
  - No head-of-line blocking
- Minimizes TLS handshake latency
  - Combines crypto and transport handshake
- Modular Congestion Control and reliability
  - Application level/User Space
  - Different algos can be plugged in
- Unambiguous ACKs
- Better RTT estimations with delay encoded in ACKs and montonically increasing packet numbers

Implications for HTTP streaming:

- Lower start-up delay
- Lower stalls [2]



Adoption Gaining Momentum

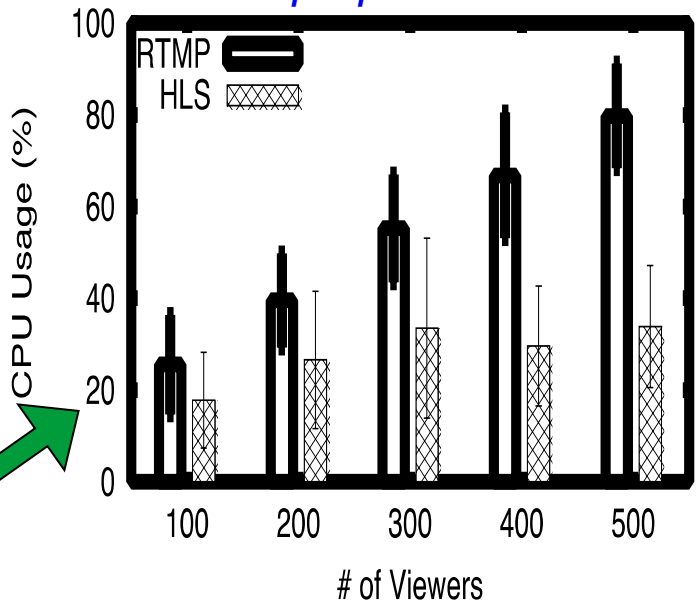
- Support in major browsers (Chrome, Firefox, Opera, Safari
- All Apps from Google, Facebook and Uber use QUIC
- As of Oct 2021 about 6% of all websites use QUIC

1. Iyengar, Jana, and Martin Thomson. "QUIC: A UDP-based multiplexed and secure transport." *Internet Engineering Task Force, Internet-Draft draftietf-quic-transport-17* (2018).
2. Langley, Adam, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang et al. "The quic transport protocol: Design and internet-scale deployment." In *Proceedings of the conference of the ACM special interest group on data communication*, pp. 183-196. 2017.

Stateful protocols: latency

- Stateful protocols provide low latency
  - No need to use long segments like with HTTP
  - Typically, separate control and media channels -> UDP for media
    - UDP latency<<TCP latency
  - Fine grained control over ABR
- 1 to 1 resource mapping for each session
- Poor scalability → high cost
  - Complicated redirection vs. HTTP
  - Resource consumption difference
  - Price of CDN delivery vs. Amazon EC2 server instances
  - Low latency stateful streams are costly

Wowza Stream Engine on a laptop



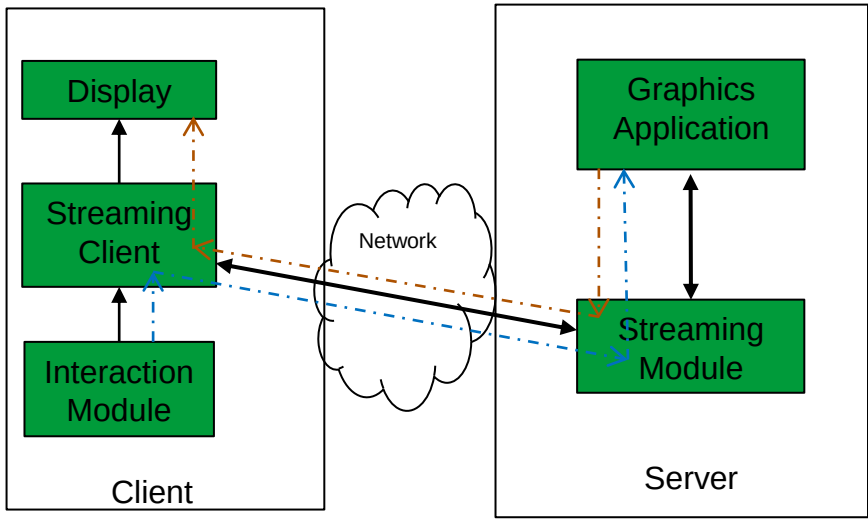
B. Wang et al: *Anatomy of a Personalized Livestreaming System.* IMC 2016.

