Impact of New Protocols: *What Happens When Everybody DASHes?*

Workshop on Media Streaming Meets 5G – Dec. 2019

Ali C. Begen, PhD

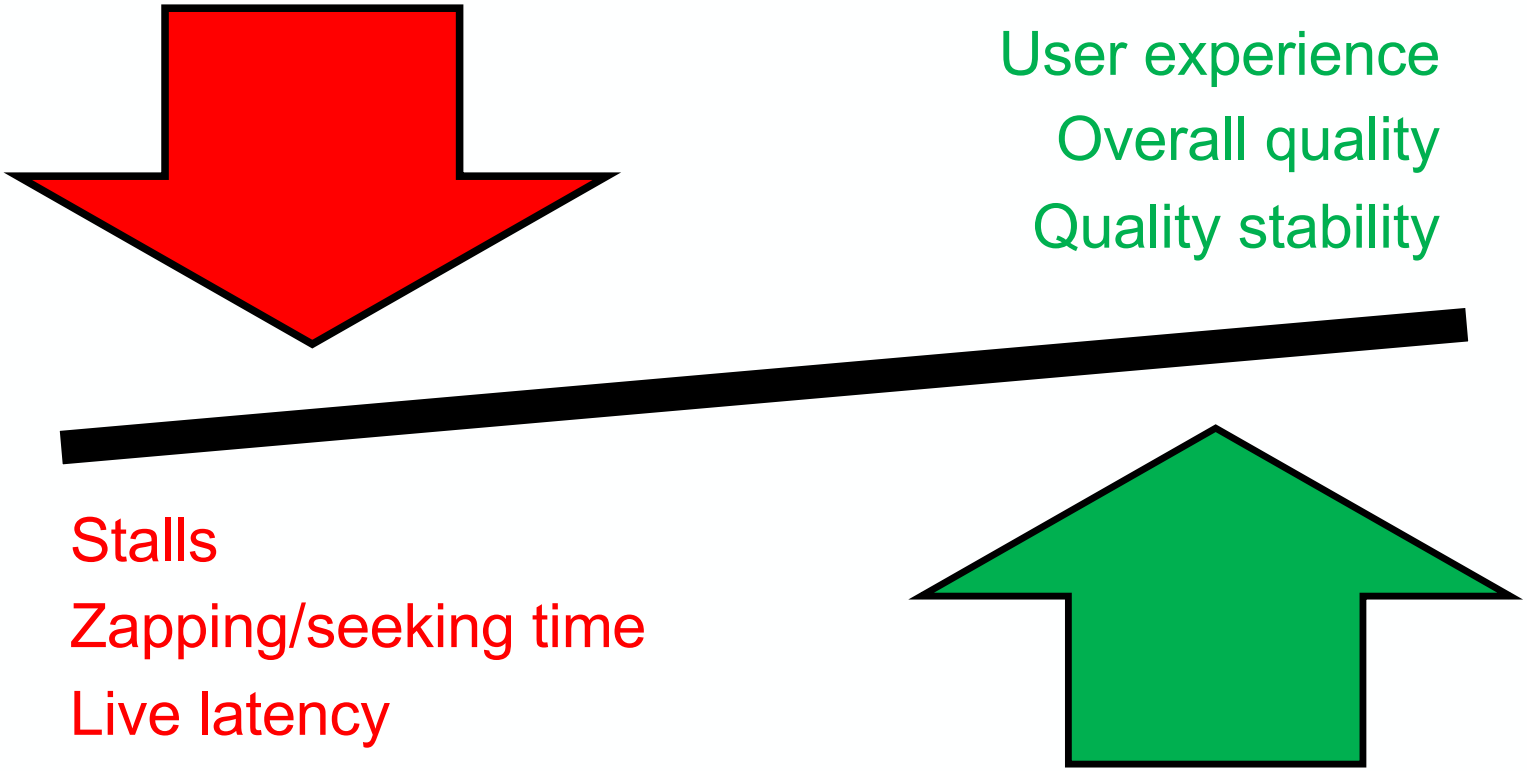
http://ali.begen.net

ÖZYEĞİN UNIVERSITY

COMCAST

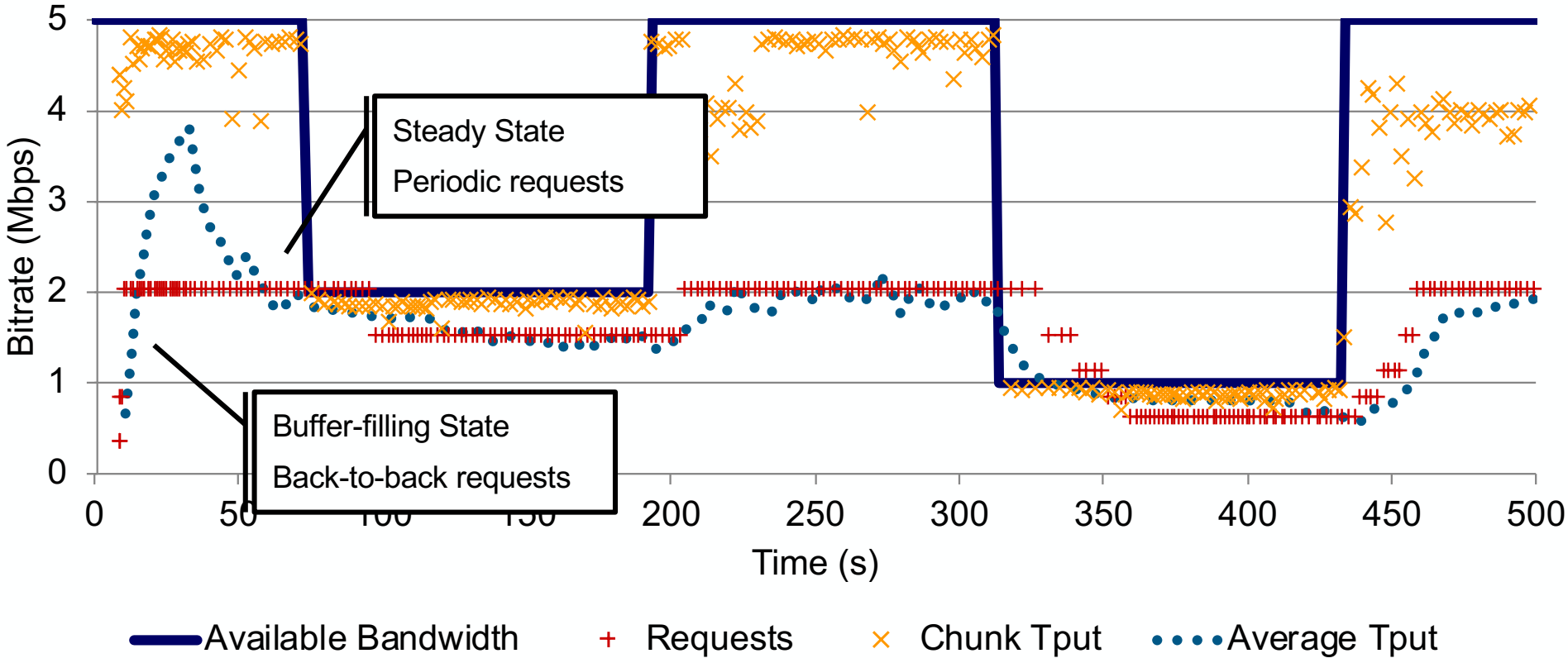
I DASH, You DASH, Everybody DASHes

Tradeoffs in Adaptive Streaming



Demystifying a Streaming Client

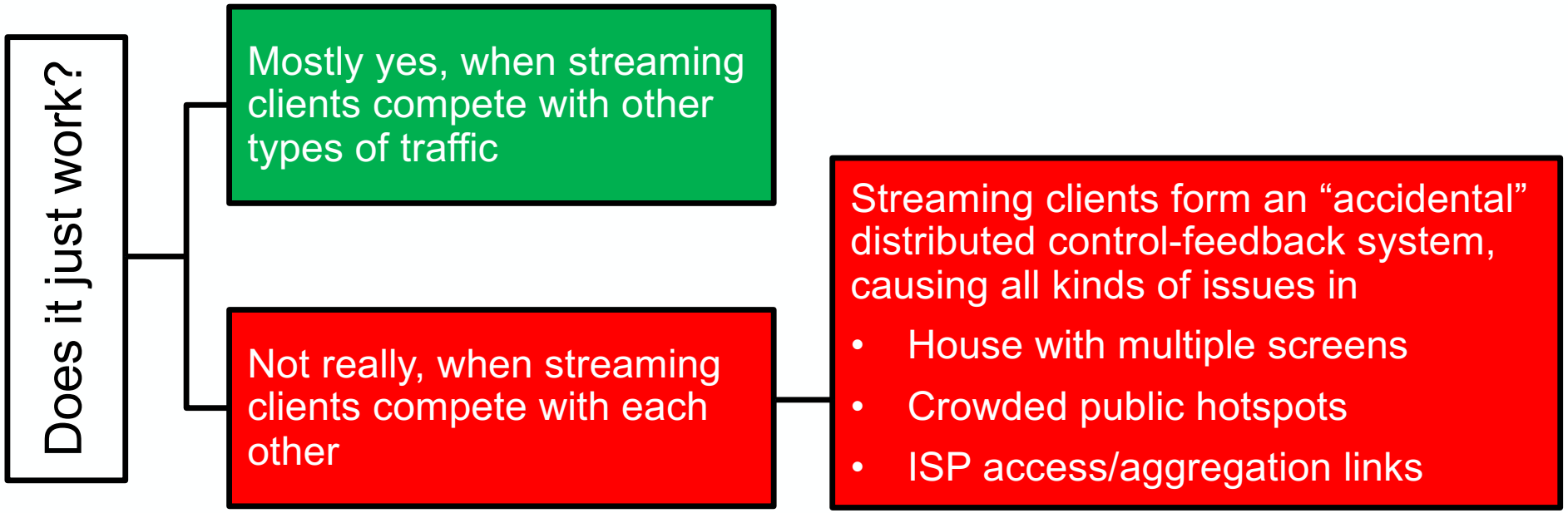
A Single Microsoft Smooth Streaming Client under a Controlled Environment



Reading: “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP,” ACM MMSys 2011

