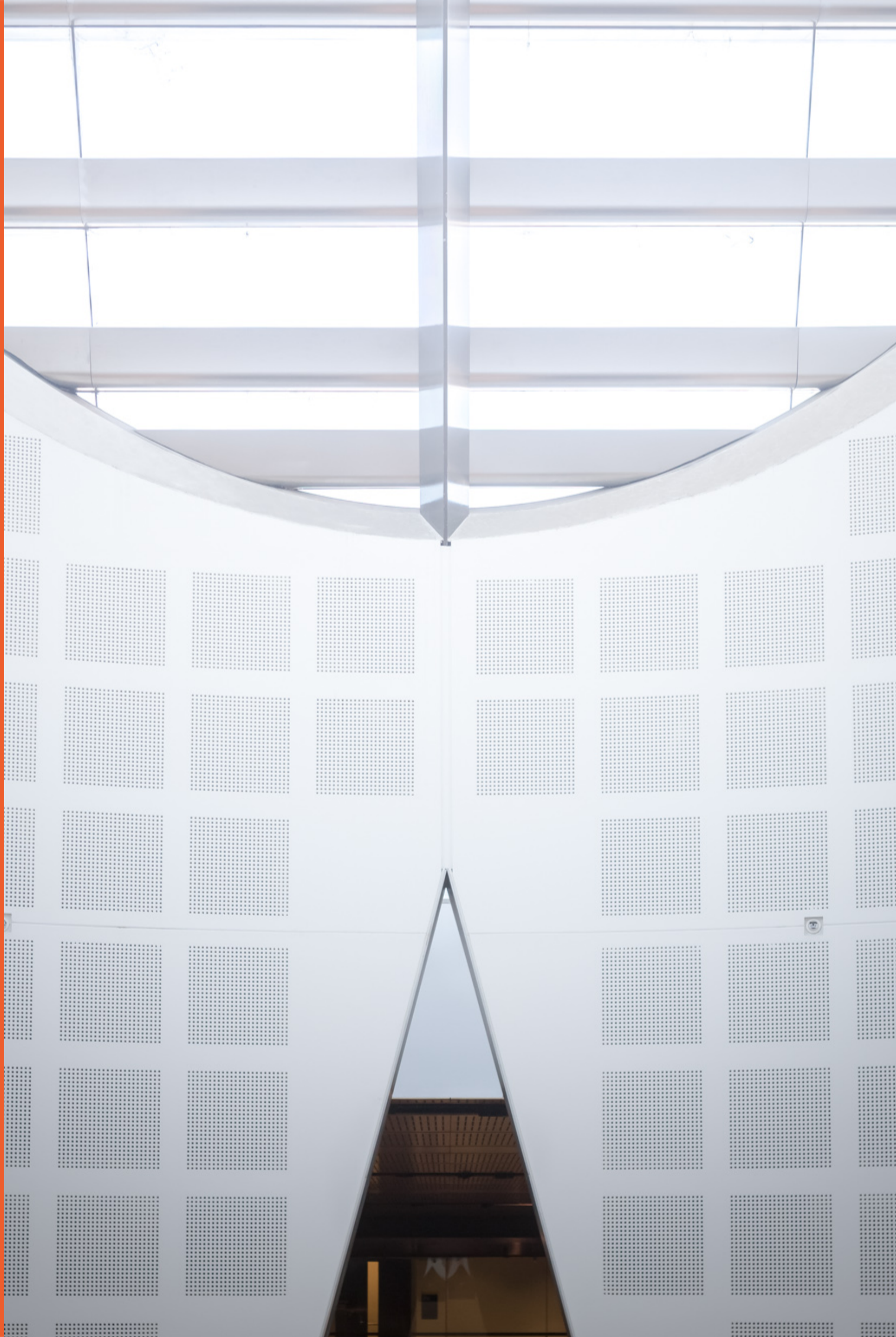
Mobile Computing
COMP5216

Week 10

Semester 2, 2020

Dr. Kanchana Thilakarathna
School of Computer Science
THE UNIVERSITY OF SYDNEY



Outline

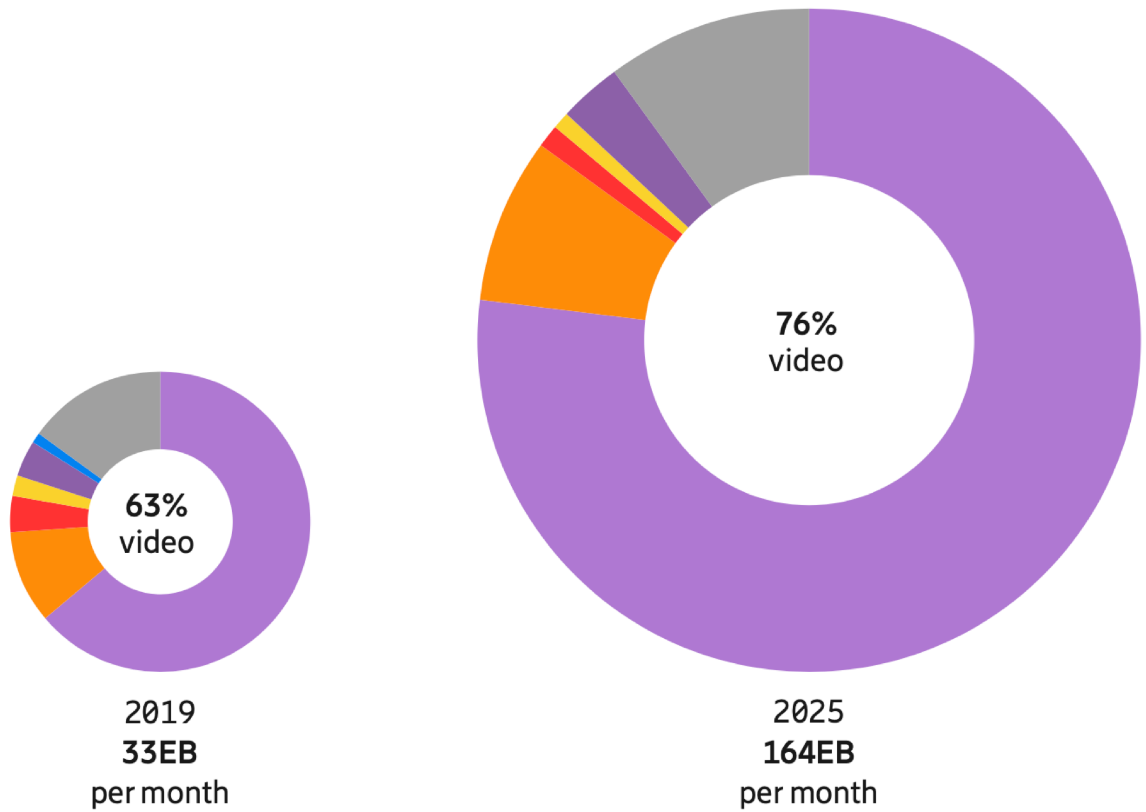
- Why mobile video?
- Part 1: Conventional video streaming
- Part 2: Next generation interactive video streaming

- Project Final

- Acknowledgement:
- Some slides/content for this presentation were obtained from
  - ACM SIGCOMM Tutorial on Adaptive Streaming of Traditional and Omnidirectional Media, Los Angeles, CA Aug. 2017
  - Computer Networking: A Top Down Approach
    7th Ed. Jim Kurose, Keith Ross
    Pearson - Addison-Wesley

Why mobile video?

Exponential Growth in Mobile Video



[1] Traffic from embedded video in web browsing and social media is included in the application category "Video"

Video | Social networking | Web browsing | Audio | Software download and update | P2P file sharing | Other segments

Main drivers for video traffic growth

- Video is part of most online content (news, ads, social media, etc.)
- Video sharing services
- Video streaming services
- Changing user behavior
  - video being consumed anywhere, any time
- Increased segment penetration, not just early adopters
- Evolving devices with larger screens and higher resolutions
- Increased network performance through evolved 4G deployments
- Emerging immersive media formats and applications (HD/UHD, 360-degree video, AR, VR)

Why is it important for Mobile Developers?

3 Application types

- *streaming, stored* audio, video
  - *streaming:* <u>can begin playout before downloading entire file</u>
  - *stored (at server):* can transmit faster than audio/video will be rendered (implies storing/buffering at client)
  - e.g., YouTube, Netflix, Hulu
- *conversational* voice/video over IP
  - interactive nature of human-to-human conversation limits delay tolerance
  - e.g., Skype
- *streaming live* audio, video
  - e.g., live sporting event (futbol)

What is streaming?

Streaming is transmission of a continuous content from a server to a client and its simultaneous consumption by the client

Two Main Characteristics

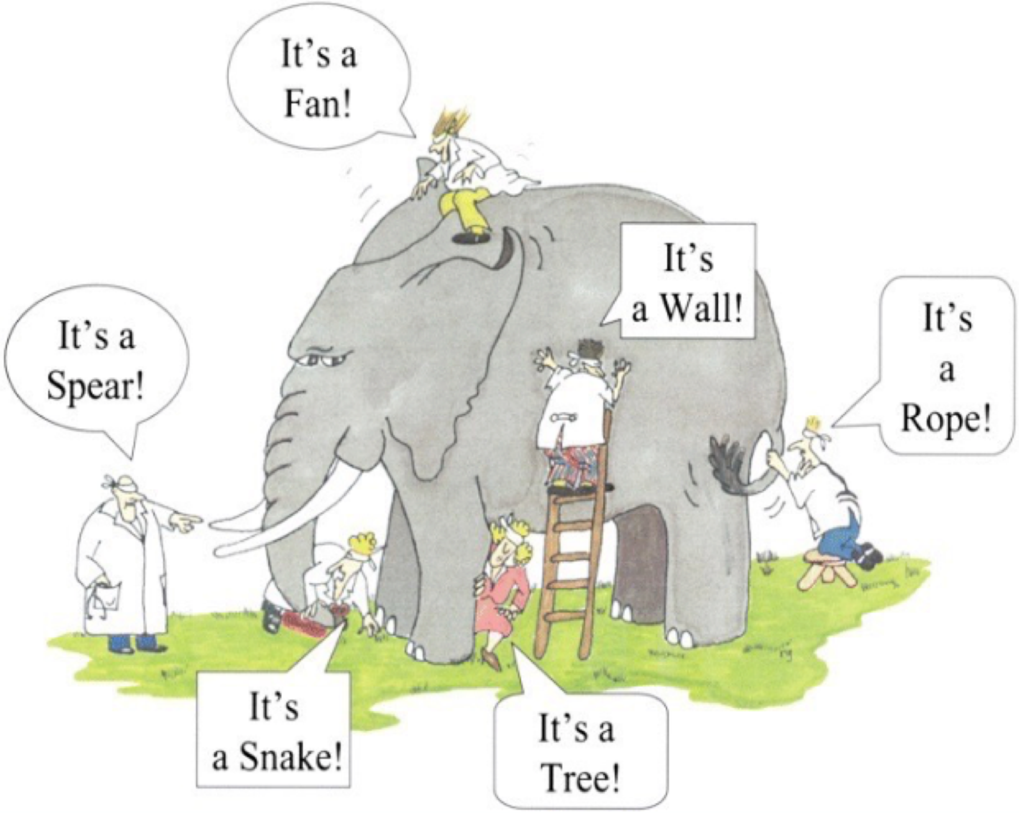
1. Client consumption rate may be limited by real-time constraints as opposed to just bandwidth availability
2. Server transmission rate (loosely or tightly) matches to client consumption rate

Eco system and requirements

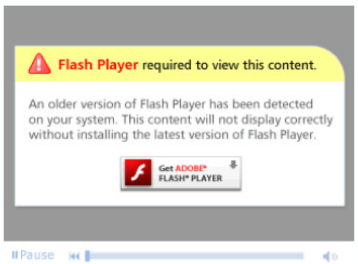
- I want to make sure that my content is protected and looks awesome on any device
- I want to make sure that my ads are viewed, trackable and measurable
- I want to make sure that my servers are properly used and latency is low
- I want to control the QoE of all my customers, differentiate my own services, make $$$ from OTT services
- I want to make sure that my device/app provides the best possible video quality
- I want the best quality for minimal $

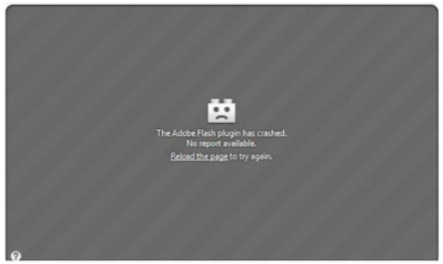
What is Quality of Experience?