Outline

- Live video streaming
  - Live video delivery
  - Protocols
  - Latency in video streaming
- Cloud Gaming and Cloud VR
  - Cloud gaming
  - VR
- Conclusions

Cloud Gaming and Cloud VR

- Examples of (Real-Time) Remote Rendered Interactive Multimedia Applications
- Remote Rendering
  - Old concept-> Remote graphical desktop environments e.g. VNC, Xpra, RDP
- Rendering done on a remote computer but displayed on a local client and responsive to inputs at the local client
- Remote Rendering + Cloud Computing
  - Real Time R R Interactive multimedia



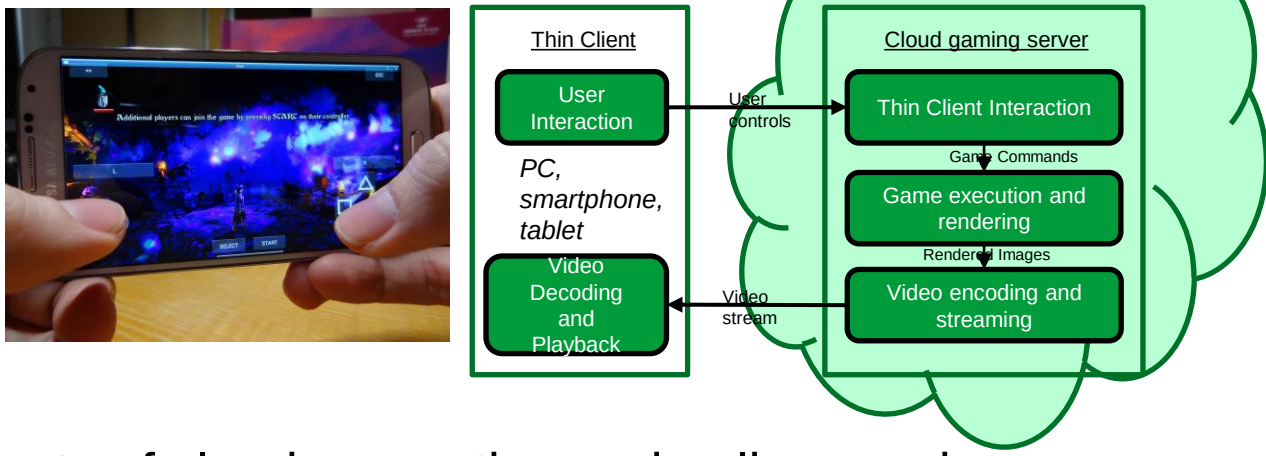
- Renewed interest for graphics heavy interactive apps
  - Graphics clouds becoming commercially available (e.g., AWS, Google, Azure, Nvidia, Tencent, Oracle, Alibaba, Baidu all offer cloud GPUs)

- End to End Latency is the key
- Benefits of offloading only visibile if the resulting visual quality is better than native rendering at the client
- Cloud Gaming and Cloud XR are main example applications

Outline

- Live video streaming
  - Live video delivery
  - Protocols
  - Latency in video streaming
- Cloud Gaming and Cloud VR
  - Cloud gaming
  - VR
- Conclusions

What is cloud gaming?



