Towards Application-aware Network Management over Encrypted Internet

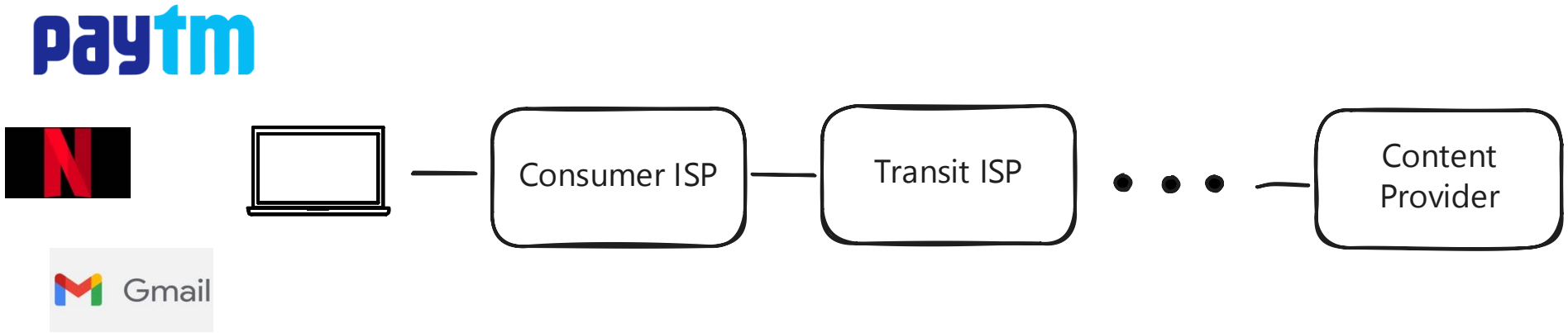
Tarun Mangla

Assistant Professor, IIT Delhi

tmangla@iitd.ac.in

Network Privacy: What and Why

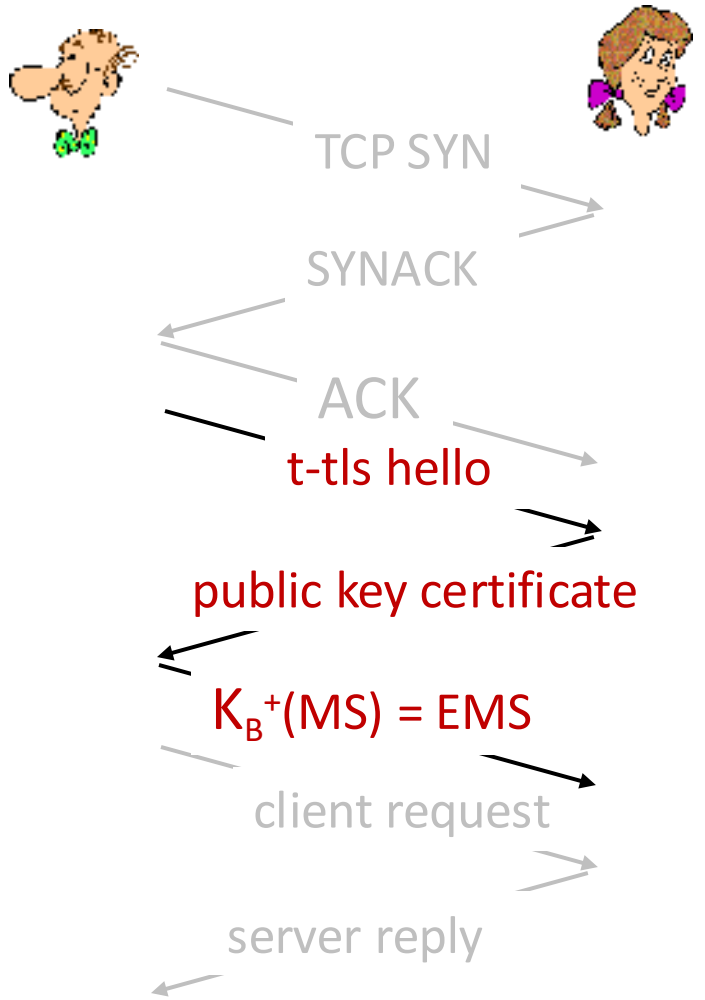
- **What**: Protection of user data in transit across networks



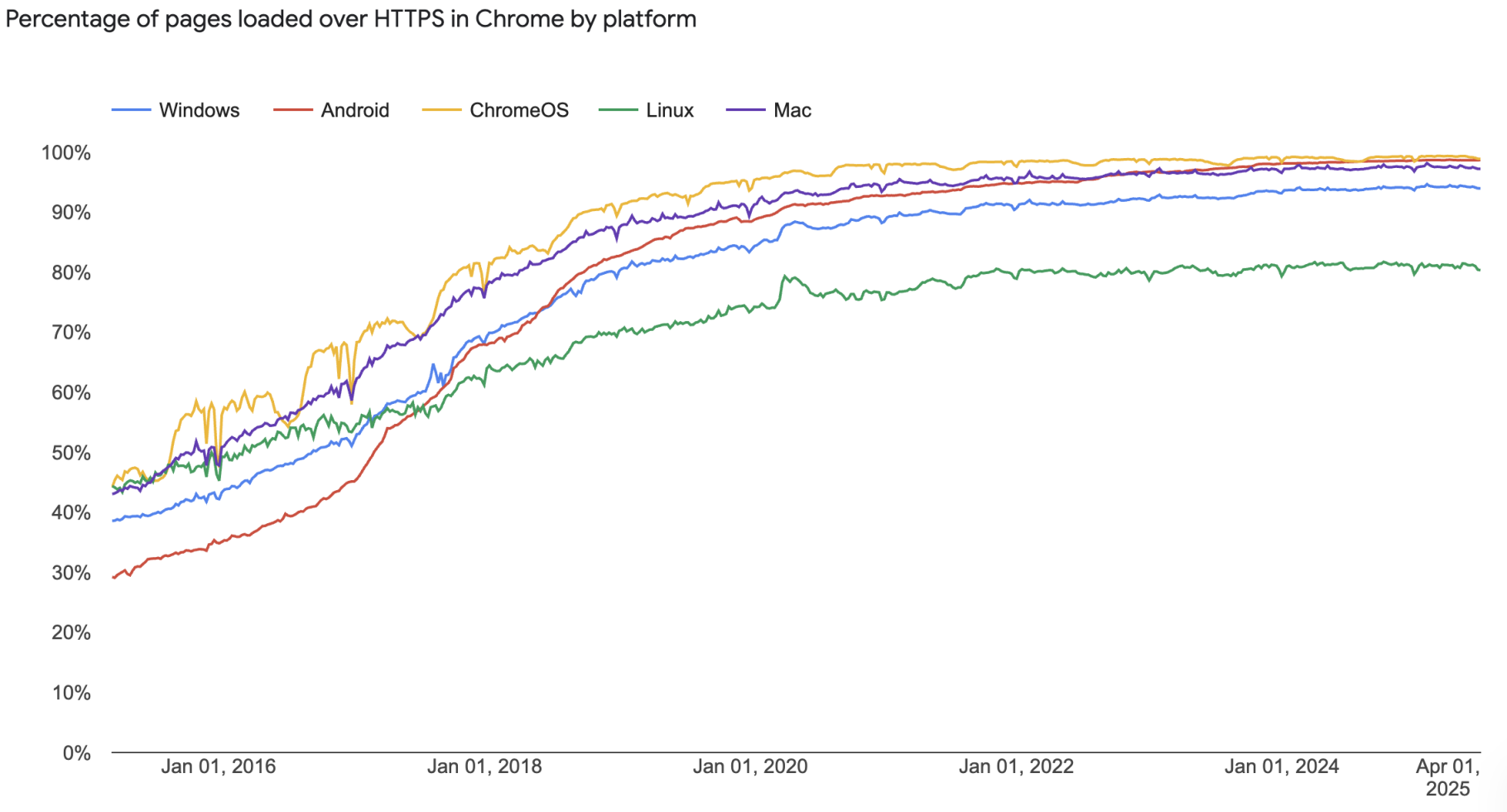
- **Why:**
  - Protect personal and sensitive data
  - Preserve freedom and autonomy
  - Mitigate security risks

Network Privacy: How

- Transport Layer Security (TLS): *de facto* standard for Internet security
- Provides:
  - confidentiality: via *symmetric encryption*
  - integrity: via *cryptographic hashing*
  - authentication: via *public key cryptography*

