SCOM/ SRSI
Video over HTTP

Ana Aguiar

DEEC, FEUP

2018-19

Contents

- Why use HTTP to deliver video?
  - Any issues with this?
- Dynamic Adaptive Streaming over HTTP
- How to choose an appropriate data rate?
- How to choose a CDN cache?

Contents

- Why use HTTP to deliver video?
  - Any issues with this?
- Dynamic Adaptive Streaming over HTTP
- How to choose an appropriate data rate?
- How to choose a CDN cache?

Why stream over HTTP

- Can take advantage of CDN
- HTTP has solved the middlebox problem
- Well-established, simple and cheap
- Can take advantage of every improvement to HTTP

Contents

- Why use HTTP to deliver video?
  - Any issues with this?
- Dynamic Adaptive Streaming over HTTP
- How to choose an appropriate data rate?
- How to choose a CDN cache?

- Monia Ghobadi and Yuchung Cheng and Ankur Jain and Matt Mathis. “**Trickle: Rate Limiting YouTube Video Streaming**”, Presented as part of the 2012 {USENIX} Annual Technical Conference ({USENIX} {ATC} 12)
- https://www.usenix.org/conference/atc12/technical-sessions/presentation/ghobadi

Contents

- Why use HTTP to deliver video?
- Dynamic Adaptive Streaming over HTTP
- How to choose an appropriate data rate?
- How to choose a CDN cache?

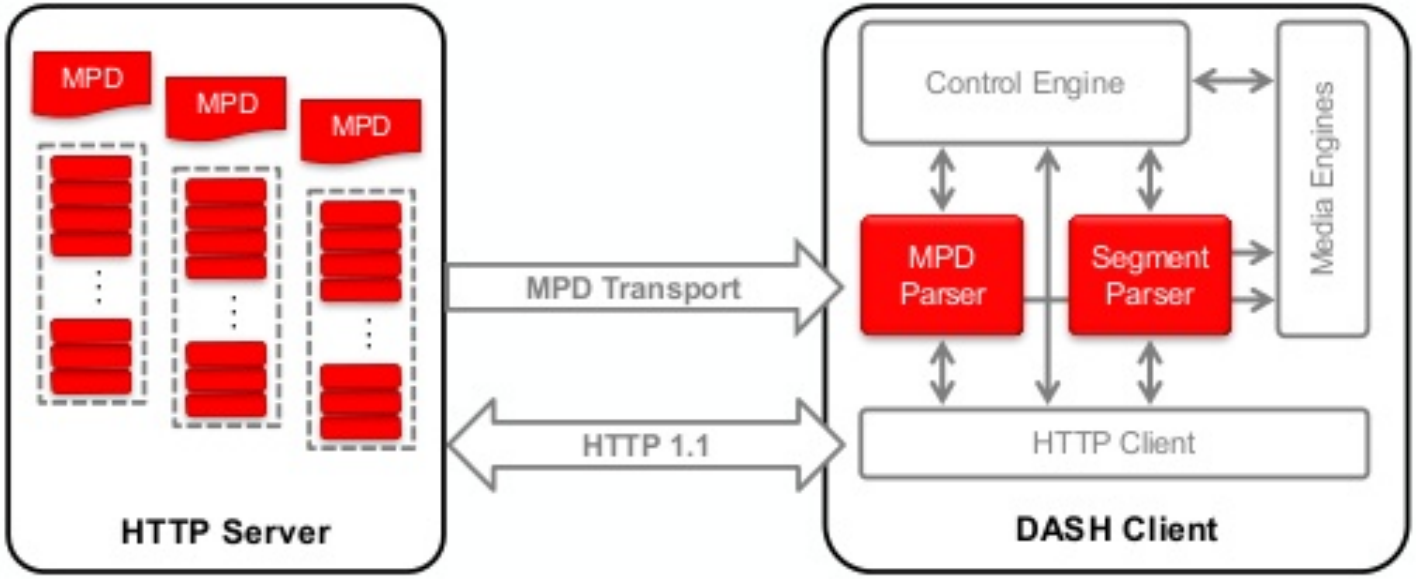
Why DASH?