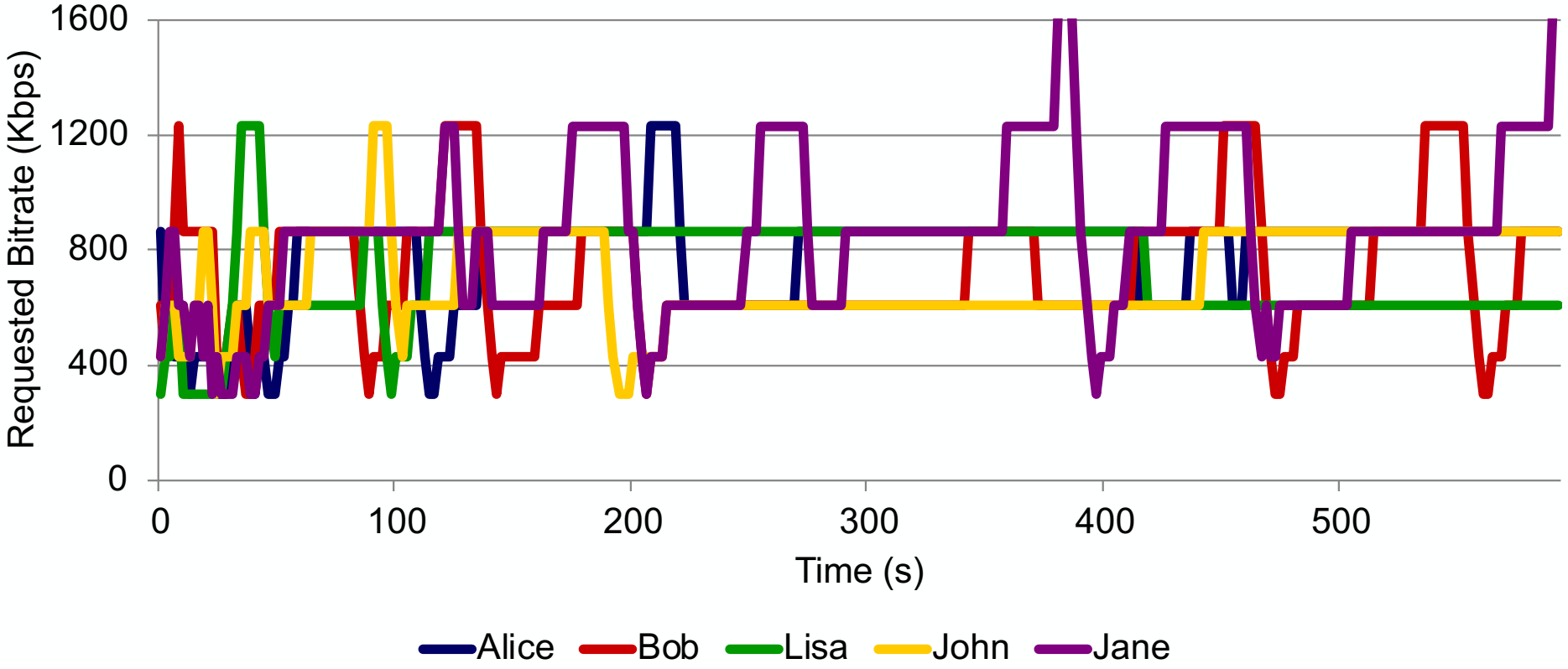
Streaming over HTTP – The Promise

- Leverage tried-and-true Web
  - Video is just ordinary Web content
- Leverage tried-and-true TCP
  - Congestion avoidance
  - Reliable transport
  - No need for special QoS for video



Selfishness Hurts Everyone

10 Streaming Clients Sharing a 10 Mbps Link

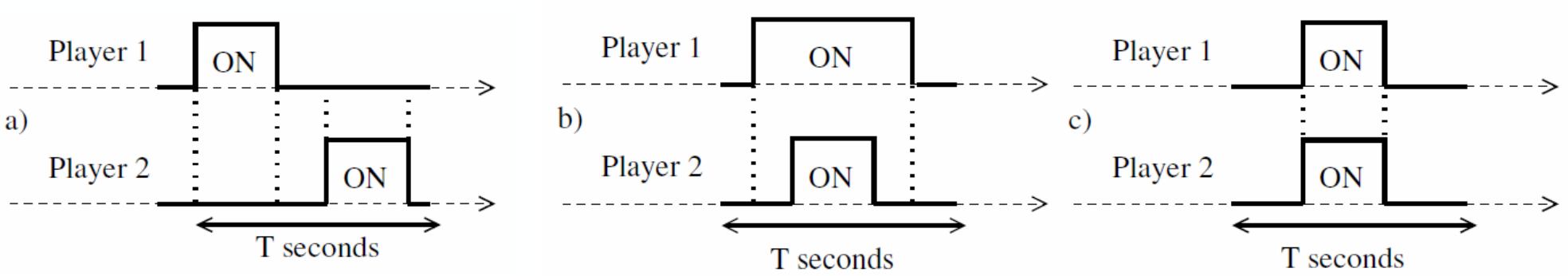


Understanding the Root Cause

Two Competing Clients

- Depending on the timing of the ON periods:
  - Unfairness, underutilization and/or instability may occur
  - Clients may grossly overestimate their fair share of the available bandwidth

Clients cannot figure out how much bandwidth to use until they use too much (Just like TCP)



Reading: "What happens when HTTP adaptive streaming players compete for bandwidth?," ACM NOSSDAV 2012

How to Solve the Issues?

Fix the clients and/or the transport

- Use a better adaptation algorithm like PANDA or BOLA
- Use machine learning or deep learning like Pensieve
- Improve the HTTP/TCP/QUIC stack, try out the new developments (HTTP/2/3)
- Adopt ideas from game/consensus theory (GTA)

Get support from the network

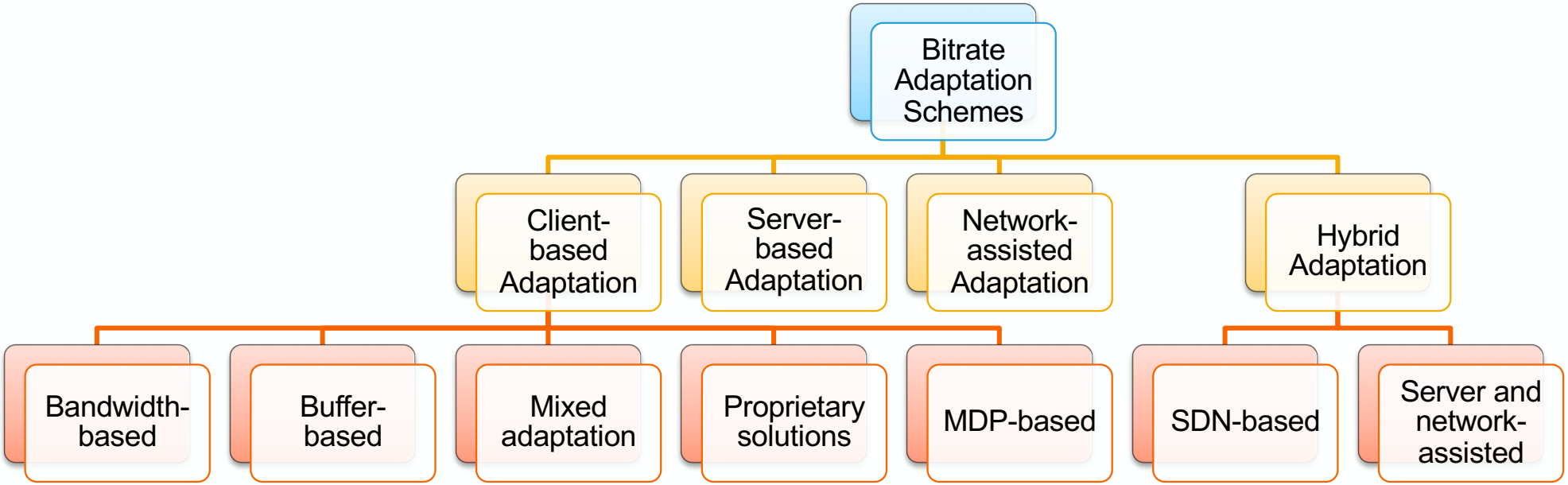
- QoS in the core/edge
- SDN

Enable a control plane

- Assist the clients and network elements through metrics and analytics

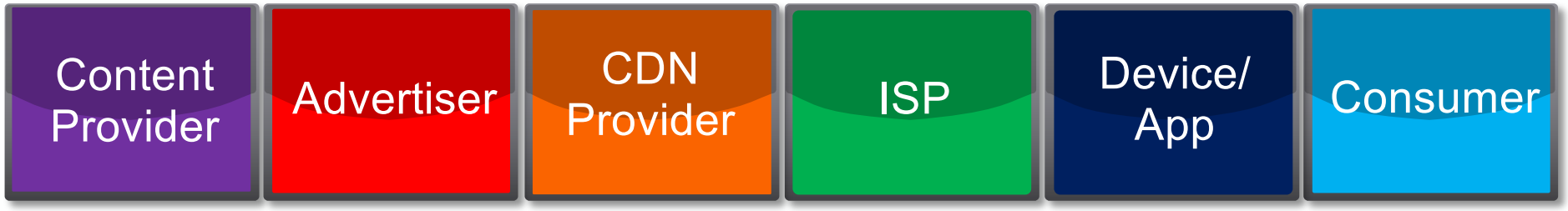
Looking for a **P**ermanent **h**ead **D**amage (Degree)?

Review of over 100 Adaptation Schemes



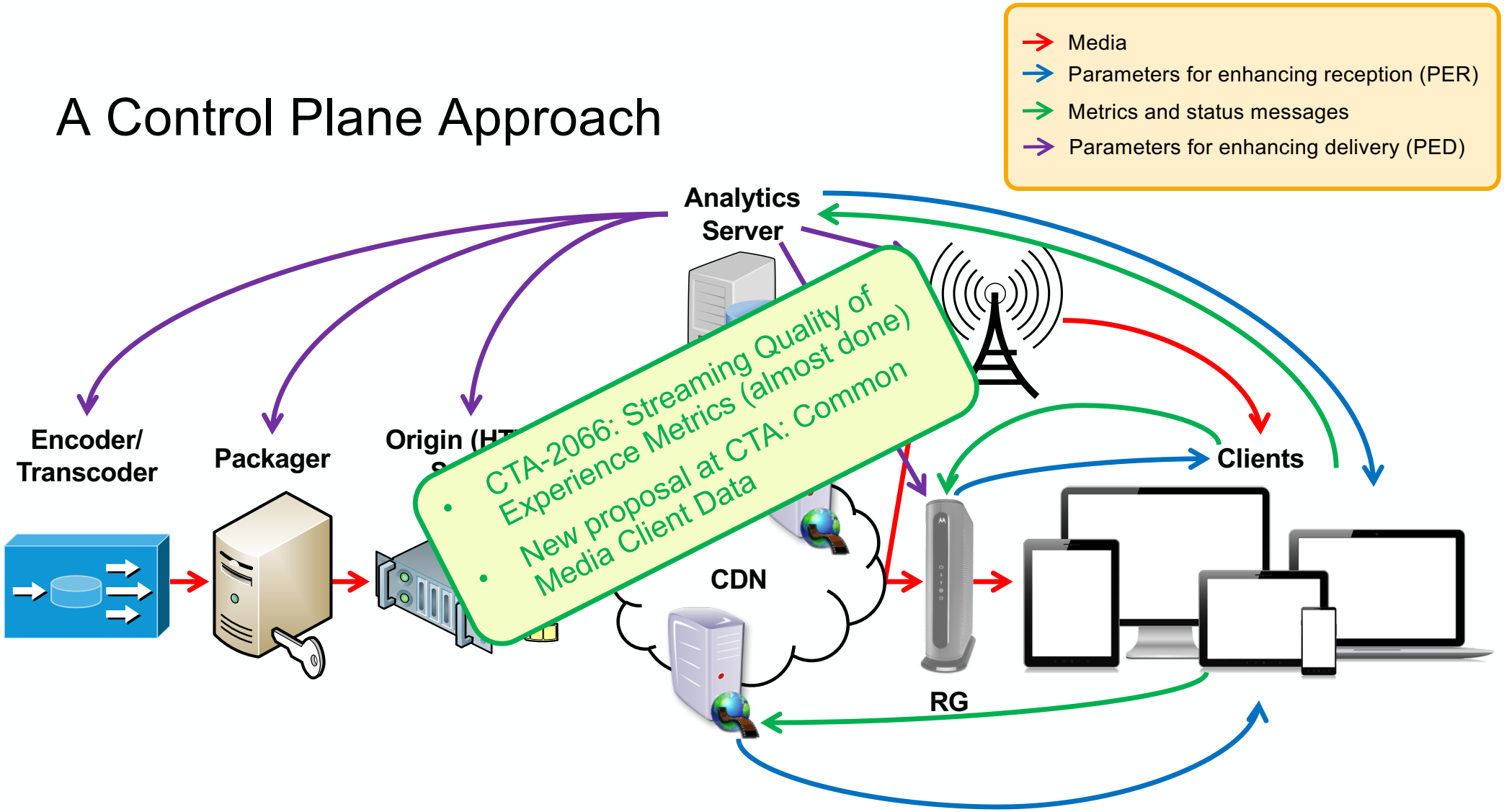
Reading: “A survey on bitrate adaptation schemes for streaming media over HTTP,” IEEE Commun. Surveys Tuts., 2019

