EECE5512
Networked XR Systems

Last Class - Recap

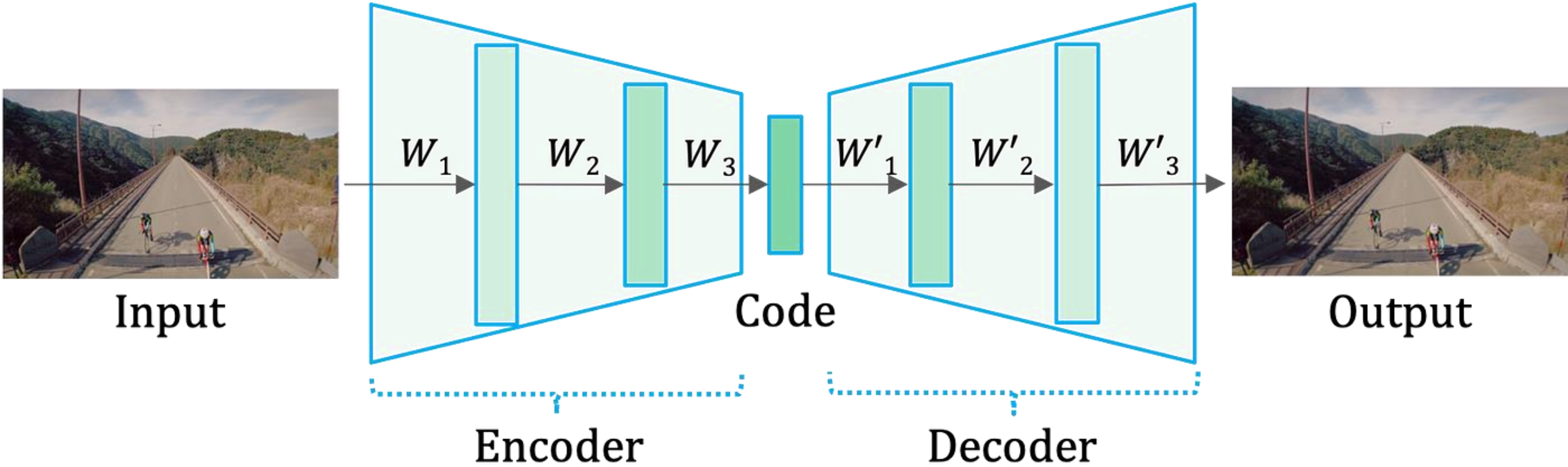
- Quiz
- Limitations of traditional Compression
- Machine Learning based Compression
    - Video
    - Point cloud
    - Mesh

Lecture Outline for Today

- Leftover topics from compression

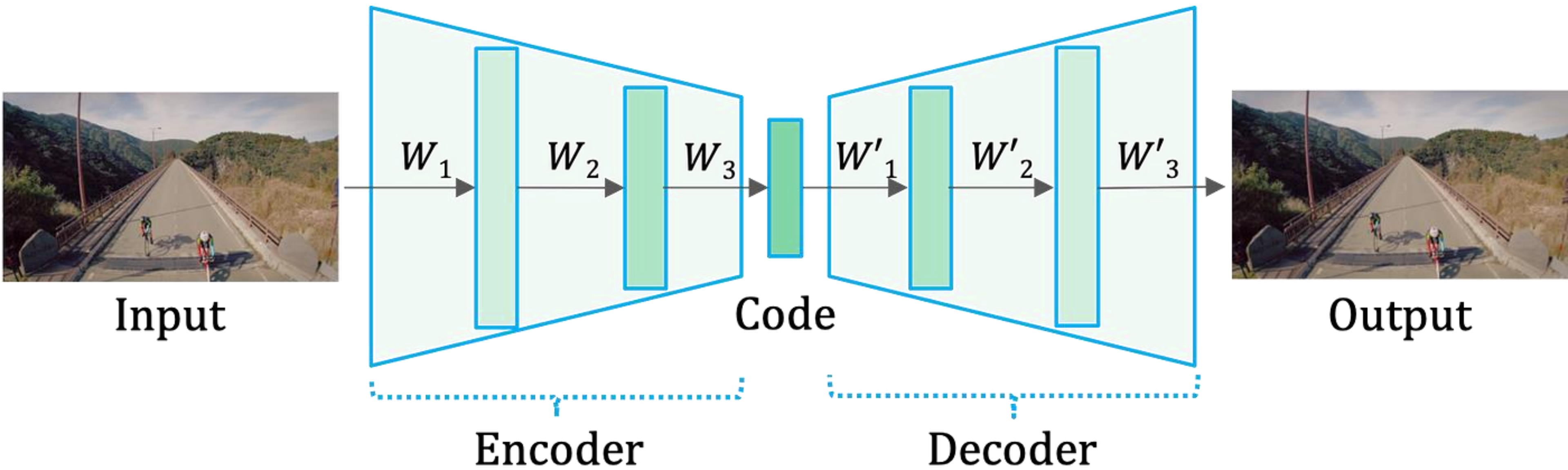
- Streaming Fundamentals
- On-demand Video Streaming
- Live Streaming
- Video Conferencing

Learned Image Compression



Spatial redundancy – Convolutional neural networks (CNNs)