- Enable
  - Very high user experience
  - Deployment on top of HTTP and CDN
  - Adaptation based on network conditions, device and user preferences
  - Seamless switching
  - Client differentiation
  - Technology re-use
  - Support multiple types of streaming (live, on-demand, …)
  - …
- See presentation for details of MPEG-DASH

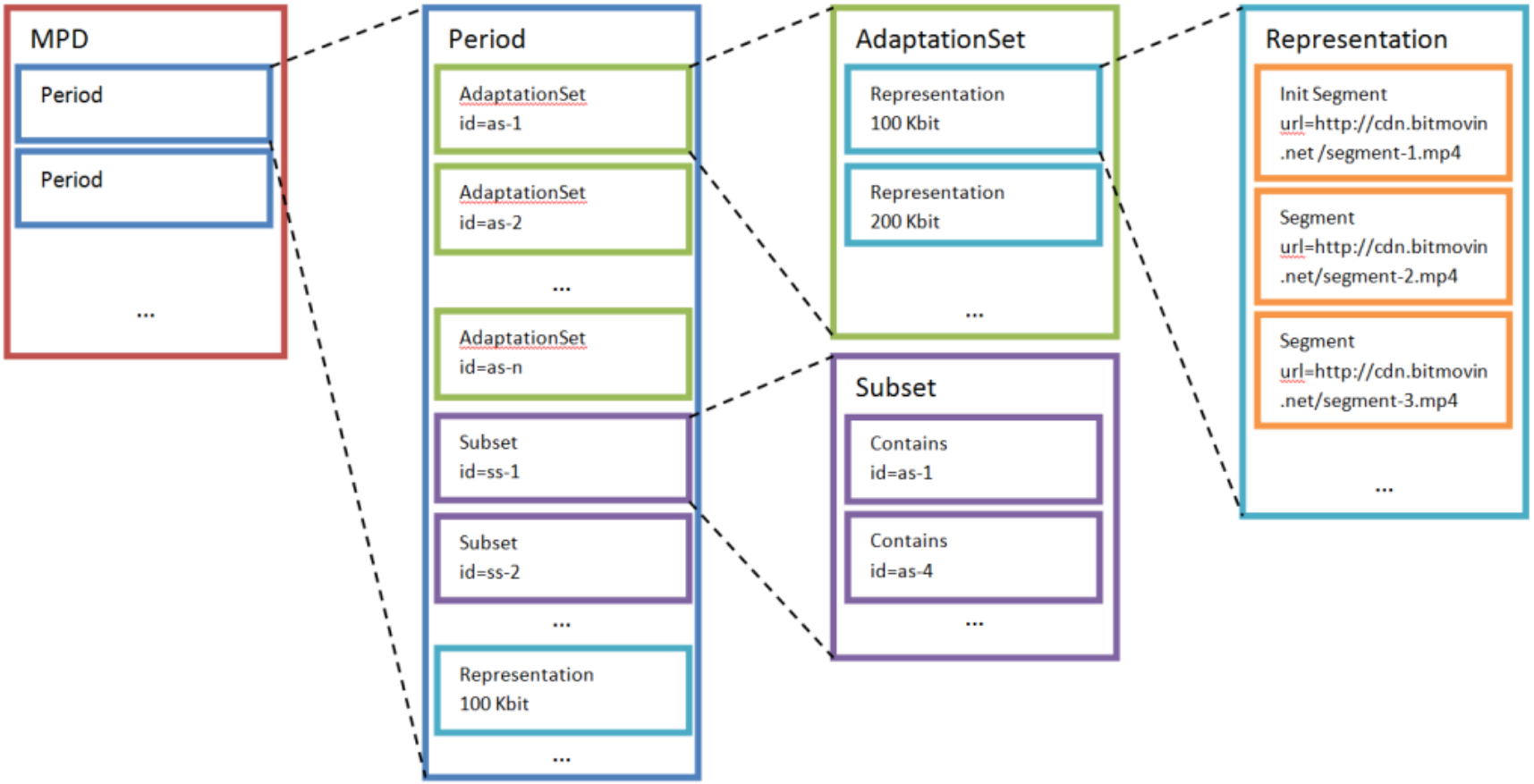
Dynamic Adaptive Streaming over HTTP

- Framework to enable client side adaptation
- Media Description Profile
  - Redundant information of media streams
    - E.g. codec, language, DRM, resolution, bandwidth
  - Access and timing information
    - URLs and byte range of segments
    - Start time and duration
    - Live service: instructions to start playout
    - …

