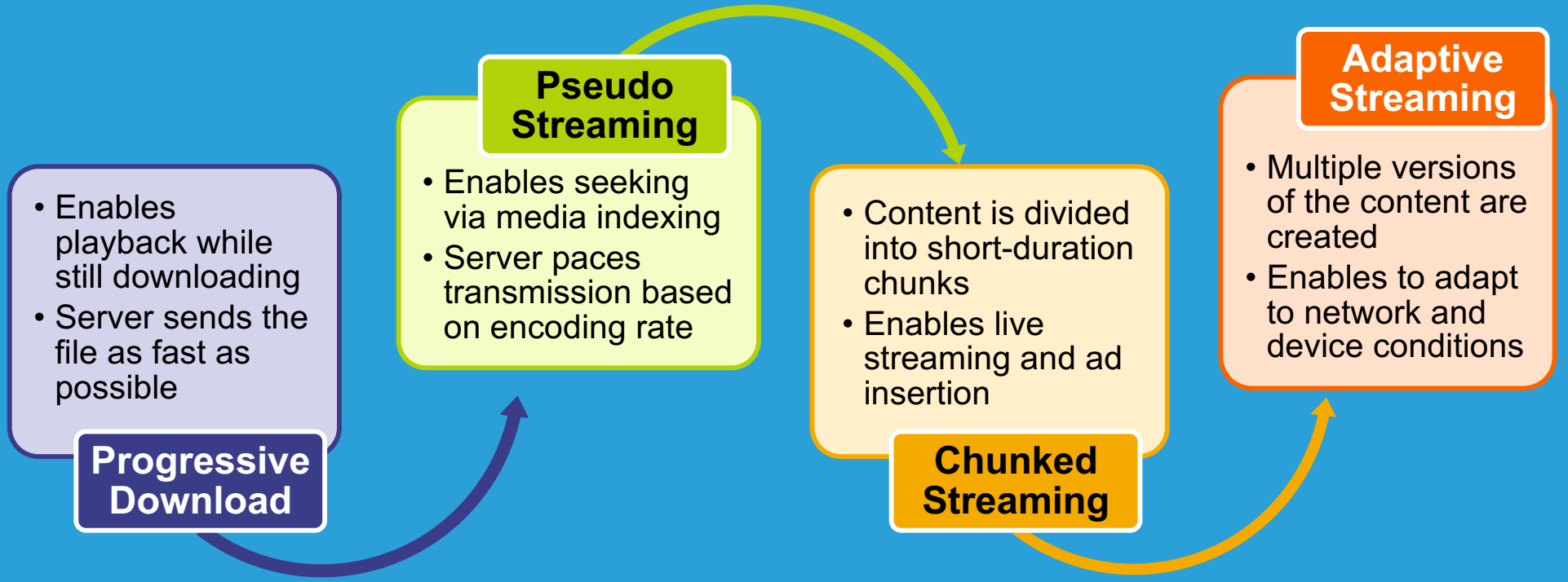

ANATOMY OF A DASH CLIENT
1. TASKS
2. SCALING
3. ANALYTICS
4. COOL FEATURES
ALI C. BEGEN, PH.D.
HTTP://ALI.BEGEN.NET
COMEDY CENTRAL

Video Delivery over HTTP



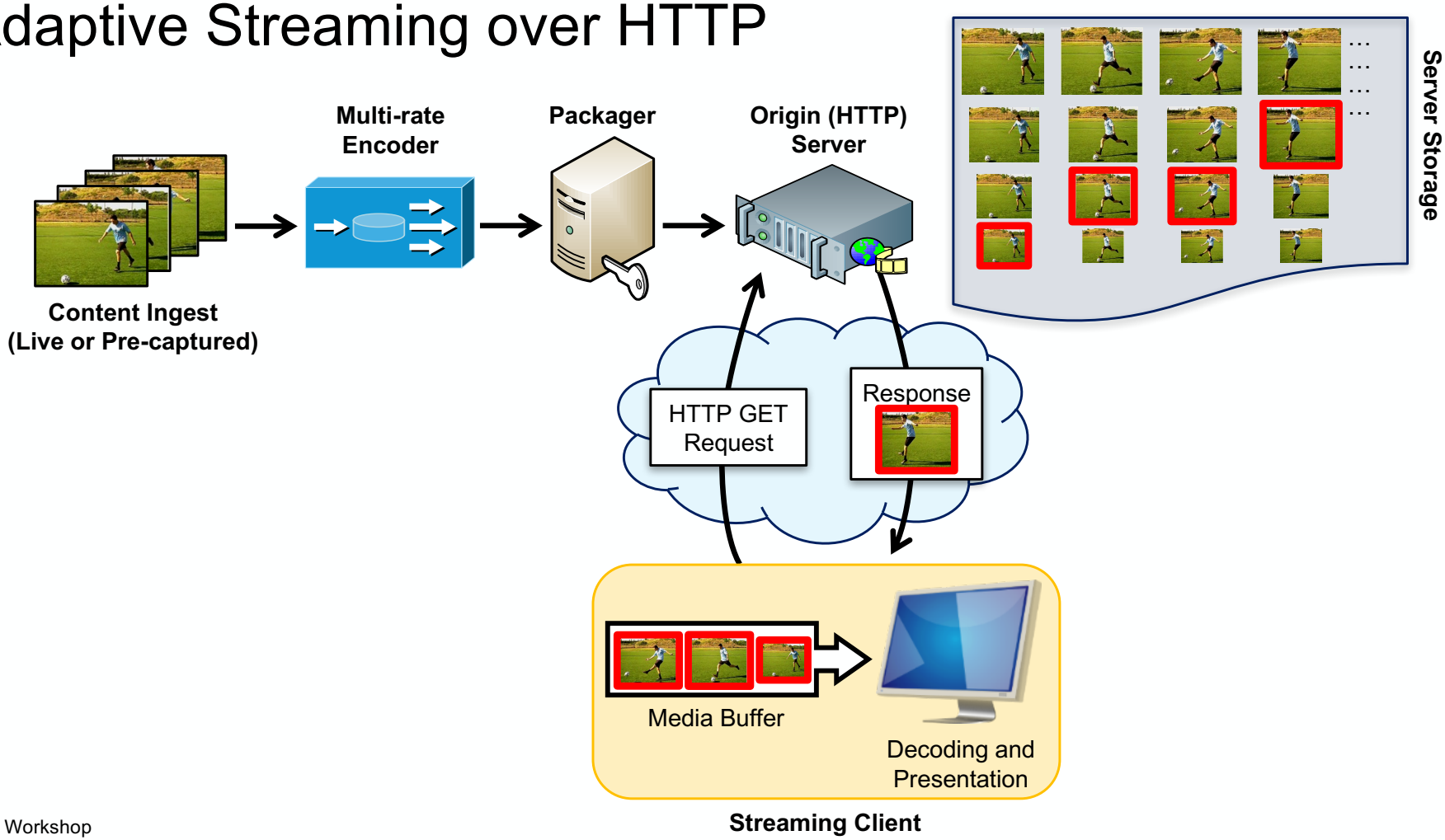
Dead, Surviving, Maturing and Newborn Technologies

- Move Adaptive Stream (Long gone)
  - http://www.movenetworks.com
- Microsoft Smooth Streaming (Legacy)
  - http://www.iis.net/expand/SmoothStreaming
- Adobe Flash (Almost dead)
  - http://www.adobe.com/products/flashplayer.html
- Adobe HTTP Dynamic Streaming (Legacy)
  - http://www.adobe.com/products/httpdynamicstreaming
- Apple HTTP Live Streaming (The elephant in the room)
  - http://tools.ietf.org/html/draft-pantos-http-live-streaming (Soon to be an RFC)
- MPEG DASH and CMAF (The standards)
  - http://mpeg.chiariglione.org/standards/mpeg-dash
  - http://mpeg.chiariglione.org/standards/mpeg-a/common-media-application-format