Conflicting Goals in the Media Delivery Chain



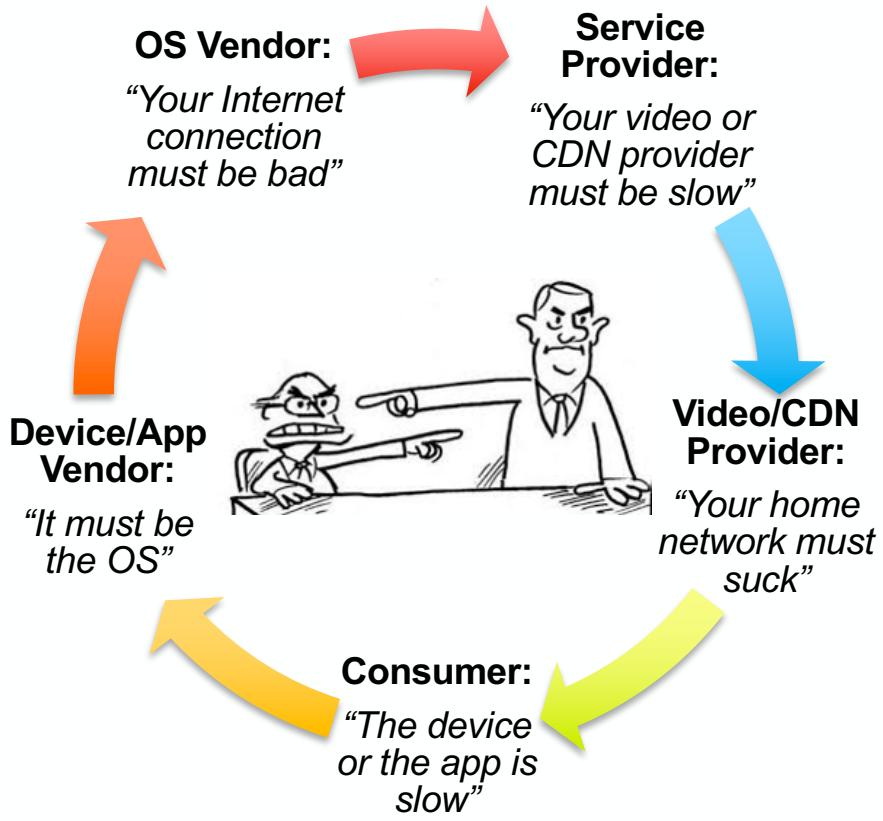
- I want to make sure that my content is protected and looks awesome on any device
- I want to make sure that my ads are viewed, trackable and measurable
- I want to make sure that my servers are properly used and latency is low
- I want to control the QoE of my customers, make $ from and differentiate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

A Control Plane Approach



Do We Need Analytics?

Yes, at the Source, Encoder, Packager, Origin, Cache, GW, Player, etc.



- Use a common language (e.g., CTA-2066) across players
- Beware infobesity

"Overabundance of information implies a scarcity of user attention"

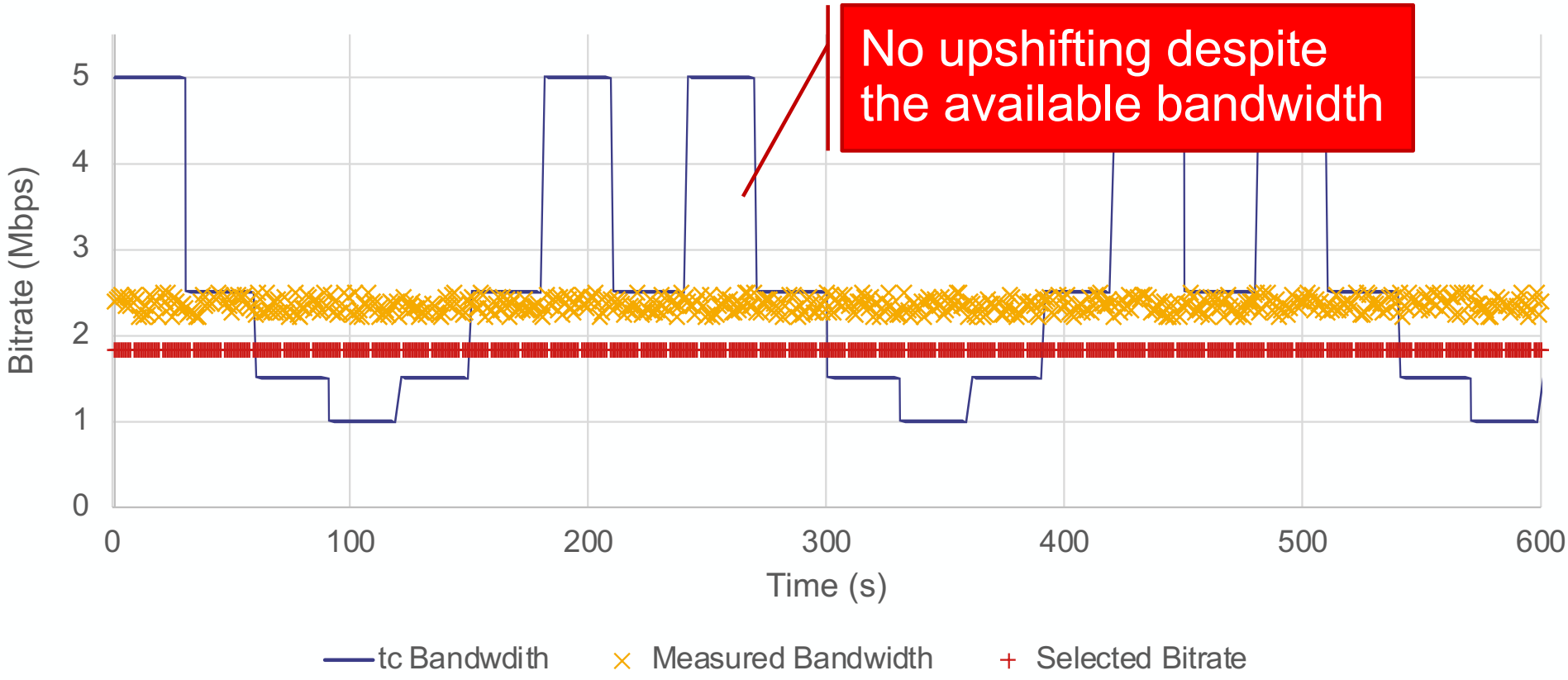
What about Latency

Predicting the Bandwidth Accurately in Low-Latency Apps

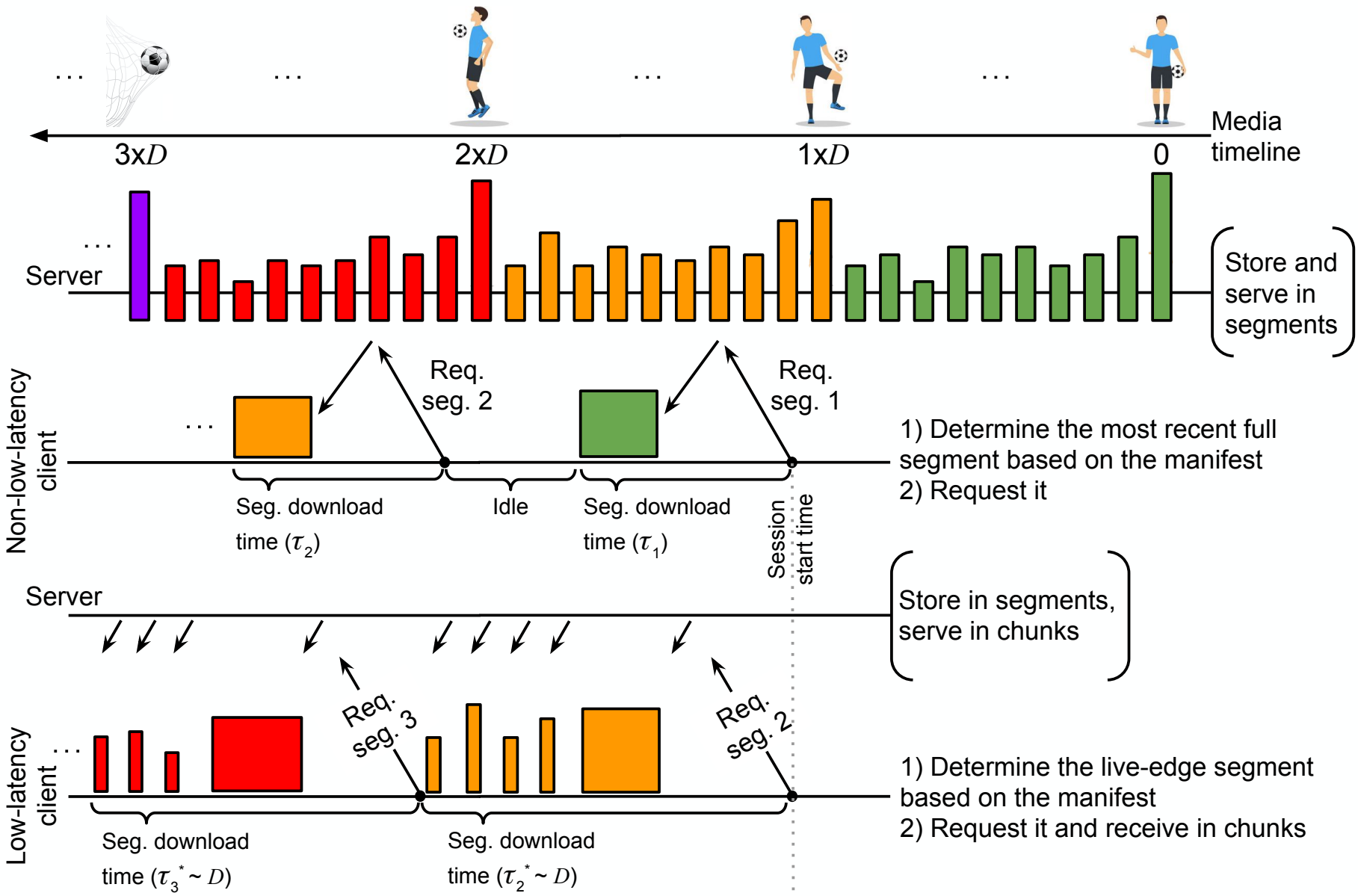
Reading: “Bandwidth prediction in low-latency chunked streaming,” ACM NOSSDAV 2019