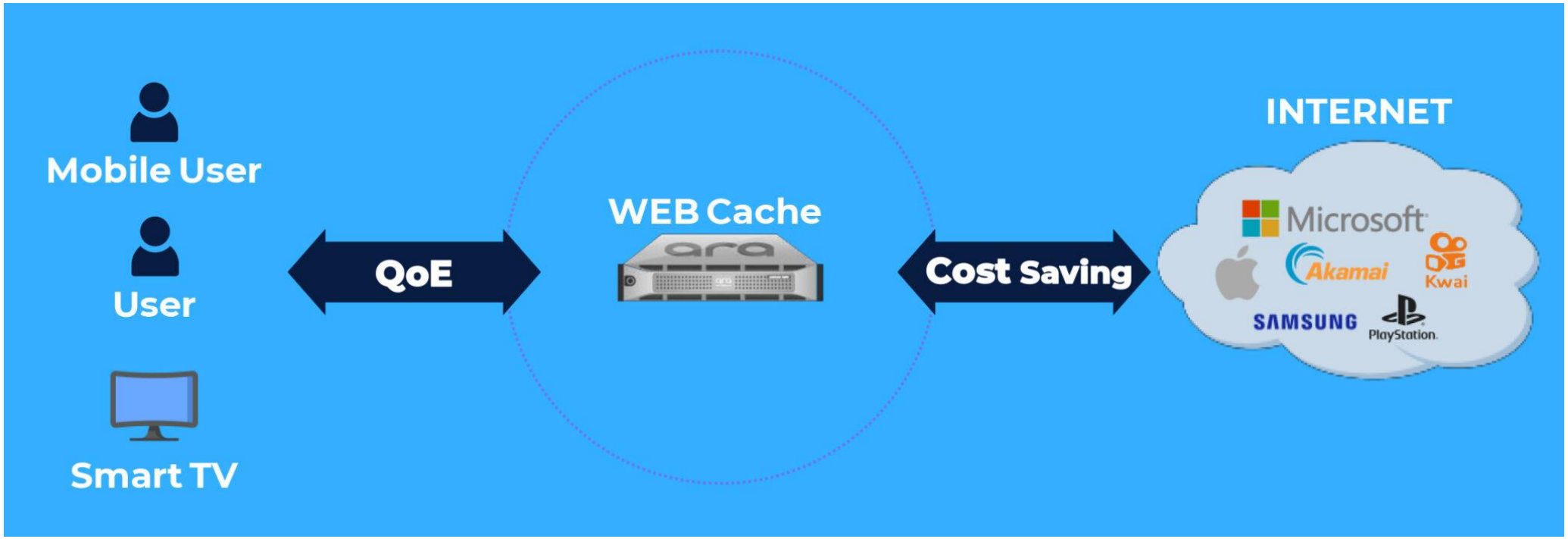


(Most) Internet Traffic is Encrypted



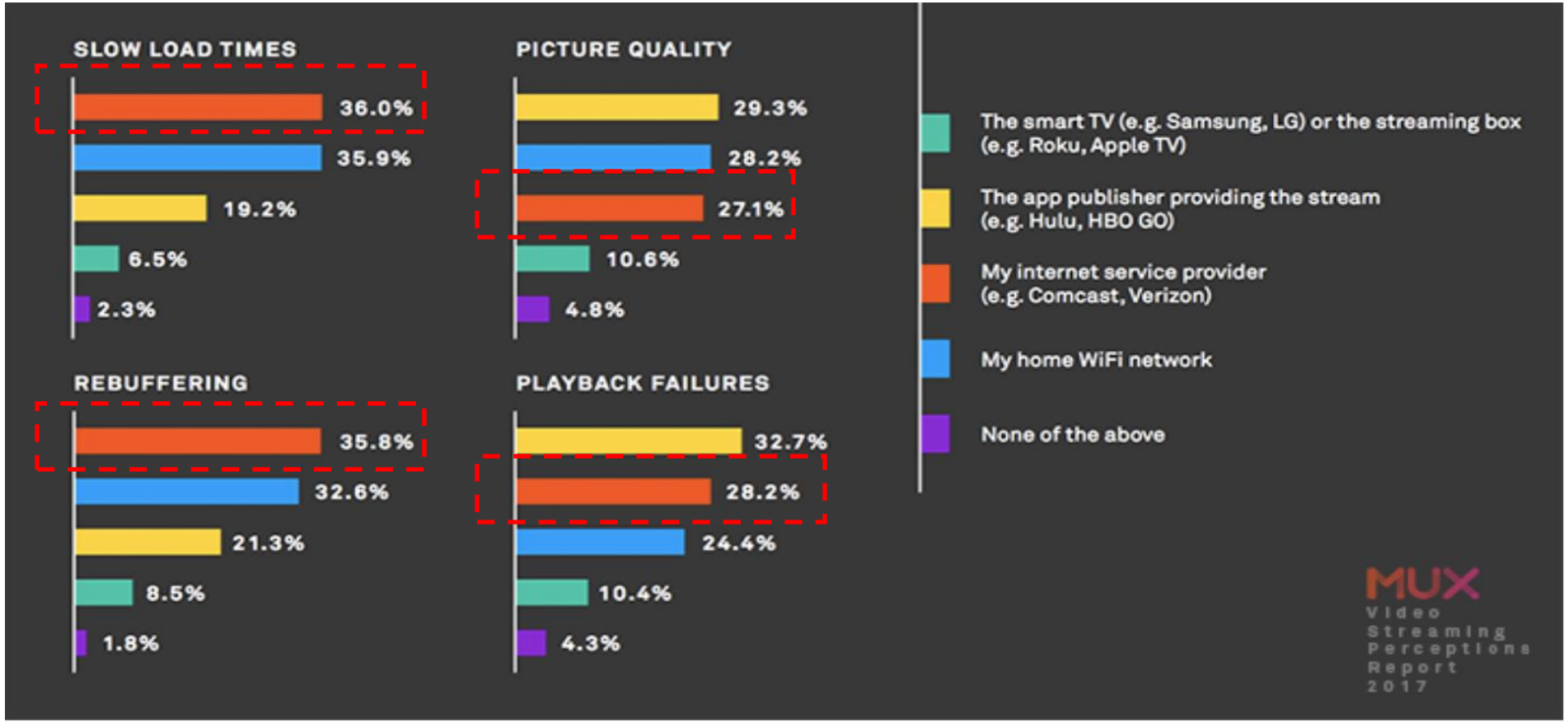
Side Effect: ISPs Lose Visibility

- Internet Service Providers (ISPs) can't see application-level data due to encryption
- No longer able to do network optimizations such as web caching



So what?

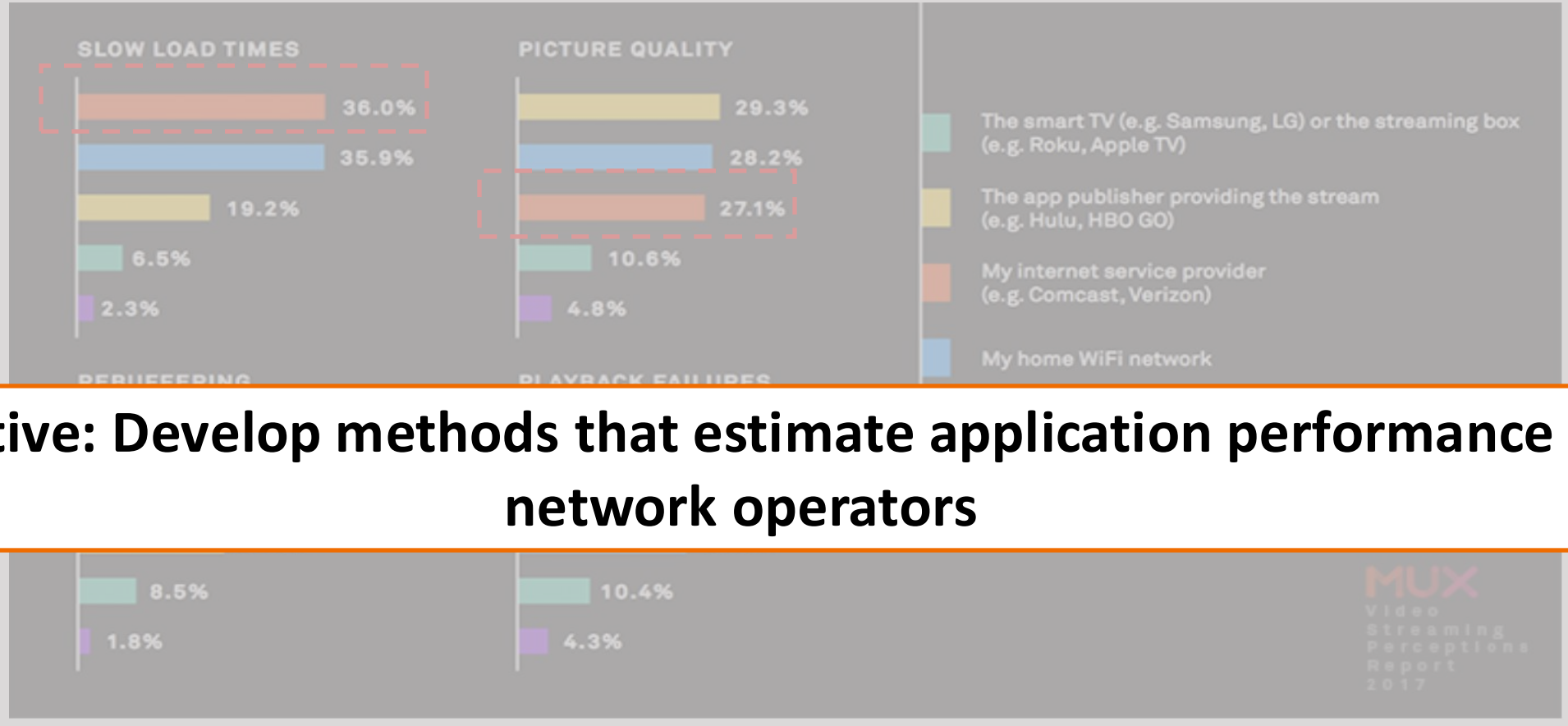
ISPs Often Get Blamed for Application Performance Issues



https://mux.com/blog/isps-and-publishers-get-the-blame-for-video-streaming-problems/