Scope of MPEG DASH



DASH Data Model



Some Vocabulary

- Period: time sequence
- Adaptation set: set of switchable representations
- Representation: encoded version of media
  - Audio/ video parameters
  - Codec, container
  - Bandwidth
  - URL construction
  - …
- Adaptation subset: enabes creator to restrict combination of adaptation sets

Switching Point Alignment

- Segments can use different representations
- Stream Access Points (SAP) enable seamless switching between representations

Challenges

- Bandwidth estimation
- Scheduling segment requests
- Adaptation logic

Contents

- Why use HTTP to deliver video?
- Dynamic Adaptive Streaming over HTTP
- How to choose an appropriate data rate?
- How to choose a CDN cache?

How to pick the streaming data rate?

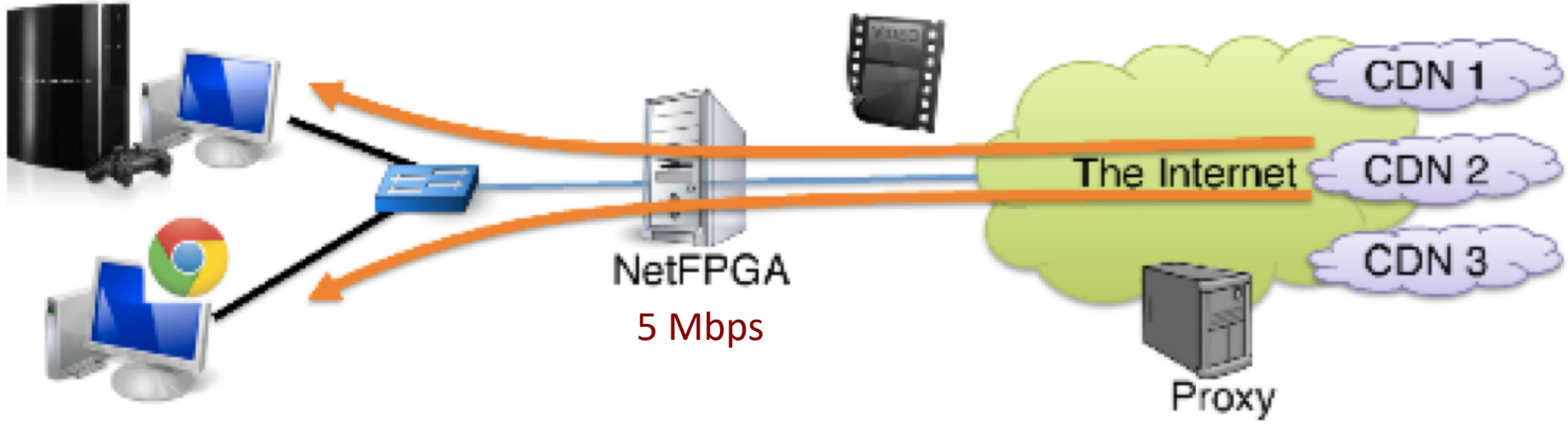
- Depends on estimated available bandwidth
  - Too high: many re-buffering events
  - Too low: lower video quality
- Estimation is done above TCP
- Rate picking is usually conservative
- Rate picking algorithm is proprietary
  - Differs from provider to provider

How do Streaming Services work?

Service	Client	Segment	TCP	Playout buffer	# bitrates
A	browser	4s	persistent	Change request rate	9
B	PS3	8s	Non-persistent	Change TCP rcv window	6
C	PS3	Whole file	Open connection	Change TCP rcv window	7