Bandwidth Measurement is Tricky

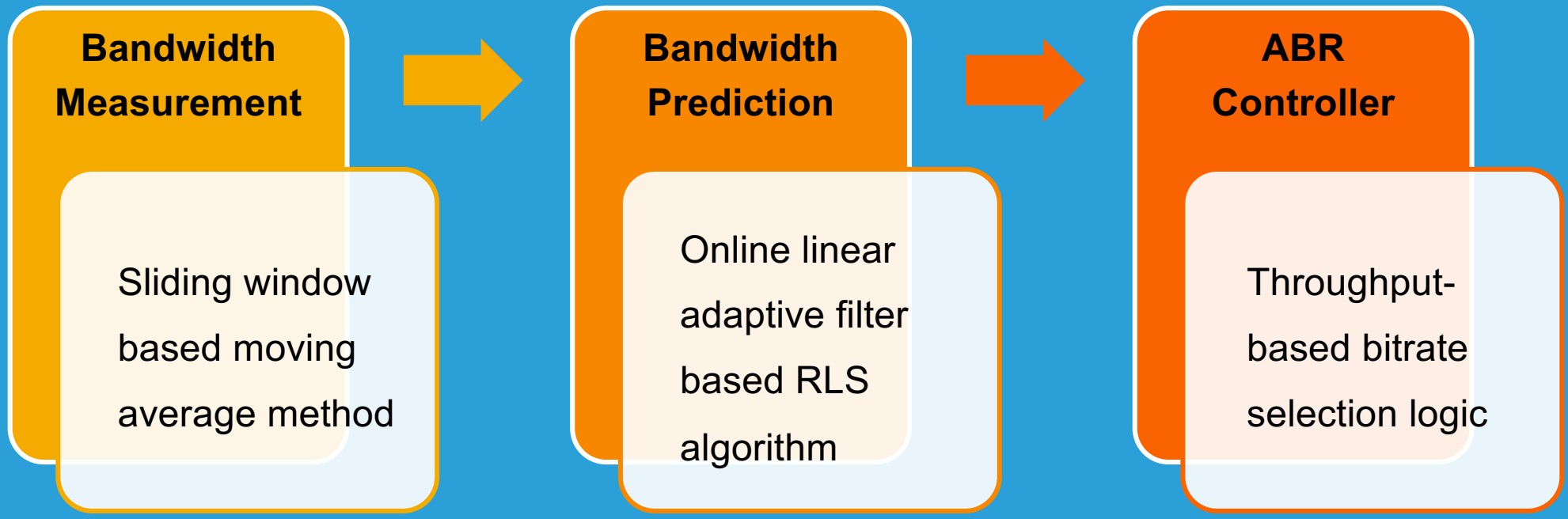
Live Twitch Data* (Nov. 2018)



* Encoded at {0.18, 0.73, 1.83, 2.5, 3.1, 8.8} Mbps with three resolutions of {540p, 720p, 1080p}, and packaged with CMAF

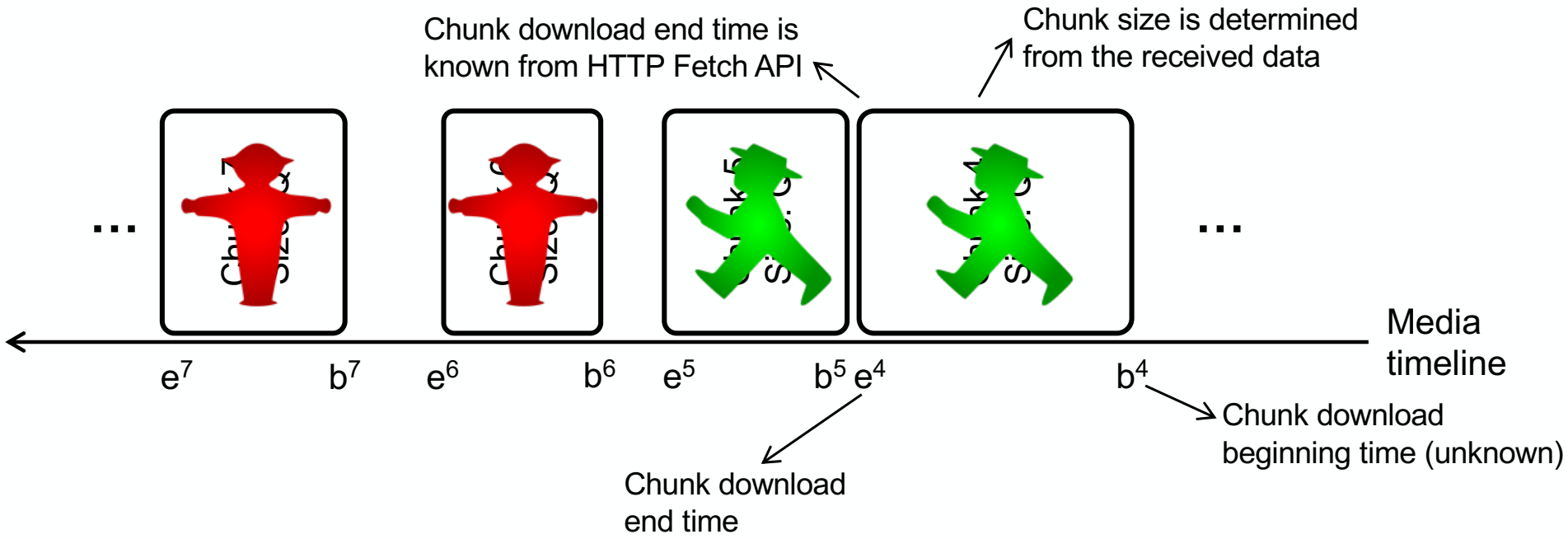

3xD
2xD
1xD
0
Media timeline
Server
Store and serve in segments
Non-low-latency client
Req. seg. 2
Req. seg. 1
Seg. download time (τ_2)
Idle
Seg. download time (τ_1)
Session start time
1) Determine the most recent full segment based on the manifest
2) Request it
Server
Store in segments, serve in chunks
Low-latency client
Req. seg. 3
Req. seg. 2
Seg. download time ($\tau_3^* \sim D$)
Seg. download time ($\tau_2^* \sim D$)
1) Determine the live-edge segment based on the manifest
2) Request it and receive in chunks

ABR for Chunked Transfer Encoding (ACTE)



Bandwidth Measurement

Identifying the “Good” Chunks



- Compute the download rate for the chunks where the transmission is network limited
  - If there is a negligible idle period after a chunk download, use that chunk, otherwise disregard it

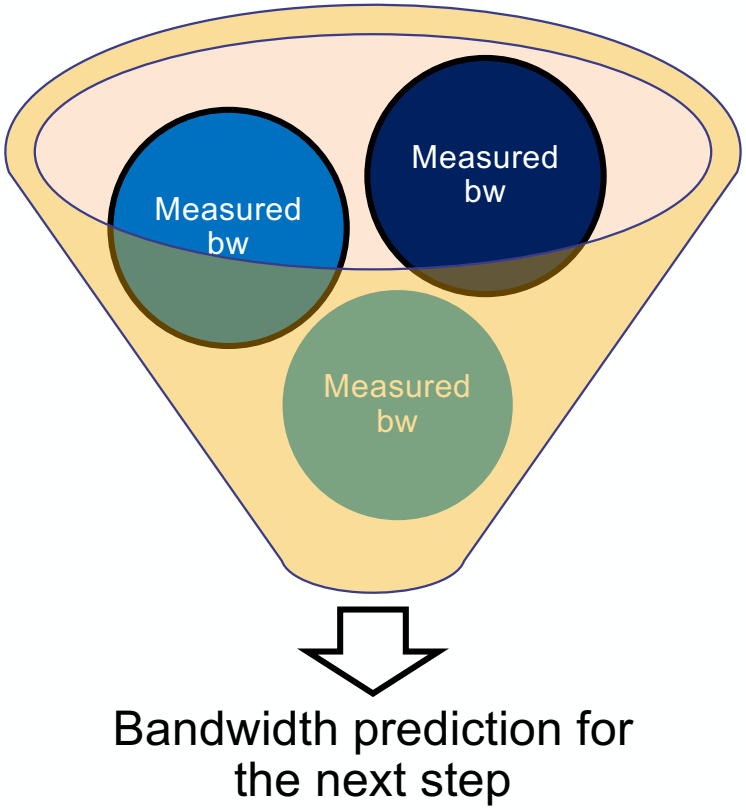
Bandwidth Measurement

Identifying the "Good" Chunks without the b^n Values

- Reasonable assumption: Idle periods cannot happen within a chunk, happen only between the chunks
  - Since the server pushes the chunks at full network speed
- For each chunk, compute its download rate, which equals its size divided by this chunk's end time minus previous chunk's end time
  - If this download rate is close (+/- 20%) to the average segment download rate, there must be significant idle time between these two chunks
    - Transmission is source limited
    - Disregard the current chunk
  - Else, the idle time is negligible
    - Transmission is network limited
    - The current chunk's download rate is a good approximation of the available bandwidth
- Use a sliding window based moving average method over the last three successful chunk downloads

(Future) Bandwidth Prediction

Online Linear Adaptive Filter Using Recursive Least Squares (RLS)



Performance Evaluation

Schemes Implemented

Bandwidth Measurement	ABR Schemes		
	Throughput-based	Buffer-based	Hybrid
SLBW	TH_{sl}	-	-
EWMA	TH_{ew}	-	-
SWMA	TH_{sw}	$BOLA_{sw}$	$Dynamic_{sw}$
WSSL	TH_{wss}	-	-

SLBW: Segment-based last bandwidth

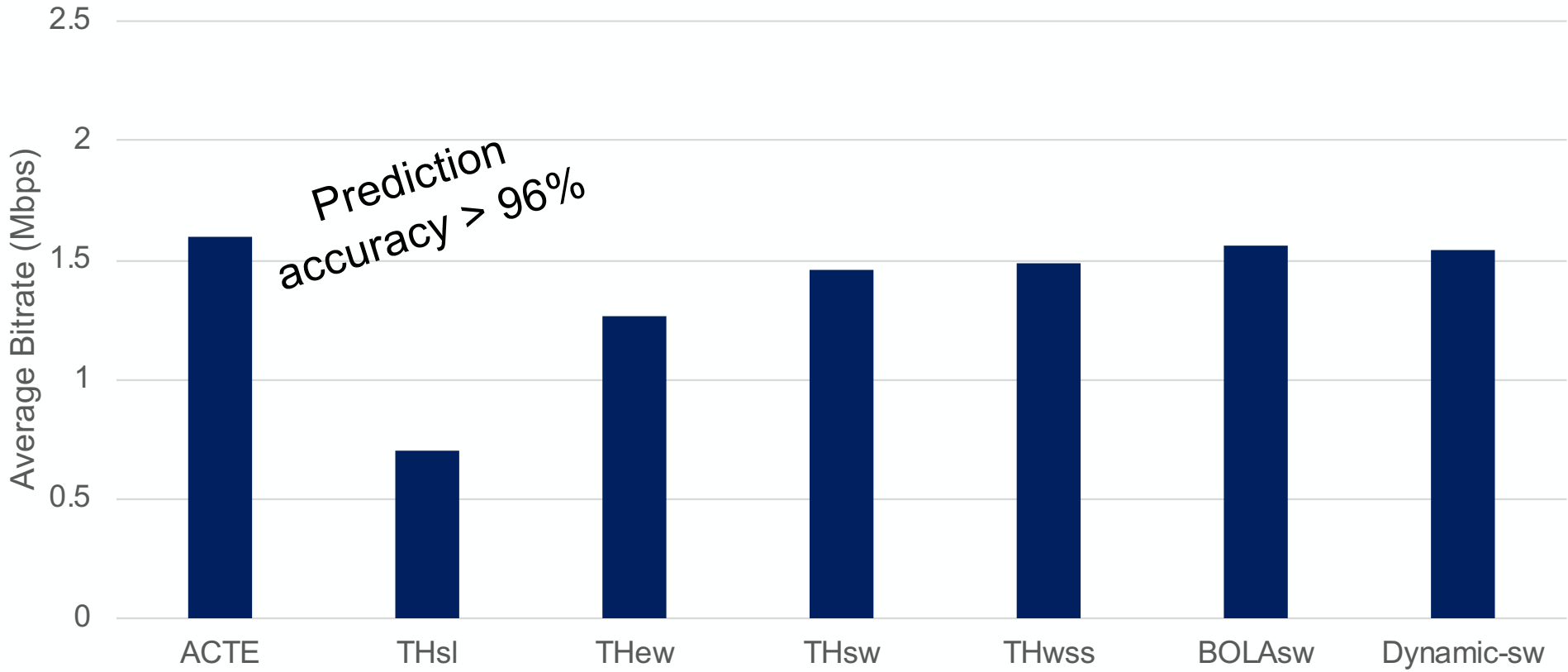
EWMA: Chunk-based exponentially weighted moving average