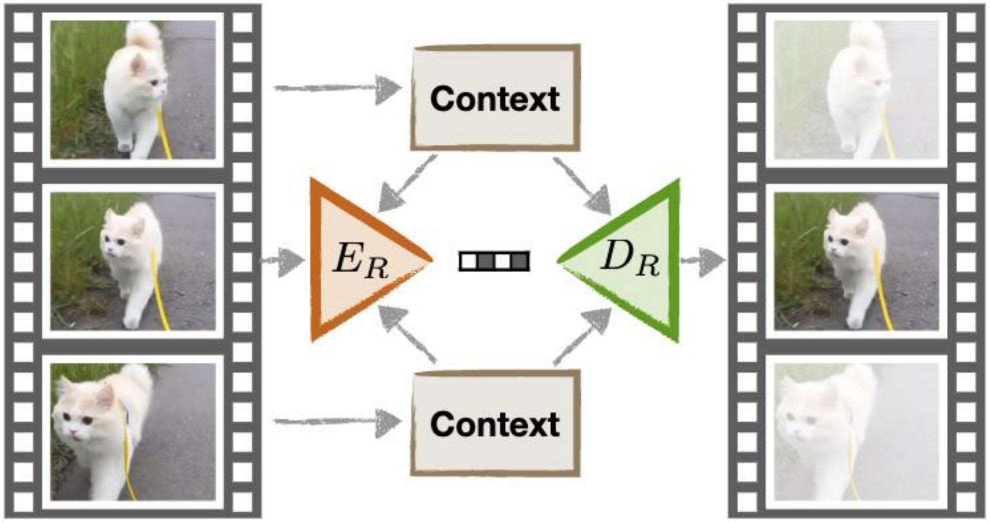
Learned Video Compression



Spatial & Temporal redundancy – 3D CNNs or LSTMs, need to estimate residuals

Learned Video Compression

- Example

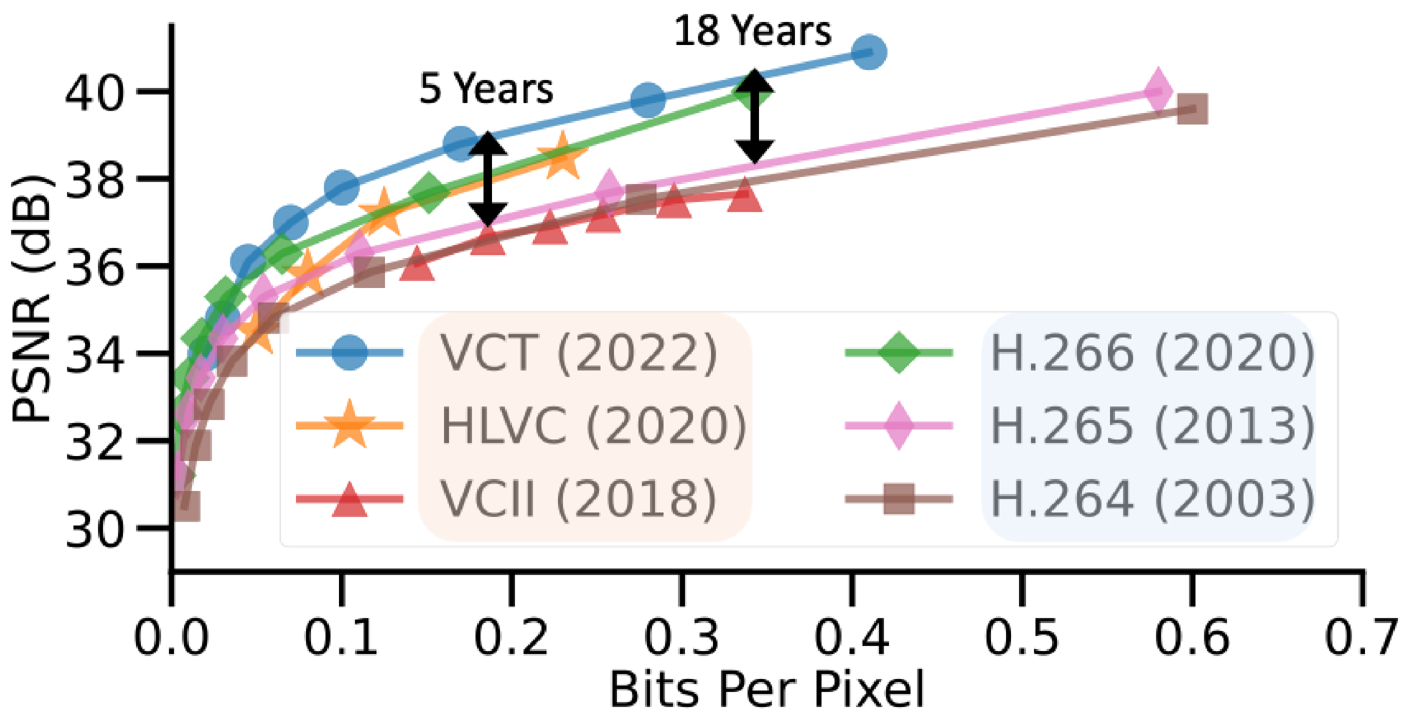


Video compression through image interpolation

Predict in-between frames from two reference frames

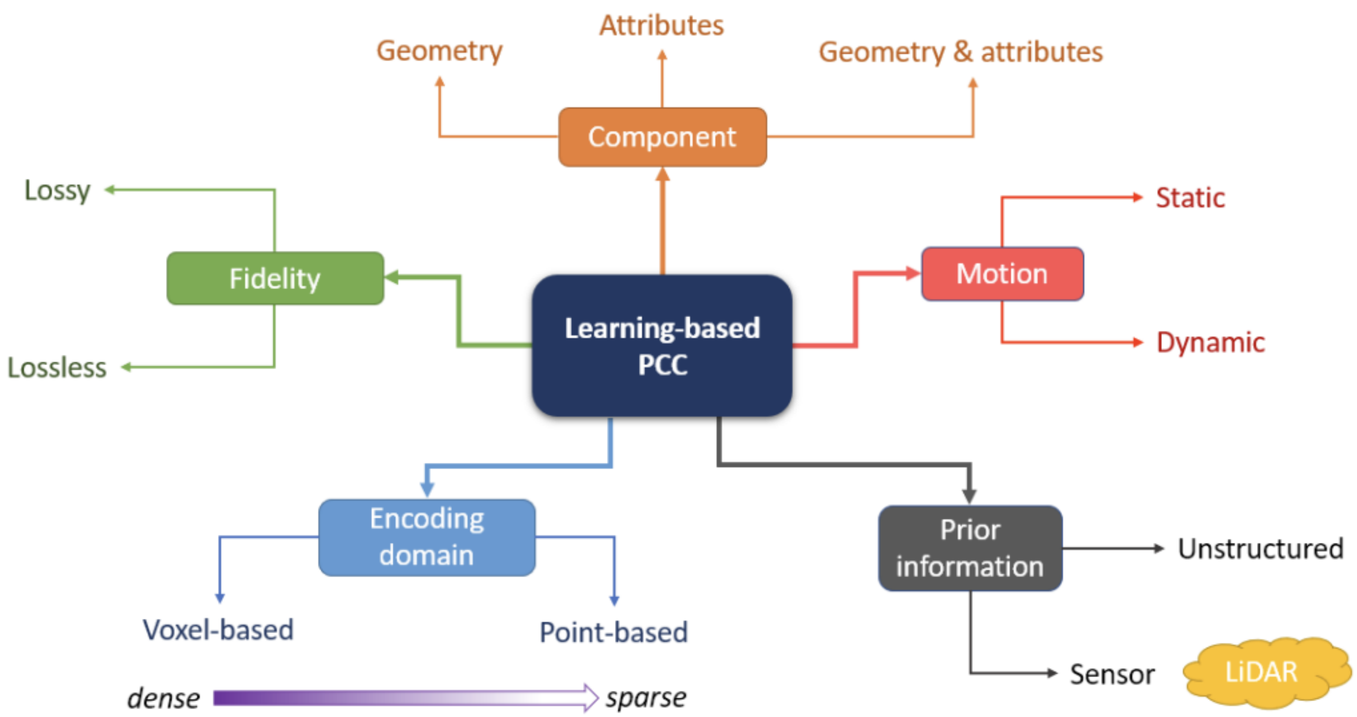
https://arxiv.org/pdf/1804.06919

Evolution of Video Codecs

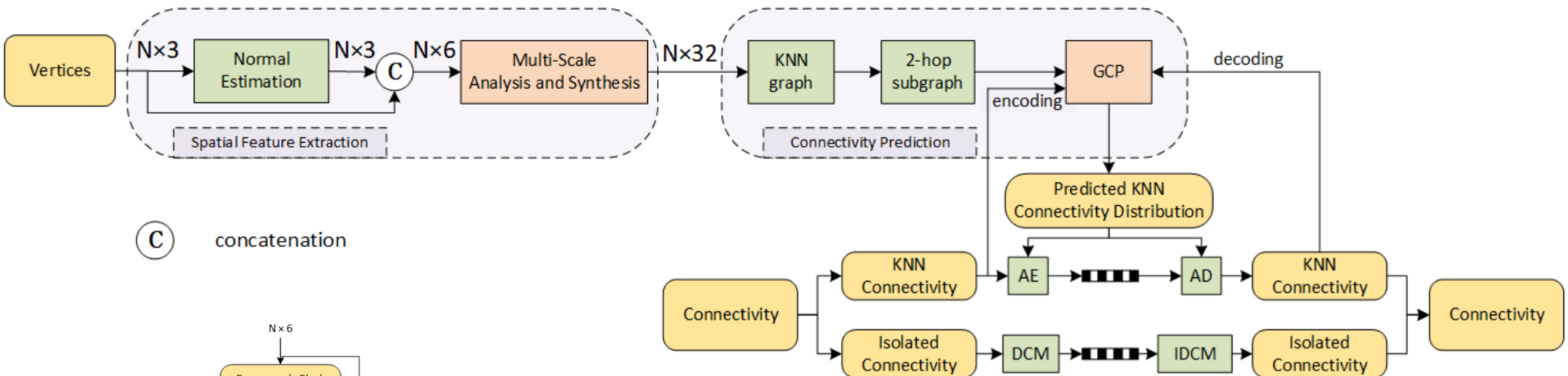


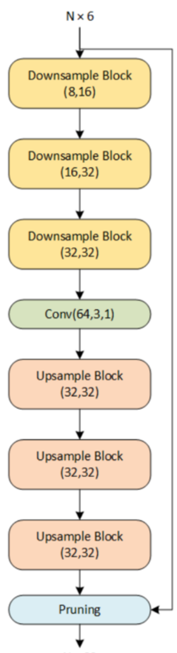