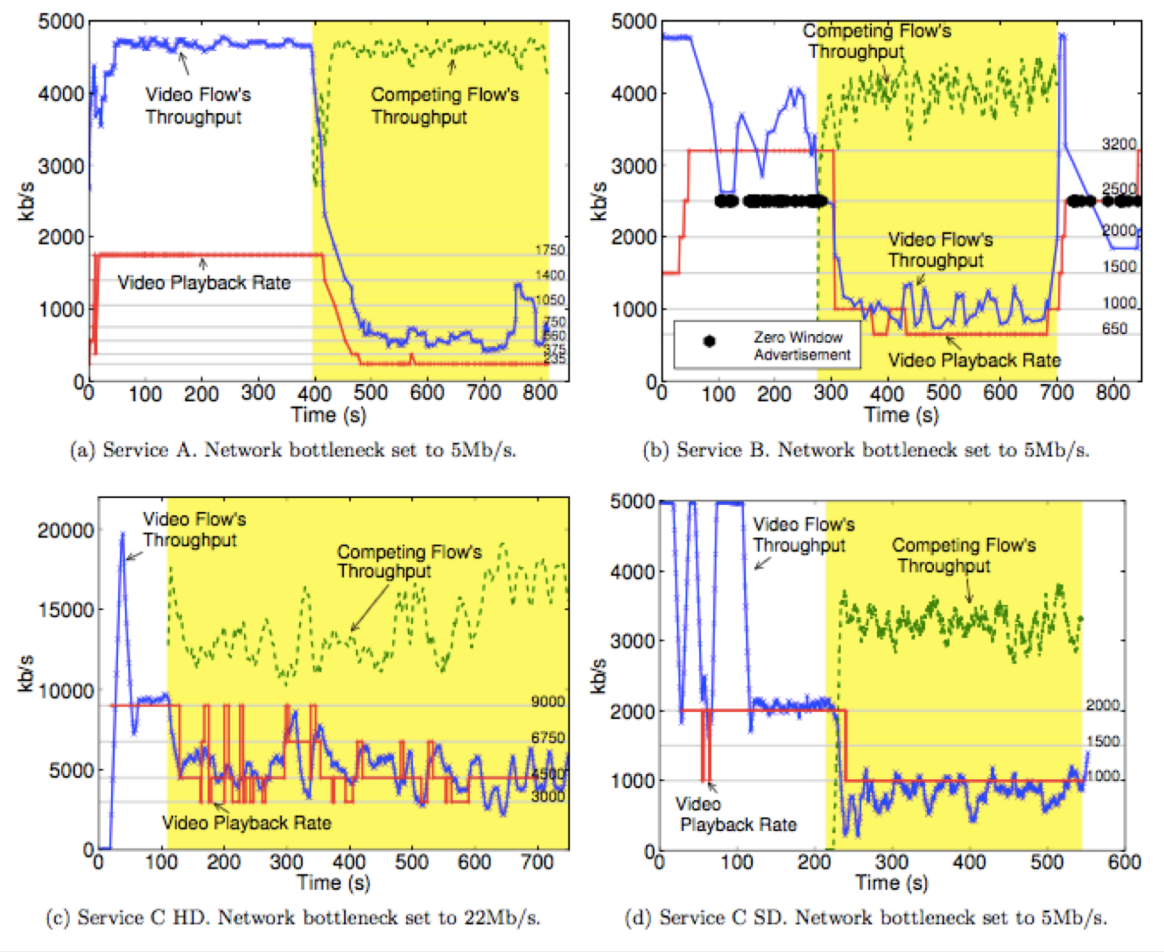
Te-Yuan Huang, Nikhil Handigol, Brandon Heller, Nick McKeown, and Ramesh Johari. 2012. Confused, timid, and unstable: picking a video streaming rate is hard. In *Proceedings of the 2012 Internet Measurement Conference* (IMC '12). ACM, New York, NY, USA, 225-238. DOI: https://doi.org/10.1145/2398776.2398800

Experimental Setting



- NetFPGA creates a controllable bottleneck

Video Behaviour with Competing Flow



(a) Service A. Network bottleneck set to 5Mb/s.

(b) Service B. Network bottleneck set to 5Mb/s.