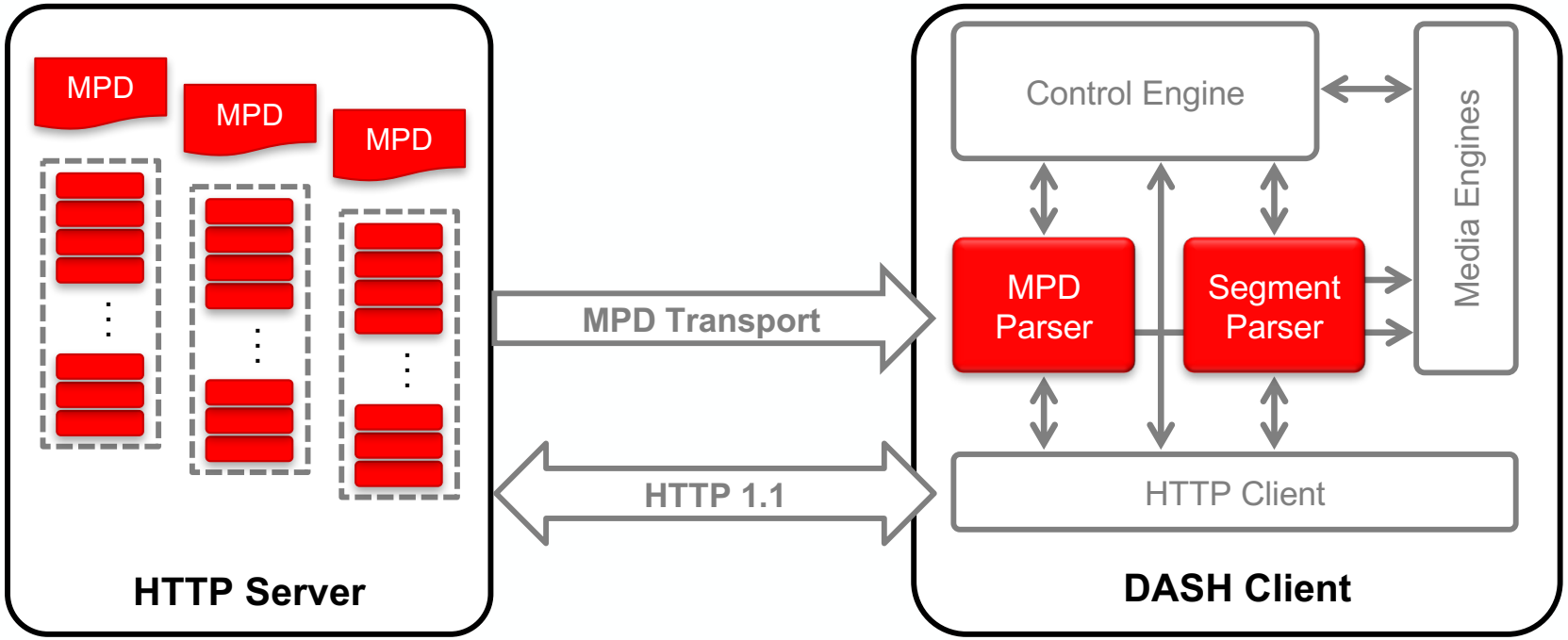


Adaptive Streaming over HTTP



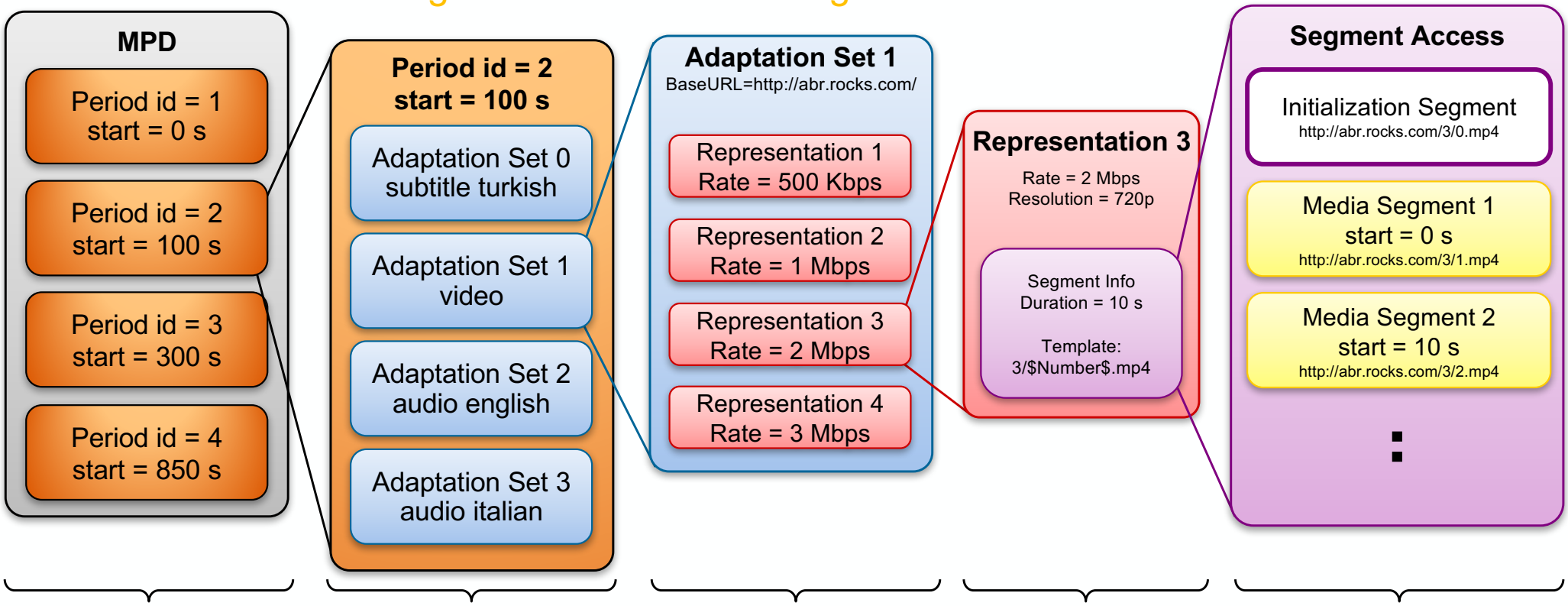
Scope of MPEG DASH

Shown in Red



DASH Manifest – Media Presentation Description (MPD)

List of Accessible Segments and Their Timings



Smart Clients

- Client fetches and parses the manifest
- Client uses the OS-provided HTTP stack (HTTP may run over TCP or QUIC)
- Client uses the required decryption tools for the protected content

Client monitors and measures
- Size of the playout buffer (both in bytes and seconds)
- Chunk download times and throughput
- Local resources (CPU, memory, window size, etc.)
- Dropped frames

Request

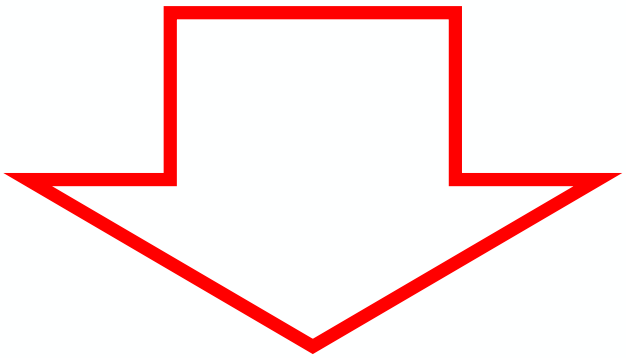
Response

(One can also multicast media segments)