neural and classical video codecs showing
compression efficiency across generations.

Learned Point Cloud Compression



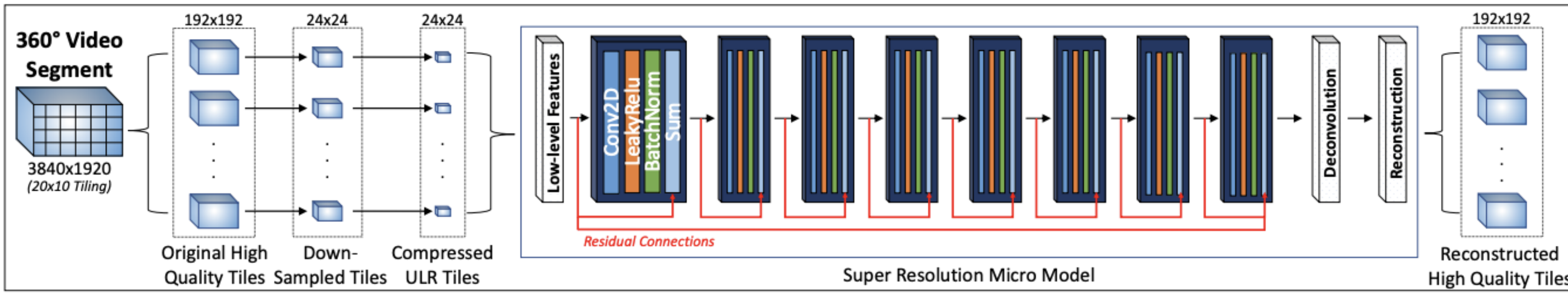
Learned Mesh Compression

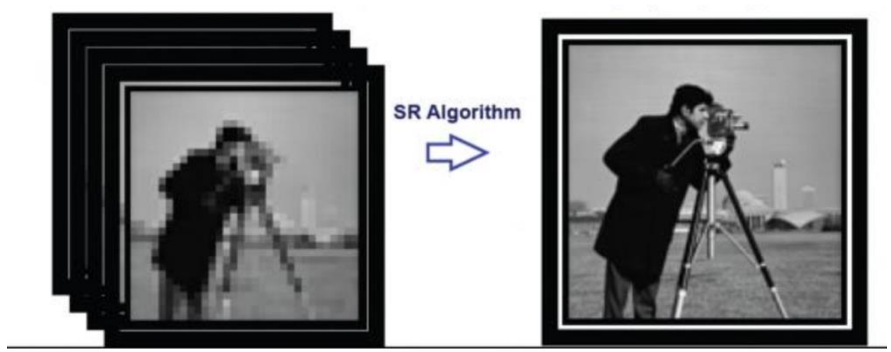




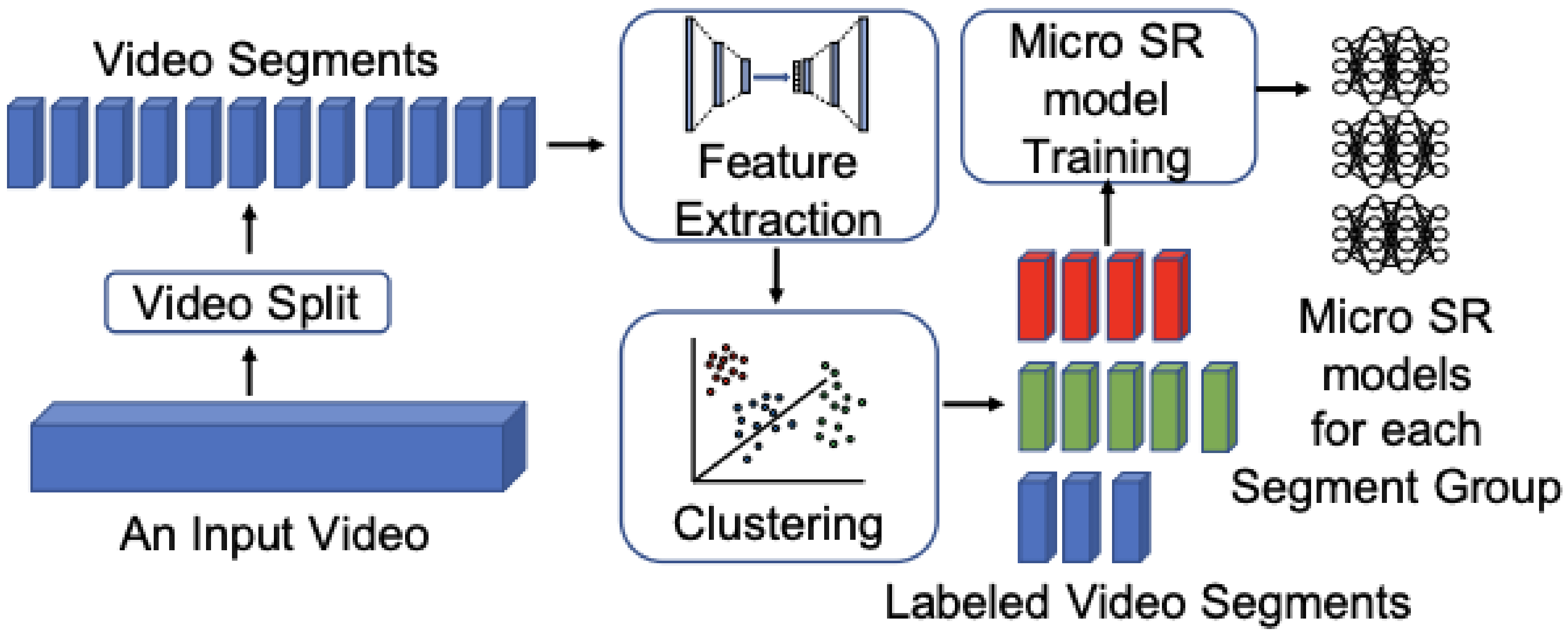
Vertex & Connectivity prediction

Super Resolution of Low-Res Content to High Res





Super Resolution of Low-Res Content to High Res



Performance Metrics

- Quality
  - PSNR
  - SSIM
  - VMAF - Netflix
- Compression ratio
- Latency
- Power consumption