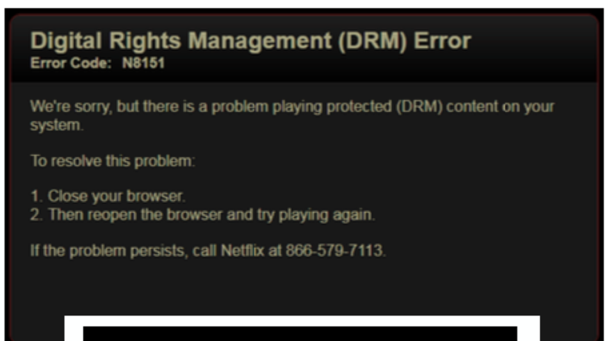
Everyone has different perspective

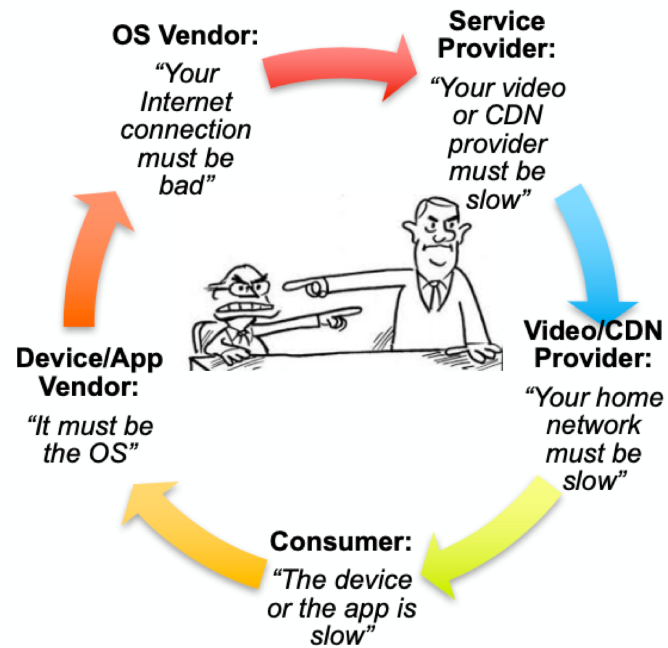


Video streaming: Challenges





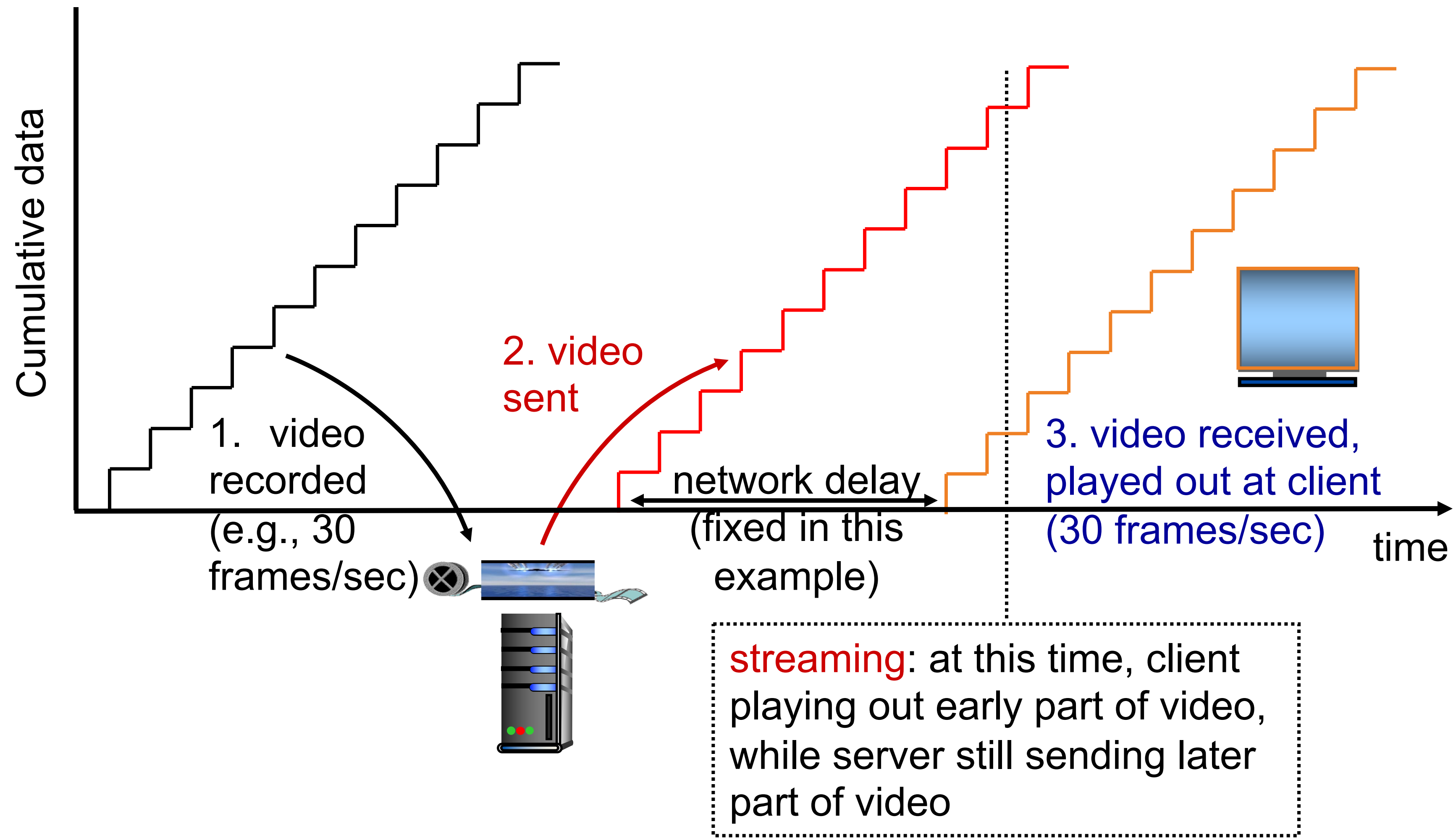




Video Streaming: challenges

- video traffic: major consumer of Internet bandwidth
  - Netflix, YouTube: 37%, 16% of downstream residential ISP traffic
  - ~1B YouTube users, ~75M Netflix users
- challenge: scale - how to reach ~1B users?
  - single mega-video server won't work (why?)
- challenge: heterogeneity
  - different users have different capabilities (e.g., wired versus mobile; bandwidth rich versus bandwidth poor)
- *solution:* distributed, application-level infrastructure

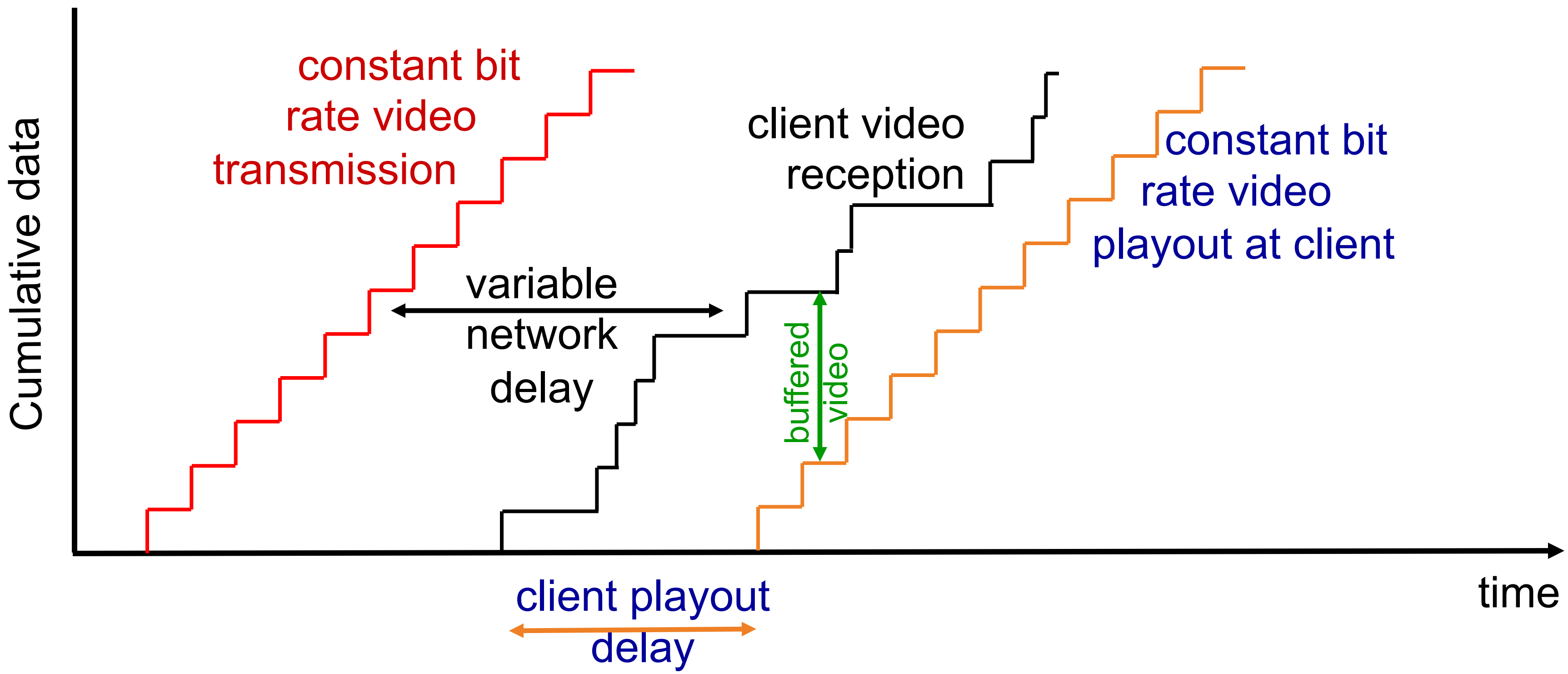
Streaming stored video:



Streaming stored video: challenges

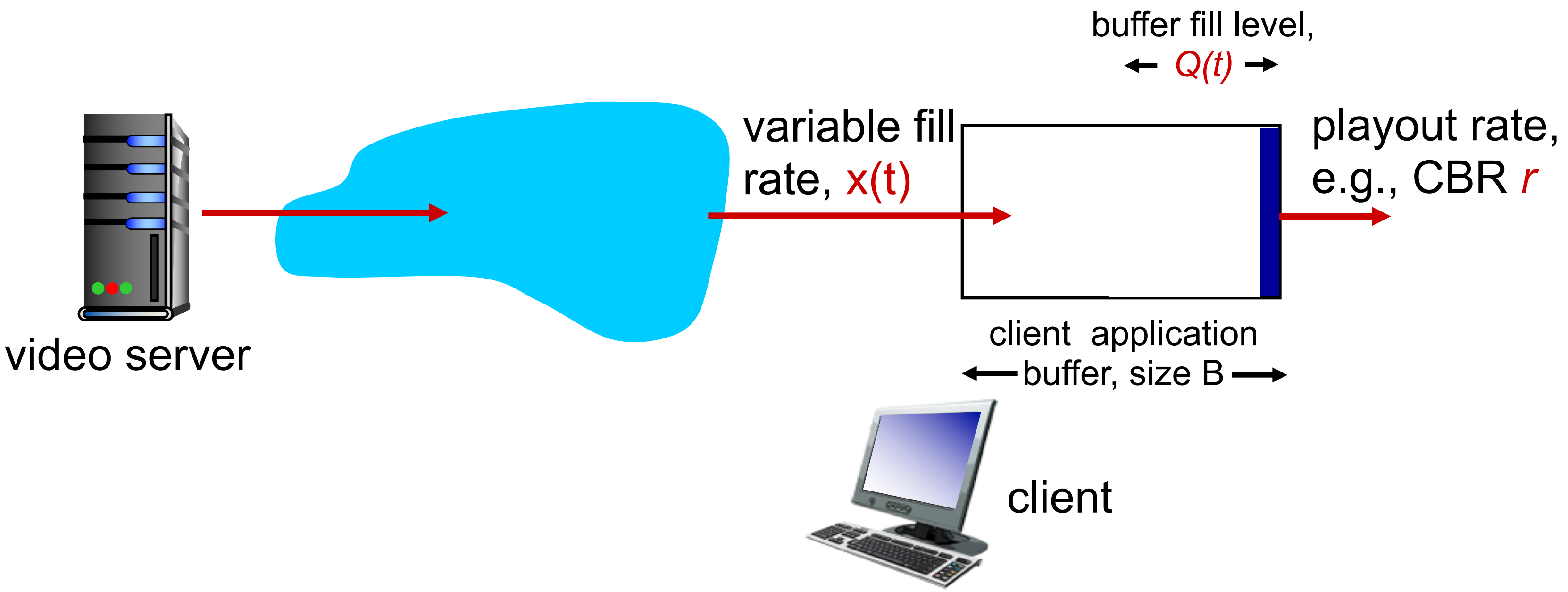
- continuous playout constraint: once client playout begins, playback must match original timing
  - … but network delays are variable (jitter), so will need client-side buffer to match playout requirements
- other challenges:
  - client interactivity: pause, fast-forward, rewind, jump through video
  - video packets may be lost, retransmitted

Streaming stored video: revisited



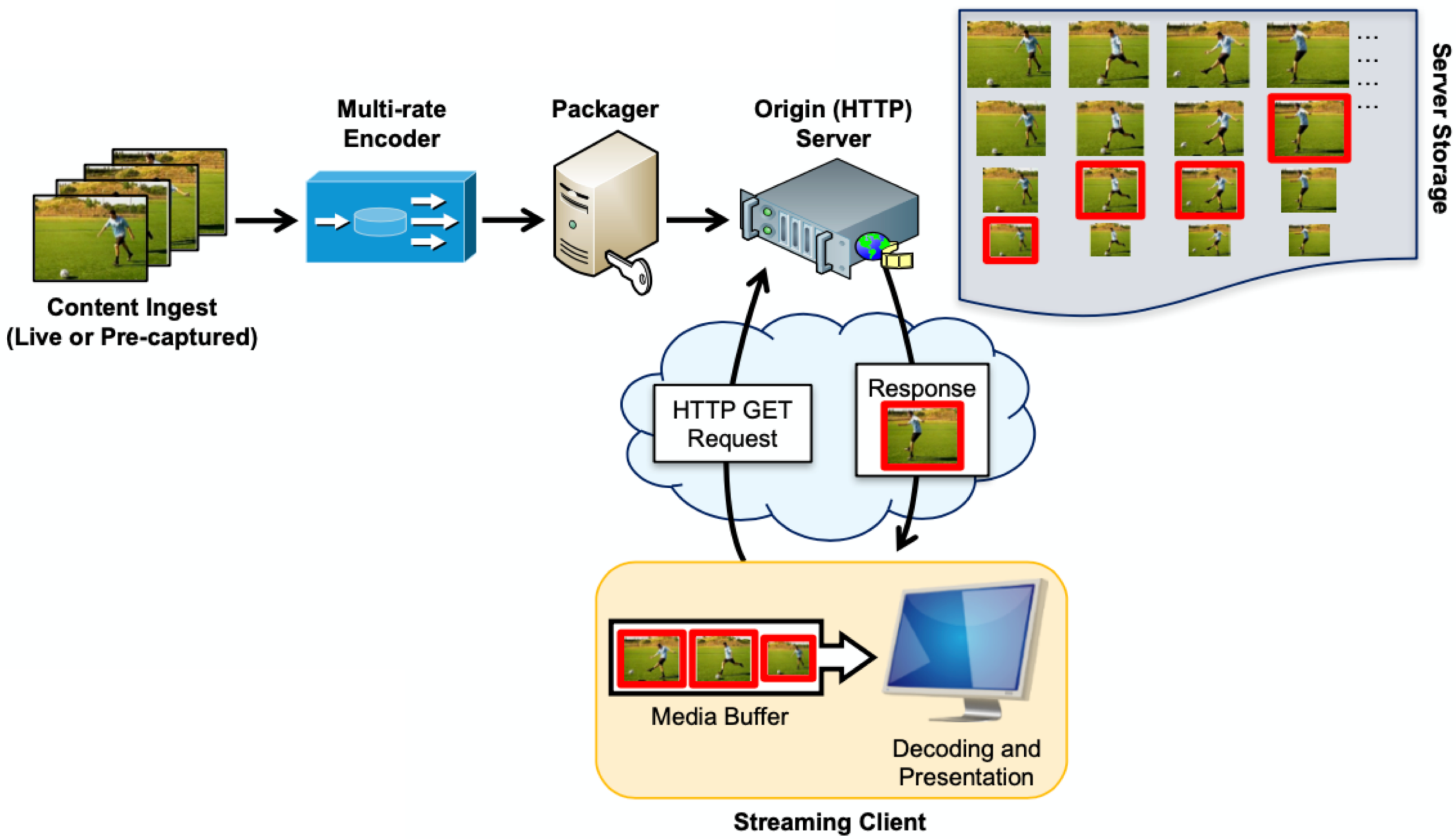
- *client-side buffering and playout delay:* compensate for network-added delay, delay jitter

Client-side buffering, playout



1. Initial fill of buffer until playout begins at t_p
2. playout begins at t_p,
3. buffer fill level varies over time as fill rate x(t) varies and playout rate r is constant

DASH: Dynamic, Adaptive Streaming over HTTP

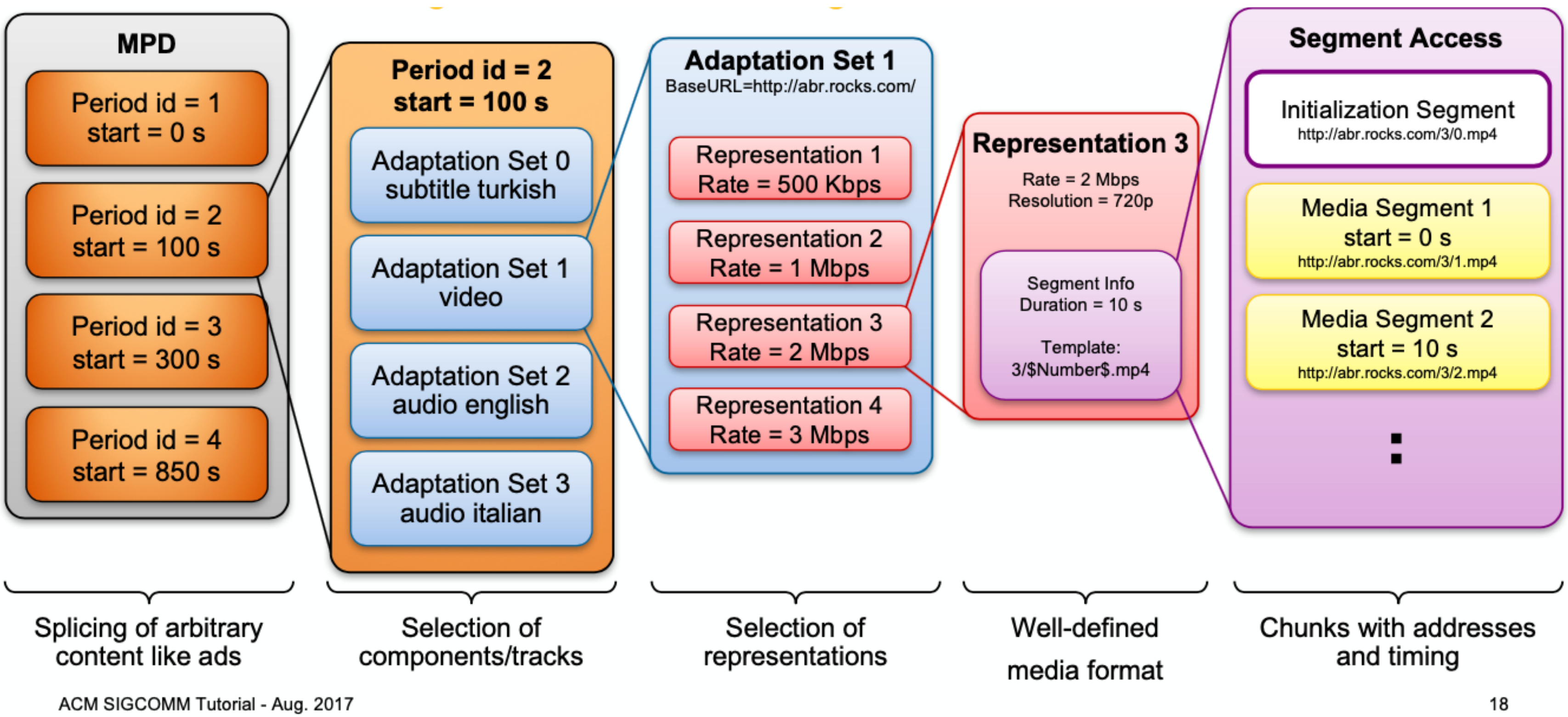


https://mpeg.chiariglione.org/standards/mpeg-dash

Streaming multimedia: DASH

- *DASH: D*ynamic, *A*daptive *S*treaming over *H*TTP
- *server:*
  - divides video file into multiple chunks
  - each chunk stored, encoded at different rates
  - *manifest file:* provides URLs for different chunks