ISPs need an in-depth understanding about application performance

ISPs Often Get Blamed for Application Performance Issues



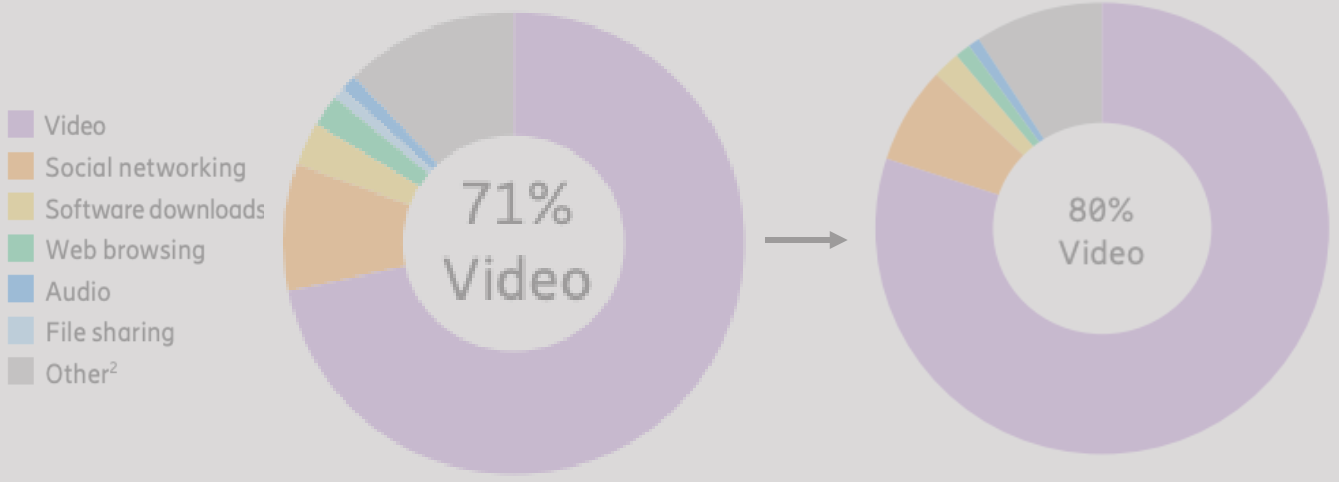
Objective: Develop methods that estimate application performance for network operators

https://mux.com/blog/isps-and-publishers-get-the-blame-for-video-streaming-problems/

ISPs need an in-depth understanding about application performance

Focus of Today's Talk: Internet video

Extensively used for entertainment, education, remote work, telehealth

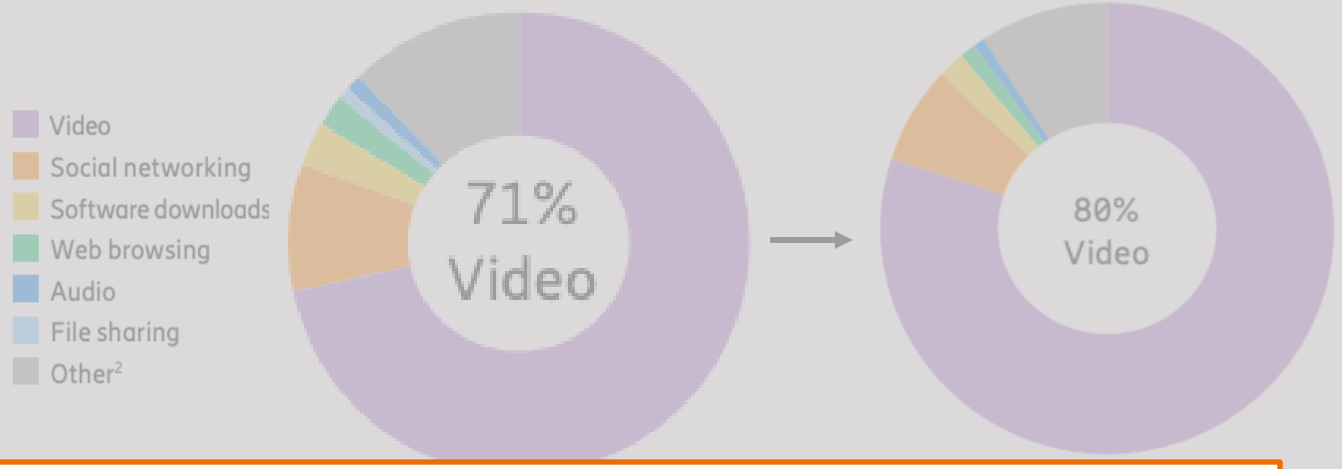


2022: 90 EB per month

2028: 324 EB per month

Largest Contributor to the Internet Traffic

Focus of Today's Talk: Internet video



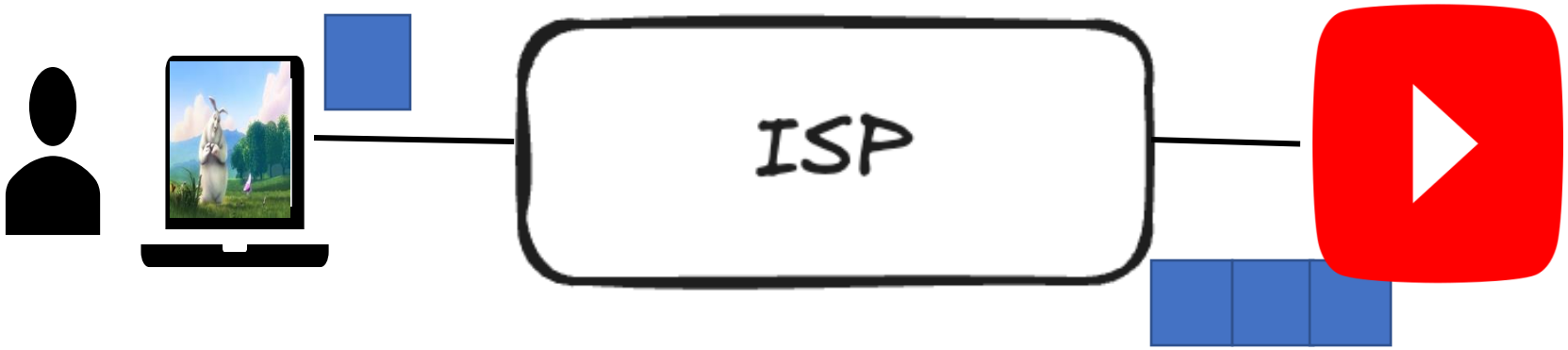
Objective: Develop methods that estimate ~~application~~ video performance for network operators

Extensively used for entertainment, education, remote work, telehealth

Largest Contributor to the Internet Traffic

Challenging for ISPs to obtain video performance

- ISPs lack access to the end-points including end-user device, streaming application, or server



ISPs need to rely on the limited view of network measurement data to *estimate* QoE

Objective: Develop methods that estimate video QoE using passive network measurements

Talk Outline

① Inferring **video streaming** performance for ISPs

ISP

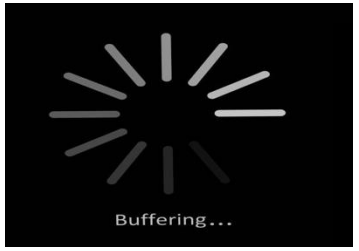
② Inferring **video conferencing** performance for ISPs

ISP

Video Streaming Background

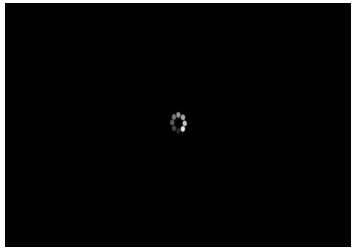
- HTTP Adaptive Streaming (HAS) is the *de facto* standard for video streaming
- HAS performance can be approximated by following objective metrics [Sigcomm'13, ToB'15]

Average bitrate



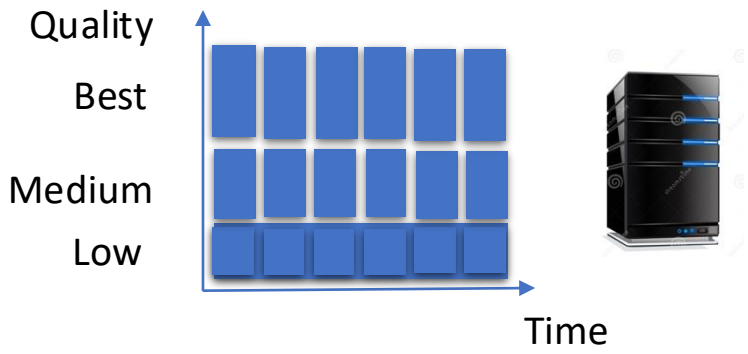
Re-buffering ratio

Bitrate switches



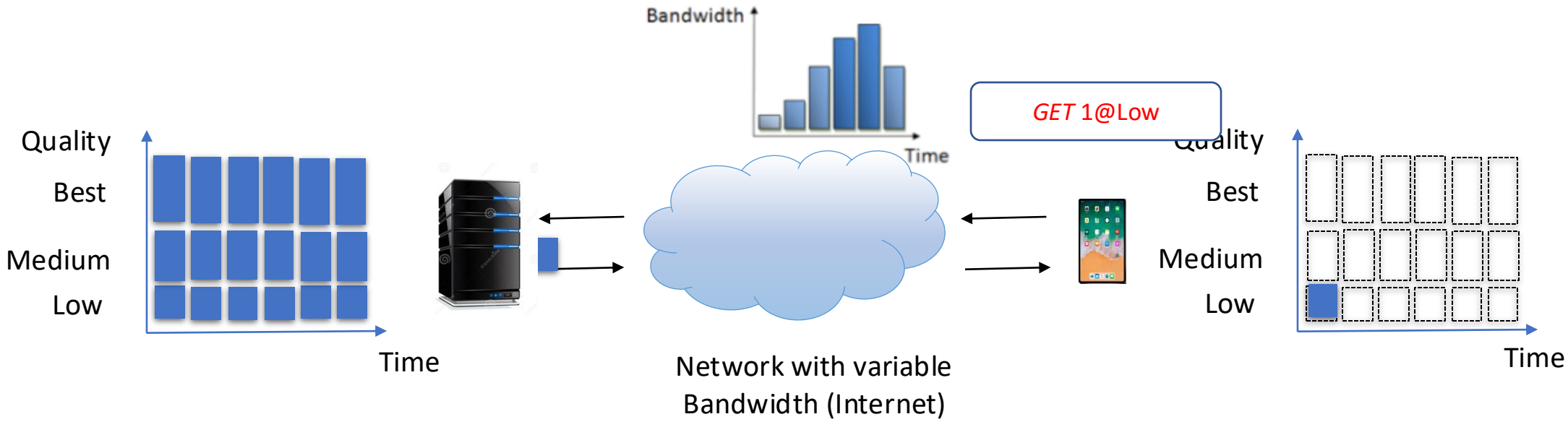
Startup Time

HTTP Adaptive Streaming (HAS)