Type of ML Codecs

- Generalized model
  - Train on a large-scale dataset – as much data as possible
  - Complex model
- Category-specific model
  - Train on a particular class of dataset e.g., sports or Netflix database
- Video-specific model
  - Model specific to video – memorize the conent

Limitations

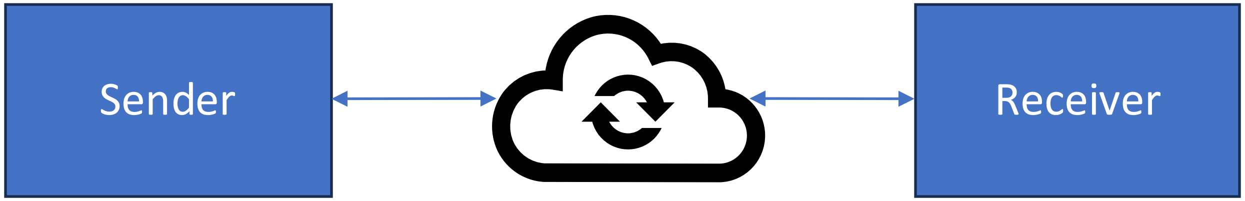
- Difficult to generalize
  - There is never enough data to train a model
  - We can circumvent this problem in certain scenarios (e.g., when streaming on-demand stored content like Netflix or YouTube)
- Not many devices have GPUs in practice
- High Power consumption

Lecture Outline for Today

- Leftover topics from compression

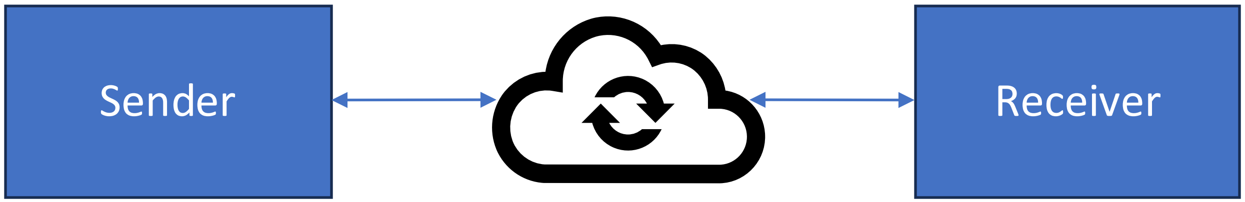
- Streaming Fundamentals
- On-demand Video Streaming
- Live Streaming
- Video Conferencing

Networked XR System

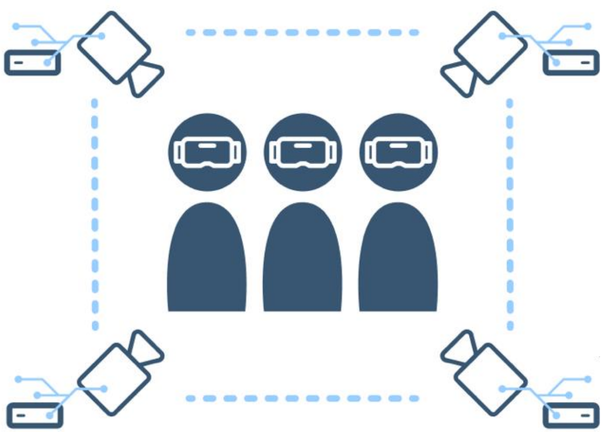
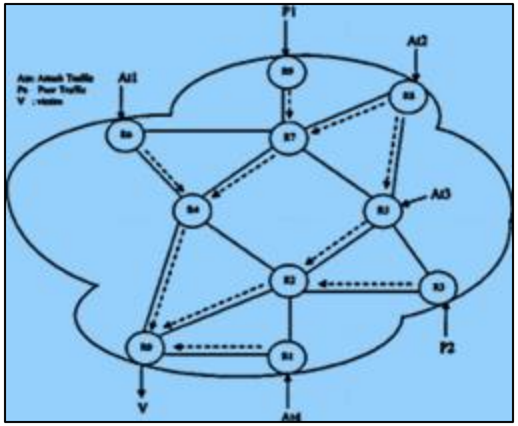