Client performs adaptation

Client measures and reports metrics for analytics

Tradeoffs in Adaptive Streaming



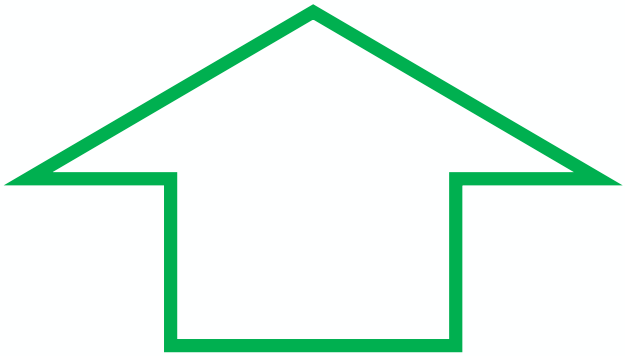
Overall quality

Quality stability

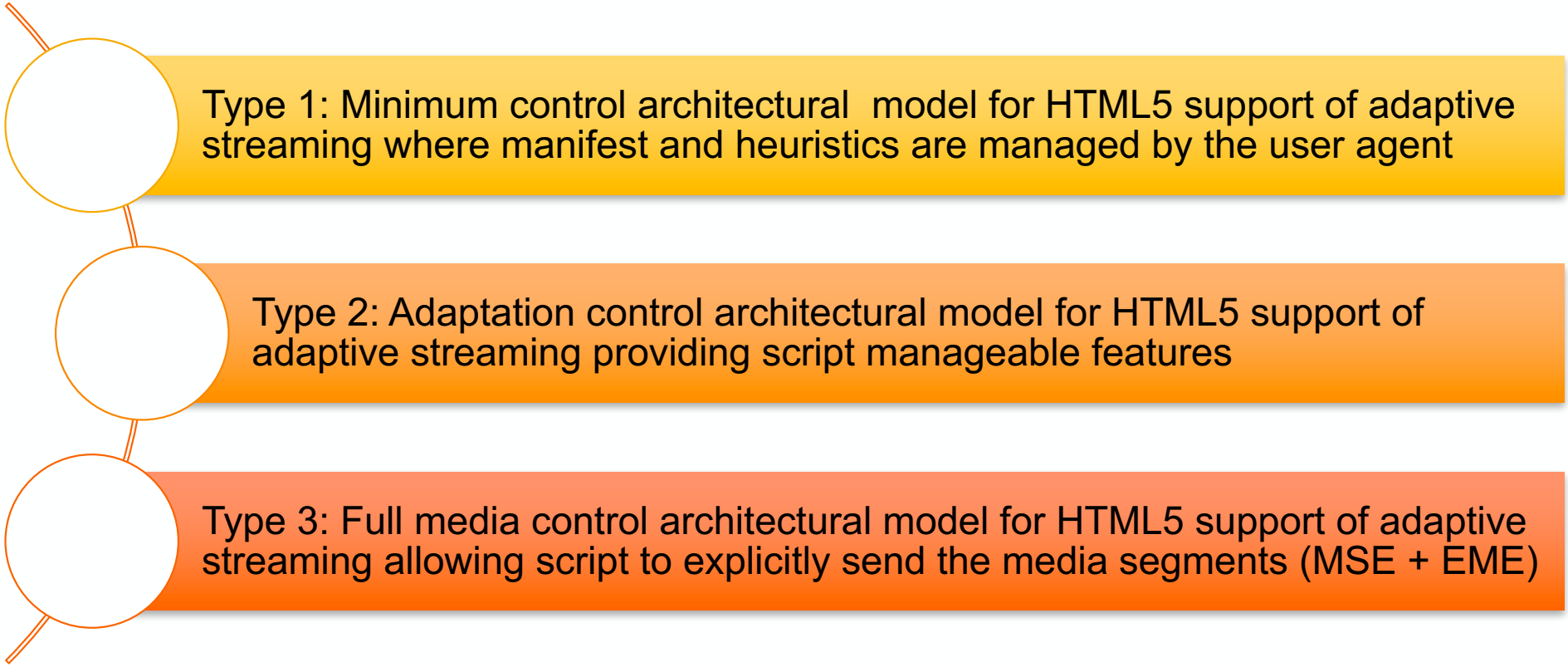
Proximity to live edge

Stalls

Zapping/seeking time

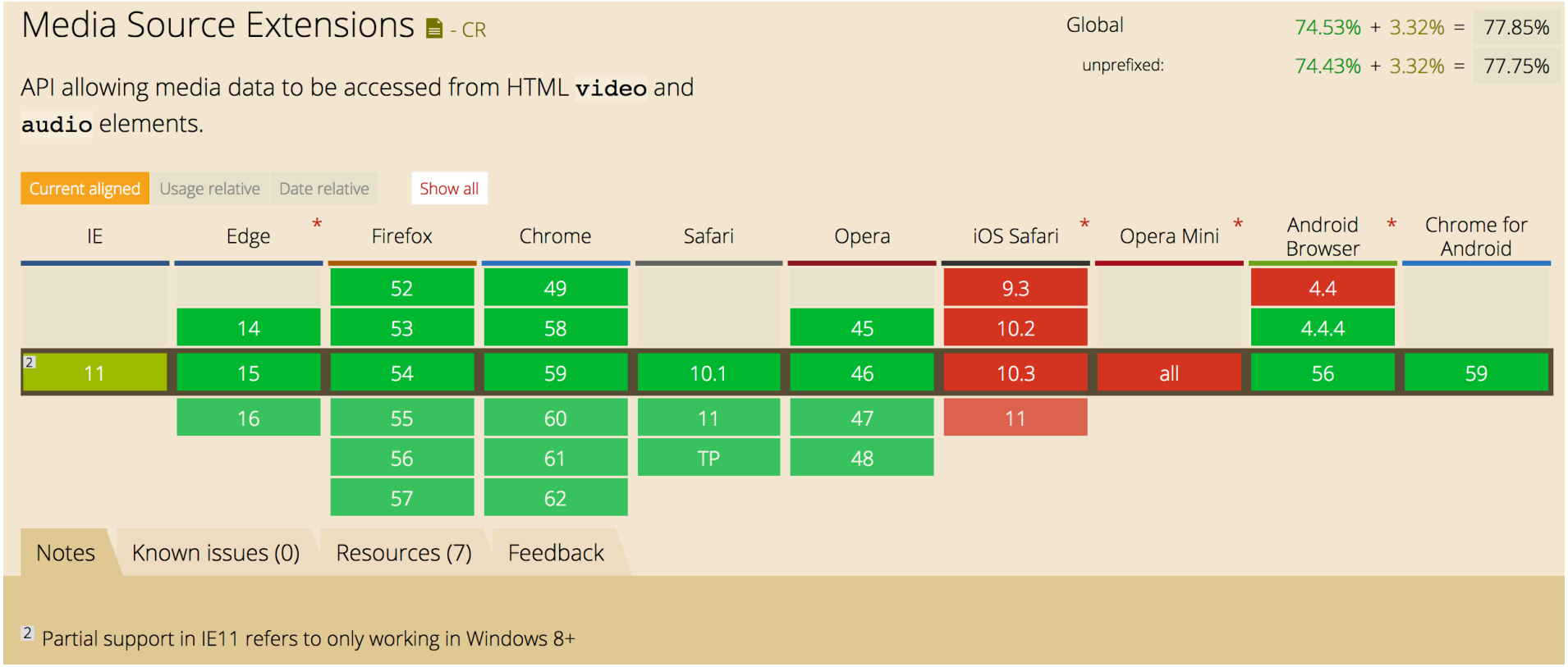


Types of Browser-Based Playback



Source: CTA WAVE

MSE Support in Web Browsers



Source: http://caniuse.com/#search=mse

DRM Support on Desktop Browsers

Browser	OS	EME/CDM	Flash Player	NPAPI/ Silverlight 5
Chrome	Win	Widevine	(Yes)	No
	OS X		(Yes)	No
	Linux		(Yes)	No
Firefox	Win	Widevine	Yes	No
	OS X		Yes	No
	Linux		Yes	No
Safari	> OS X Yosemite	(FairPlay)	Yes	Yes
	< OS X Yosemite	No	Yes	Yes
IE/Edge	< Win 7	No	Yes	Yes
	Win 10	PlayReady	Yes	No