Streaming multimedia: DASH – Manifest file



Streaming multimedia: DASH

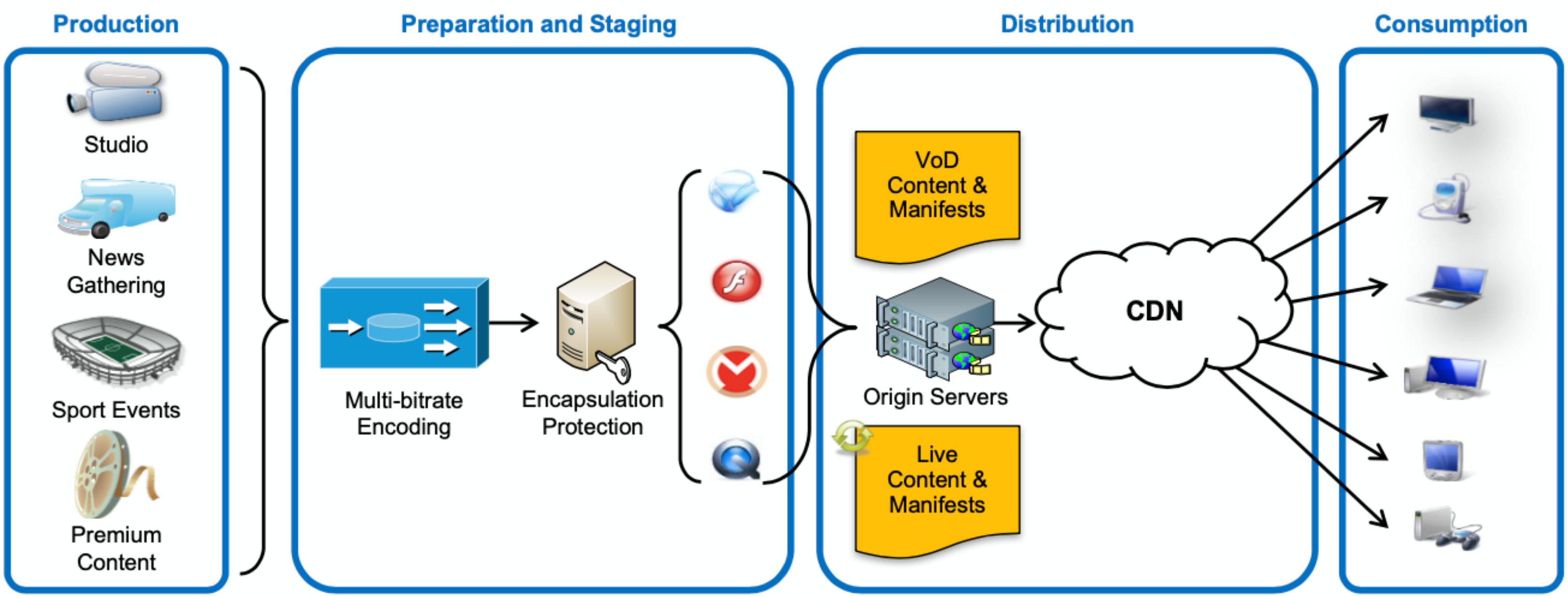
- *client:*
  - periodically measures server-to-client bandwidth
  - consulting manifest, requests one chunk at a time
    - chooses maximum coding rate sustainable given current bandwidth
    - can choose different coding rates at different points in time (depending on available bandwidth at time)

Streaming multimedia: DASH

- *"intelligence"* at client: client determines
  - *when* to request chunk (so that buffer starvation, or overflow does not occur)
  - *what encoding rate* to request (higher quality when more bandwidth available)
  - *where* to request chunk (can request from URL server that is "close" to client or has high available bandwidth)

This is why we (mobile developers) need to understand video streaming

It is not just stored video…



Content distribution networks

- *challenge:* how to stream content (selected from millions of videos) to hundreds of thousands of *simultaneous* users?

- *option 1:* single, large “mega-server”
  - single point of failure
  - point of network congestion
  - long path to distant clients
  - multiple copies of video sent over outgoing link

….quite simply: this solution *doesn’t scale*

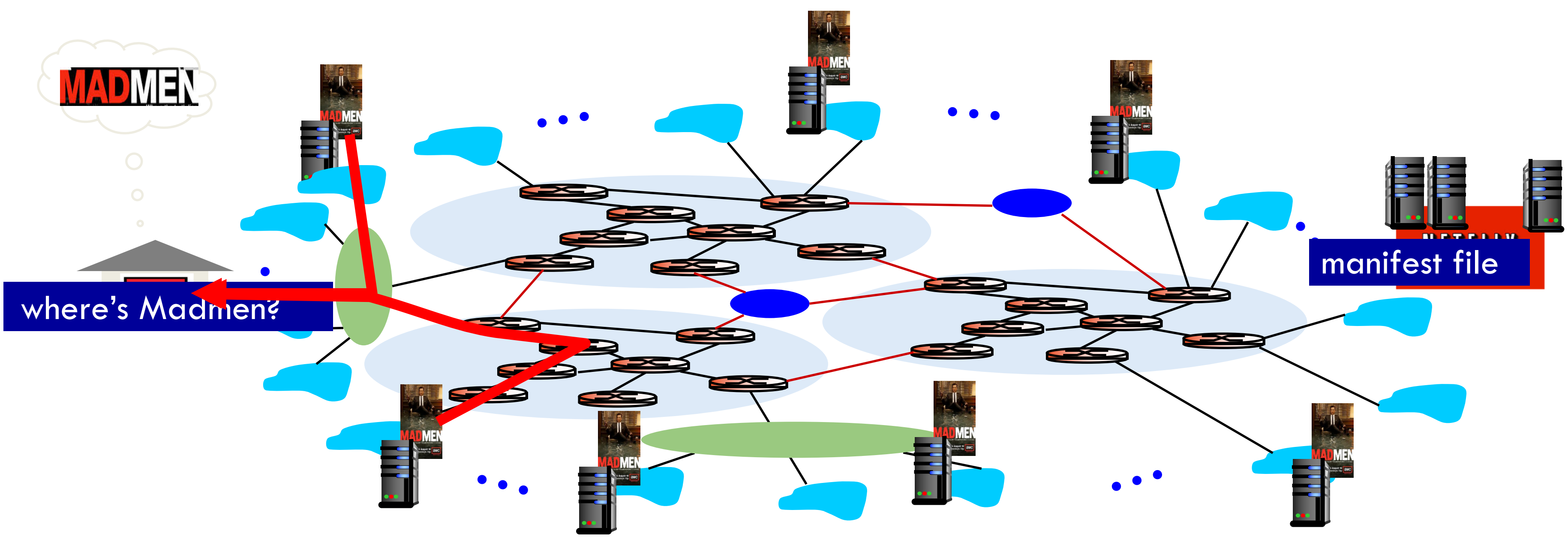
Content distribution networks

- *challenge:* how to stream content (selected from millions of videos) to hundreds of thousands of simultaneous users?

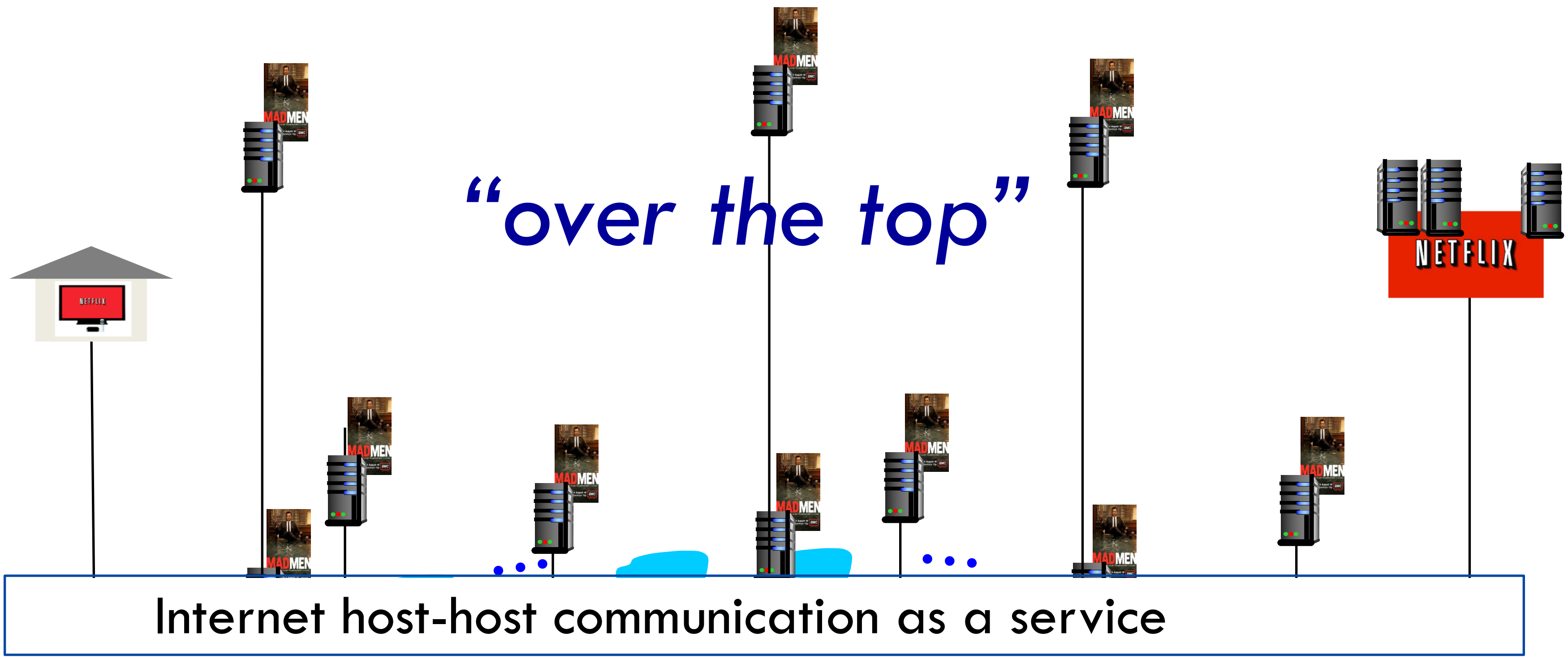
- *option 2:* store/serve multiple copies of videos at multiple geographically distributed sites *(CDN)*
  - *enter deep:* push CDN servers deep into many access networks
    - close to users
    - used by Akamai, 1700 locations
  - *bring home:* smaller number (10's) of larger clusters in POPs near (but not within) access networks
    - used by Limelight

Content Distribution Networks (CDNs)

- CDN: stores copies of content at CDN nodes
  - e.g. Netflix stores copies of MadMen
- subscriber requests content from CDN
  - directed to nearby copy, retrieves content
  - may choose different copy if network path congested



Content Distribution Networks (CDNs)



OTT challenges: coping with a congested Internet

- from which CDN node to retrieve content?
- viewer behavior in presence of congestion?
- what content to place in which CDN node?