SWMA: Chunk-based sliding window moving average

WSSL: Will's simple slide-load

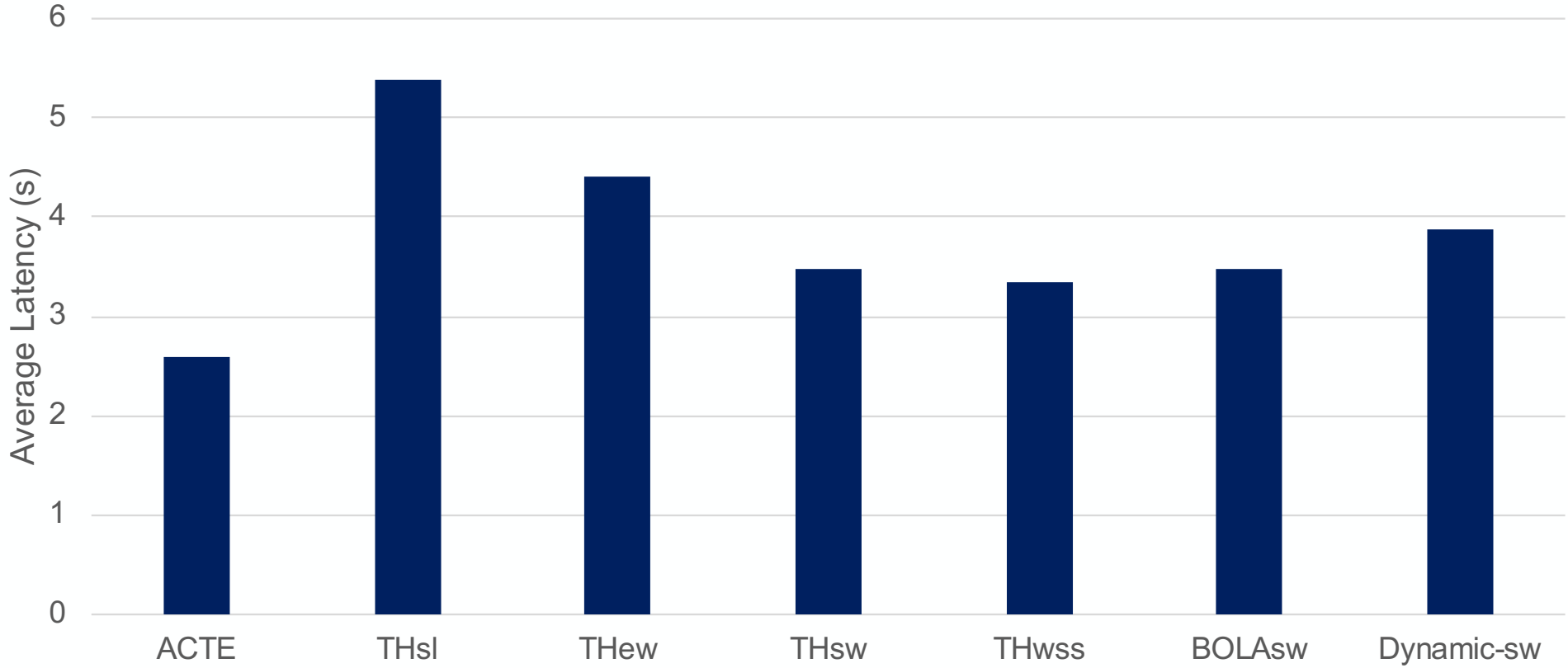
Average Selected Bitrate

28.6% Improvement by ACTE over Other Schemes



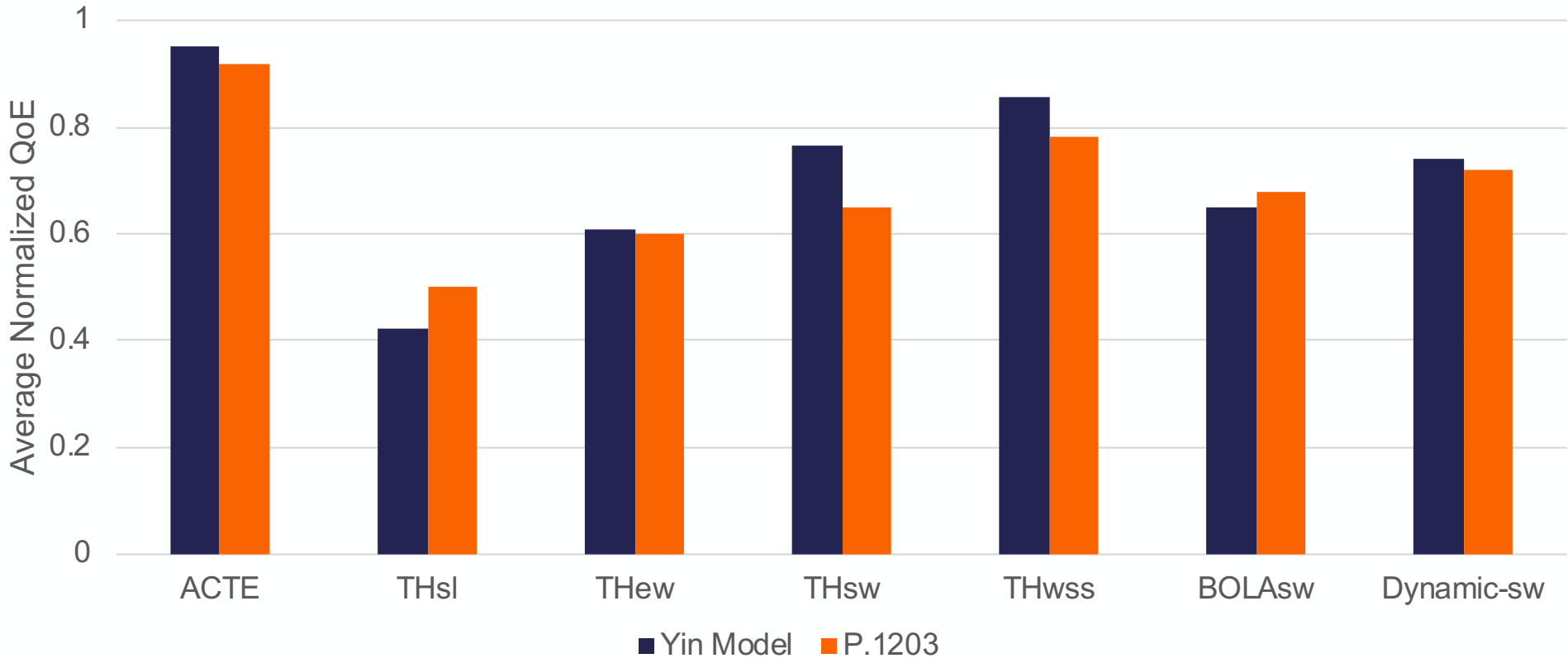
Average Live Latency

36.2% Improvement by ACTE over Other Schemes



Average Normalized QoE

49.3% Improvement by ACTE over Other Schemes



What about Quality

Encoding and Streaming in a Quality-Aware Fashion

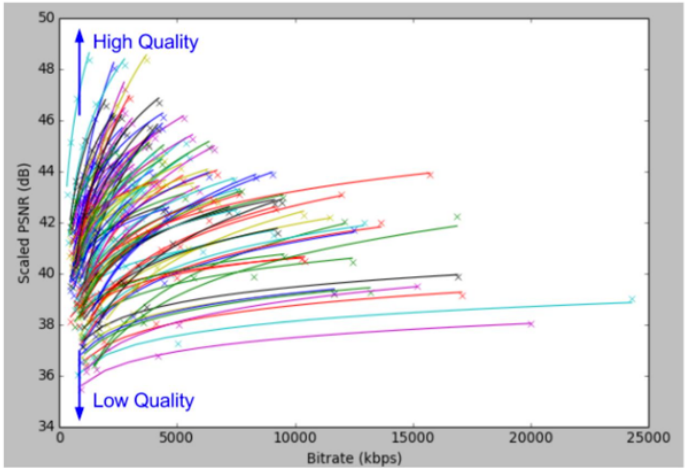
Reading: “Spending quality time with the Web video,” IEEE Internet Comput., 2016, and “Quality-aware HTTP adaptive streaming,” IBC 2015

Content-Based (Content-Aware) Encoding

Picking the Bitrate Ladder Based on the Content

Increased bandwidth brings new opportunities

Drives overall maturity of IP/ABR delivery technologies, introduces new opportunities – Per Title Encode Optimization

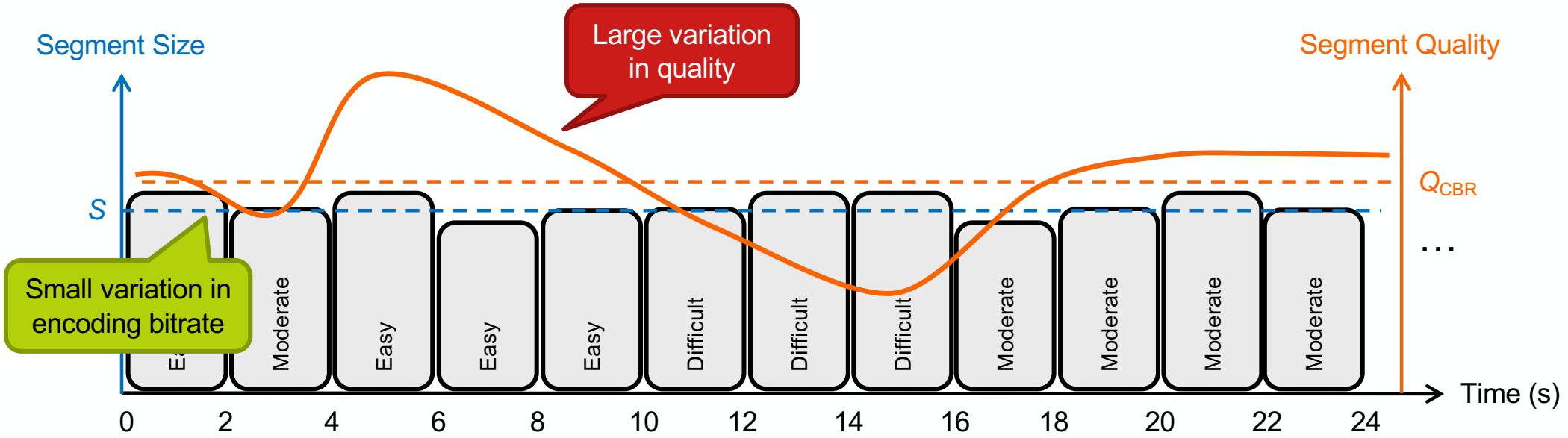


Source: Netflix Tech Blog, December 2015

Content-aware encoding gives us fairness in quality as opposed to fairness in bitrate