DRM Support on Mobile Platforms

Platform	Browser	Native App / DRM
iOS	No MSE/EME yet HLS (AES-128 CBC) via <video>	Native SDK and WebView: FairPlay
Android	MSE/EME	Native + Android SDK (MediaDRM APIs) (Widevine + OMA v2) ExoPlayer Native + WebView with Widevine
Windows 10	MSE/EME	WebViews/Hosted Web Apps with PlayReady

Streaming over HTTP – The Promise

- Leverage tried-and-true Web infrastructure for scaling
  - Video is just ordinary Web content!
- Leverage tried-and-true TCP
  - Congestion avoidance
  - Reliability
  - No special QoS for video

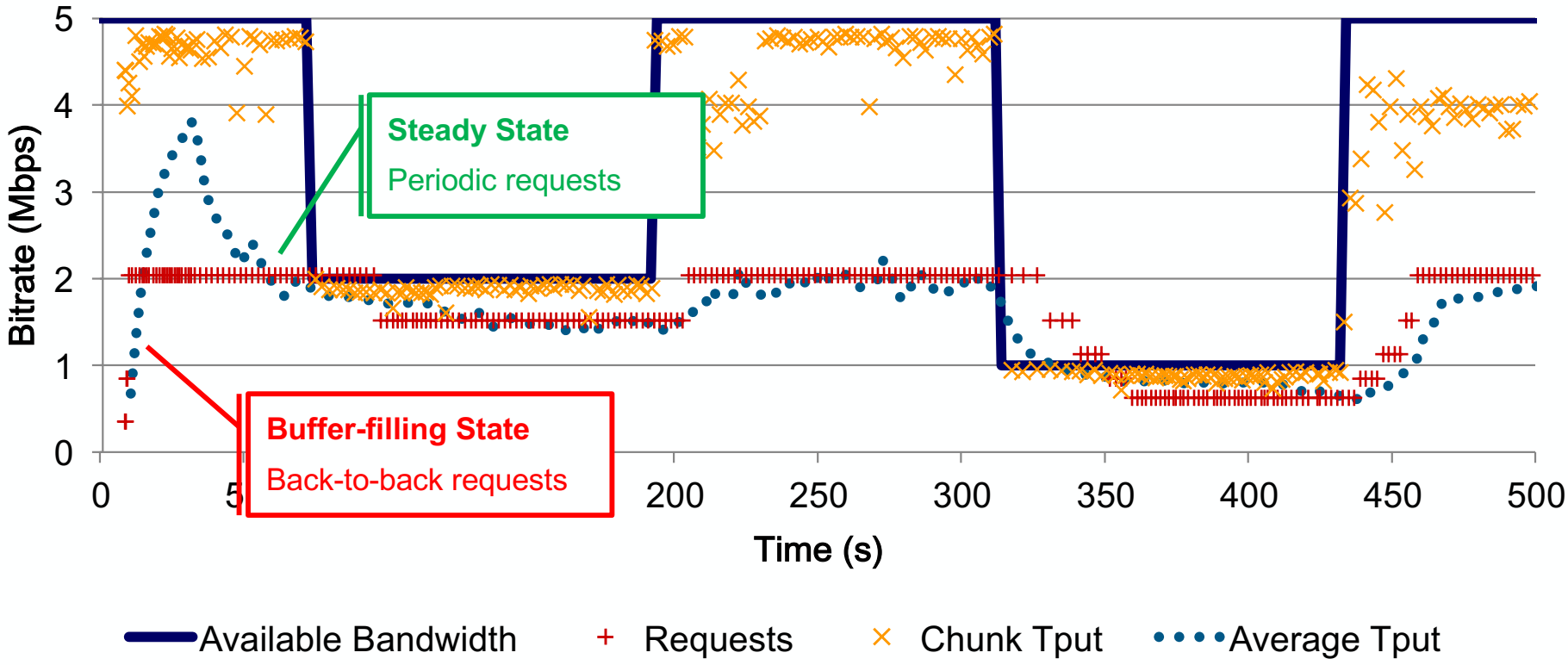
It should all "just work" ☺

Does It Just Work?

- When a streaming client competes with other types of traffic, mostly yes
- When streaming clients compete with each other, we begin to see problems:
  - The clients' adaptation behaviors interact with each other
  - The competing clients form an "accidental" distributed control-feedback system
- Unexpected behaviors will result in places like
  - Multiple screens within a household
  - ISP access and aggregation links
  - Small cells in stadiums and malls

Demystifying a Streaming Client

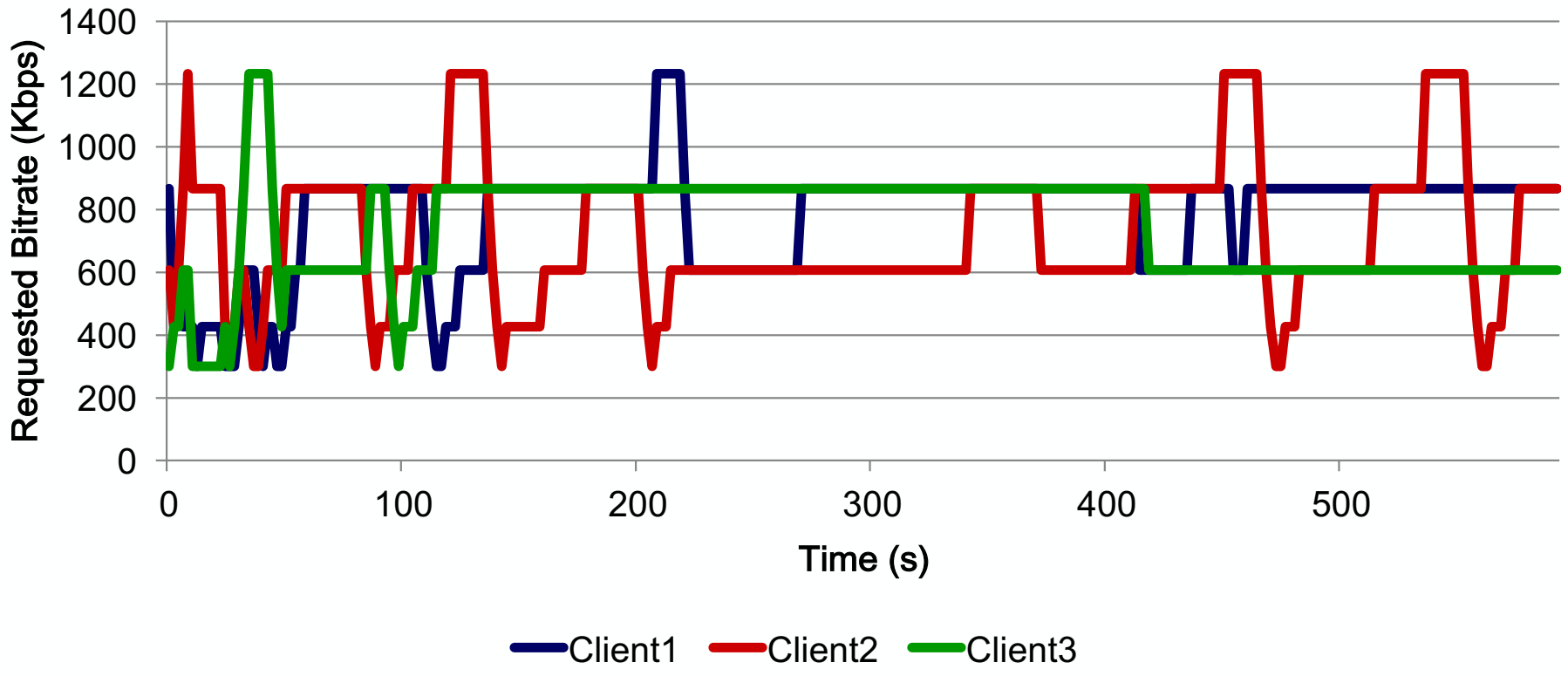
A Single Microsoft Smooth Streaming Client under a Controlled Environment