- Combines the concepts of cloud computing and online gaming
- Server runs game and renders graphics on behalf of thin client
  - Client → server: Control input
  - Server → client: Video stream
- Challenges
  - Touch-to-photon latency should not be perceived
  - High downstream bandwidth required for high quality video stream

Benefits of cloud gaming

Users

- Play games with high-end graphics on resource constrained devices
- Access to games on any device anywhere
- Avoid hardware updates

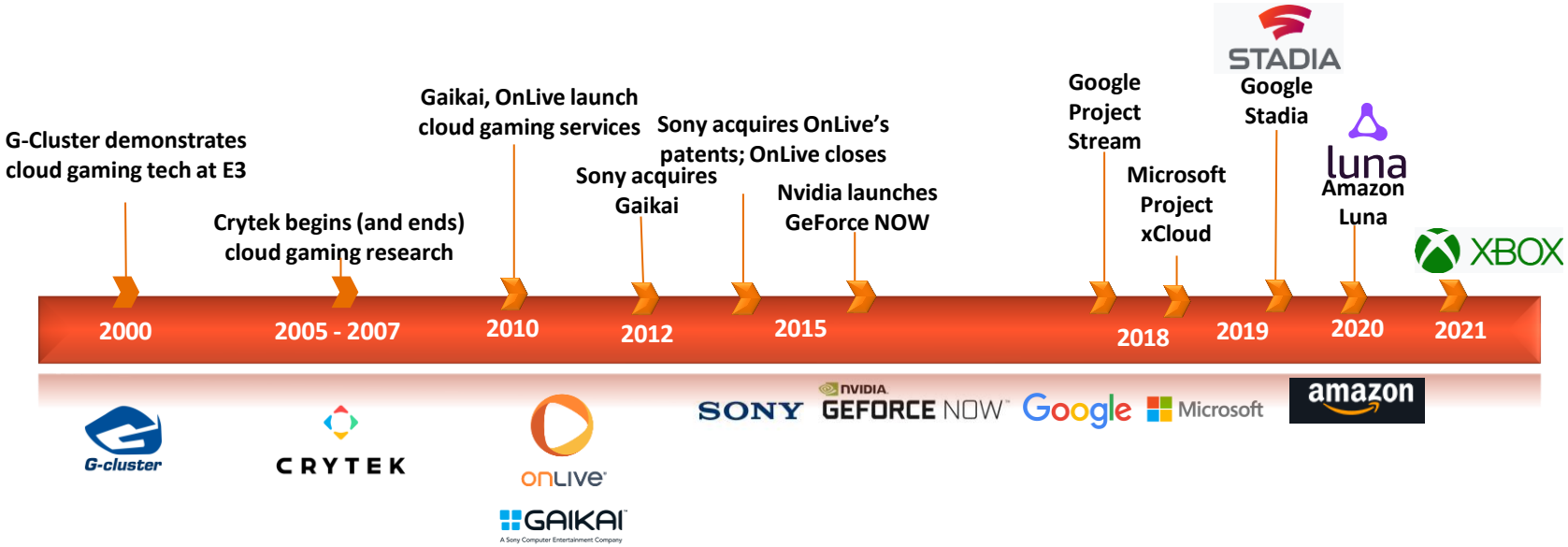
Game developers

- Concentrate on a single platform
- Reach out to more gamers
- Mitigate piracy

Service providers

- Enable new business models
- Create more demand on already-deployed cloud resources

Brief history of cloud gaming



Outline

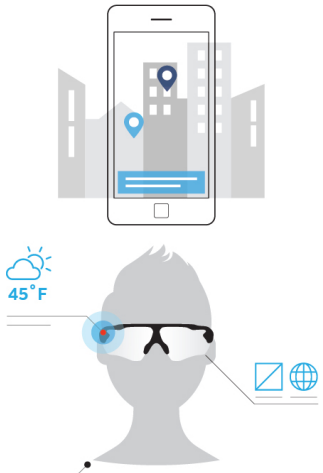
- Live video streaming
  - Live video delivery
  - Protocols
  - Latency in video streaming
- Cloud Gaming and Cloud VR
  - Cloud gaming
  - Cloud VR
- Conclusions

Immersive technologies

Physical world

AR: Augmented Reality

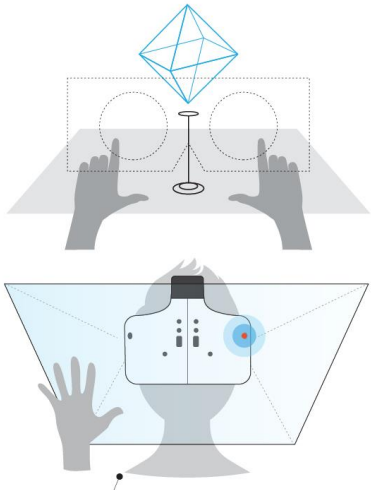
Real world with digital information overlay



Real world remains central to the experience, enhanced by virtual details.