Classical networked system pipeline

Networked XR System

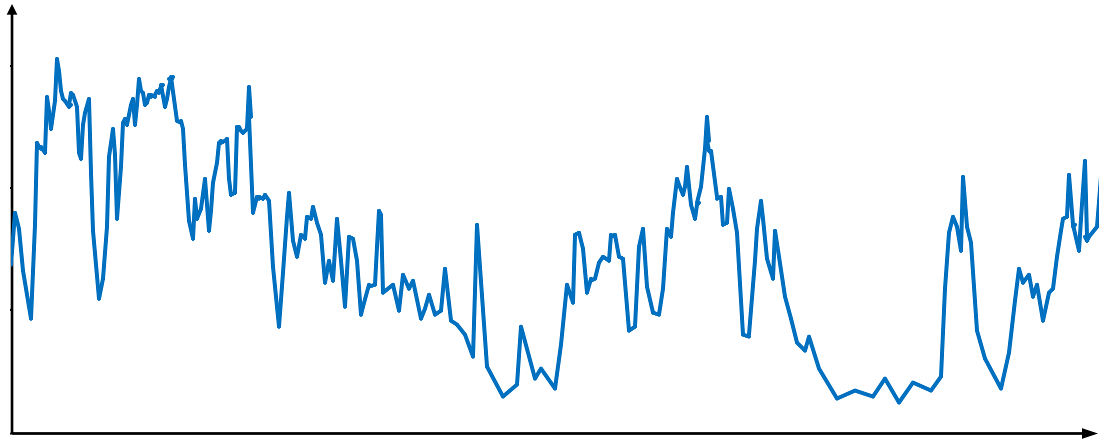


Classical networked system pipeline

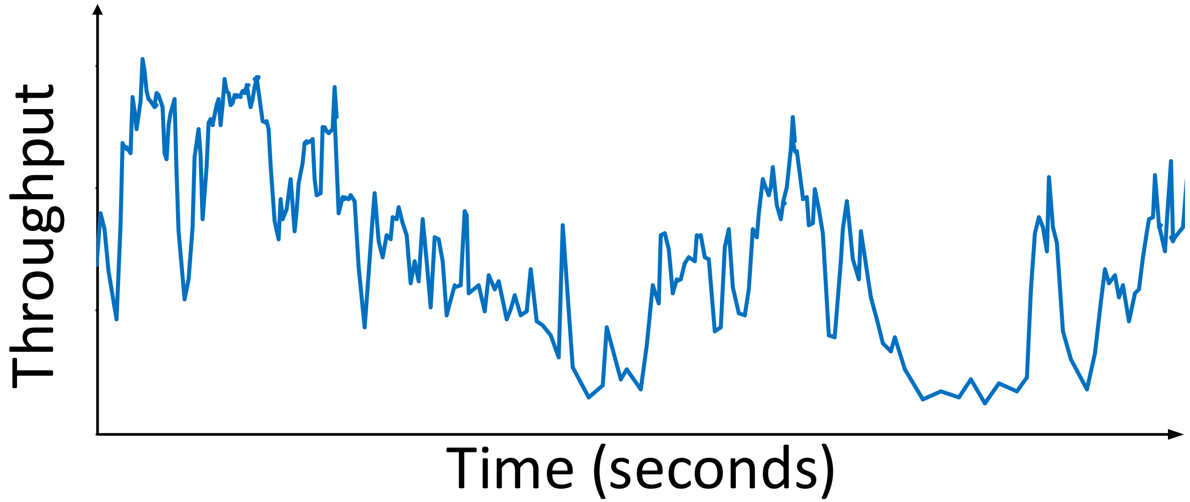

Modern day pipeline

Streaming Fundamentals



What is this graph? And what's going on here?

Streaming Fundamentals



Streaming Fundamentals

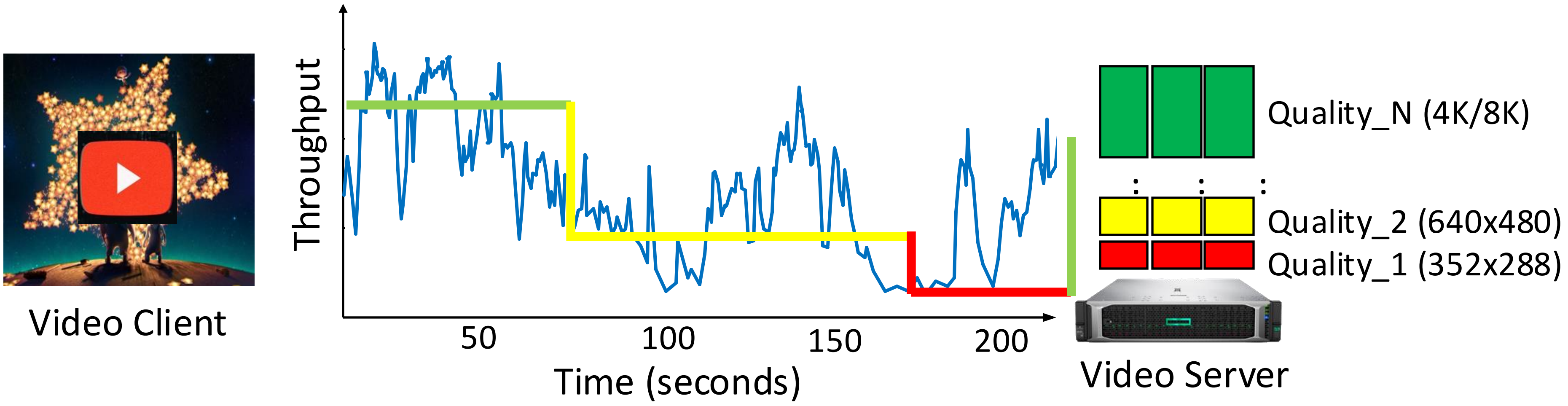
- Bandwidth
  - Wide area, wireless
- Latency
  - Transmission, packet processing, propagation
  - Router bottleneck
- Variability of bandwidth
  - Wide area, wireless
- Synchronization between network and application
  - TCP vs. application traffic control

Streaming Fundamentals

- Solutions
  - Compression
  - Streaming protocol
  - Improve network throughput
  - Tighter integration of apps with network protocols

Streaming Experience

- Given these compression principles, what's the best way to compress the content for streaming and/or storage?



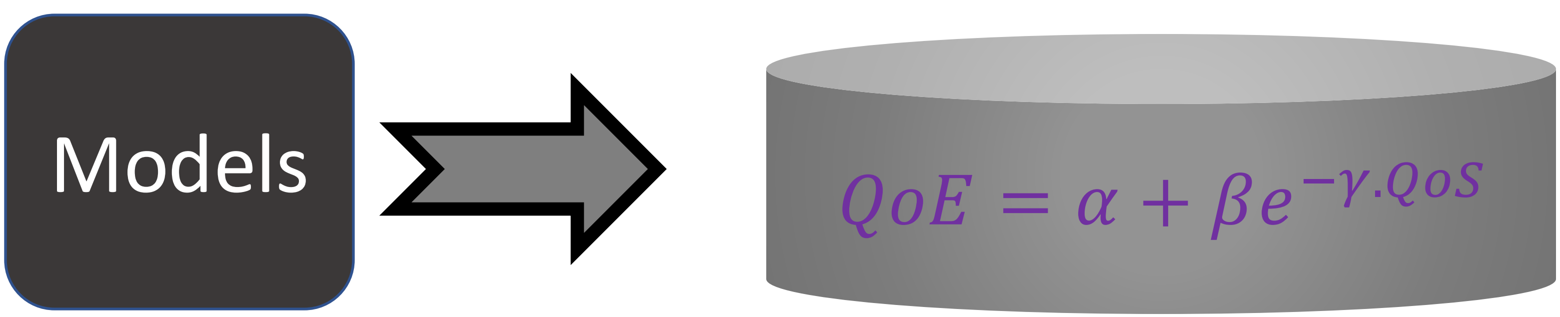
Streaming Experience

- Subjective – user quality of experience (QoE)

Streaming Experience

- Objective – user quality of experience (QoE)

QoS = *f*(network latency, throughput)



$$QoE = \alpha + \beta e^{-\gamma . QoS}$$

Fielder et. al, IEEE TON Mar’2010

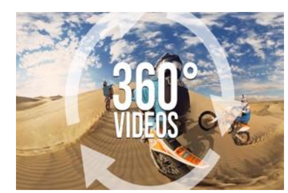
Streaming Experience

- Objective – user quality of experience

F(QoE Metrics)