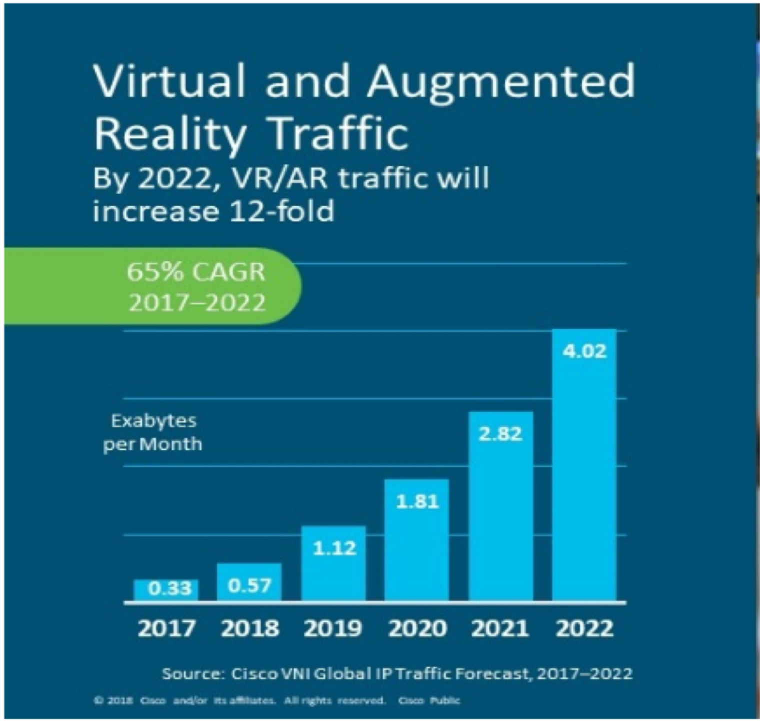
Part 3: Next generation interactive media delivery

Week 10 - COMP5216

THE UNIVERSITY OF SYDNEY

VR/360° videos

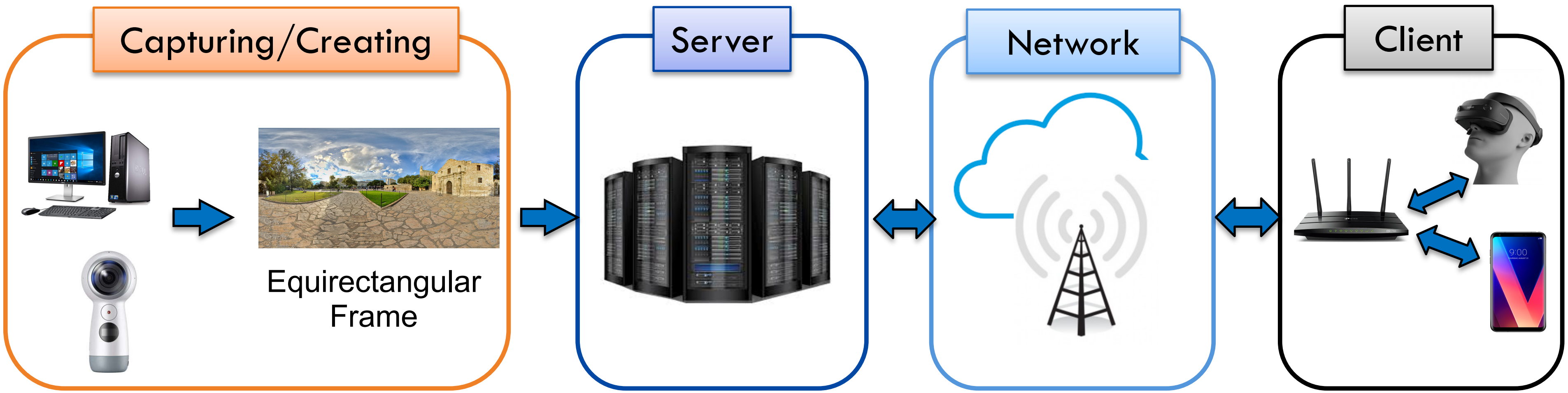
- VR/AR/360 video traffic will increase by **12-fold** of the value in 2017, by 2020 Bullet point
- Immersive video experience from **360° video** streaming.
- Many applications
  - Entertainment
  - Education
  - Telemedicine



Sample 360° video

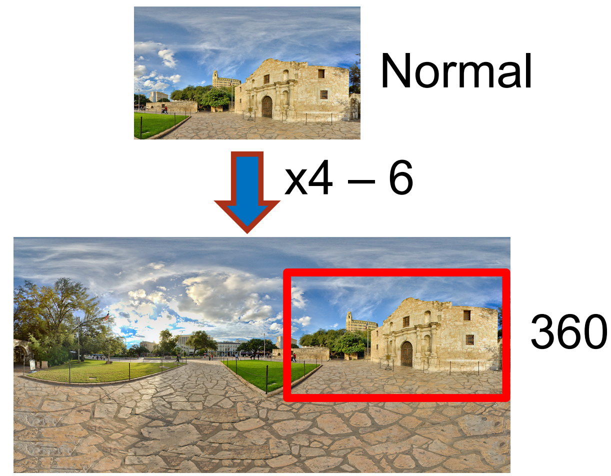
How current 360° video streaming works?

- YouTube, Facebook, Vimeo, Netflix … currently host 360 videos.
- General architecture of 360° video streaming used by majority of the services



Challenges: High bandwidth consumption

- 360 video frames are 4-6 times larger than normal frame.
- Redundant data transmission
  - Only selected portion of the video is visible at a given time
- High bandwidth usage



- For 4K resolution smooth playback*
  - On-demand BW: 100 Mbit/s
  - Live-telecasting BW: 83.2 Mbit/s
  - Typical 4G LTE BW: 20 Mbit/s

30K x 24K x 36 x 60 x 2 / 600

~5.2 Gbps

Resolution: 30K x 24K; Bit depth: 36; Frame rate: 60; Stereoscopic: 2; Compression gain: 600

** Whitepaper on the VR-Oriented Bearer Network Requirement(2016) - Huawei*

Challenges: Delay

Delay results in Stalling/Rebuffering

For ~50% of 360 video sessions in YouTube & Facebook can stall at least 5s per minute in total[1].

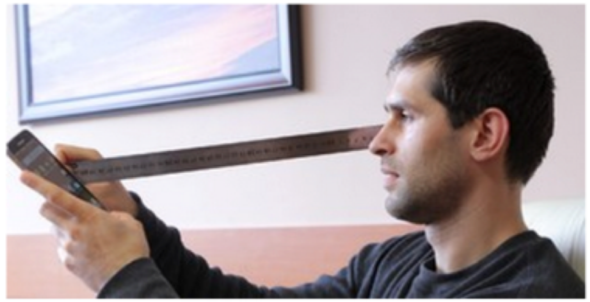
Excess delay causes cyber-sickness/VR sickness

Different symptoms[2]: general discomfort, headache, nausea, vomiting etc …

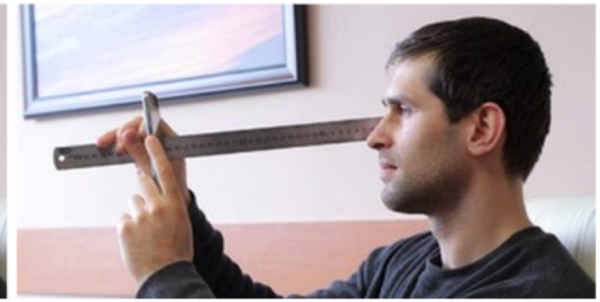
[1] *LIME: Understanding Commercial 360° Live Video Streaming Services – Liu et al. (MMsys 2019)*

[2] *Measurement of exceptional motion in VR video contents for VR sickness assessment using deep convolutional autoencoder – Kim et al. (VRST '17)*

Challenges: Poor QoE

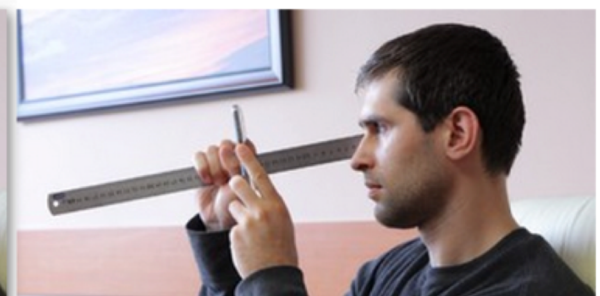


480p, average person starts noticing pixelization at around 14.7 inches (here - 37.4cm) in a 4-inch 480p phone



720p, average person starts noticing pixelization at around 11 inches in a 4.7-inch 720p phone

1080p, average person starts noticing pixelization at around 7.8 inches in a 5-inch 1080p phone



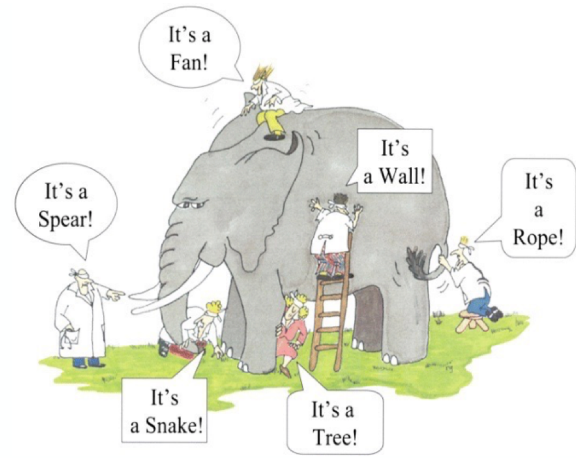
Quad HD, average person starts noticing pixelization at around 6.44 inches in a 5.5-inch 1440p phone

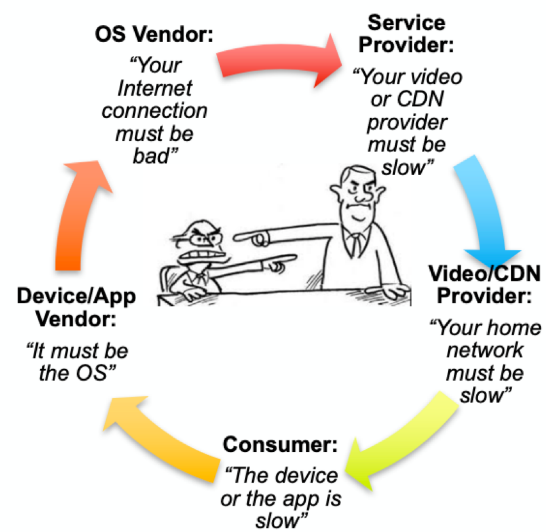
Source: Thomas Stockhammer

Challenges: Poor QoE

Poor QoE Quality of Experience is due to several facts

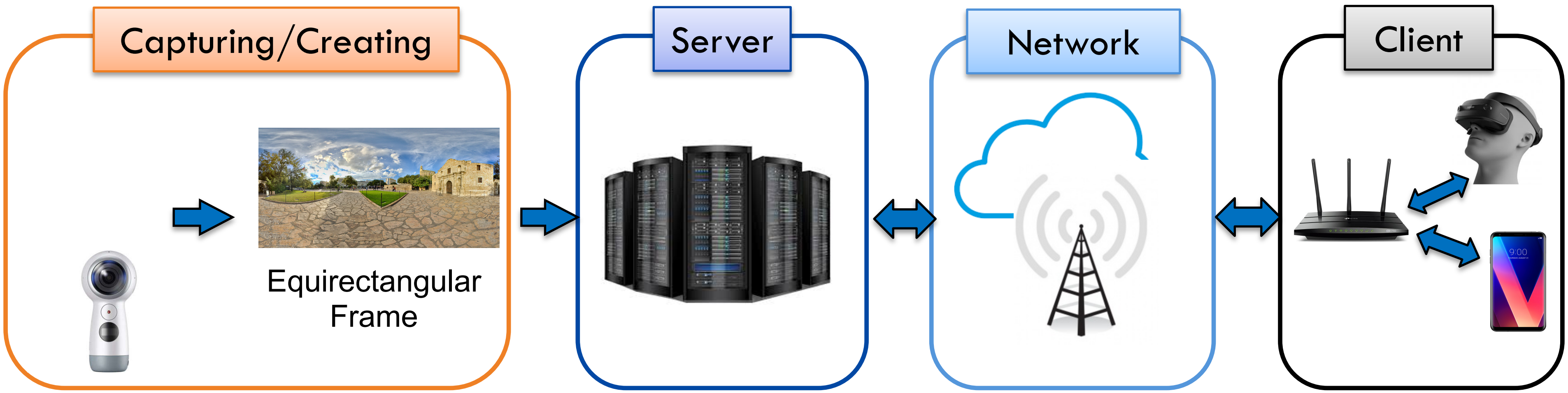
- Lower bit-rate due to low bandwidth
  - Blurred image
- Video quality fluctuations
- Rebuffering
- High thermal and power dissipation at client device