MR: Mixed Reality

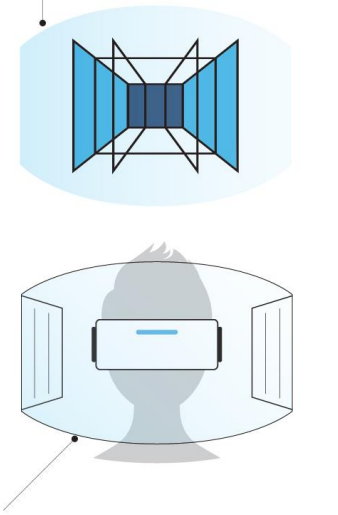
Real and the virtual are intertwined



Interaction with and manipulation of both the physical and virtual environment.

VR: Virtual Reality

Completely digital environment



Fully enclosed, synthetic experience with no sense of the real world.

Virtual world

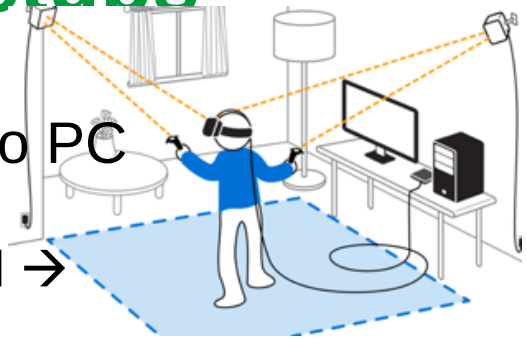
XR (eXtended Reality) = AR+MR+VR

What is virtual reality (VR)?

- Goal is immersive experience
  - High pixel resolution per eye needed
- Use of head mounted display (HMD)
  - Graphics rendered in real-time
  - Head tracking to update field of view
- Handheld controller(s)
  - Point and select, control virtual objects, implement locomotion, etc.
- Stringent “motion-to-photon” latency requirements
  - VR/simulator sickness (nausea, vomiting...)
  - Mismatching sensory input (visual vs. vestibular and proprioceptive)
  - Some say <20ms, others say even less

VR equipment and setups

- HMD normally cable connected to PC
  - PC has dedicated GPU
  - Stepping on cable while immersed → accidents
  - Wireless adapters exist (e.g., HTC)
  - VR backpack for increased mobility
- Different HMDs exist
  - Rift and Vive are the traditional kinds
  - Some can track gaze (FOVE, Varjo)
  - Mobile VR → no PC required
    - Smartphone with headmount (e.g., Daydream View)
    - Standalone headset (e.g Quest, Focus)
  - Varjo's multi-resolution display



https://vr.google.com/daydream/standalonevr/

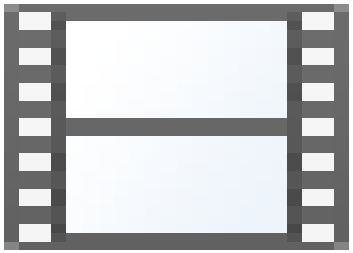
Tracking and Degrees of Freedom (DoF)

- HMD and controllers either 3 or 6 DoF
- Traditionally 6DoF enabled by outside-in tracking
  - Require external base stations
  - Oculus Rift and HTC Vive
- Some HMDs provide 6DoF with inside-out tracking
  - Camera + motion sensing
  - Oculus Quest, HTC Cosmos, Oculus Rift S, Lenovo Mirage Solo
- Also standalone 6DoF controllers exist
  - E.g. Oculus Quest provides fully standalone 6DoF experience