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

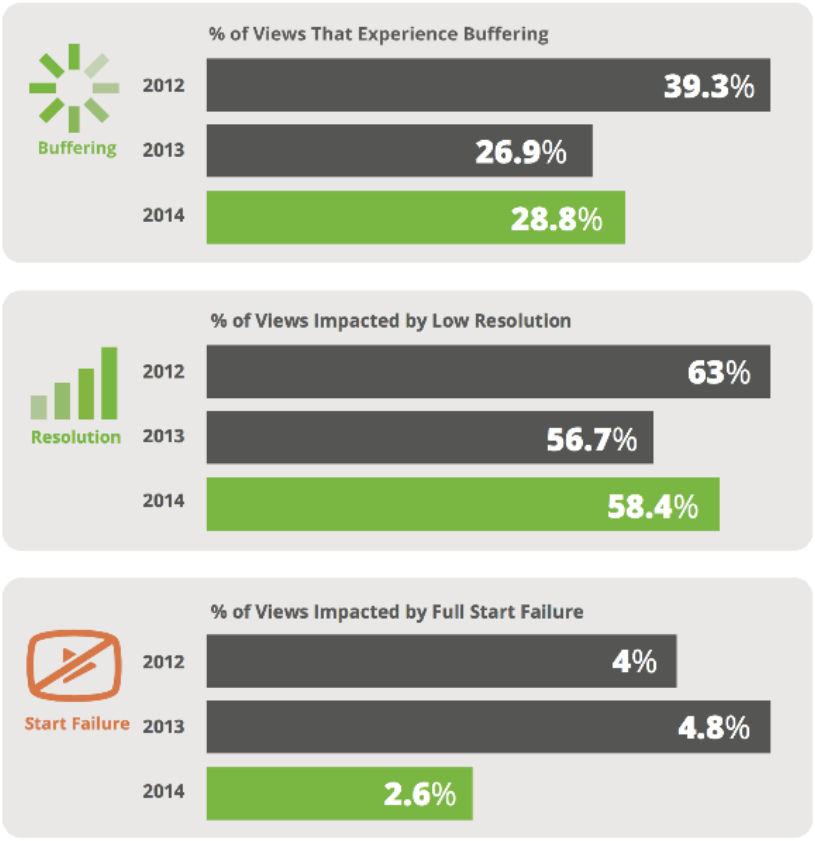
A Simple Test Scenario

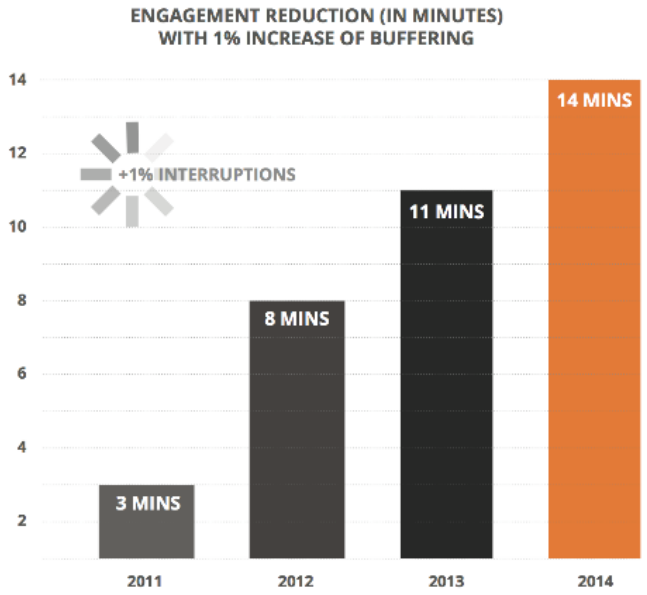
10 (Commercial) Streaming Clients Sharing a 10 Mbps Link

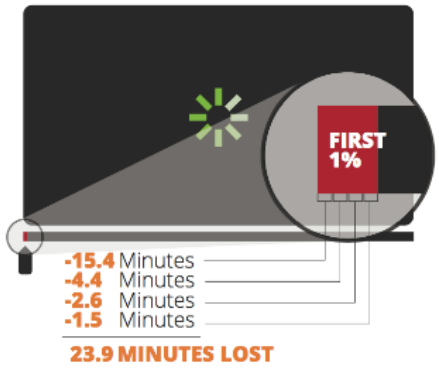


Selfishness Hurts Everyone

Viewer Experience Statistics

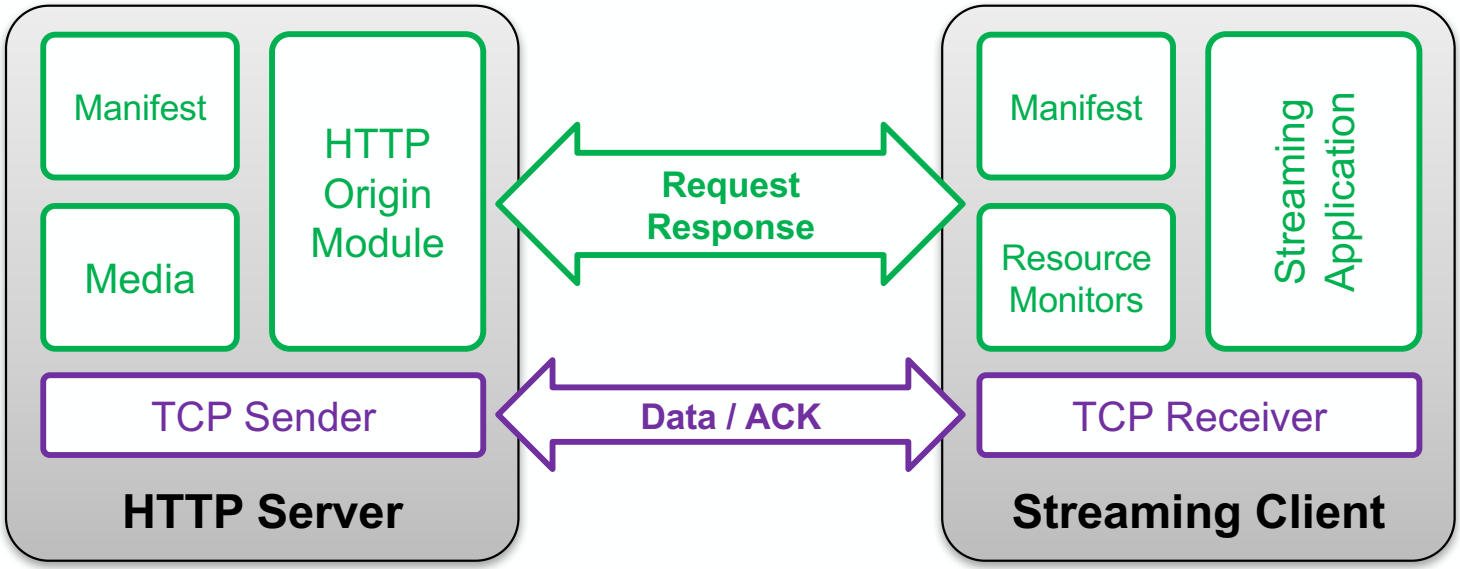






Source: Conviva Viewer Experience Report, 2015

Inner and Outer Control Loops



There could be multiple TCPs destined to the same or different servers

Streaming with Multiple TCP Connections

- Using multiple concurrent TCPs
  - Should not be used to greedily get a larger share of the bandwidth
  - Can help mitigate head-of-line blocking
  - Allows fetching multiple (sub)segments in parallel
  - Allows to quickly abandon a non-working connection without having to slow-start a new one