$$QoE = \alpha + \beta e^{-\gamma . QoS}$$

Fielder et. al, IEEE TON Mar'2010

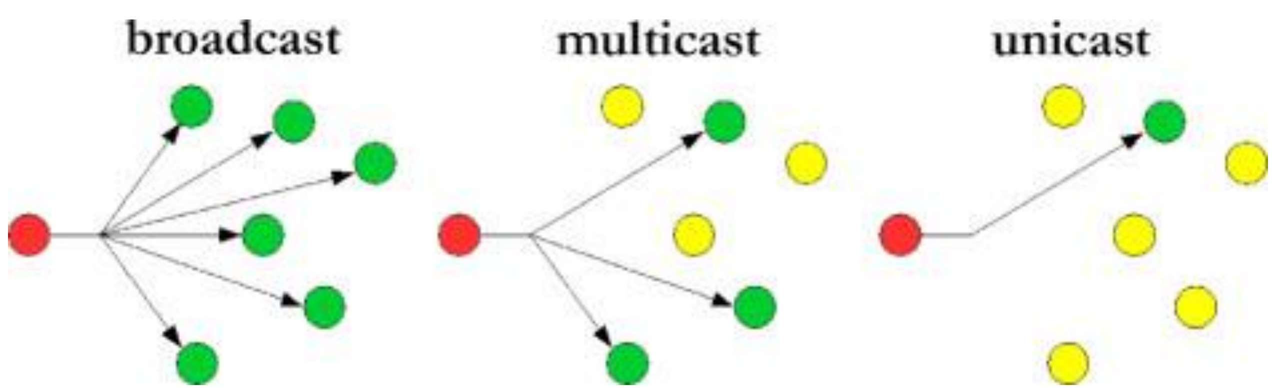
Video on Demand	Video Conferencing	Live	360º Videos	3D
NETFLIX, YouTube	Skype, FaceTime, zoom	Twitch		
Video quality Stalls Quality changes	Video quality Latency Frame rate	Quality Frame rate	Quality Stalls M-P latency	Quality Stalls M-P latency Geometry

Streaming Fundamentals

- Overall Streaming Pipeline
  - Get the video content and compress it
  - Identify the constraints (e.g., Network)
  - Define objective (user QoE)
  - Make download decisions based on the constraints maximizing the objective

Streaming Fundamentals

- Unicast
  - To one user
- Multi-cast
  - To a group of selected users
- Broadcast
  - To anyone



On-Demand Streaming

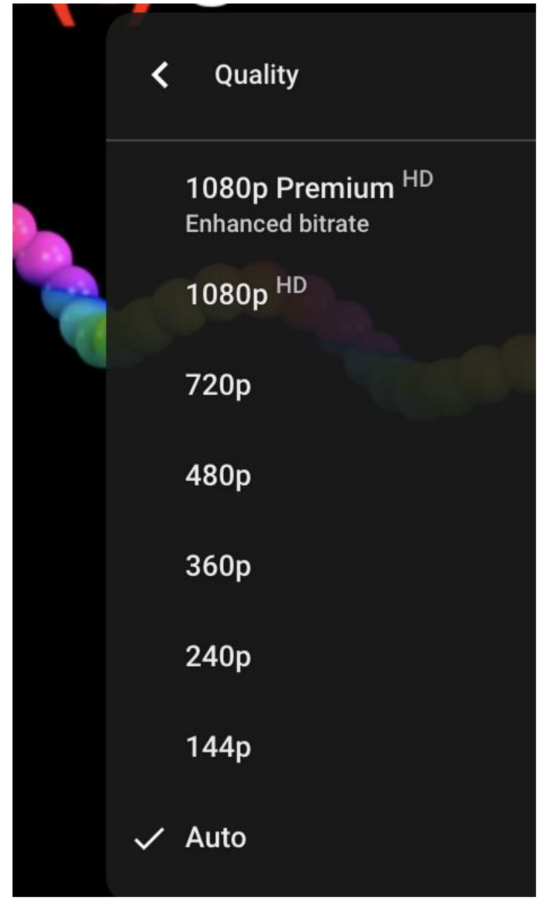
- Users can stream videos any time they want
- Opportunity to cache or pre-fetch when network conditions are good

On-Demand Streaming

YouTube

NETFLIX

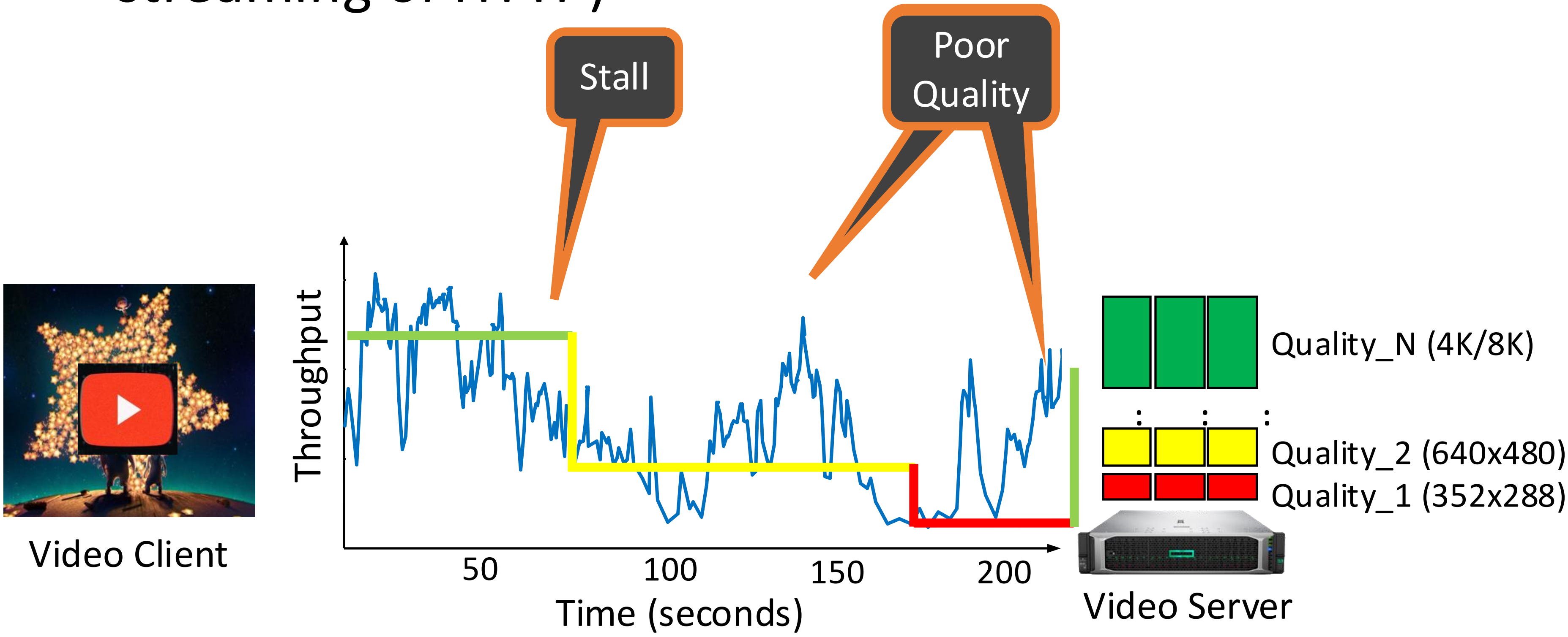
TikTok



Media is stored in different resolutions at the server

On-Demand Streaming

- Adaptive streaming – DASH (dynamic adaptive streaming of HTTP)



On-Demand Streaming

- Quality of experience metrics
- Startup latency
  - Should load the video as quickly as possible
- Stalls
  - Buffer should not be empty for playback
- Visual quality
  - More quality the better
- Fluctuations in visual quality
  - Shouldn’t change quality too frequently

On-Demand Streaming

- Need to support different user actions
  - Pause
  - Forward
  - Rewind
  - Skip or jump to a certain part of the video

- Need to re-buffer all over again

On-Demand Streaming

- Storage costs
  - E.g., Netflix stores thousands of different versions <resolutions, file formats, bitrates, …> for each video
  - Can quickly explode storage costs

Guess how many versions each movie should have?