Mobile VR challenge

	Headset + PC	Headset + backpack	Smartphone + head mount	Standalone headset
Render Capacity	High	Medium-High	Low	Low-Medium
Hardware cost	~$2000	~$3000	$100 (+phone)	$200-400
Mobility	No	Yes	Yes	Yes
Setup	Complex	Complex	Simple	Simple

60 fps



6 fps

Cloud gaming approach to enable high quality mobile VR

Latency in Cloud Gaming & Cloud VR

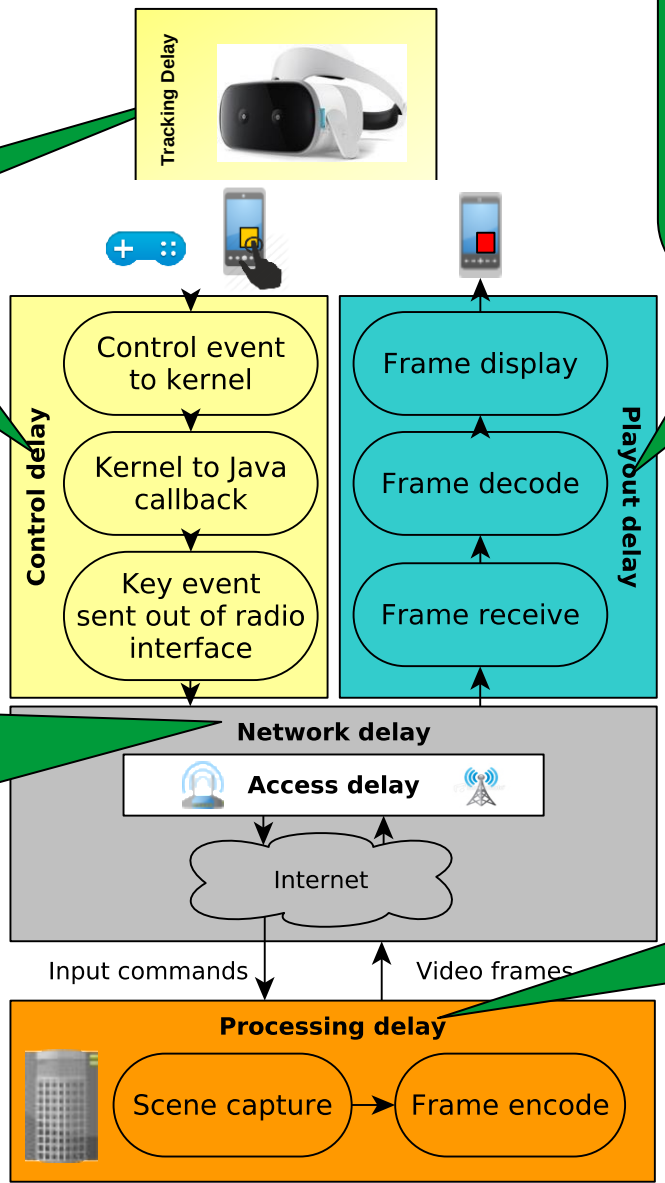
- Noticeable latencies should be avoided
  - Annoying to the user
  - May degrade gameplay performance and QoE
  - Latency in cloud VR, may cause VR sickness
- Humans are able to perceive some latencies
  - Depends on many things (task, user interface specifics, viewing modality…)
  - Touch screen tapping 50-100ms and dragging 10-60ms [1]
  - VR: Motion to photon latency of ~15ms [2]

1. J. Deber et al. *How Much Faster is Fast Enough?: User Perception of Latency & Latency Improvements in Direct and Indirect Touch*. ACM CHI 2015.
2. Draper MH, Viirre ES, Furness TA, Gawron VJ. Effects of Image Scale and System Time Delay on Simulator Sickness within Head-Coupled Virtual Environments. *Human Factors*. 2001;43(1):129-146.