New approaches

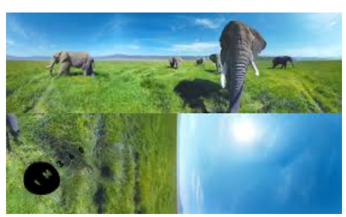
– At every point in end-to-end delivery



New approach – Tile based streaming

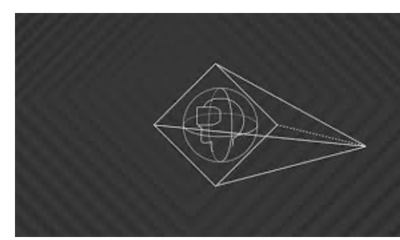
- Tile based video representation and streaming
- Panoramic frame is divided into tiles.
- Only selected tiles in user's FoV streamed from the server.
- Store panoramic frames in multiple quality levels at the servers
- Project 360° video frames in different formats
  - YouTube: Equirectangular (EQ, almost deprecated), Equi Angular cubemap (EAC)
  - Facebook: Cubemap, Pyramidal projection

EQ



EAC

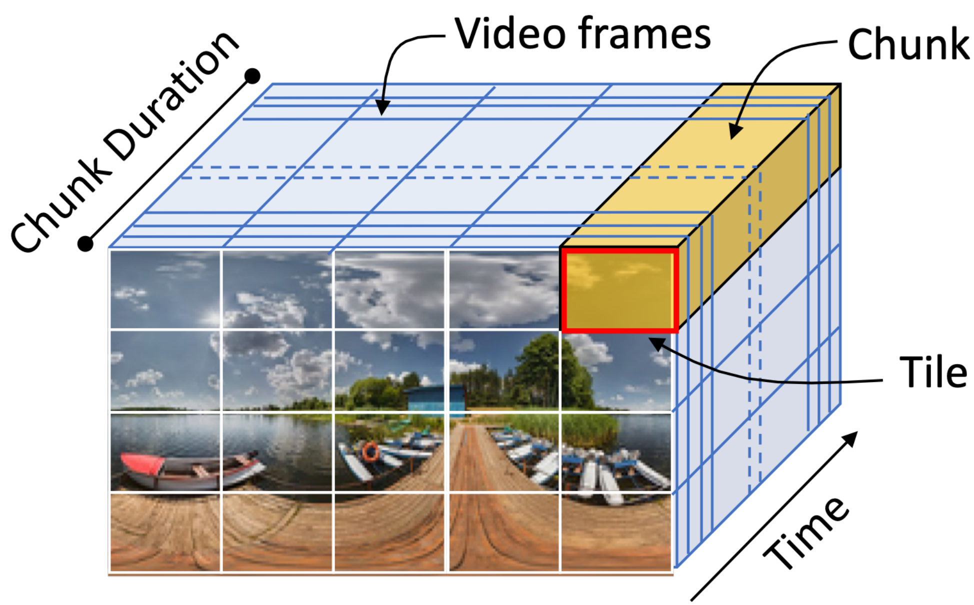
Cubemap

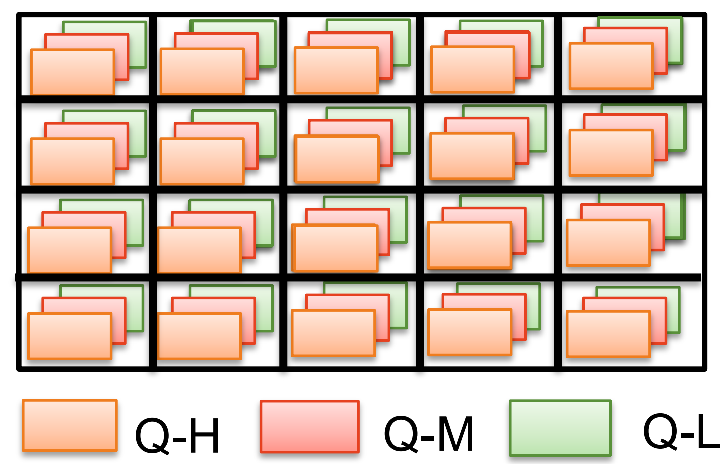


Pyramidal

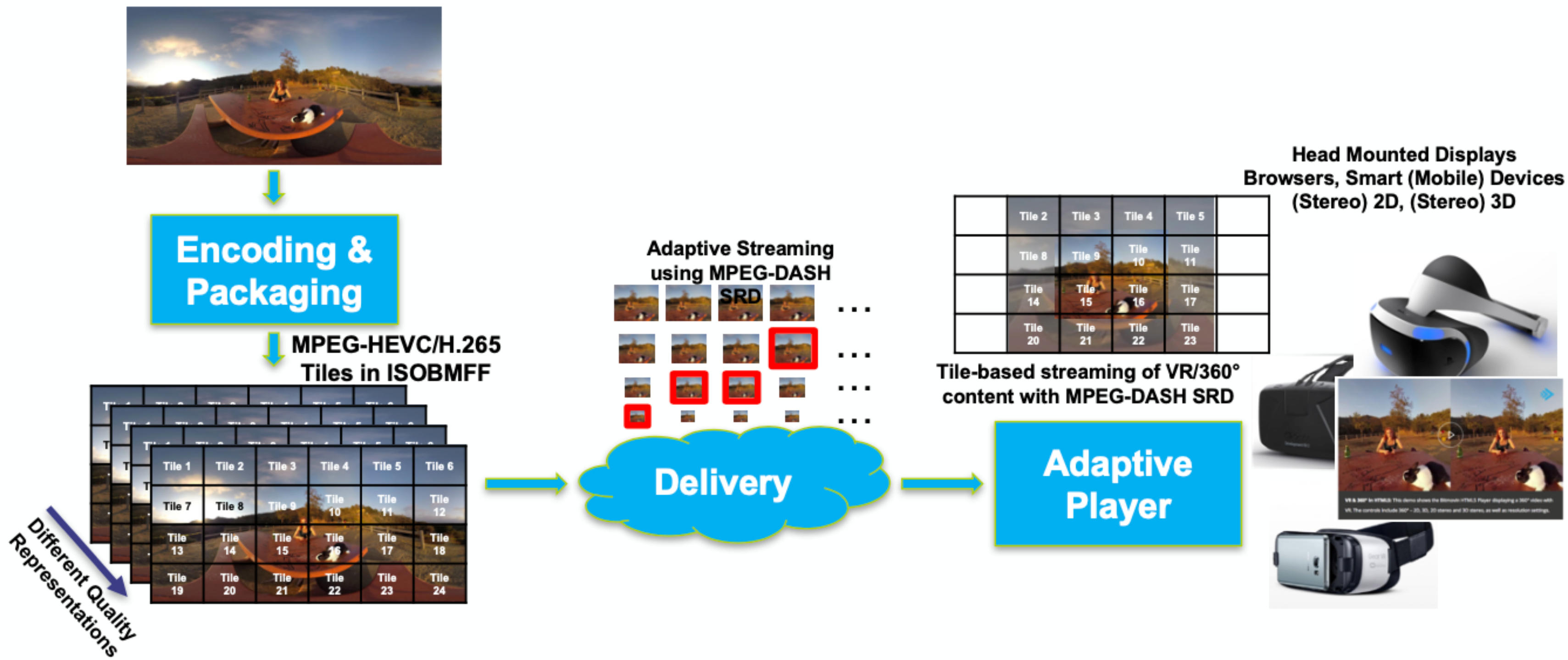
Tile based streaming

- Only selected tiles in user’s FoV streamed from the server.
- Every tile is pre-encoded in multiple bitrates/qualities
- DASH for fetching tiles



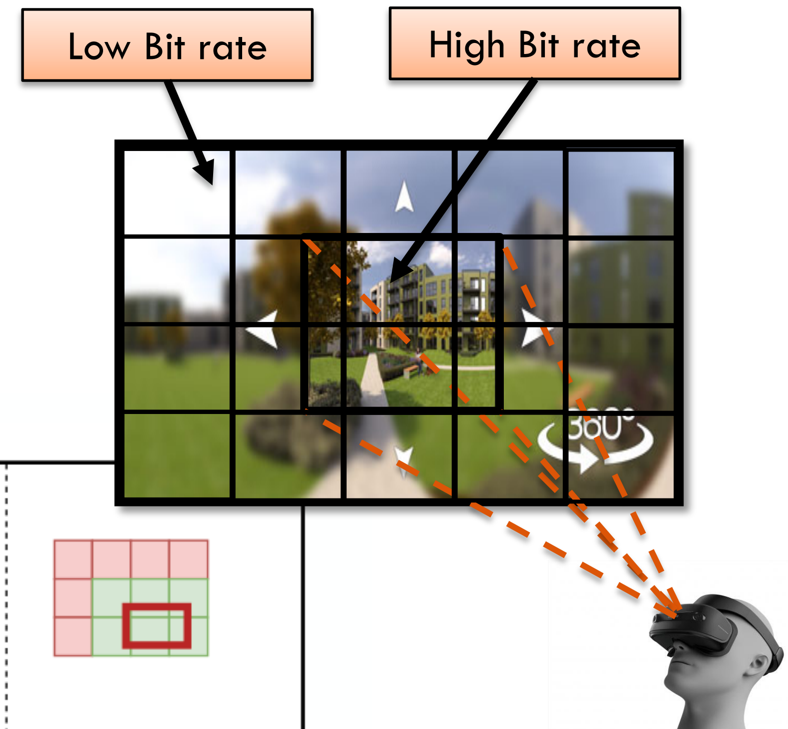


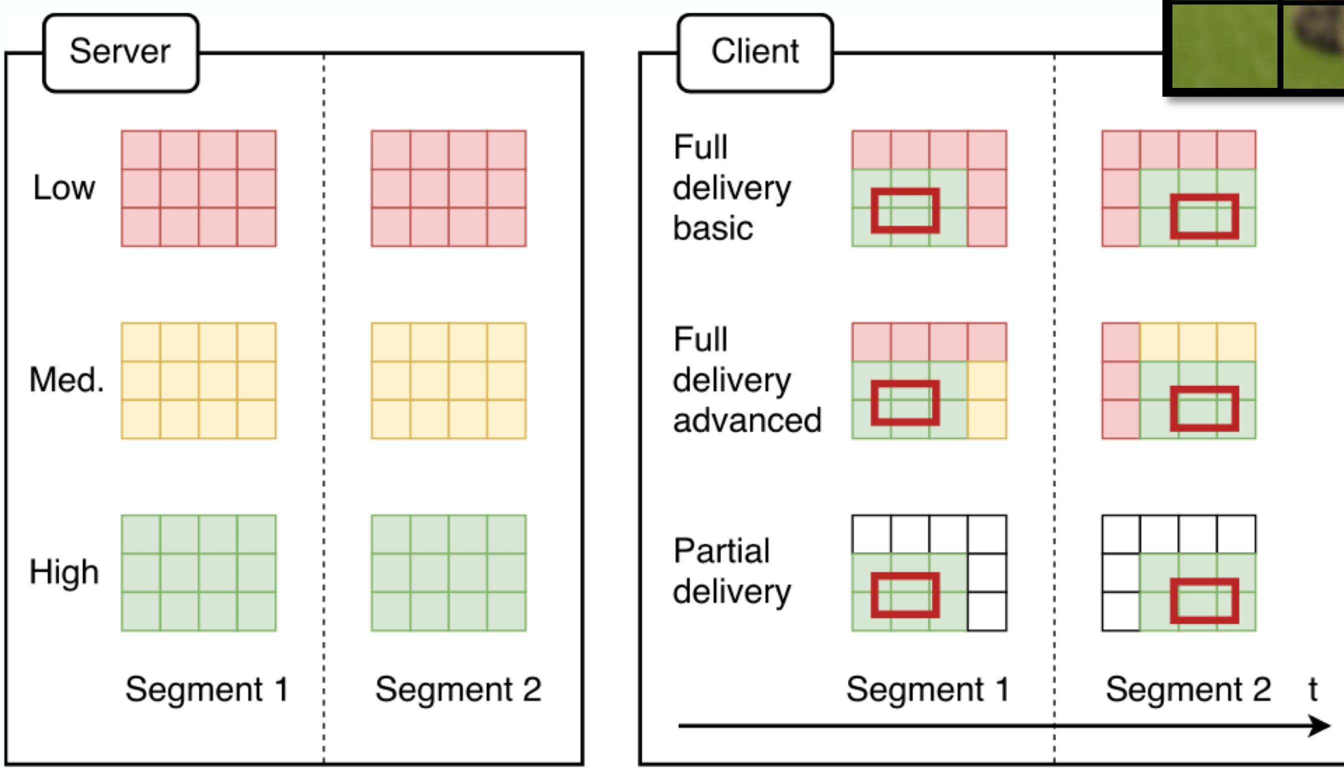
Tile based streaming – DASH enabled streaming