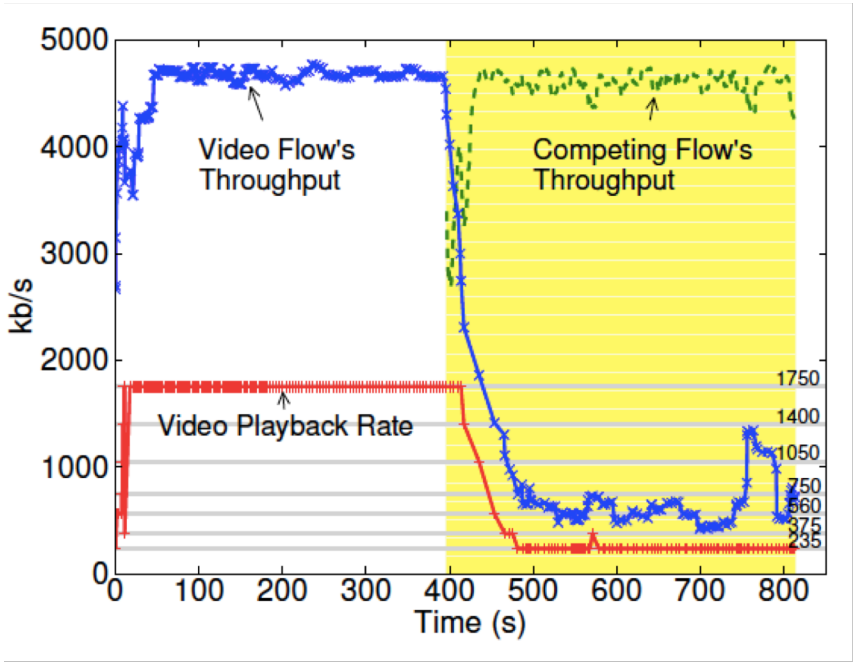
(c) Service C HD. Network bottleneck set to 22Mb/s.

(d) Service C SD. Network bottleneck set to 5Mb/s.

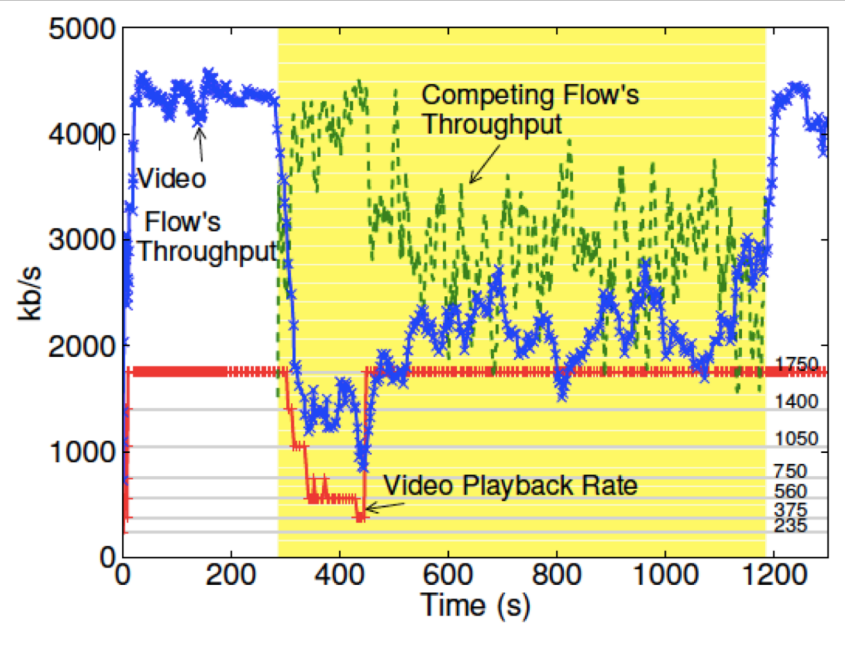
Bandwidth should be divided in half among flows, but observations tell a different story

Is the adaptive rate algorithm guilty?