Latency in Cloud Gaming & Cloud VR

Time from push/touch to event in application code
- Can also be hmd movement or separate controller (USB/Bluetooth)

Time to decode video frame and draw it on screen
- Decoding hw accelerated
- Display refresh rate and buffering mode (e.g., double)



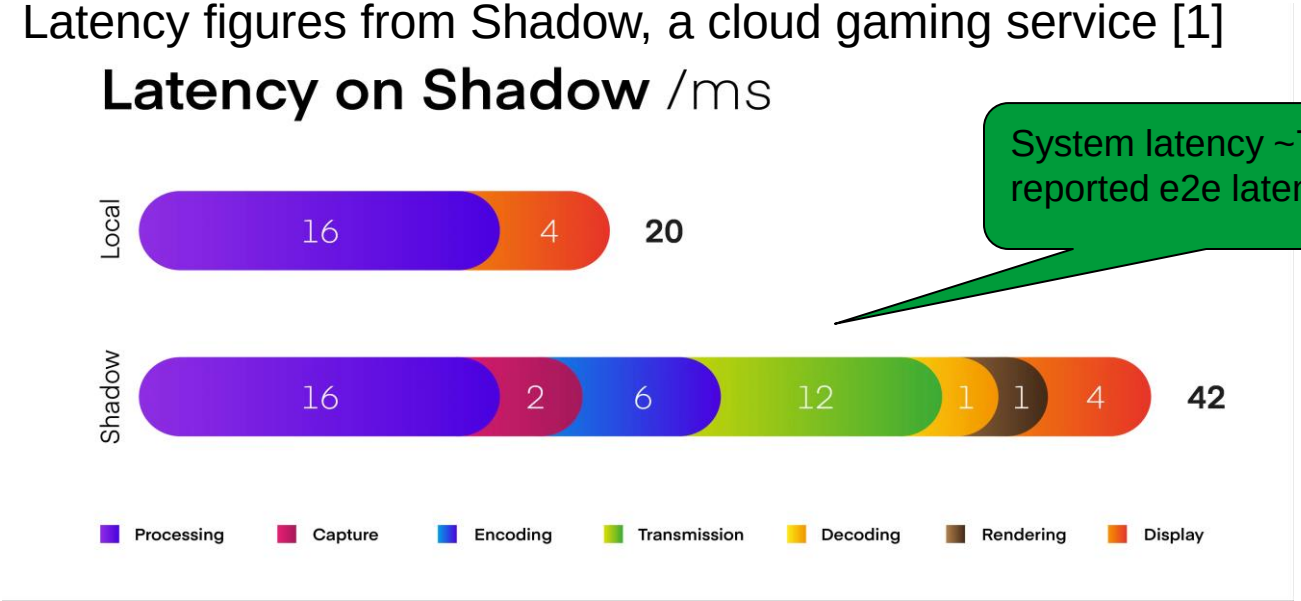
Depends on distance and (access) network technology used
Only a component of the total end to end latency

Depends on server capabilities and game

Latency in Cloud Gaming & Cloud VR

- Network latency is only one part of the problem
- System Latency at server and client: Capture, Encode, Decode, game logic processing, (client side minimal) rendering, interaction and display make up the bulk of the latency
- Additional in VR: Tracking Latency-> head rotation(3DoF), (6DoF) point of observation changes

Latency figures from Shadow, a cloud gaming service [1]



1. https://web.archive.org/web/20210817121609/https://shadow.tech/en-gb/blog/news/roadmap-cloud-gaming-without-latency

Masking latency

- Mask latency with speculative execution: Outatime[1]
  1. Predict user input/motion based on earlier ones (Markov model/NN)
  2. Render speculated frame
     - Sometimes render multiple speculative outcomes when user input is difficult to predict (e.g., fire a weapon)
  3. Correct misprediction (image based rendering for warping)
- Advantages:
  - Can potentially mask over 100ms of network latency
  - No need to replicate computing to many locations
- Disadvantages:
  - Bandwidth overhead: bitrate is 1.5-2 x original with all optimizations
  - Computing overhead: bounded to 4x rendering in the paper but this depends on game (what kind of events are possible)
  - Need to modify the game engine