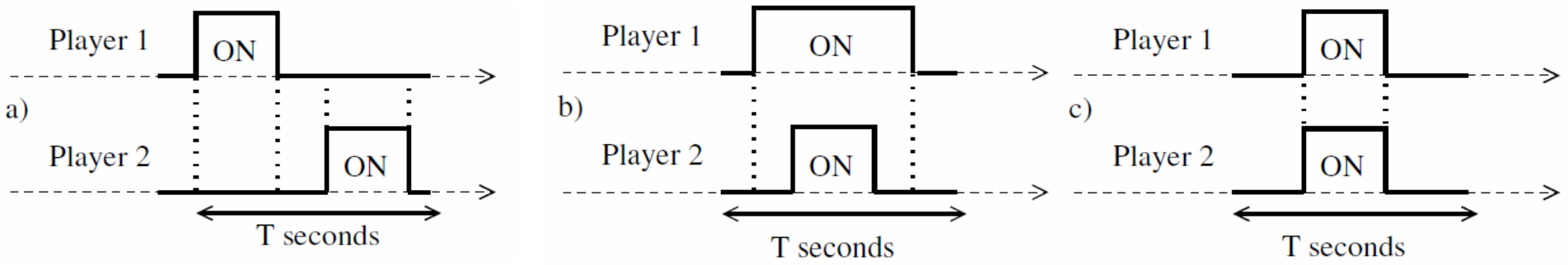
System performance deteriorates very quickly if many clients adopt this approach without limiting the aggregated bandwidth consumption

Understanding the Root Cause

Two Competing Clients

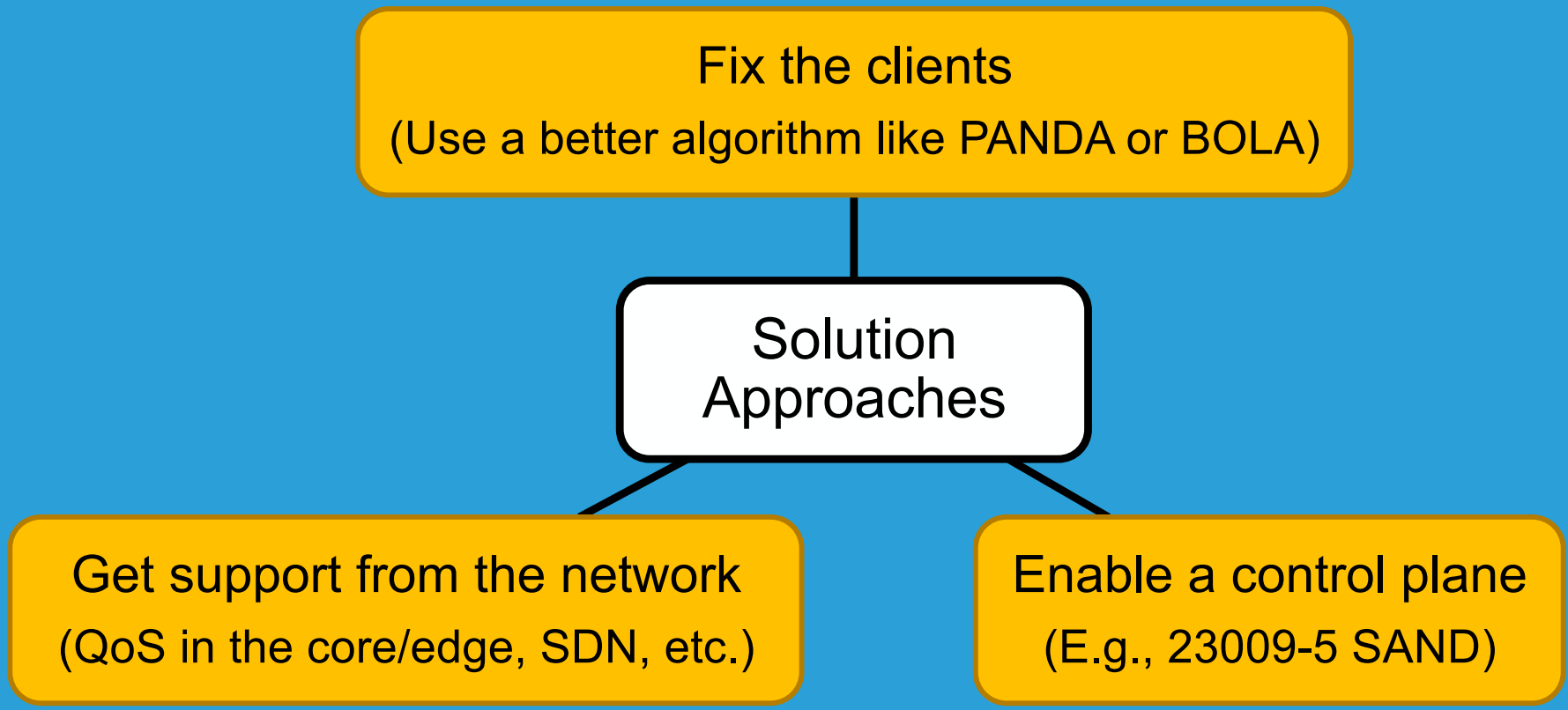
- Depending on the timing of the ON periods:
  - Unfairness, underutilization and/or instability may occur
  - Clients may grossly overestimate their fair share of the available bandwidth

Clients cannot figure out how much bandwidth to use until they use too much



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

How to Solve the Issues?

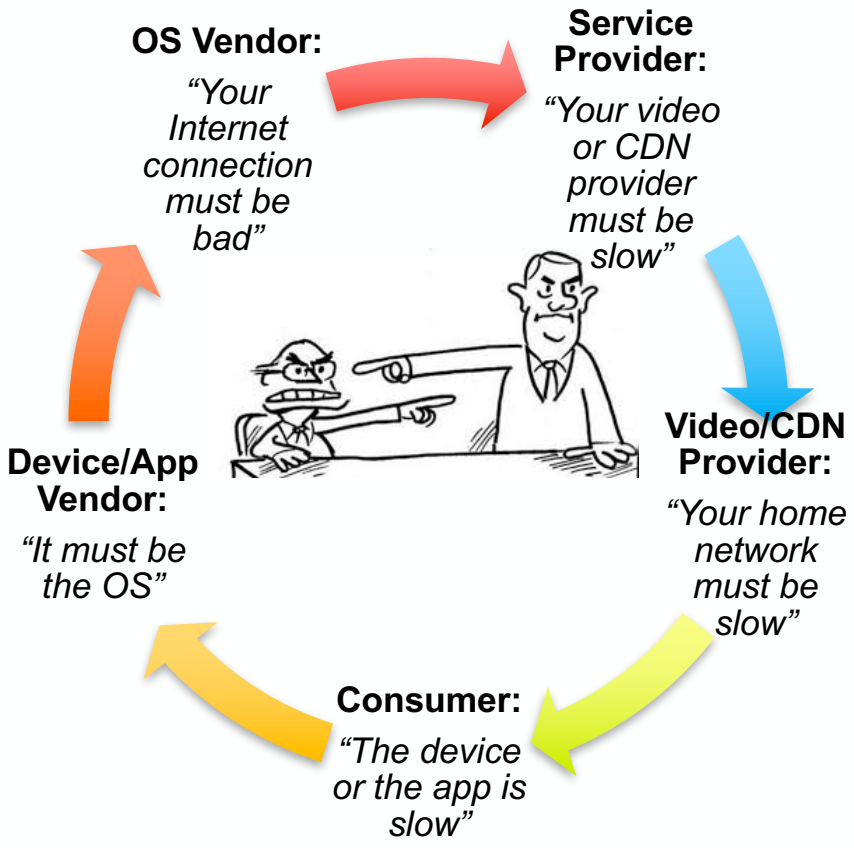


Commercial OTT Issues

- Content format issues
  - Each asset is copied to multiple media formats
    - different video codecs
    - different audio codecs
    - different (regional) frame rates
  - Cost to content creators and distributors
  - Inefficiencies in CDNs and higher storage costs
- Platform issues
  - Lack of consistent app behavior across platforms
  - Varying video features, APIs and semantics across platforms