With rate adaptation



Forced manual rate

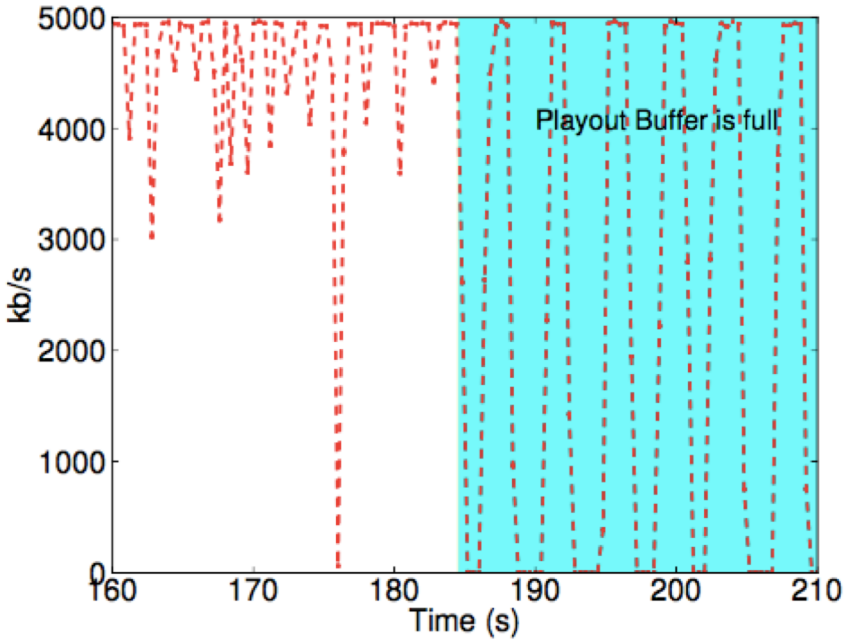


Reverse engineering throughput estimation

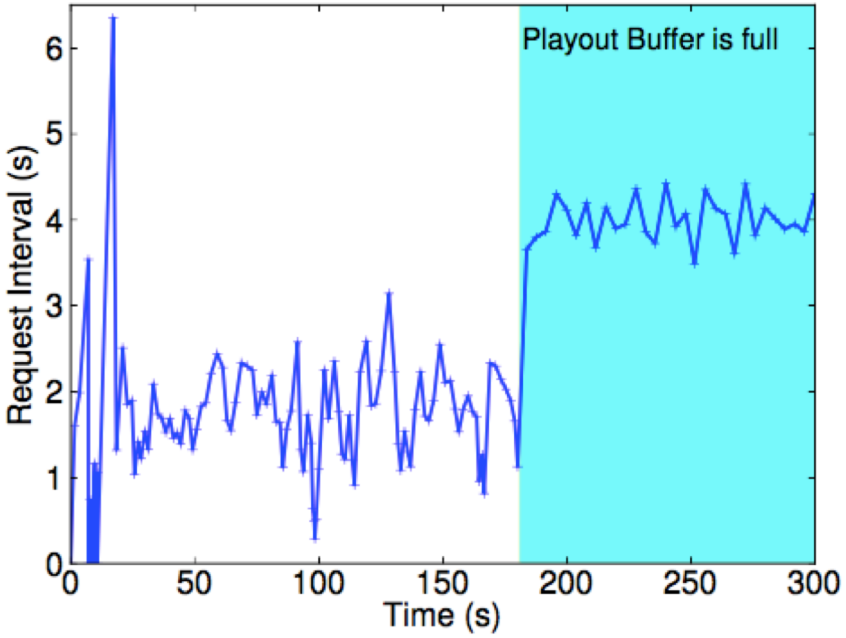
- Perceived throughput ≈ L/T
  - L: video segment size
  - T: time to download
  - Good approximation to chosen rate

- Hypothesis: Throughput measured by HTTP file transfer is not a good estimate of available bandwidth
  - Verify hypothesis. How?
  - Why?
  - Same or different problems for the 3 services?

Observing Service A Before Competing Flow



(a) TCP throughput before and after the buffer fills.



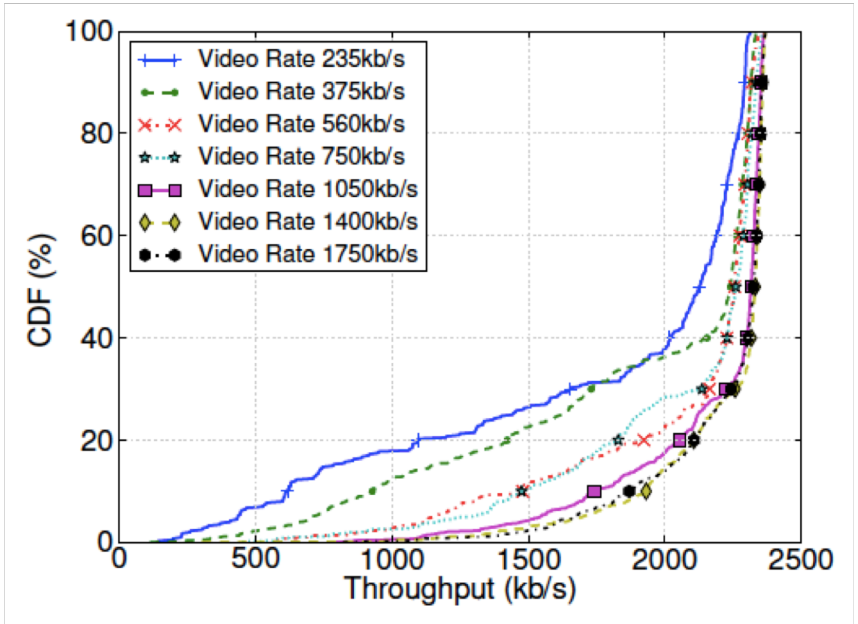
(b) Request interval before and after the buffer fills.