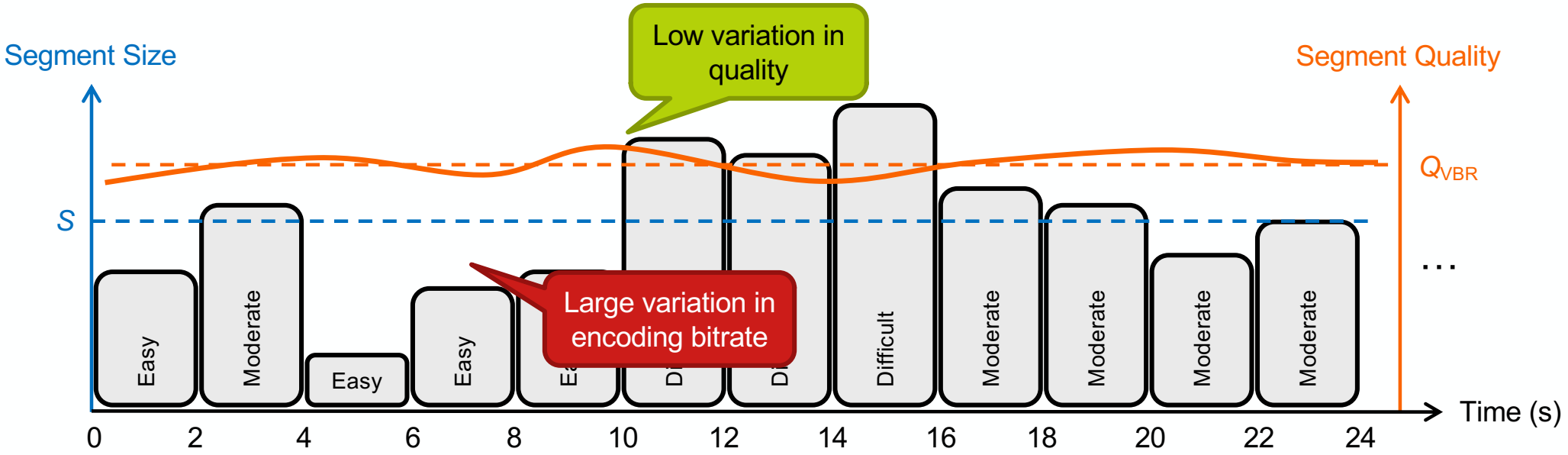
Adaptation Feature Delivers Inconsistent Quality

Guidelines Limited Bitrate Variability to (Mostly) 10% So Far



If there is something worse than having to watch a video at a lousy quality, it is to watch that video with varying quality

What If We Encode in a More Subtle Fashion?

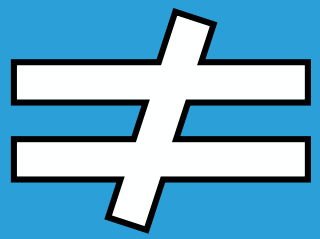


While we spend the same total amount of bits, we not only increase average quality but also reduce quality variation

HLS authoring spec for ATV allows 2x capping rate for VoD. For linear content, variability is limited to 10-25% range.

Generating VBR-encoded segments is easy, but streaming them is not!

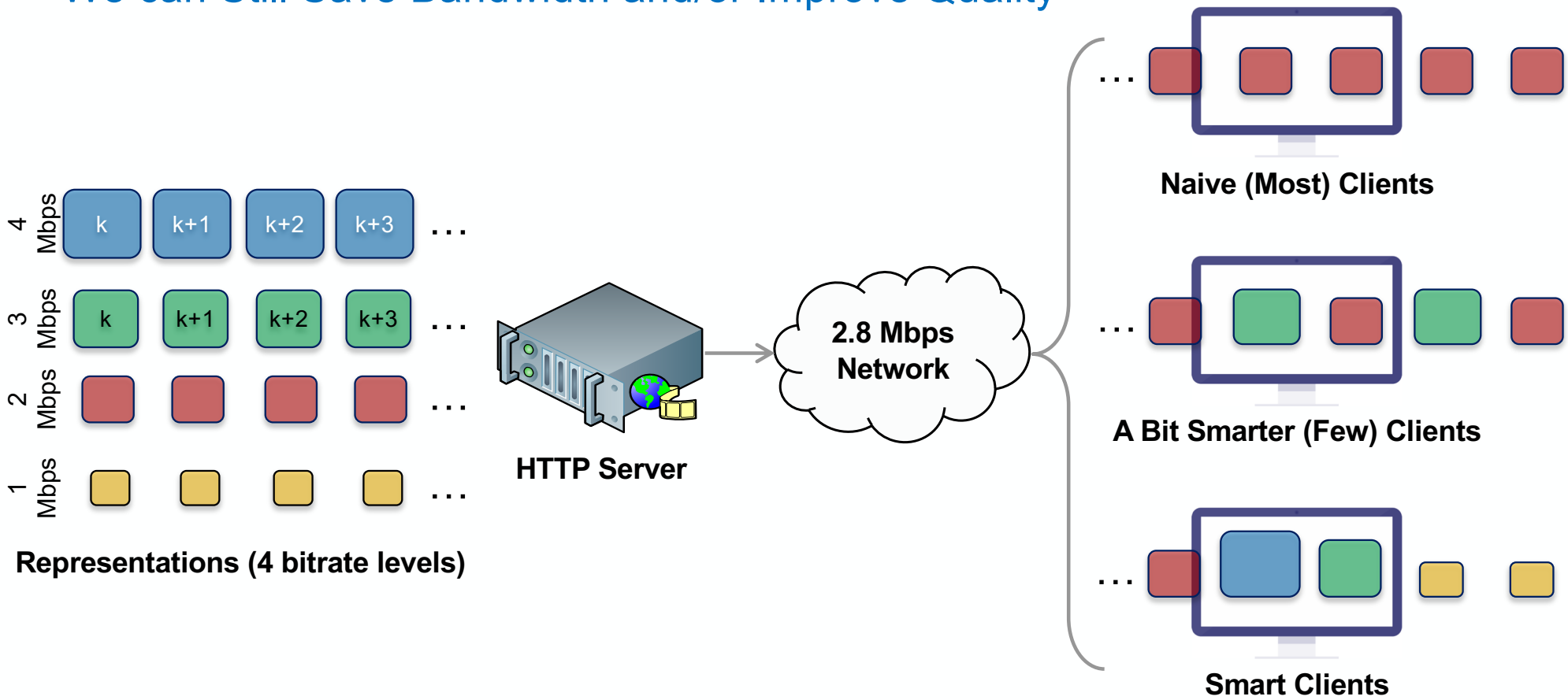
Content-aware Encoding



Content-aware Streaming

What If the Content is Already CBR Encoded

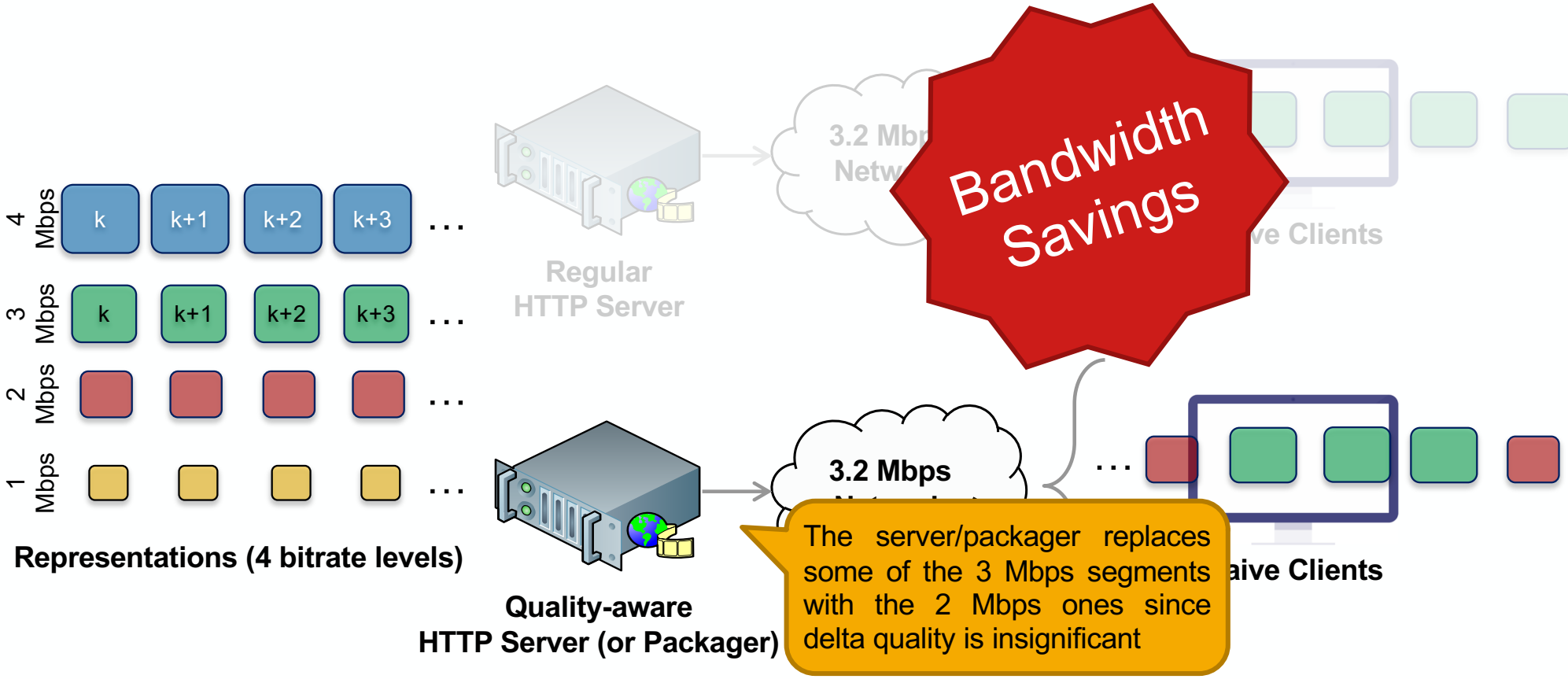
We can Still Save Bandwidth and/or Improve Quality