- **Playback issues**
  - **Codec incompatibility**
  - **Partial profile support**
  - **Switching bitrate glitches**
  - **Audio discontinuities**
  - **Ad splicing problems**
  - **Long-term playback instability**
  - **Request protocol deficiencies**
  - **Memory problems, CPU weaknesses**
  - **Scaling (display) issues**
  - **Variable HDR support**
  - **Unknown capabilities**

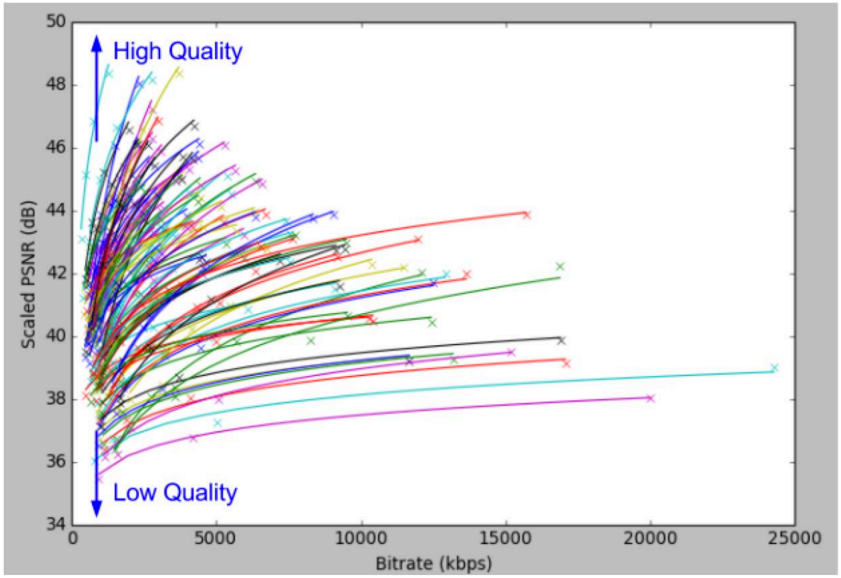
That is Why We Need Analytics



Netflix's Content-Based Encoding Method

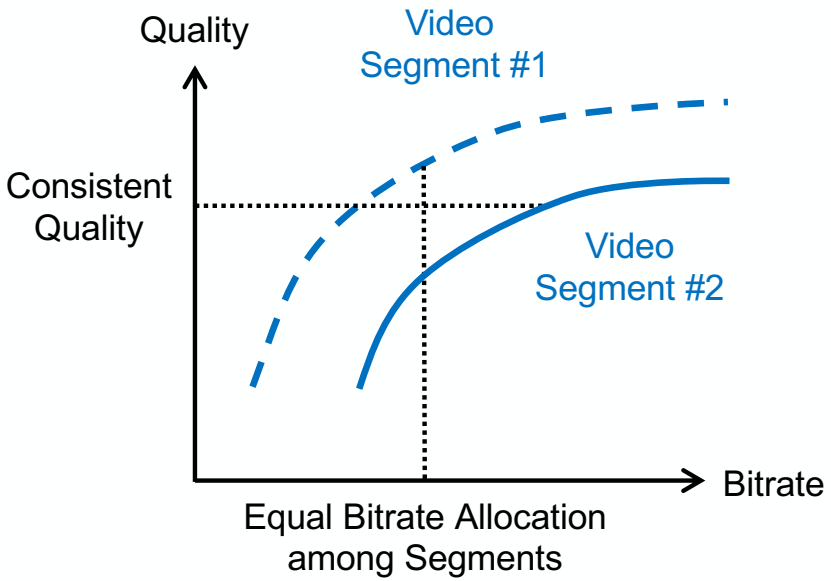
Increased bandwidth brings new opportunities

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



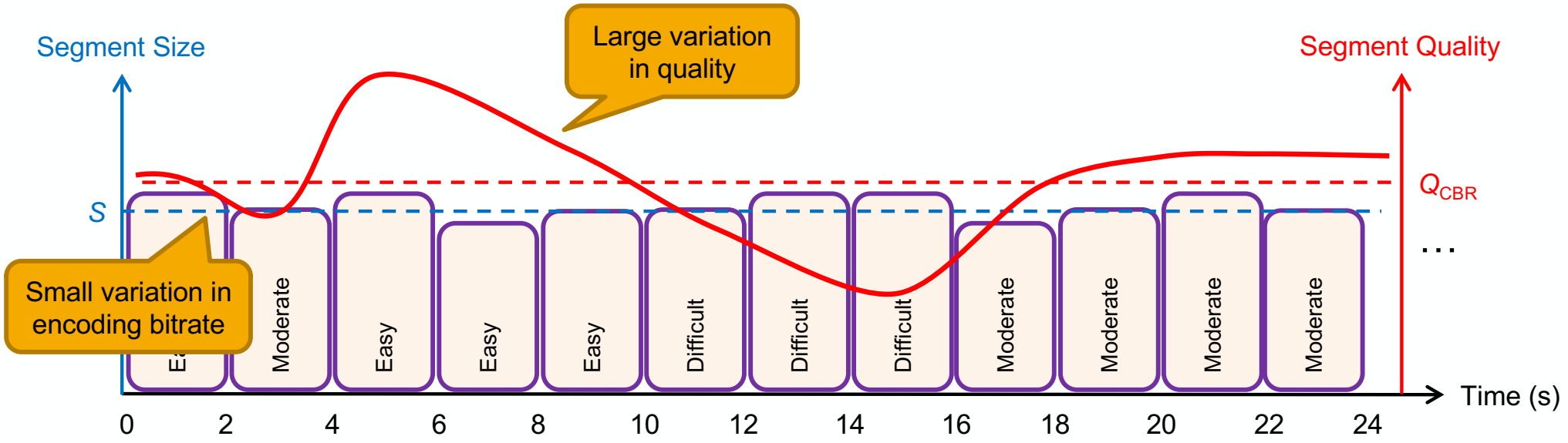
Source: Netflix Tech Blog, December 2015

Segments Have Different Complexities



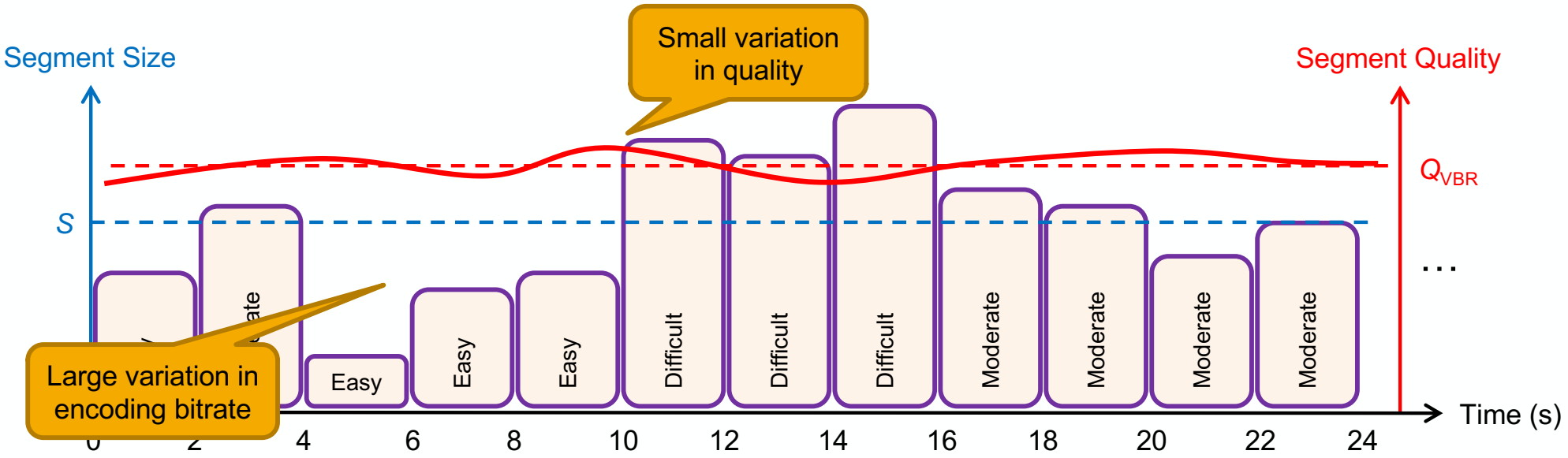
Adaptation Feature Does Not Deliver Consistent Quality

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?