Observing Service A Before Competing Flow

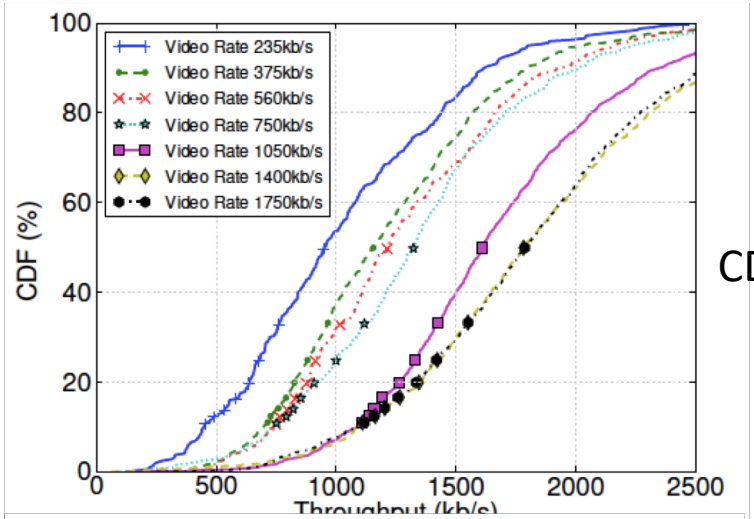
- Playout buffer full -> segments requested every 4s: on-off scheduling
- On-off scheduling -> TCP cwin times out because of no activity
- Each request starts with closed window
- Short requests do not even open the window