Tile based streaming

- Quality of the tiles
  - High : in FoV
  - Low : in OoV/ near FoV



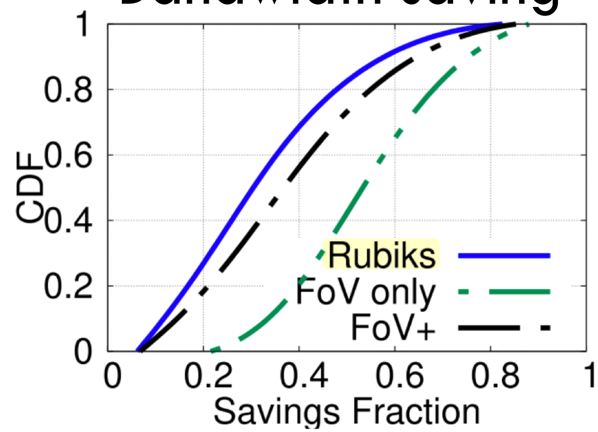


Tile based streaming - Savings

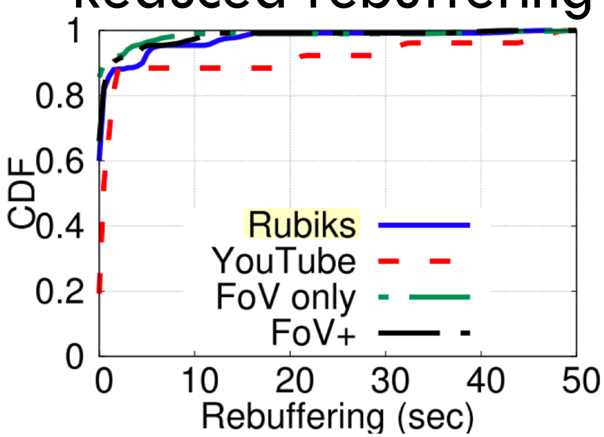
Advantage of Tile based streaming

- High **bandwidth** saving
- Reduced **rebuffering**
- Reduced **Quality changes**
- Improved **Video quality**
- Improved **QoE**

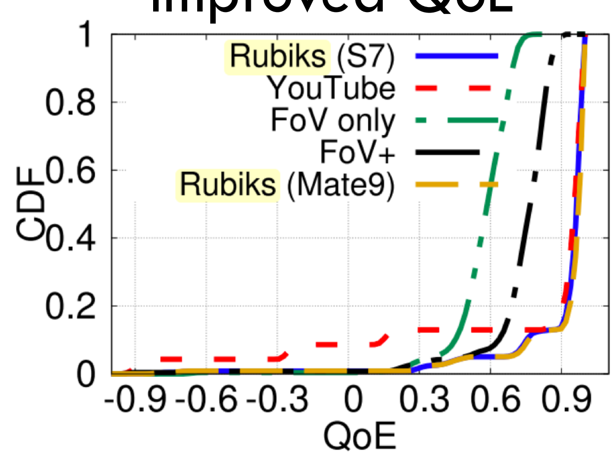
Bandwidth saving



Reduced rebuffering



Improved QoE



Tile based streaming – Network adaptation

Edge caching for tile-based streaming.

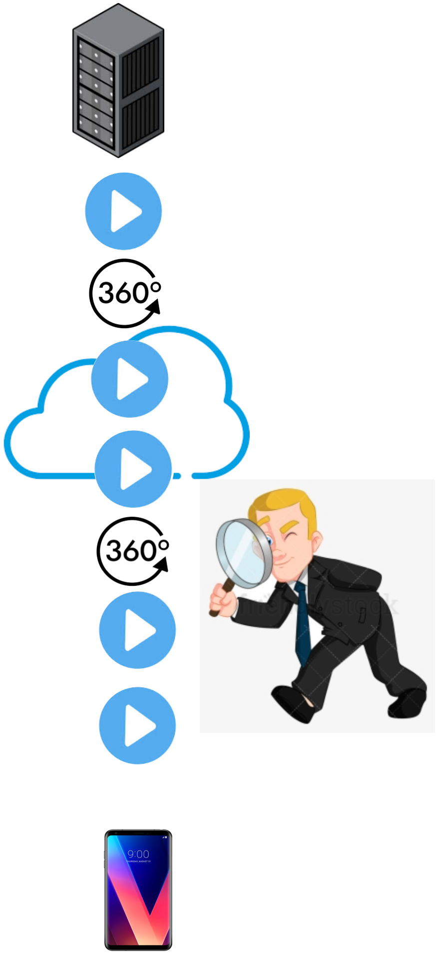
- Edge servers are more closed to the client
- Keep a track of
  - Tile location
  - Tile quality
  - Demand for the tile
- Store tiles with high priority
- When the same tile is requested by a different user, send data without connecting to main content server[1].
  - Reduce delay in network data transmission

[1] *Edge-Assisted Rendering of 360° Videos Streamed to Head-Mounted Virtual Reality – Lo et al. (ISM 2018)*

Tile based streaming – Network adaptation

Identification of 360/VR in advance

- Efficient network resource allocation
- Increase user QoE
- Identify geographical distribution of 360° video streaming.
- Enhance traffic shaping and policing



Tile based streaming – Network adaptation

Why prior identification is difficult?

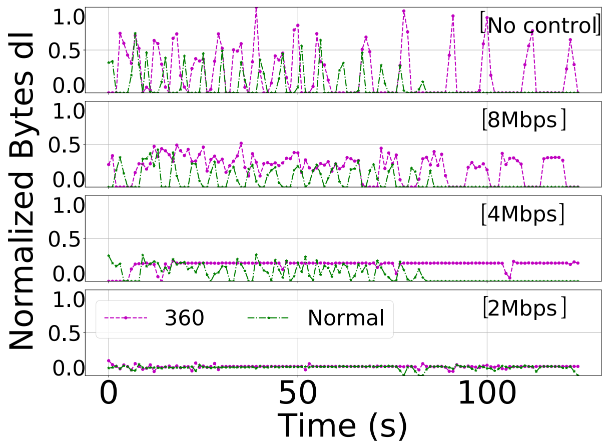
Network data is encrypted

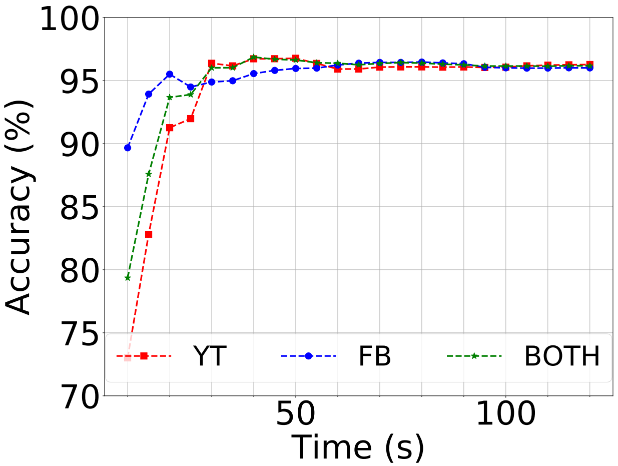
But?

Still different patterns for 360°/normal

ML based solutions are viable[1]

- With in first 30s, given YT/FB video can be classified to 360°/normal
- For YT: 97% & FB: 95% accuracy





Tile based streaming – Client side

Highly optimized HMDs (Head Mounted Device)

Majority of client improvements are for smartphones, but HMDs play vital role in VR streaming

- Facebook Oculus
  - Dedicated VR HW system
  - VR/360 video

- Samsung Galaxy VR
  - Smartphone mounted
  - VR/360 video

- Microsoft HoloLens/Magic Leap one
  - Dedicated MR (Mixed Reality) system
  - MR/VR/360 video

Tile based streaming – More work at the client

New apps/libs for tile-based streaming

Head movement prediction

– Predict the yaw, pitch and roll head movement

– Optimal values: from1-2s past window to 0.5-1s future window

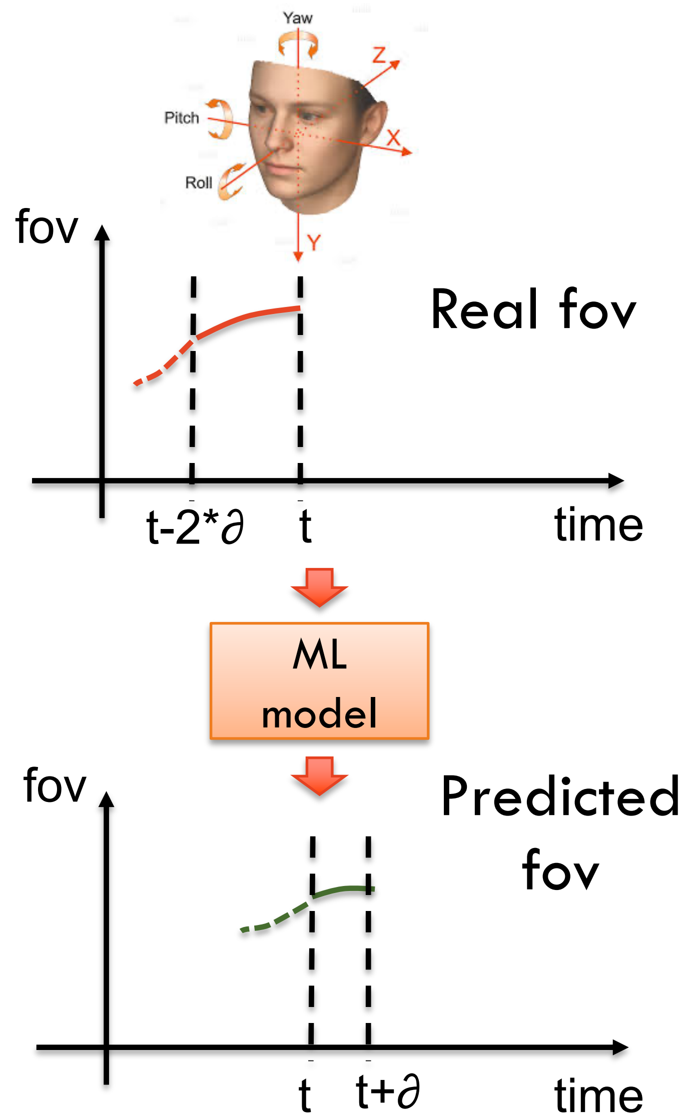
– Different ML/NN approaches

  – LR (Linear Regression)

  – RNN/LSTM

  – User based FoV trajectory clustering

– Select tiles based on the predicted FoV

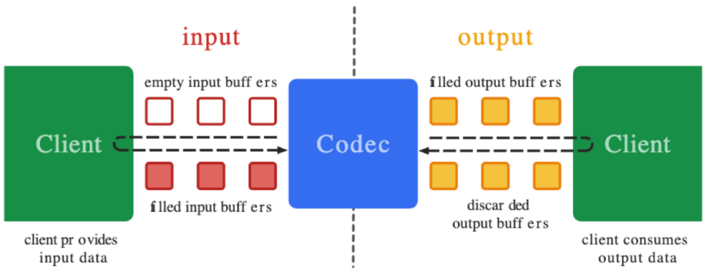


Tile based streaming – More work at the client

New apps/libs for tile-based streaming

High efficient parallel decoding

- Tile are encoded (H264/H265)
- Single decoder can not meet required motion-to-photon delay[1] (25ms)
- Parallel decoders : optimum 4
- **Android MediaCodec** HW decoders
- https://developer.android.com/reference/android/media/MediaCodec



[1] Cutting the Cord: Designing a High-quality Untethered VR System with Low Latency Remote Rendering – Luyang et al. (MobiSys 18)

New approaches – Client implementation (cont.)