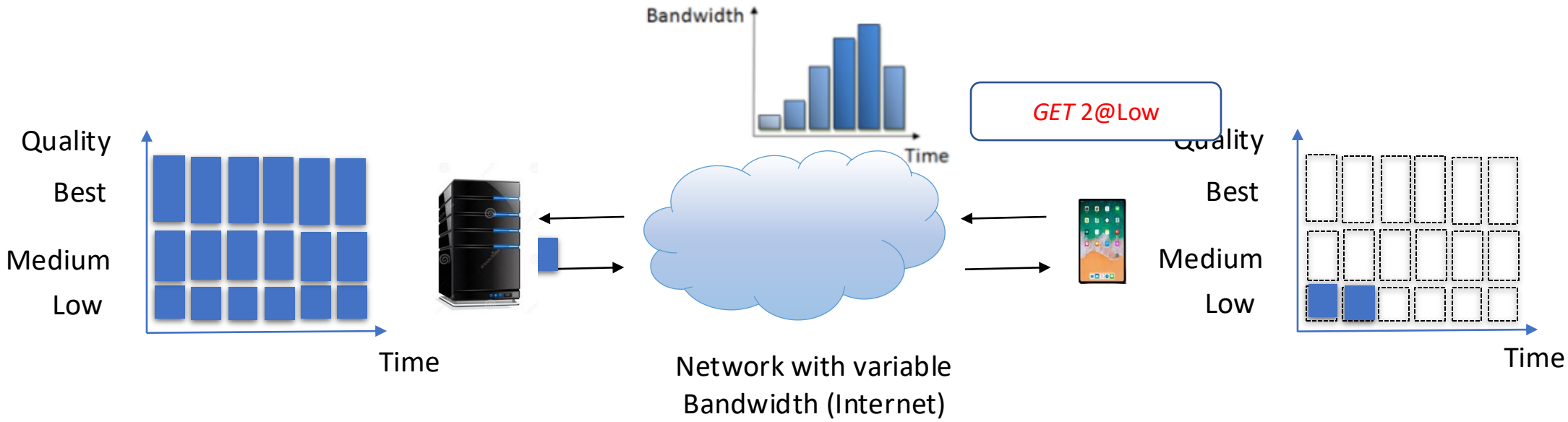
- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server

HTTP Adaptive Streaming (HAS)



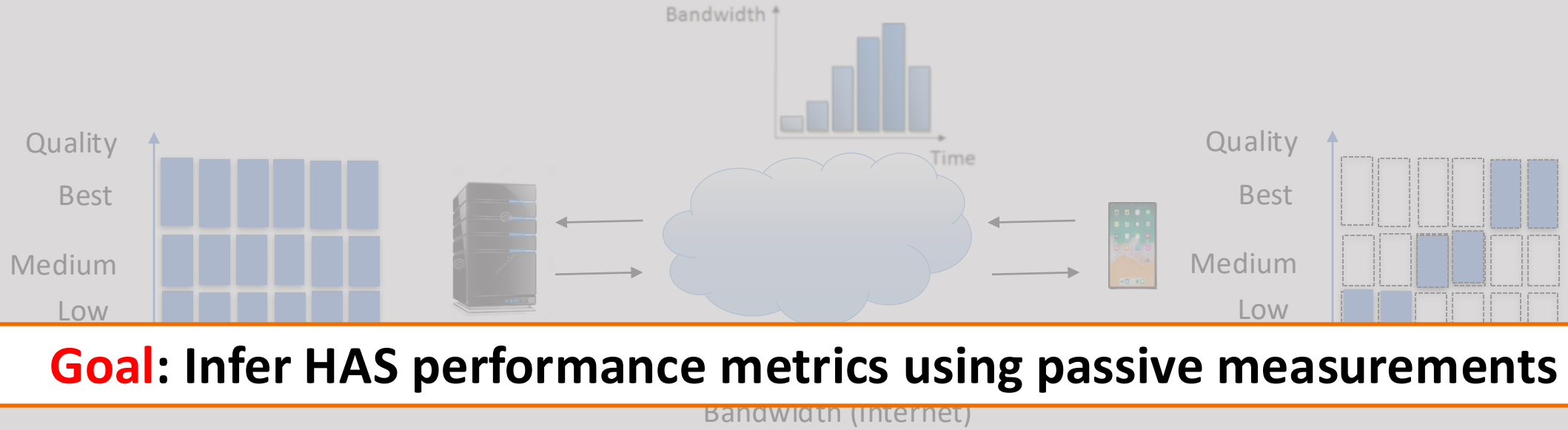
- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server
- Client sends HTTP GET requests for chunks of the quality that matches network conditions

HTTP Adaptive Streaming (HAS)



- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server
- Client sends HTTP GET requests for chunks of the quality that matches network conditions

HTTP Adaptive Streaming (HAS)



Goal: Infer HAS performance metrics using passive measurements

- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server
- Client sends HTTP GET requests for chunks of the quality that matches network conditions

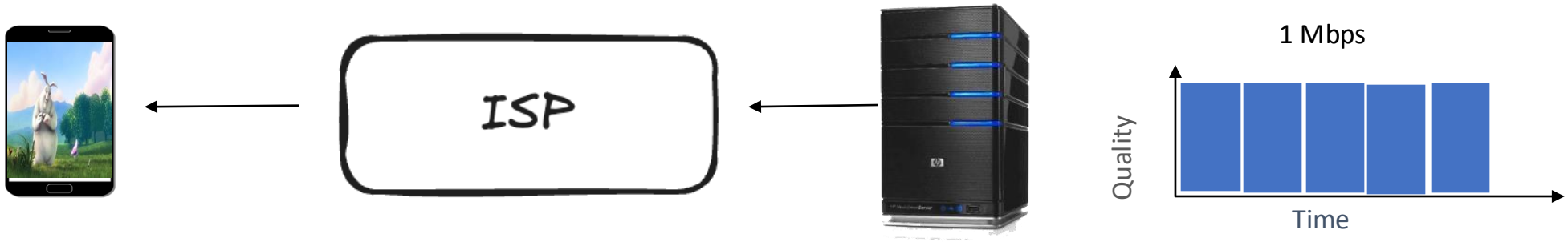
Approach Overview

- DNS
- TLS Server Name Indication (SNI)
- Machine Learning

- Heuristics based on flow activity/end-points

QoE Inference Approach Overview

Session modeling: Use network properties of streaming protocol to model a video session



Hypothetical example

Performance can be inferred by simply logging the amount of downloaded data on network

How to design a session modeling approach for HTTP Adaptive Streaming?

eMIMIC: Key Idea and Challenges

Use packet traces to model a video session as a sequence of video chunk downloads