Reading: “Streaming video over HTTP with consistent quality,” ACM MMSys 2014

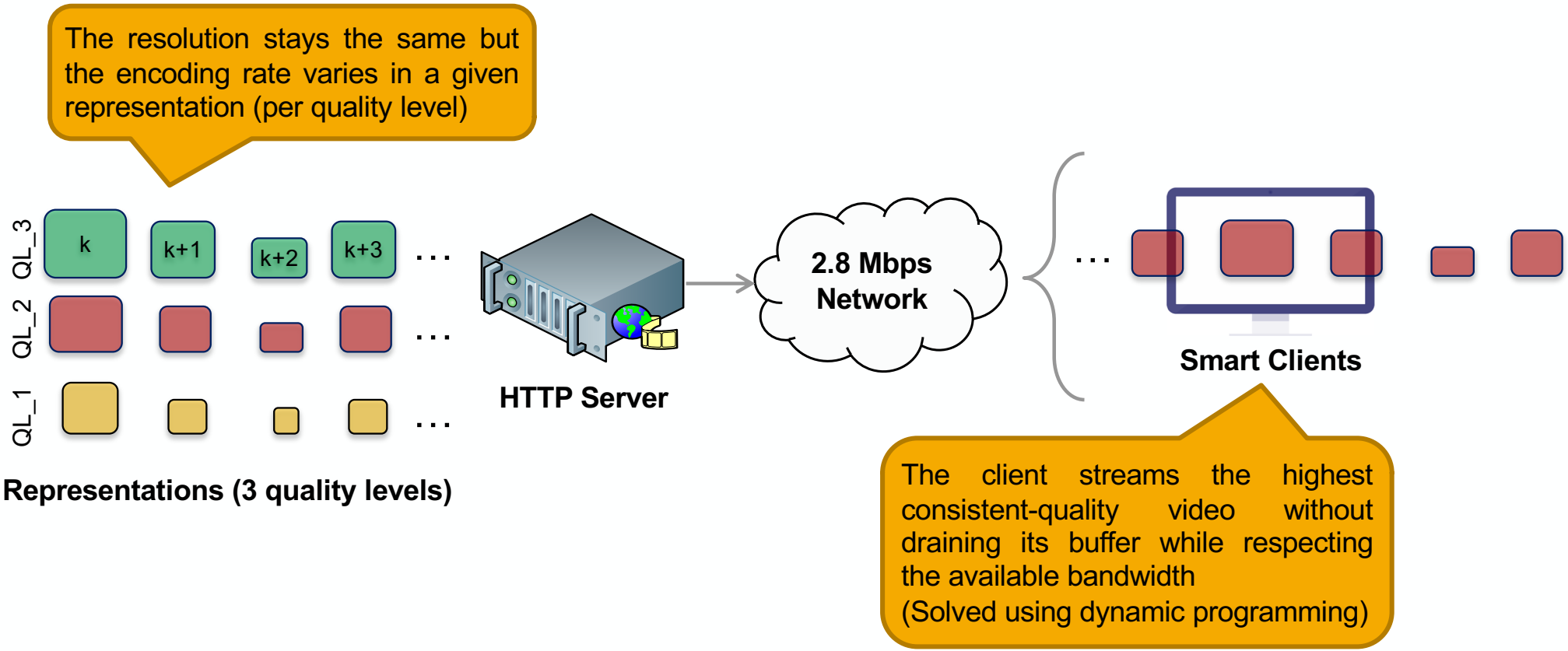
What If There is No Smartness in the Client

We can Still Save Bandwidth



Reading: “More juice less bits: content aware streaming,” ACM MMSys 2016

What If the Content is VBR Encoded



What If We Have Both CBR and VBR Encoded Content

Deployment Scenarios

- Most common scenario
  - CBR encoders and naive streaming clients
  - Capped VBR encoders (small caps) and naive clients
- Emerging scenario
  - Capped VBR encoders (small cap) and a bit smarter clients
- Best possible scenario
  - Capped VBR encoders (large cap) and smart clients

If you do not control both ends, follow Postel's Law (RFC 1122)

Be liberal in what you accept, conservative in what you send

If there are unknown clients:
- Send CBR encoded segments
- Set minBufferTime sufficiently large

Otherwise, the clients might get confused, break, stall or even crash

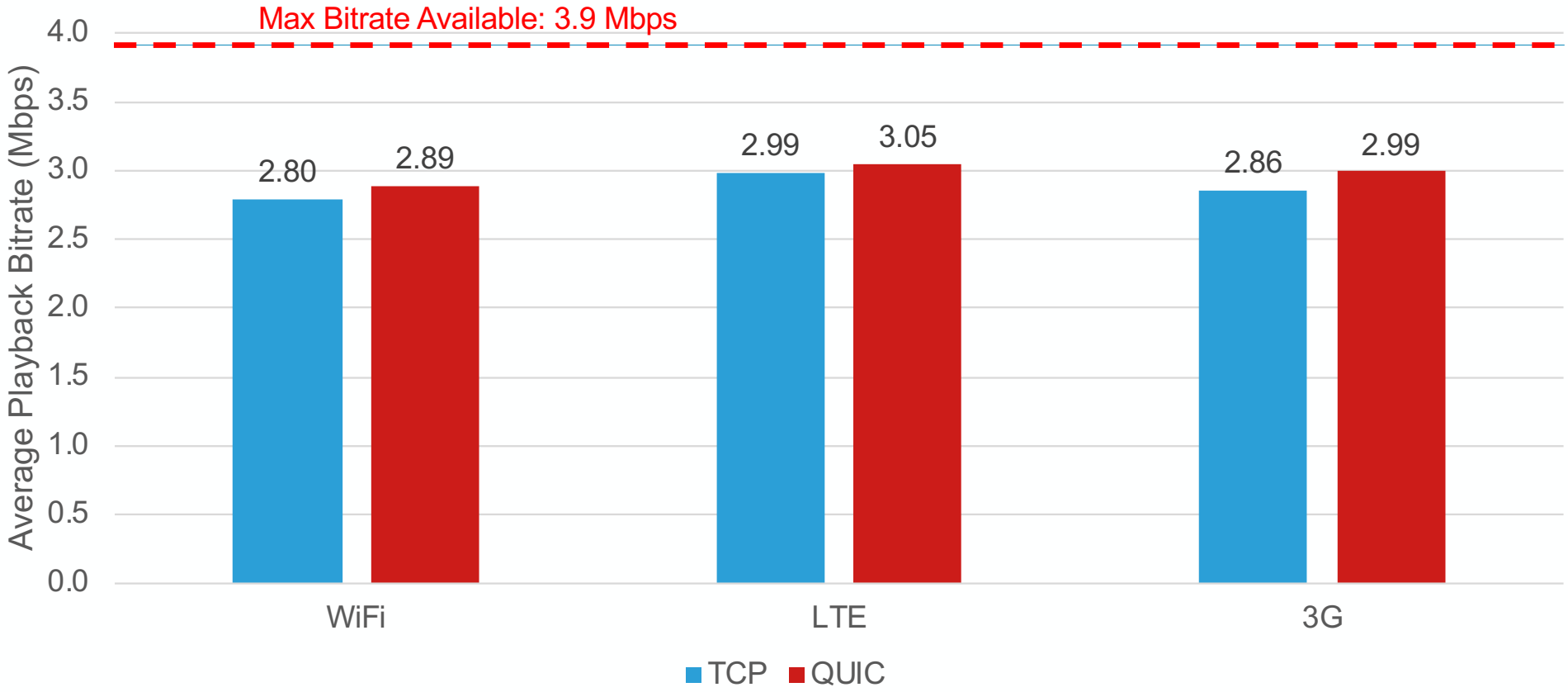
What about QUIC

Addressing Slow Startups, High Latency and Frequent Connection Changes

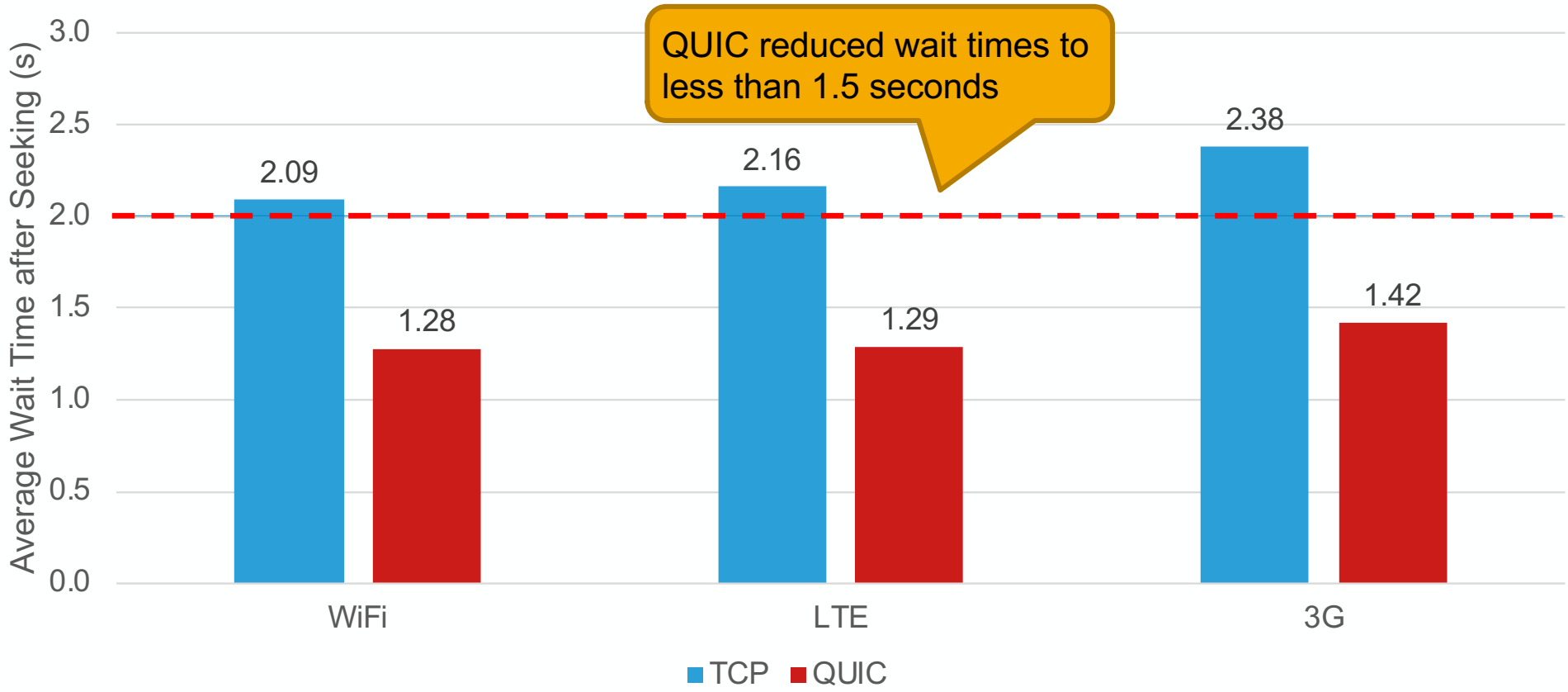
Reading: “Quickly starting media streams using QUIC,” Packet Video Wksp. 2018