Live Streaming

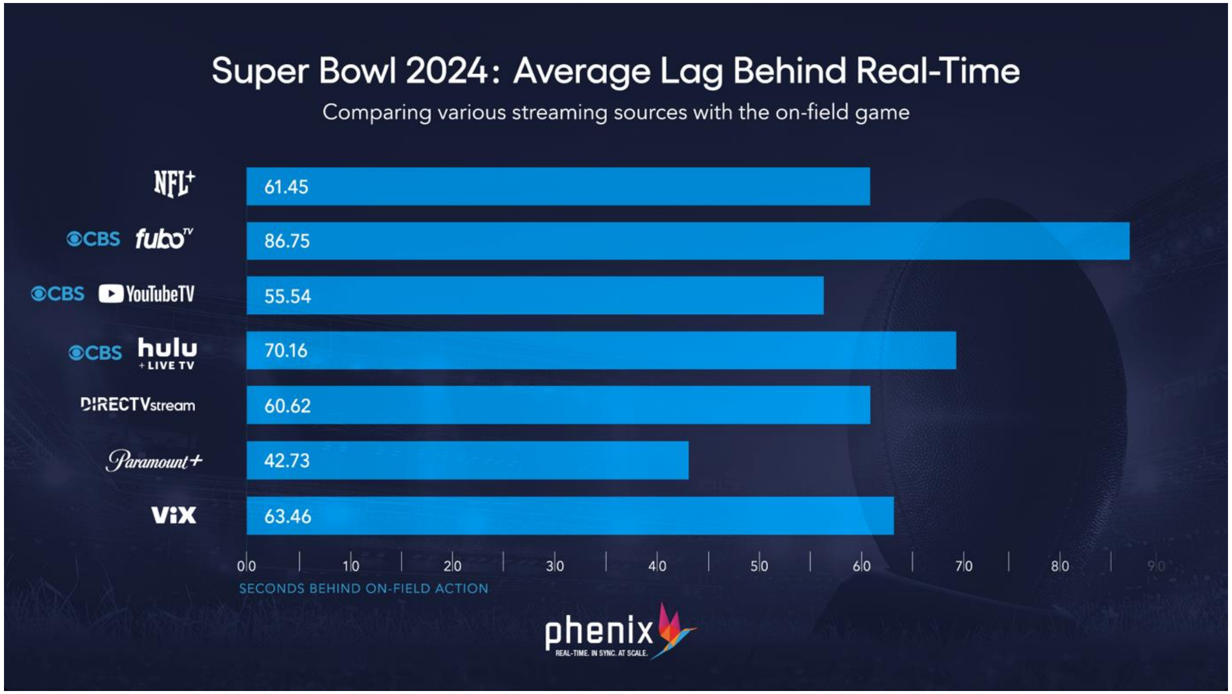
- Live (non interactive)
  - Need to support a variety of devices
  - Can afford some delay but not much

Live Streaming

- Important factors
  - Scale – how many users does the server support?
    - Transcode the video to multiple servers & distribute
  - How long the stream will be?
  - What kind of scenario?
    - Live streaming from a phone?
    - Live streaming at a concert or game?
    - Remote assistance application?

Live Streaming

- Recent super bowl live streaming latency numbers

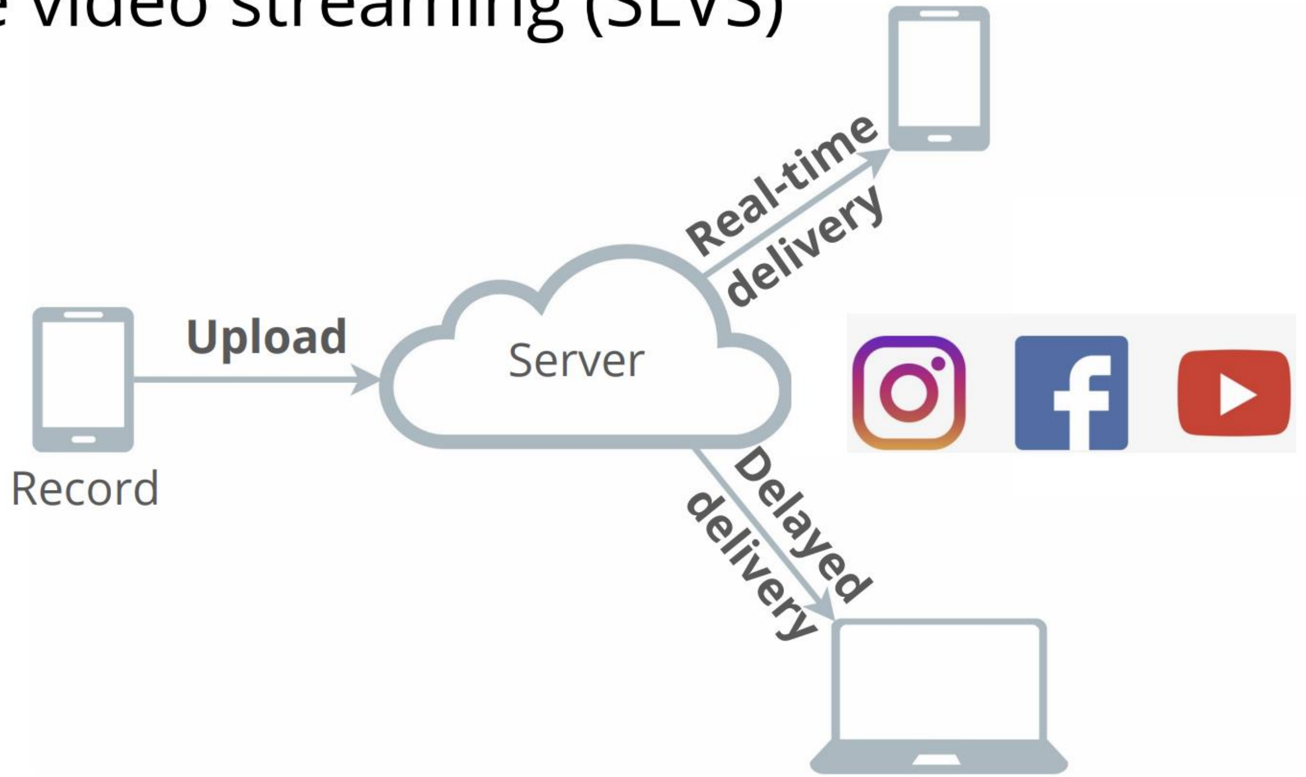


How much can you tolerate?