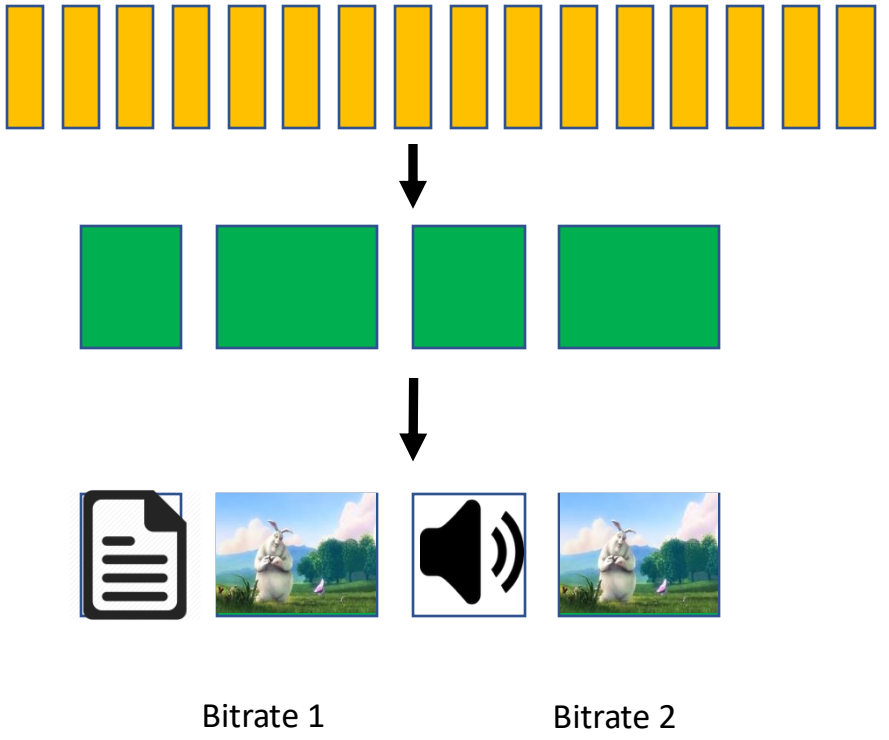
Reconstruct HTTP transactions

↓

Media type classification

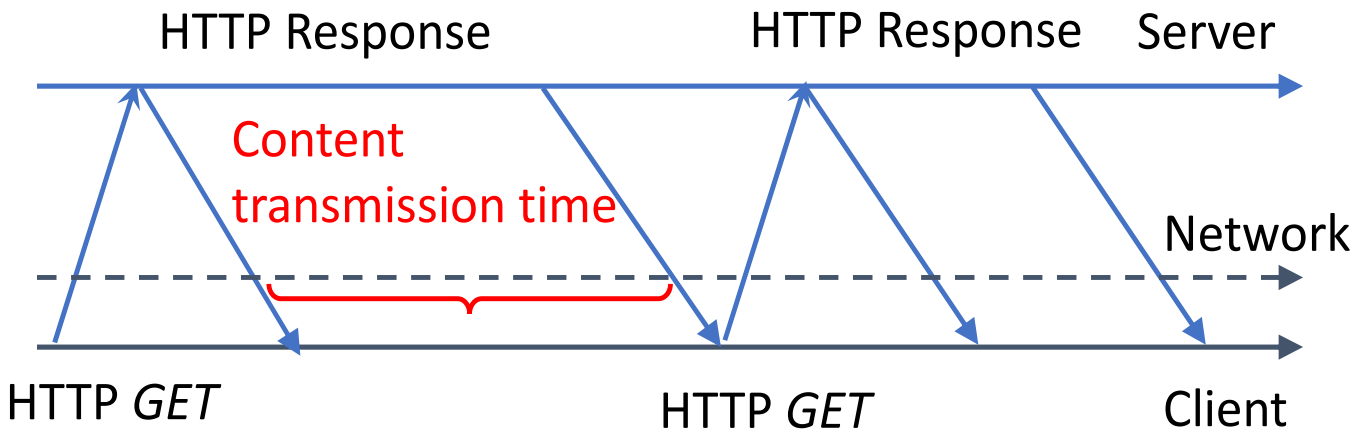
↓

Estimating video chunk quality



Challenge 1: Reconstruct HTTP Transactions

Insight: *Use directionality of data flow in an HTTP transaction*

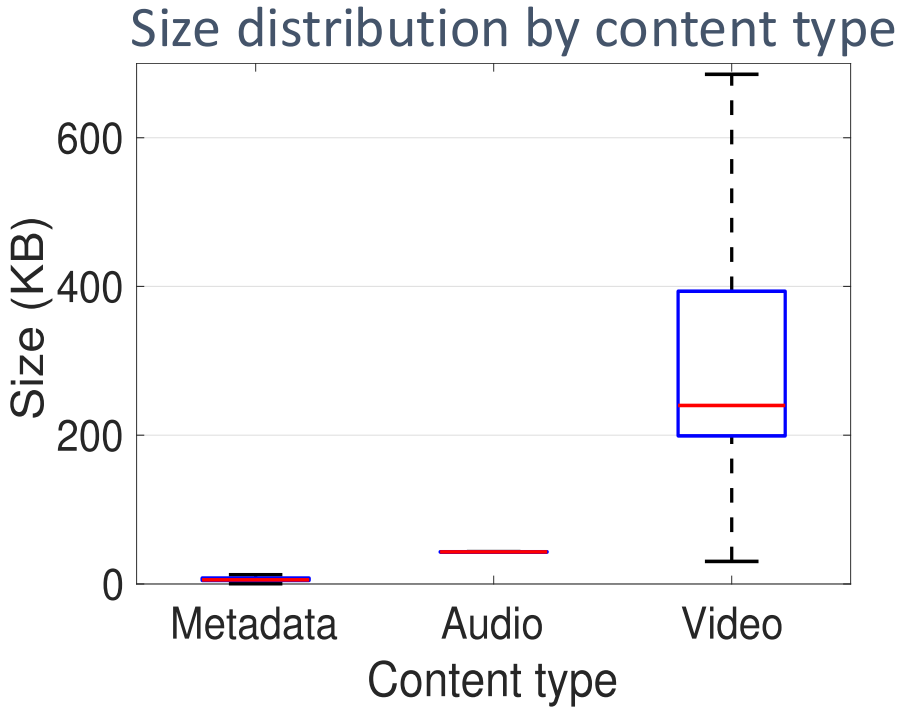


Challenge 2: Media Type Classification

Insight 1: *Use the response size of reconstructed logs*

- Metadata → Small
- Audio → Constant bitrate

Insight 2: *Track audio and video buffer to avoid misclassifying video chunks as audio*



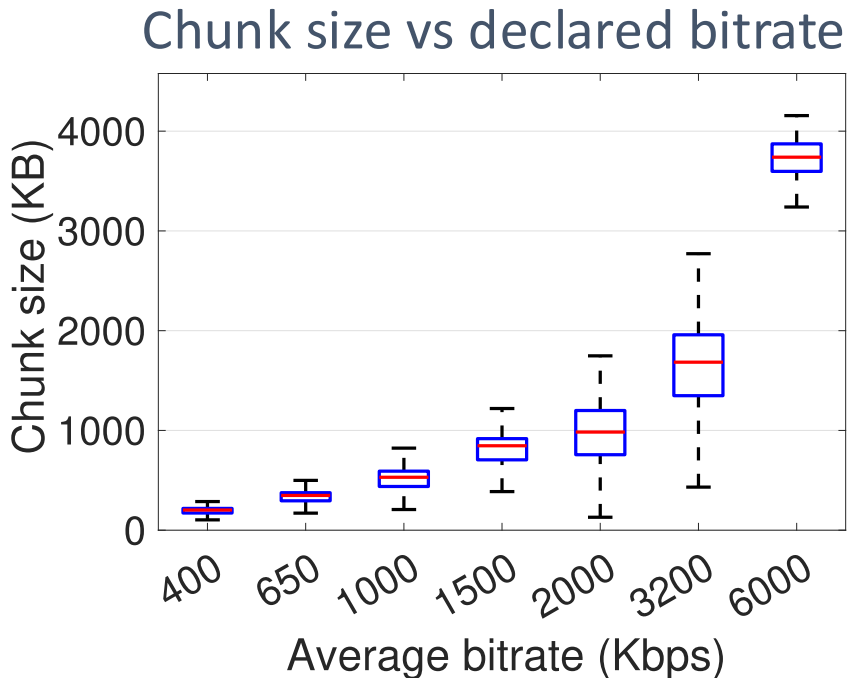
Challenge 3: Estimating Video Chunk Quality

Insight 1: *Chunk size indicates average bitrate*

- However, not enough!

Insight 2: *Bitrate switch accompanied by change in network bandwidth*

- Compare past chunk throughput



QoE Inference from HTTP Transactions

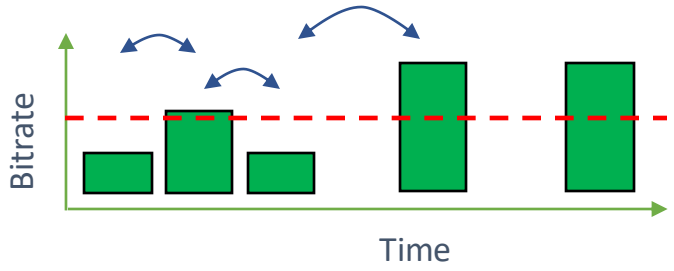
- Average bitrate: $\frac{\sum_{i=1}^{N} Q_i}{N \times L}$,

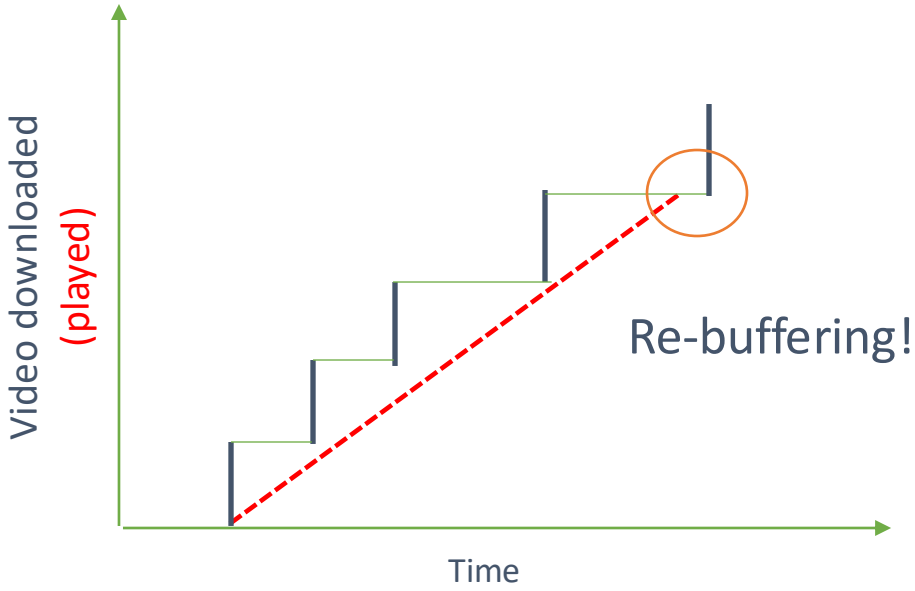
  Q_i: bitrate of i^{th} chunk, N: # of chunks in sessions, L: chunk duration