Frame-Seek Scenario Results

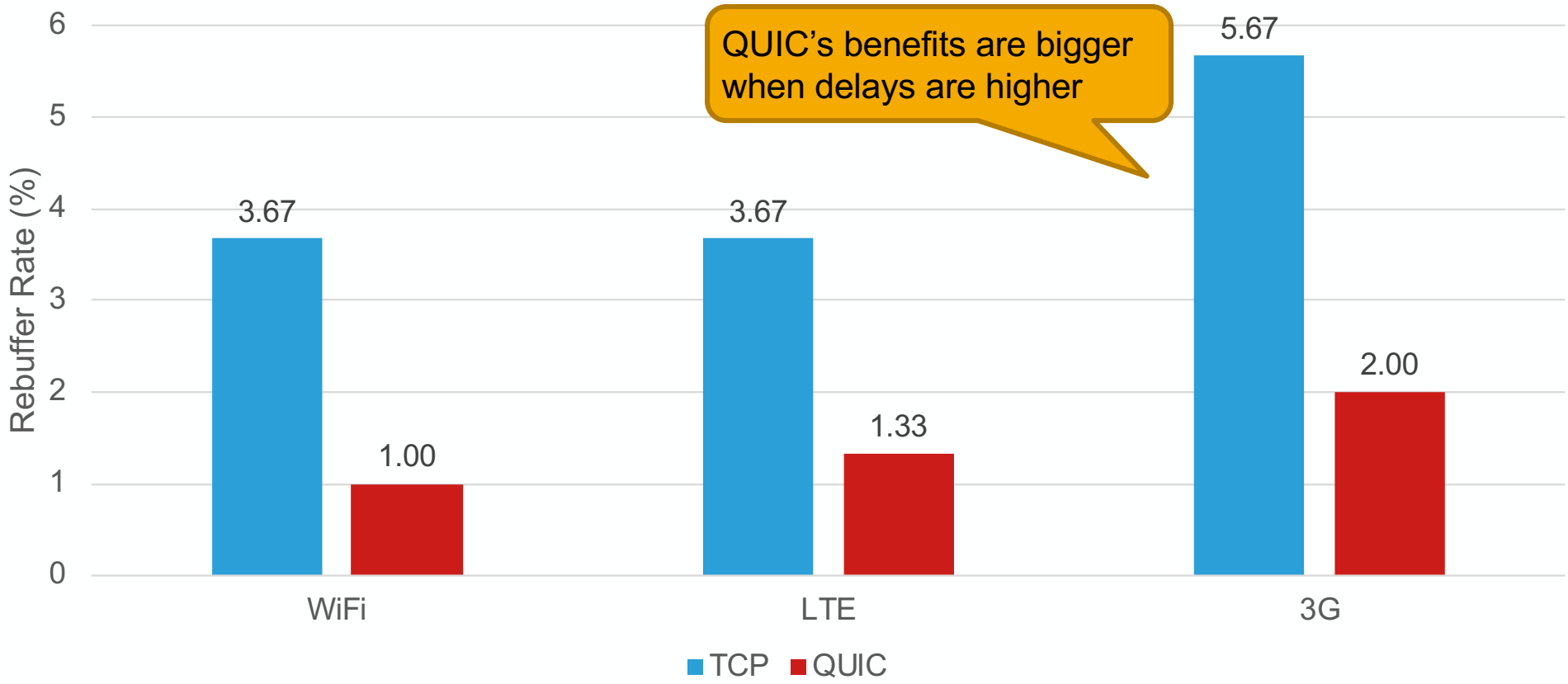
QUIC provided a higher (or at least an equal) average playback bitrate in all cases and for all algorithms



Frame-Seek Scenario Results

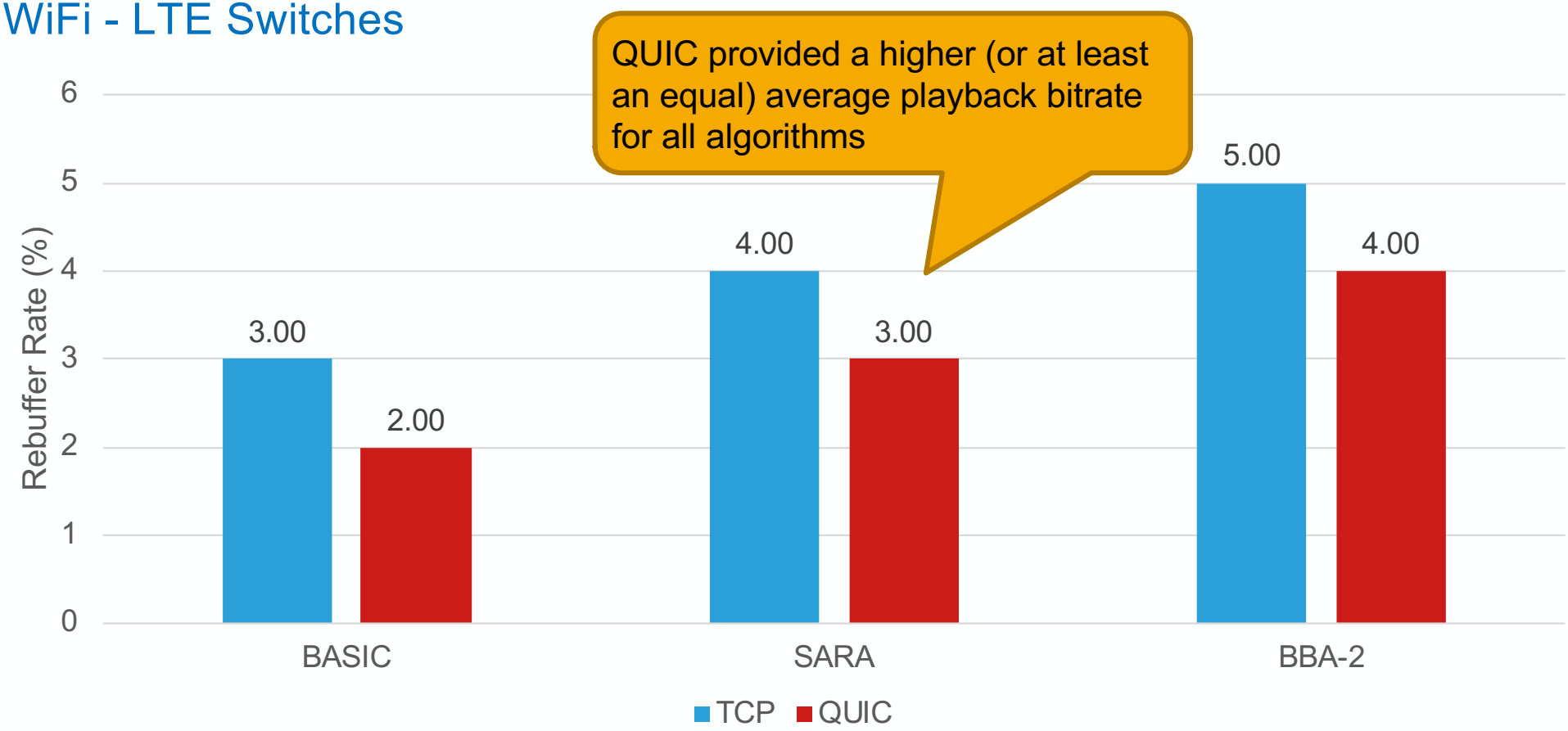


Frame-Seek Scenario Results



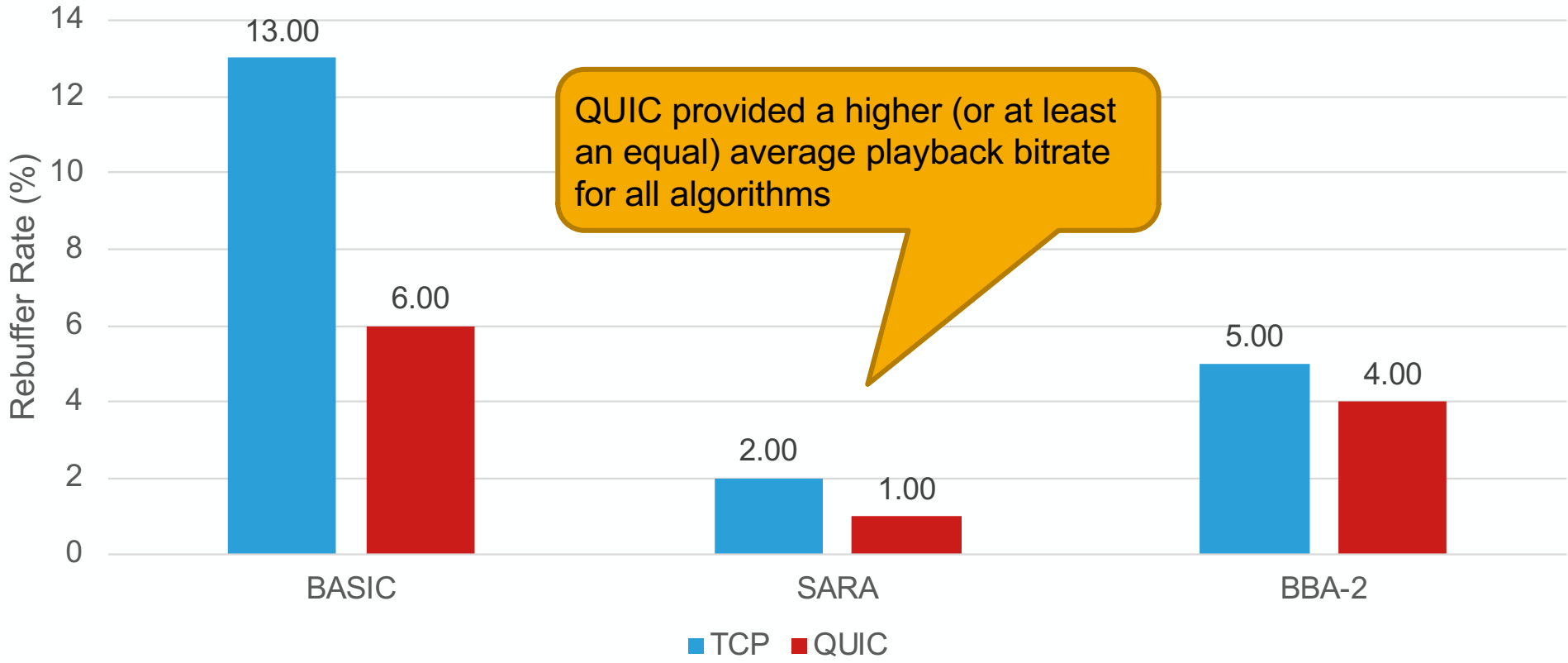
Connection-Switch Results

WiFi - LTE Switches



Connection-Switch Results

WiFi - 3G Switches



Bridging Deep Media and Communications

- Two grand challenges
  - Improving open-source HEVC encoding NEW
  - Low-latency live streaming
- Focus areas in 2020
  - Machine learning and statistical modeling for video streaming
  - Volumetric media: from capture to consumption
  - Fake media and tools for preventing illegal broadcasts
- A challenge dedicated to high-school students NEW
- Three confirmed keynotes from Google, MIT and Tsinghua
- Expecting reduced registration fees thanks to strong support

Important Dates	Submit by
Research Track	Jan. 10 **(firm)**
Demo Track	Feb. 29
Open Source/Dataset	Feb. 29
Workshops	Mar. 27
Conference	June 8-11

Visit http://acmmmsys.org today!

DON'T DRINK AND DRIVE, JUST STREAM AND FLY!

VISIT

HTTP://ALI.BEGEN.NET

FOR TUTORIALS AND PAPERS

ALSO AMAAAS

ALI_BEGEN@COMCAST.COM