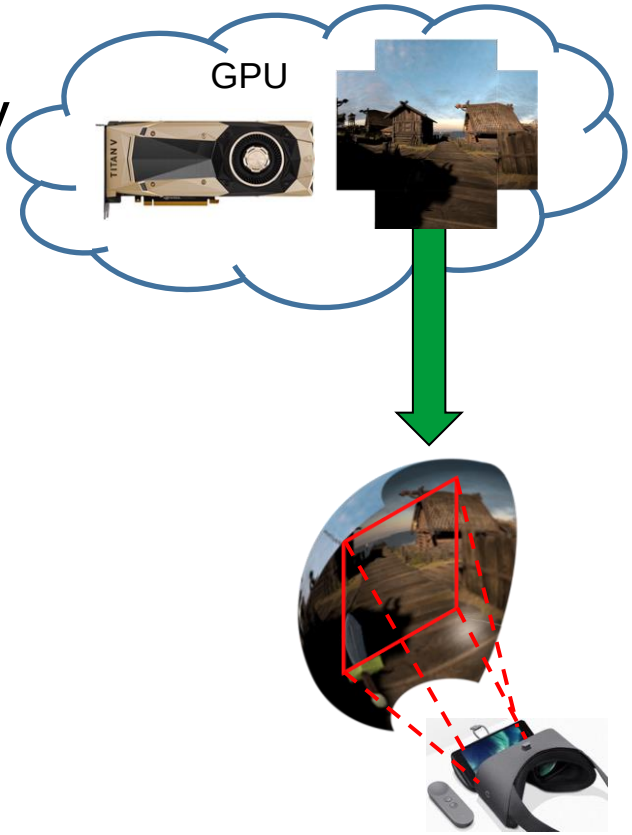
[1] Kyungmin Lee et al. *Outatime: Using Speculation to Enable Low-Latency Continuous Interaction for Mobile Cloud Gaming.* Mobisys 2015.

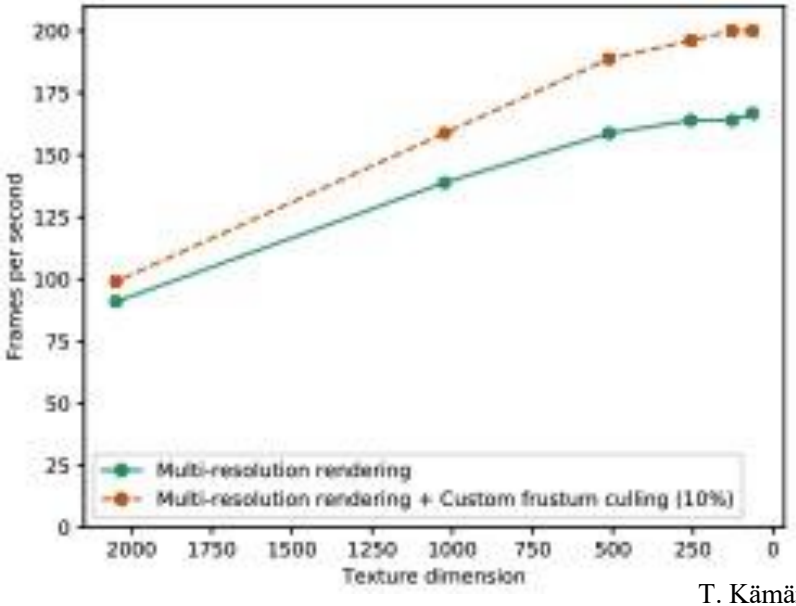
Masking latency:Cloud VR

- Pre-render content → enable local navigation at client
- Collaborative rendering → client and server share work

Pre-render content

- Render more scene than immediately visible at client
  - No need to render full 360 degrees
  - Adjusting scene size and resolution improve performance
- Works for 3DoF scenarios