- Number of switches: $\sum_{i=2}^{N} I(Q_i \neq Q_{i-1})$

  I is the indicator function

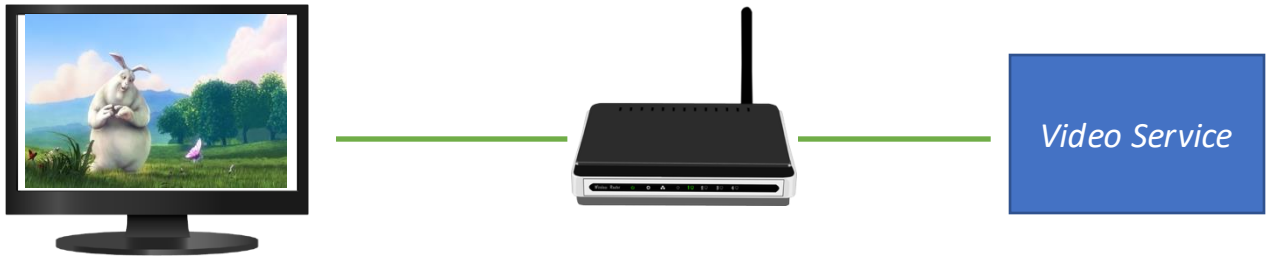
- Re-buffering ratio: **Model buffer occupancy** by accounting downloaded video chunks and total time elapsed





Evaluation

Experimental Setup



- Browser-based automated streaming
- Collect HTTP logs using trusted proxy
- Collect ground truth QoE metrics

- Emulate bandwidth
- Collect packet traces

- An automated browser-based framework to stream video sessions
- Evaluate using two popular video streaming services: VoD1 and VoD2
- Obtain packet traces and ground truth QoE metrics and HTTP logs
- 985 sessions of VoD1 and 1005 sessions of VoD2 streamed under various network conditions

Estimation Accuracy: Re-buffering Ratio

- **Baseline**: Implemented an ML-based approach [1]
- Classify Re-buffering Ratio (RR) into *zero*, *mild* (RR < 10%), and *high* stall (RR > 10%)

Actual RR	Estimated RR		
	zero	mild	high
zero	87.6%	12.0%	0.4%
mild	51.5%	44.9%	3.6%
high	3.1%	8.4%	88.4%

VoD2: eMIMIC

Actual RR	Estimated RR		
	zero	mild	high
zero	61.1%	37.0%	1.9%
mild	39.4%	48.5%	12.1%
high	26.9%	30.8%	42.3%

VoD2: ML Baseline

eMIMIC predicts low and high re-buffering ratio with significantly higher accuracy compared to ML-based approach

[1] Dimopoulos, Giorgos, et al. "Measuring video QoE from encrypted traffic." *Proceedings of the 2016 Internet Measurement Conference*. ACM, 2016.

Summary

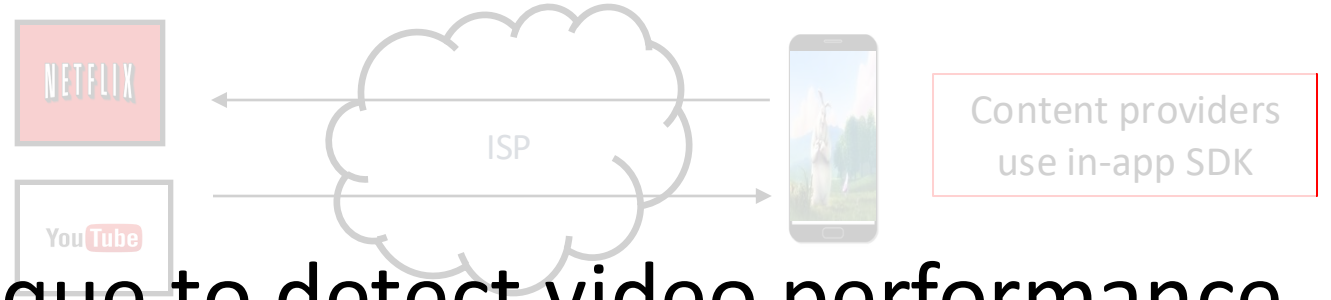
- Develop **a session modeling-based** approach for **encrypted traffic**
- Other work focusing on **system challenges**:
  - VideoNOC: End-to-end system for QoE inference in a cellular network
  - QoE Inference using coarse-grained network data
- Open problems:
  - Designing inference algorithms for newer video streaming formats
  - Continual validation framework and towards lightweight approaches

VideoNOC

CoNEXT: Drop the Packets

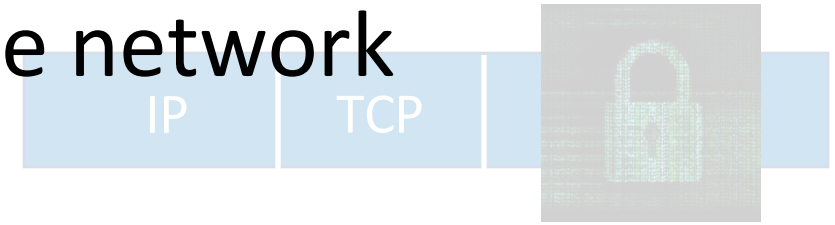
Estimating video QoE is challenging for ISPs

- No access to streaming app, device, or server



Develop a lightweight technique to detect video performance issues in the network