Source: Phoenix

Live Streaming

Social live video streaming (SLVS)



Devedeep et,al: Sigcomm'19

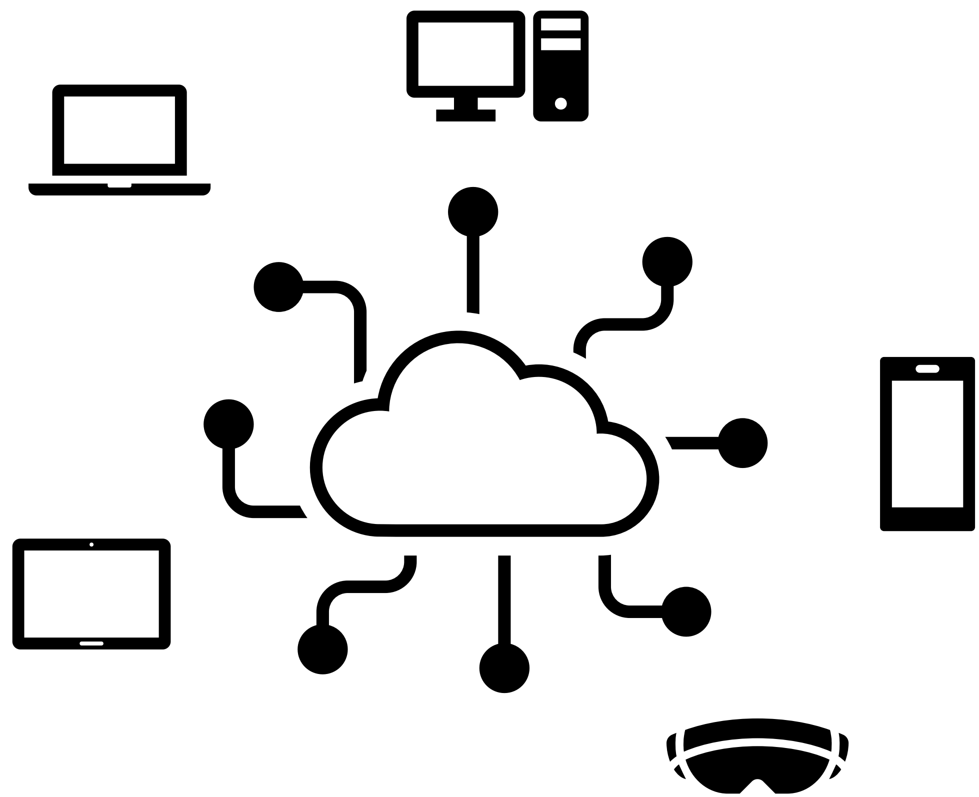
Conferencing

- Interactive
  - Need to support a variety of devices
  - Low latency

FaceTime

zoom

Conferencing

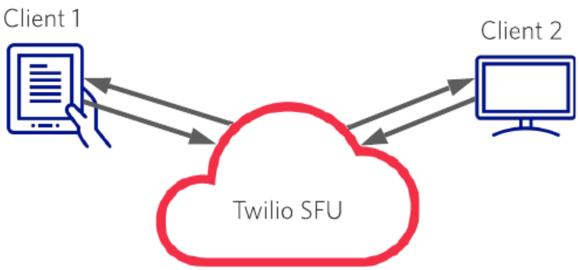


1. Peer to peer systems
2. Server relay
   Transcodes input bitstreams into different versions live and sends them to clients based on their network conditions

Conferencing

- Fewer clients – p2p is okay
- Server based is efficient for large number of clients

Client 1 communicates directly with Client 2



Client 1 communicates directly with the Twilio Selective Forwarding Unit (SFU)

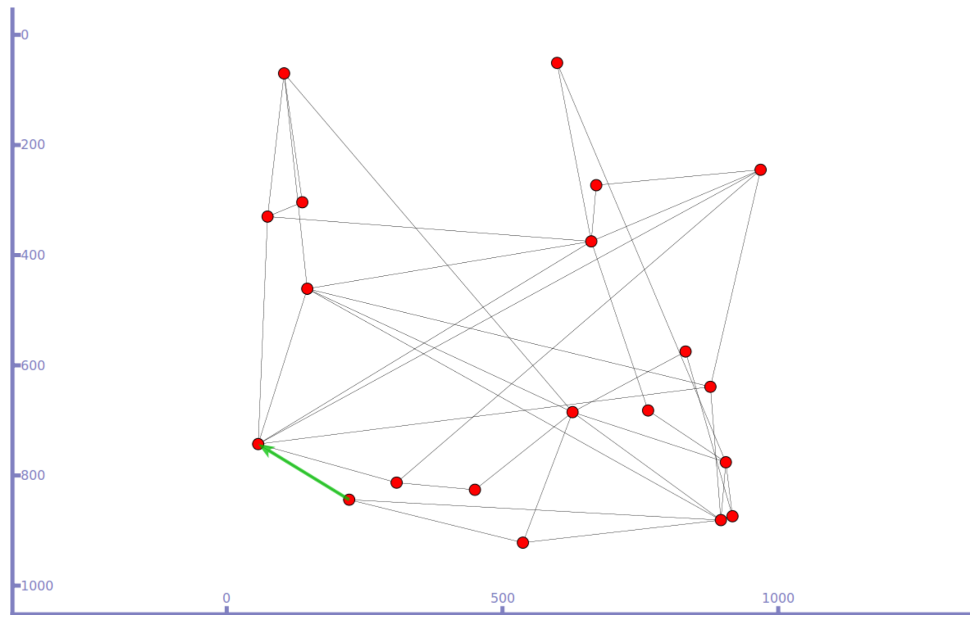
US Presidential Elections Zoom Meeting - Over 160,000 people on a single call

Conferencing

- Metrics
  - Latency (e.g., Zoom or Facetime applications have 100s of ms latency)
  - High frame rate, no freezes
  - High quality
- No option for pause, rewind, or jumping to a different parts of the video

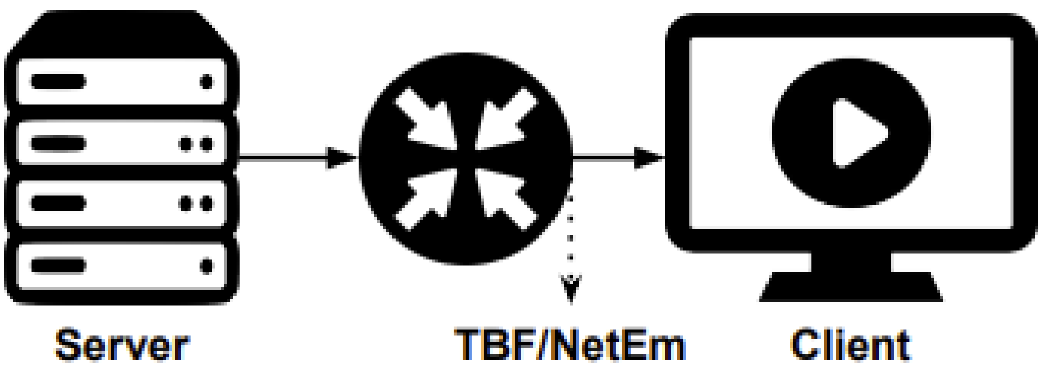
Building and Testing Streaming Protocols

- Simulation
  - Model traffic
  - Model network
  - Model compression
  - Build protocol
  - Test and evaluate



Building and Testing Streaming Protocols

- Emulation – slightly more realistic



Stream videos over realistic network conditions
Record & Replay real world network traces

Summary of the Lecture

- Streaming fundamentals
- On-demand video streaming
- Live streaming
- Video conferencing
- Building and testing streaming protocols

Next up: Adaptive Bitrate Algorithms