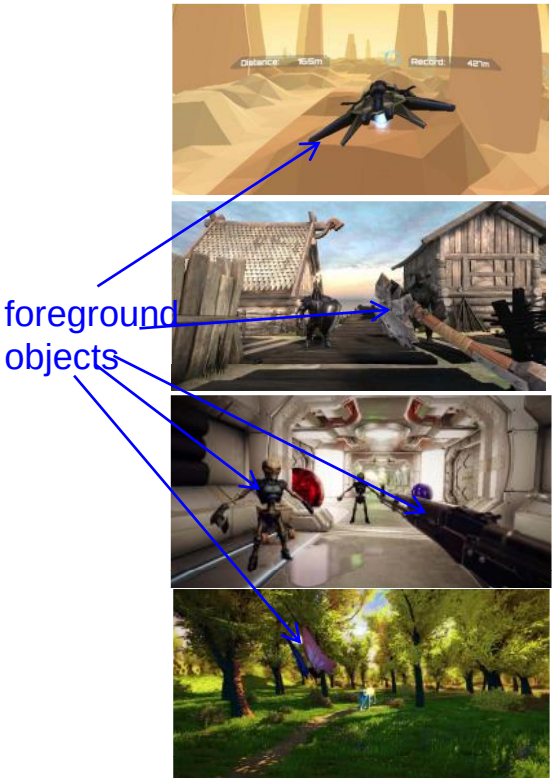


Aalto University
School of Science

T. Kämäräinen, M. Siekkinen, J. Eerikäinen, A. Ylä-Jääski.
CloudVR: Cloud Accelerated Interactive Mobile Virtual Reality.
ACM Multimedia 2018

Cooperative rendering

- Insight: rendering background often much heavier than rendering moving objects
- Cooperative rendering
  1. Divide rendering into foreground interactions and background environment
  2. Render foreground using local mobile GPU
  3. Pre-render and pre-fetch background on powerful remote server
  4. Combine foreground and background on mobile device into final frames



Zeqi Lai et al. *Furion: Engineering High-Quality Immersive Virtual Reality on Today's Mobile Devices.* MobiCom 2017.

Protocols for Cloud Gaming & Cloud VR

- Latency target is “Just Noticeable Delay”
  - Depends on Game genre, can be as high as 100ms
- Even lower for Cloud VR
  - Depends on use case and motion patterns [1] reports <16ms, [2] reports 200ms
  - Not achievable even with low latency versions of HTTP based streaming
- Stateful protocols provide low latency
  - Low protocol overhead, latency governed by network
  - With appropriate resource provisioning can provide JND for cloud gaming
  - 5G access and Edge graphics clouds, may provide JND for cloud VR/XR-> Use case under discussion by TSG-SA4

1. Ellis SR, Mania K, Adelstein BD, Hill MI. Generalizeability of Latency Detection in a Variety of Virtual Environments. Proceedings of the Human Factors and Ergonomics Society Annual Meeting. 2004;48(23):2632-2636. doi:10.1177/154193120404802306
2. Allison, R.S., Harris, L.R., Jenkin, M., Jasiobedzka, U., & Zacher, J.E. (2001) Tolerance of temporal delay in virtual environments, Proceedings, IEEE Virtual Reality 2001, pp. 247-253.

Protocols for Cloud Gaming & Cloud VR

- RTMP, RTSP conventional stateful protocols
  - Different channels for data and media
  - Control info/user actions over data channel
  - Video/audio over media channel

- WebRTC likely to be used in future with game engine support
  - Unity Render Streaming
  - Unreal Pixel Streaming

Wrapping up…

- Live video streaming
  - Outbound streams use HTTP due to scalability (CDN)
  - Ingest streams can use stateful protocols
  - Latency matters unlike with VoD
  - Low latency HTTP streaming emerged recently
- (Remote Rendered) Interactive multimedia
  - Gaming and VR as example applications
  - Graphics computing vs. latency tradeoff is the main challenge for mobile
    - Remote rendering increases latency → bring servers closer and optimize rendering
  - 5G bandwidth and latency will make these more feasible
  - WebRTC becoming protocol of choice for Interactive and conversational multimedia