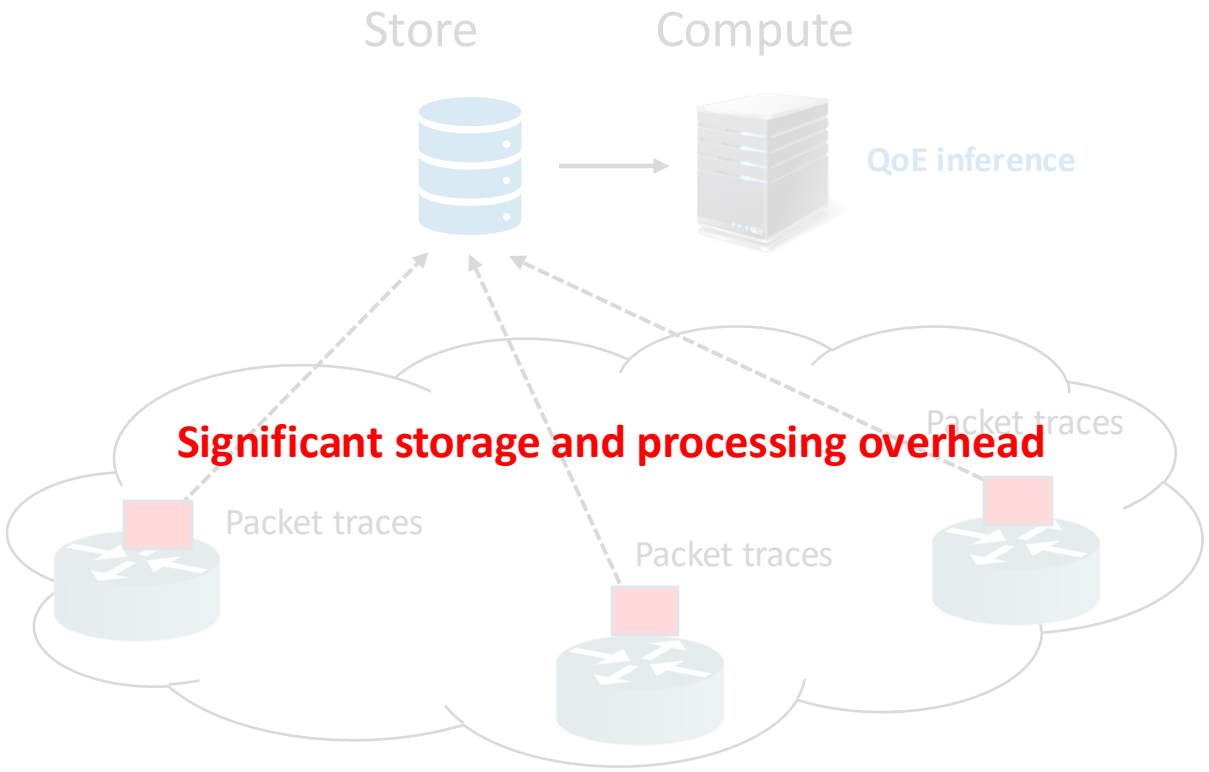
- Video traffic is end-to-end encrypted



- Inference techniques need to be scalable

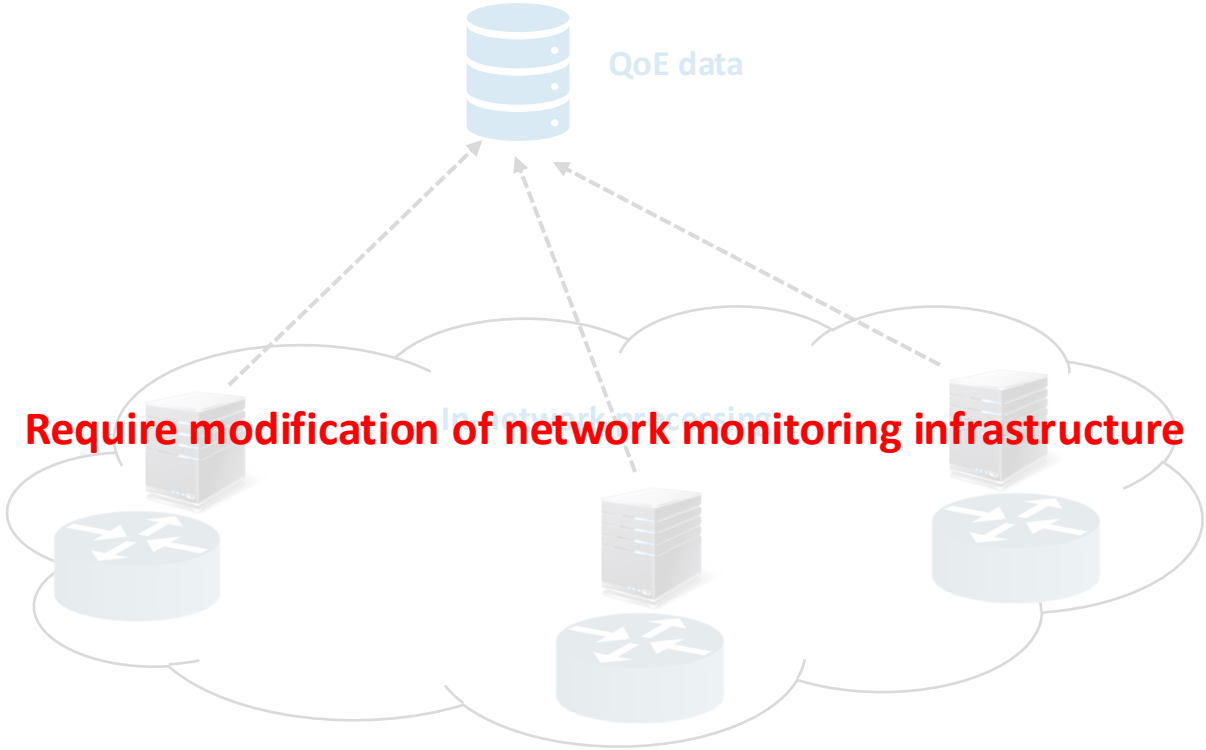
Proposed inference approaches

Assume access to packet traces



Significant storage and processing overhead

In-network processing using flexible telemetry systems



Require modification of network monitoring infrastructure

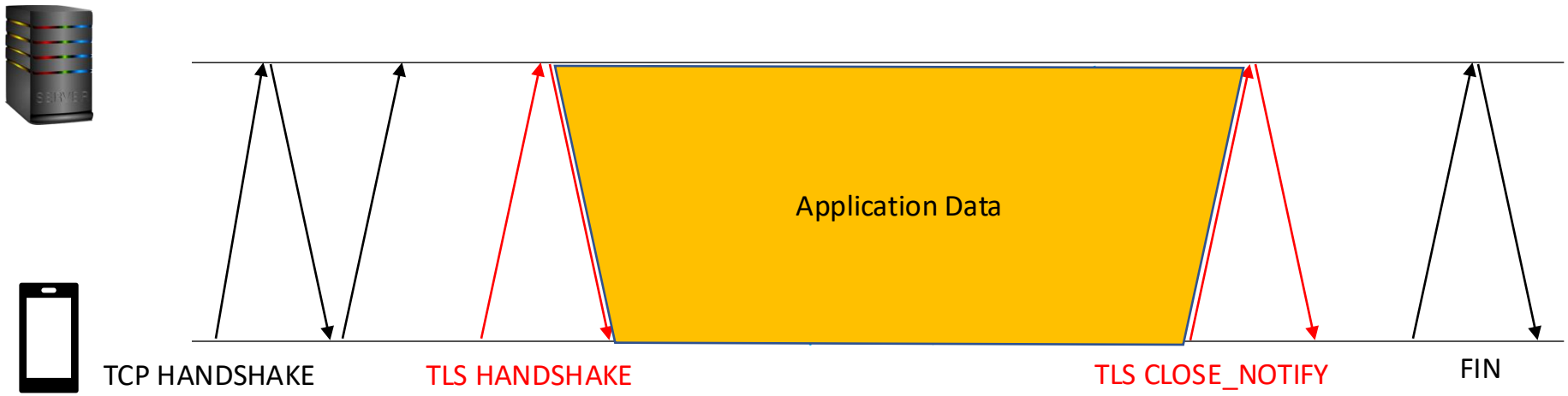
Key idea 

ISPs already collect lightweight data using standard telemetry systems.

Can we use such readily-available network data for QoE inference?

What form of network data?

- Use TLS transaction data available from a standard proxy



TLS Transaction Log

start time, end time, uplink data volume, downlink data volume, server name indication (SNI)

QoE estimation

Video traffic and session identification

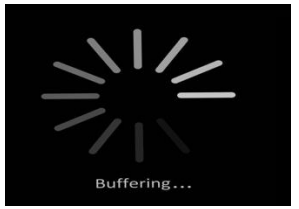
Overview of the approach

Feature kind	Example
Session level	Session duration
Transaction statistics	Median Transaction Data Rate
Temporal statistics	Data downloaded in 30s

Video quality

low, medium, high

Re-buffering ratio (rr)



zero, mild, high

Combined QoE

Min (quality, rr)

Features selected based on the semantics of HTTP-based adaptive streaming

Coarse-grained QoE estimation enables detection of video performance issues

Results: Comparison with packet traces

Combined QoE, Svc1

Data type	Accuracy	Recall	Precision	Memory overhead (# per session)	Computation overhead (in s)
TLS transactions	69%	73%	71%	19.5	8.3
Packet traces [1]	74%	82%	73%	27,689 (1400x)	503 (60x)

TLS transactions provide reasonable accuracy with significantly lower storage and processing overhead

[1] Giorgos Dimopoulos, Ilias Leontiadis, Pere Barlet-Ros, and Konstantina Papagiannaki. "Measuring video QoE from encrypted traffic." *ACM IMC 2016*

VCA Measurement

Bandwidth requirements for VCAs

VCA	Utilization (Mbps)	
	Upstream	Downstream
Meet	0.95	0.84
Teams	1.40	1.86
Zoom	0.78	0.95

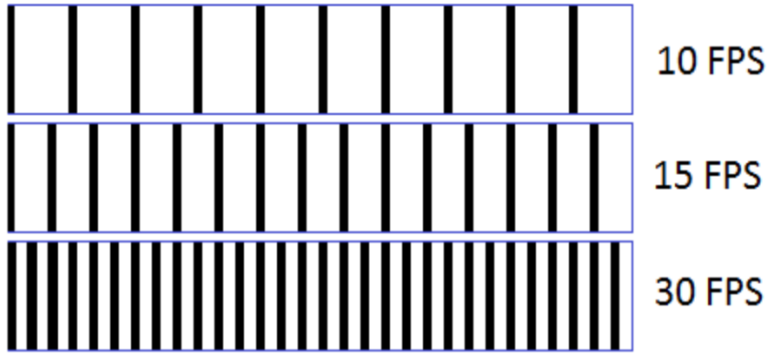
High upstream utilization. Implications on network management and broadband policy.

Performance under Varying Link Capacity

VCAs adjust video quality to match network conditions

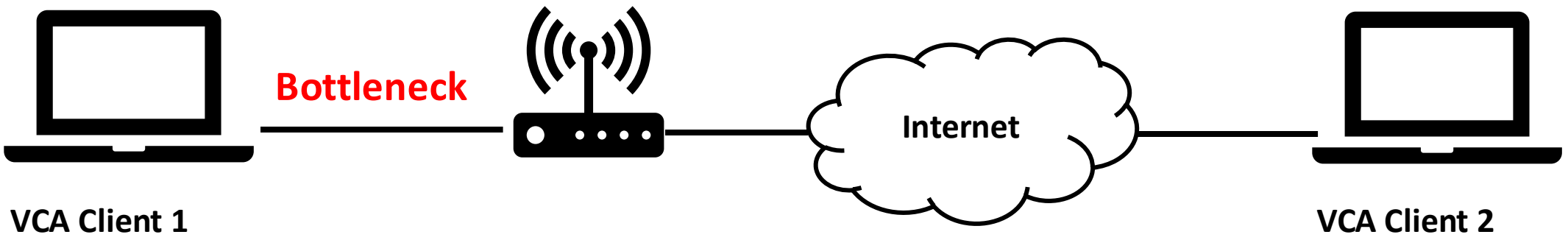
Quantization Parameter (QP)

Resolution



Frames Per Second (FPS)

Performance under varying Capacity

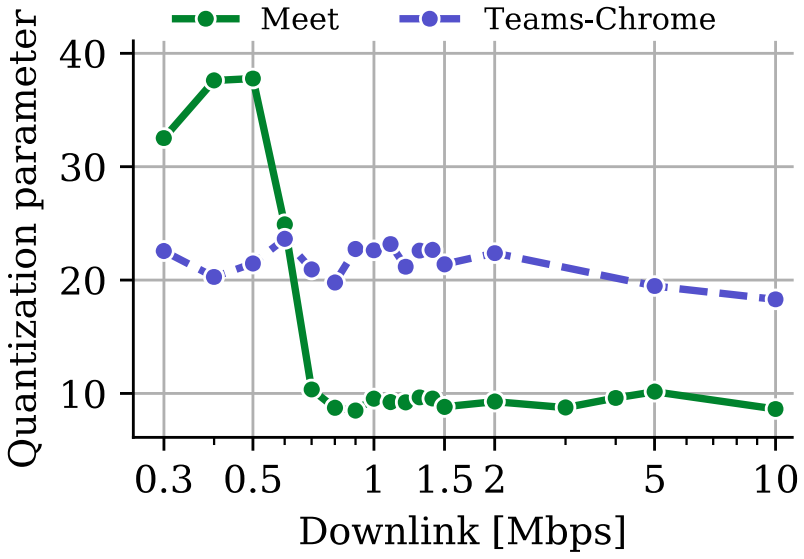


Experiment Setup

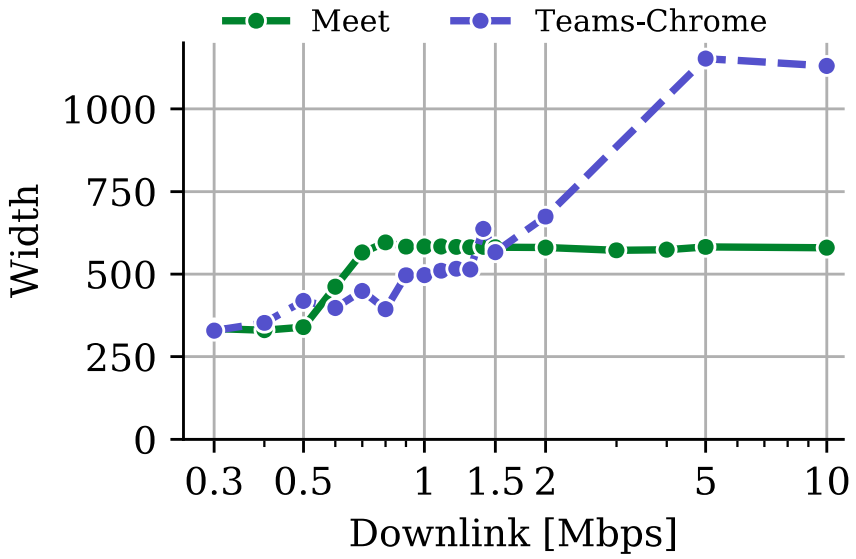
Log QoE metrics: Quantization Parameter, Frames Per second (FPS), and Video Resolution