New apps/libs for tile-based streaming

Tile based spherical players[1][2]

- Currently, entire panoramic frame is rendered to spherical canvas
- With tiles, selected area of sphere will be covered
  - Construct the frames from tiles
  - Render the frame to spherical canvas
- **Android OpenGL-ES**
  - **https://developer.android.com/training/graphics/opengl**

[1] Rubiks: Practical 360-Degree Streaming for Smartphones – He et al. (MobiSys 2018)

[2] Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices – Qian et al. (mobicom 2018)

New approaches – Client implementation (cont.)

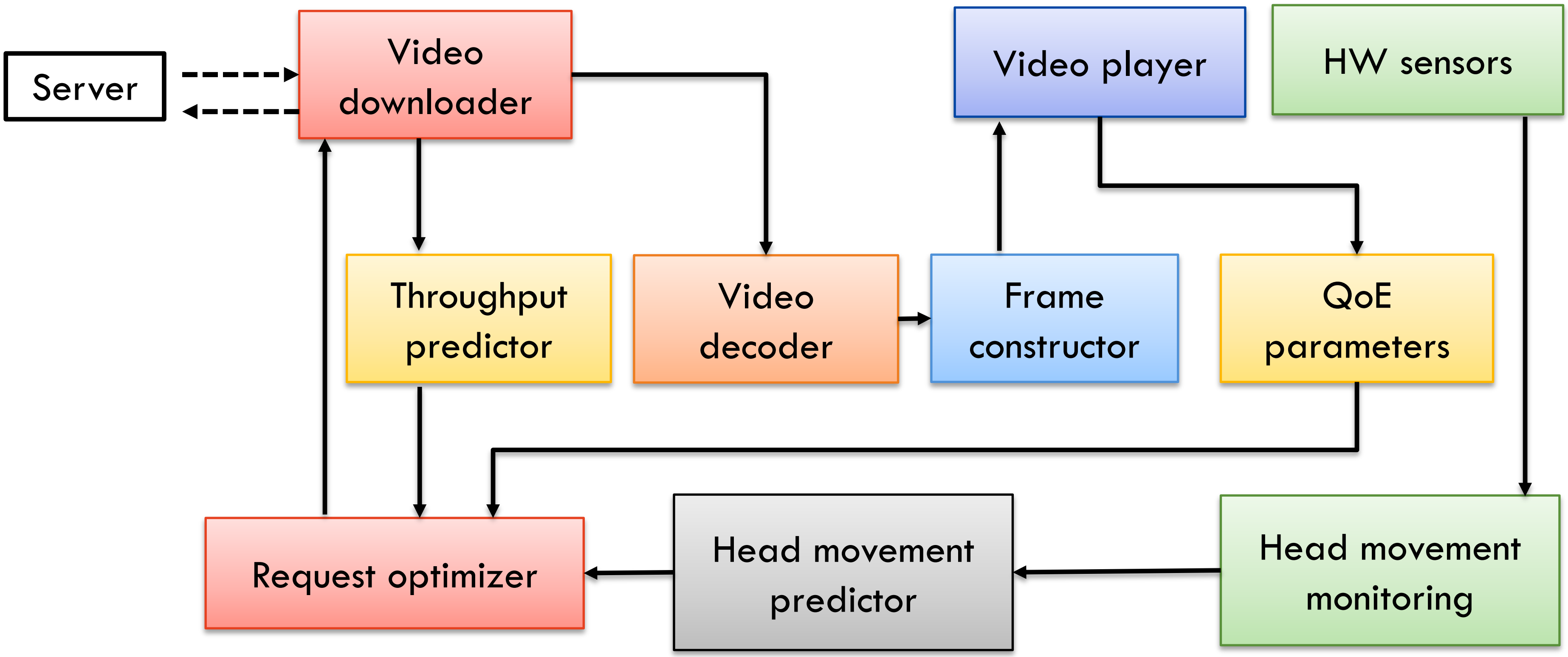
New apps/libs for tile-based streaming

Algorithm implementations in efficient way

- Network Throughput / Head movement prediction
- QoE metrics (Current Video quality/ Quality changes)
- Taking above info, Request optimizers select
  - What tiles?
  - What quality level (bit rate)?
- **Android Native NDK (C++)**
  - **https://developer.android.com/ndk**

New approaches – Client implementation (cont.)

Sample client implementation at smartphone[1]



[1] *Rubiks: Practical 360-Degree Streaming for Smartphones – He et al. (MobiSys 2018)*

So much more work to do…

E.g. How to improve Cyber sickness?

Project Final Submission & Demo

Week 10 - COMP5216

THE UNIVERSITY OF SYDNEY

Project Final Submission

- **What/When do you need to submit?**
  - Electronic submission 1 – **5:00pm, 06/11/2020 (Friday, Week10)**
  - Presentation & Demo – 5:00pm, 09/11/2020 (Week 11 lecture)
  - Electronic submission 2 – 5:00pm, 09/11/2020 (Week 11 lecture)

- **Electronic submission 1 (preliminary):**
  - Final report
  - Video
- **Electronic submission 2:**
  - Final report
  - Video
  - Presentation
  - Code
  - APK file
- Submission 2 will be used for marking

Project Final Submission

- **Presentation & Demo**
  - Week 11 – Lecture period will be three hours long (no tutorials)
  - Starts at 5:00pm.
  - We have 30 groups
  - 3 minutes to present
  - You can share your screen and do the live demo with the emulator or you can use a camera and a mobile phone or phones.
  - 2 minutes demo and Q&A.
  - It is your responsibility to arrange all technical requirements to successfully demonstrate the key features.

Presentation Schedule – Week 11

Time	Group	App name
5:06-5:11	1	Uni-CODEX
5:12-5:17	2	Food-Fair
5:18-5:23	3	Chef Group
5:24-5:29	4	Exposure
5:30-5:35	5	Focus On
5:36-5:41	6	Study help app
5:42-5:47	7	Easy to Buy
5:48-5:53	8	Study Habbits
5:54-5:59	9	TickTask
6:00-6:05	10	Game Forum
6:06-6:11	11	USYD Connect
6:12-6:17	12	Bee (Group formation)
6:18-6:23	13	Green Fingers
6:24-6:29	14	Lose Weight Helper
6:30-6:35	15	Health Keeper
6:36-6:41	16	Co-Shopping
6:42-6:47	17	Travel Guide
6:48-6:53	18	PetsH
6:54-6:59	19	Livery
7:00-7:05	20	Orphan donations
7:06-7:11	21	Pet Recipie
7:12-7:17	22	Wardrobe
7:18-7:23	23	Clinical Tasks
7:24-7:29	24	Anonymous social app
7:30-7:35	25	Wage Theft
7:36-7:41	27	Room mate
7:42-7:47	28	Pet Boss
7:48-7:53	30	Makeup Booking
7:54-7:59	33	Team Mates
8:00-8:05	35	Image search

Submissions - Final Report [7 marks]

1) Introduction to the app: succinctly describe the problem and the app.

2) Validation of the app: for each item in the minimum feature set, provide an experimental validation that you have managed to successfully implement the proposed techniques. For examples,

   - a. Results of a user study to reflect the effectiveness of the GUI.
   - b. Experimental validation of the effectiveness of the optimization techniques (bandwidth/computation/energy saving mechanisms).

3) Challenges and setbacks: explain whether you were able to achieve goals proposed at the proposal phase. If not, explain reasons for taking different paths.

4) Next steps: identify what is missing from your current implementation and explain how to plan to take your app to the next level.

5) References

6) Appendix: documentation and manual.

Submissions

6) Appendix: documentation and manual.

You also need to write a documentation/manual as an Appendix to the final report to guide a potential user on how to set up the working environment of your application and re-compile and re-deploy your app to a mobile device.

The final report must be of Adobe Acrobat Portable Document Format (*.pdf) format. No other file format is accepted. The final report must **NOT** exceed **TWELVE** pages including references (single space and font size 12 for body text).

Submissions

- **Video [3 marks]**

The introduction video must be compatible with the VLC media player and .mp4 file format is preferred. The video should not be longer than 3 minutes.

- **Presentation [2 marks]**

The project presentation is to pitch your app. Each group has maximally **3 minutes** to present the assignment using the presentation material submitted. The presentation will be generally starting at the lecture time (i.e. 5:00pm) from Group 1 in the weekly lecture room. You should be familiar with the presenting via Zoom. The presentation will be marked in terms of clarity, attitude/confidence, presentation skills and content.

- **Demo [3 marks]**

Each group has maximally **2 minutes** after the presentation to demonstrate the key features of the app. The app should be installed on a mobile device. Successful demonstration of the key features of the app, and the readiness to distribute will be evaluated.

It is your responsibility to arrange all technical requirements to successfully demonstrate the key features, e.g. internet access, multiple mobile devices in case of a collaborative app, other supporting devices such as wearables, access to cloud services, etc. prior to the presentation time slot.

Submissions

- **Source code of the app [5 marks]**

Export of the project development environment as a zip file and an APK file (or a suitable method/format to install the developed app on a mobile device) should be submitted. Successful installation of the app on a real device, testing of key features of the app, challenges and effort in coding will be evaluated.

Individual Contributions

- "Each group member must contribute to the assignment equally and the members will be awarded the same marks. Under certain circumstances, adjustment of marks may happen to group members at the discretion of the course coordinator."
- App development is always going to be group work.
- In real-life, every group member is not going to be one of your friends
  - Thus, navigating collaboration issues are part and parcel of the challenge.
  - Resolve the problems now. Talk to me !

- Due to multiple requests, I have decided to give you the option to submit a peer-feedback form. (Optional)
  - Individual submission. (No marks allocated)
  - However, if you receive entirely negative feedback from all members, it may lead to reduction in marks.

Project Final Marking

- **Mark Allocation**
  - The app [10 marks]
  - Novelty of the problem, creativity of the solution, challenges in developing the solution and the amount of effort in developing the solution will be evaluated by a panel of judges. **All deliverables will be considered in this evaluation including the in-class presentation and demo.**
  - Source code [5 marks]
  - Final report [7 marks]
  - Video [3 marks]
  - Presentation [2 marks]
  - Demo [3 marks]

- Panel of judges includes two external evaluators;
  - Suranga Seneviratne from USYD
  - Glenn Stephens from Microsoft

- **Good Luck !**