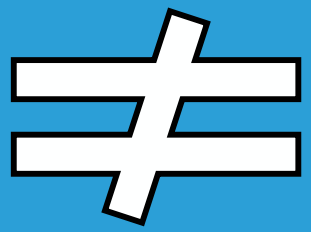


While we spend the same total amount of bits, we not only increase average quality but also reduce quality variation

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

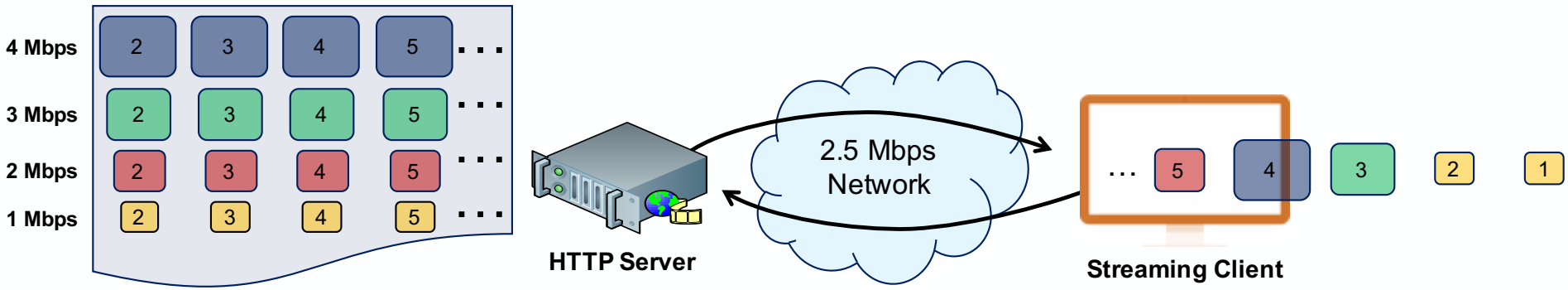
Generating VBR-encoded segments is easy, but streaming them is not!

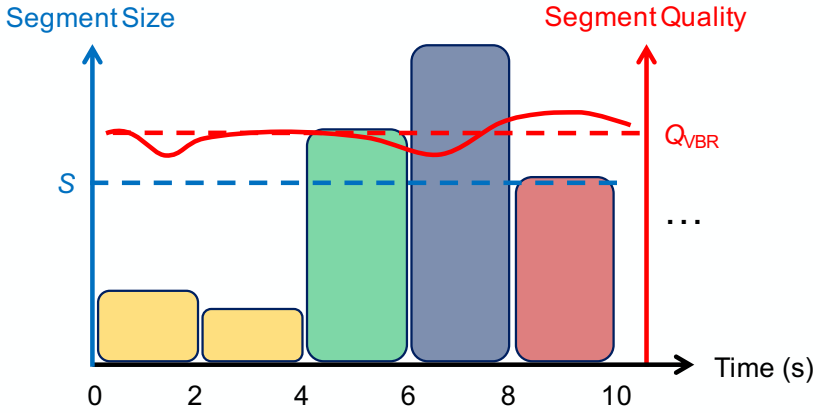
Content Aware Encoding



Content Aware Streaming

Multiple Representations Naturally Enable “Cherry-Picking”

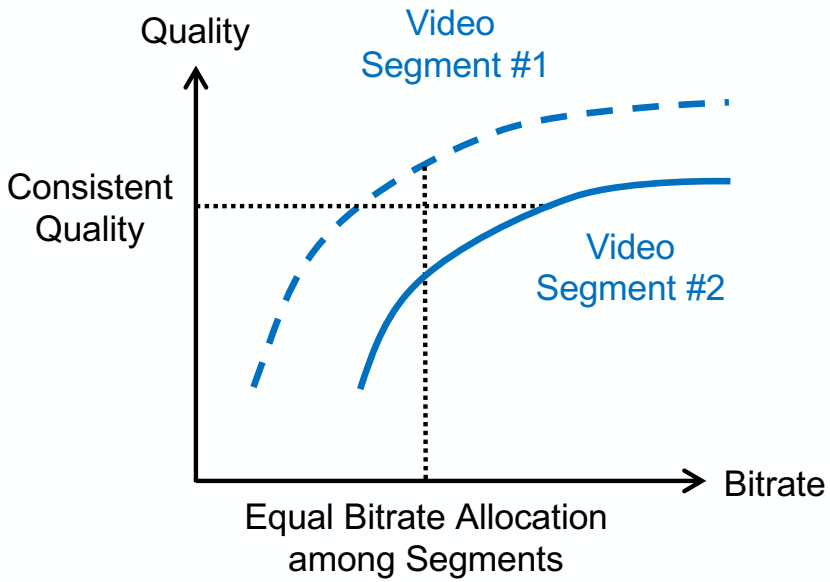


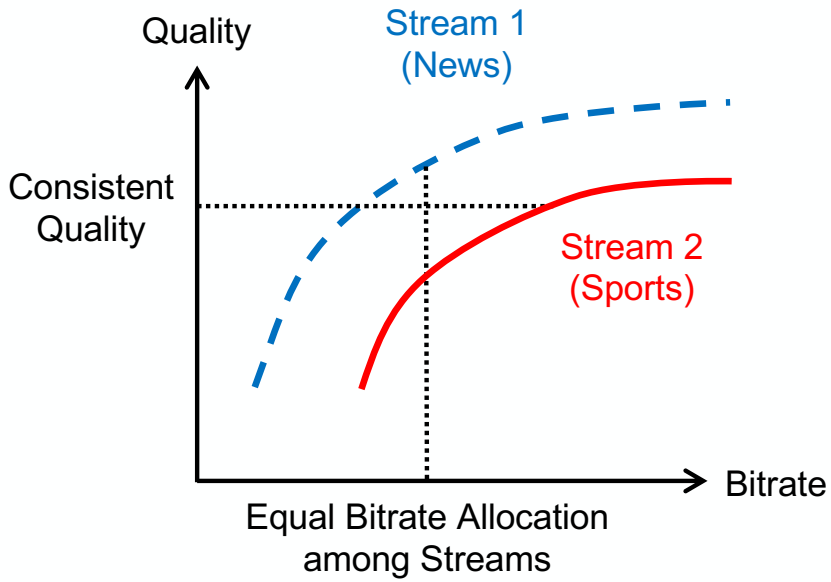


Reading: “Streaming video over HTTP with consistent quality,” ACM MMSys 2014

Dimensions – In-Stream vs. Across-Streams

- Same principle applies to both:
  - In-stream: Temporal bit shifting between segments
  - Across-streams: Bit shifting between streams sharing a bottleneck link





Reading: "Spending quality time with the Web video," IEEE Internet Comput., 2016

ANY QUESTIONS?