VCA Performance under varying Capacity

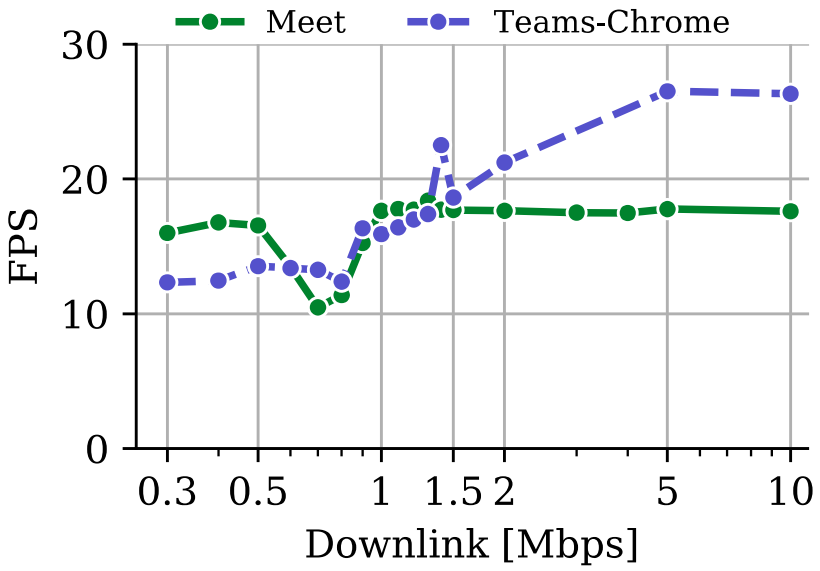
Quantization Parameter



Video Resolution

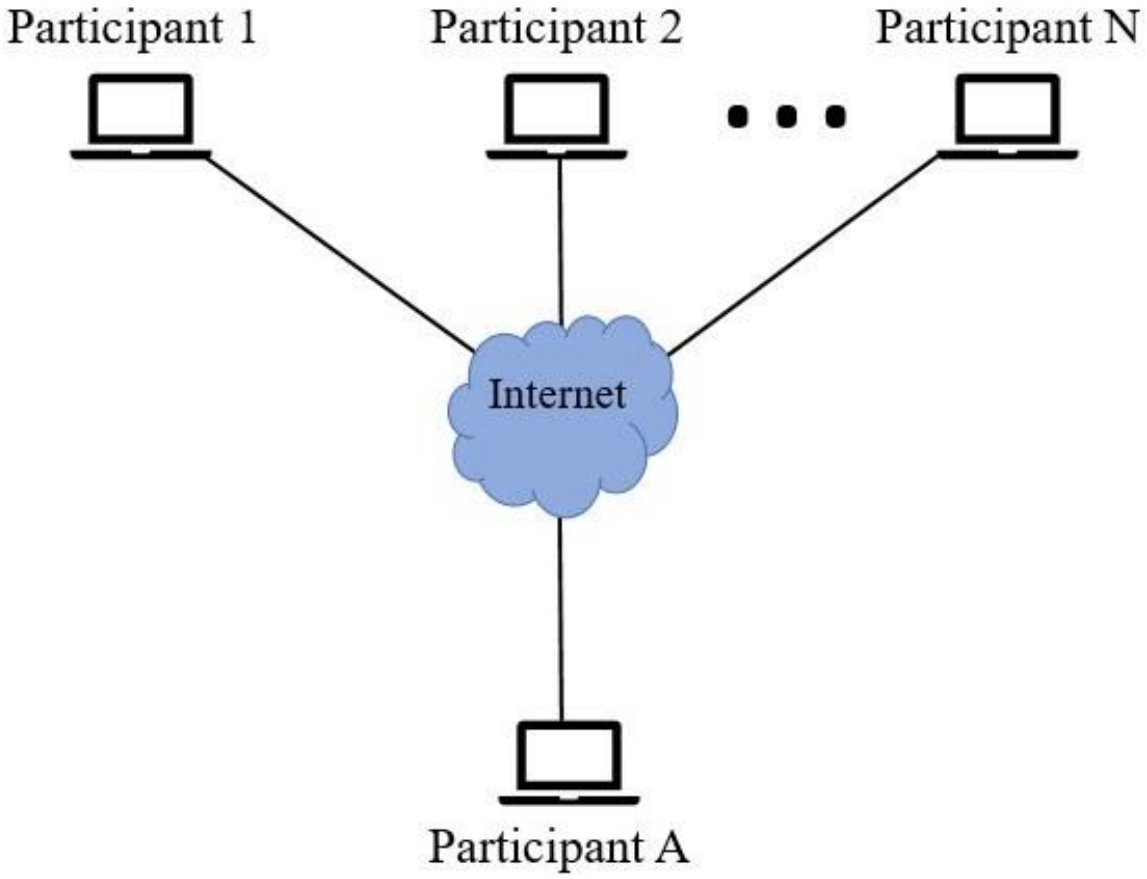


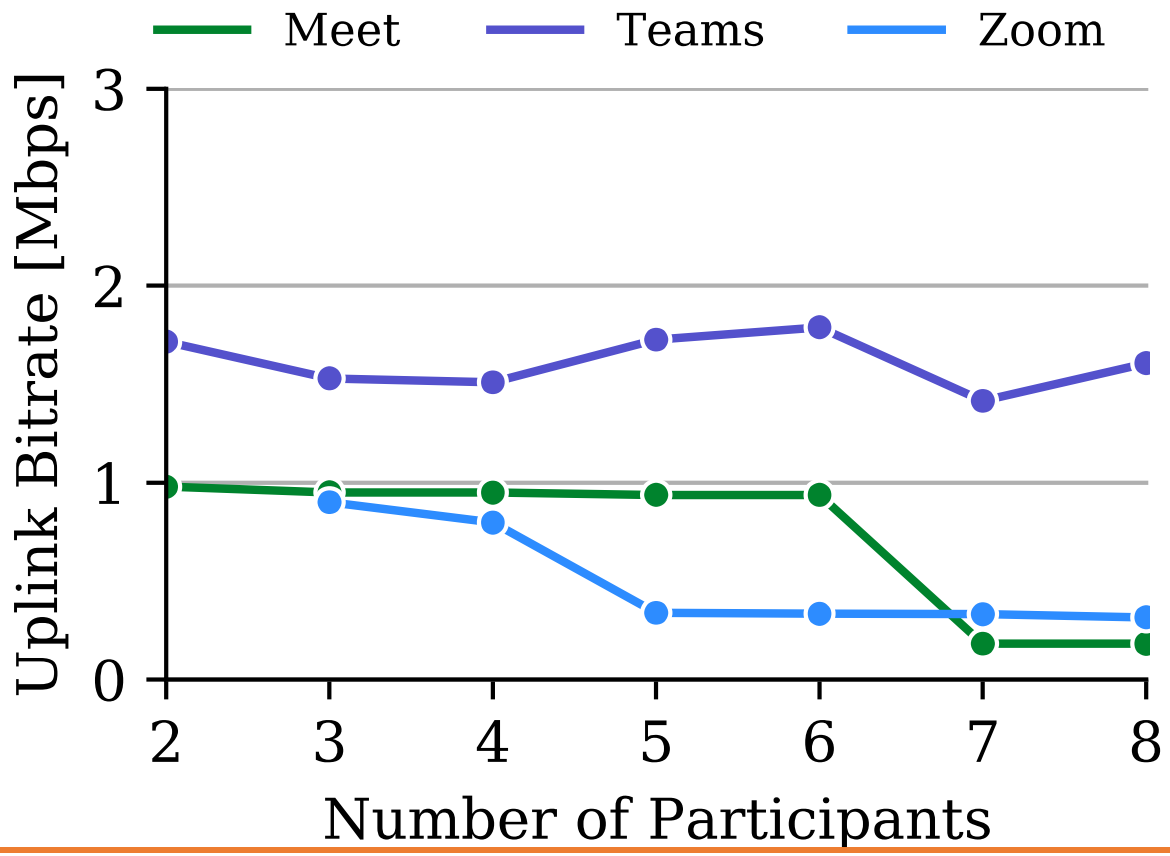
Frames Per Second



Impact of capacity degradation on QoE metrics differs across VCAs.
Why → Differences in encoding mechanisms

Impact of Number of Clients





Increase in # of participants impacts upstream utilization due to changes in video layout