Choice of data rates

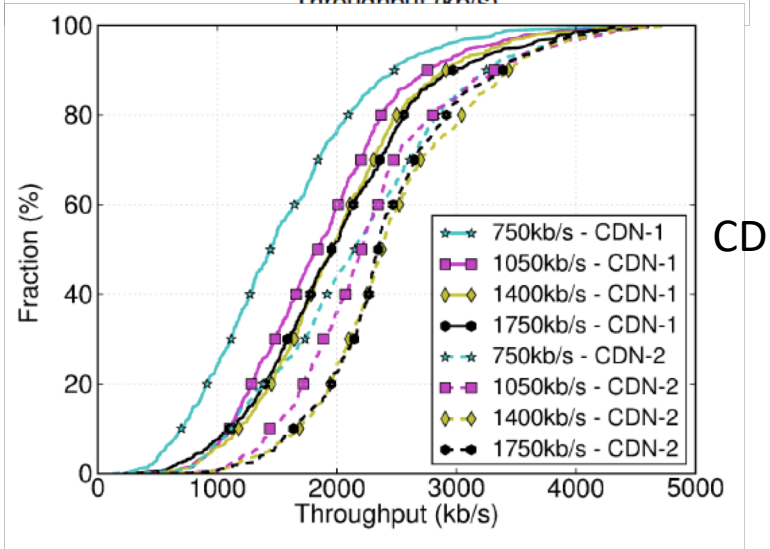
Without competing flow



With competing flow

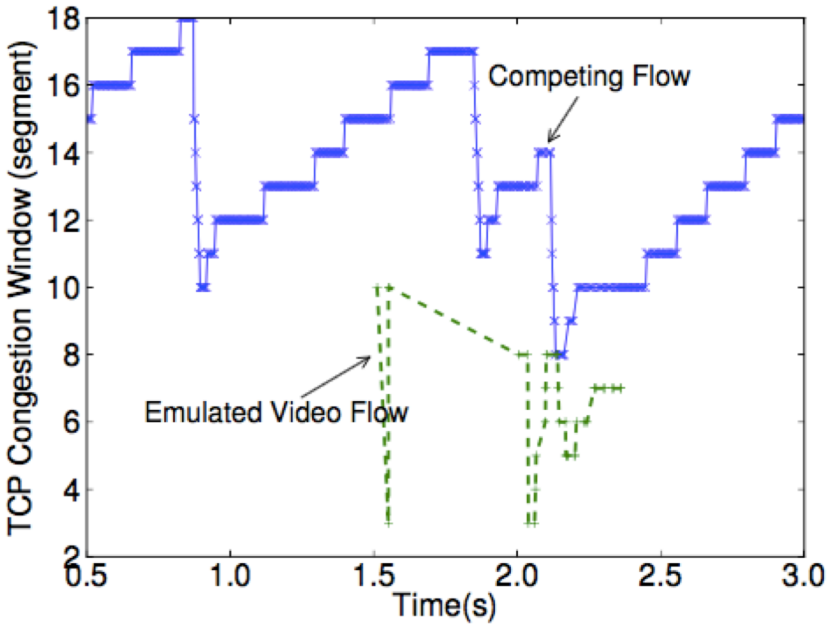


CDN 1

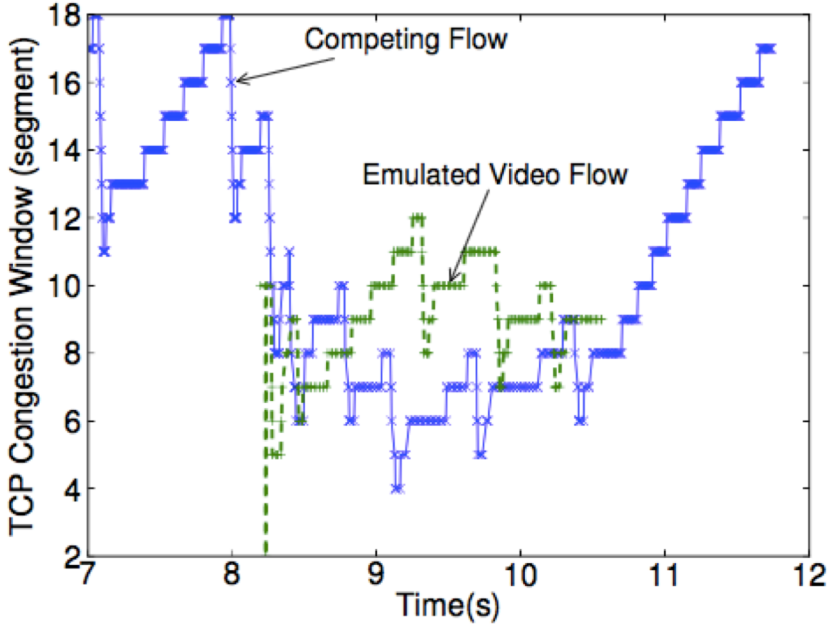


CDN 2

Observing Service A After Competing Flow



(a) A 235kbps Segment.



(b) Five contiguous 235kbps segments concatenated into one.

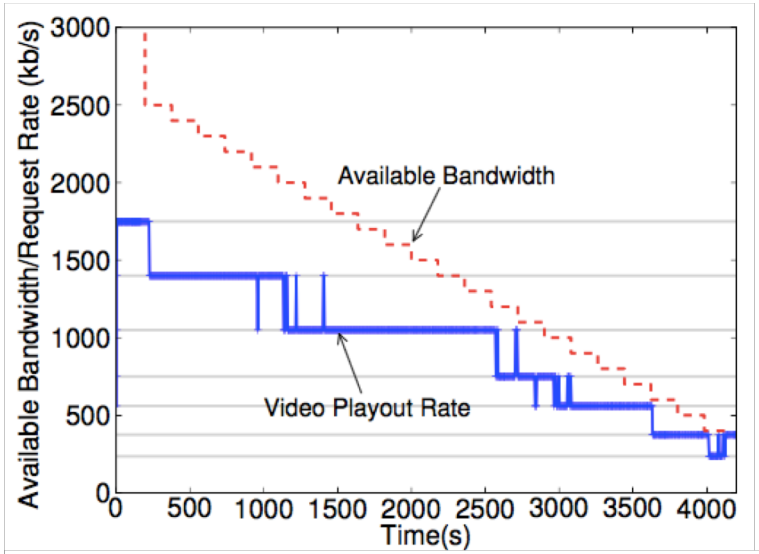
Video flow experiences losses because other flow is occupying full bandwidth and congestion window does not open

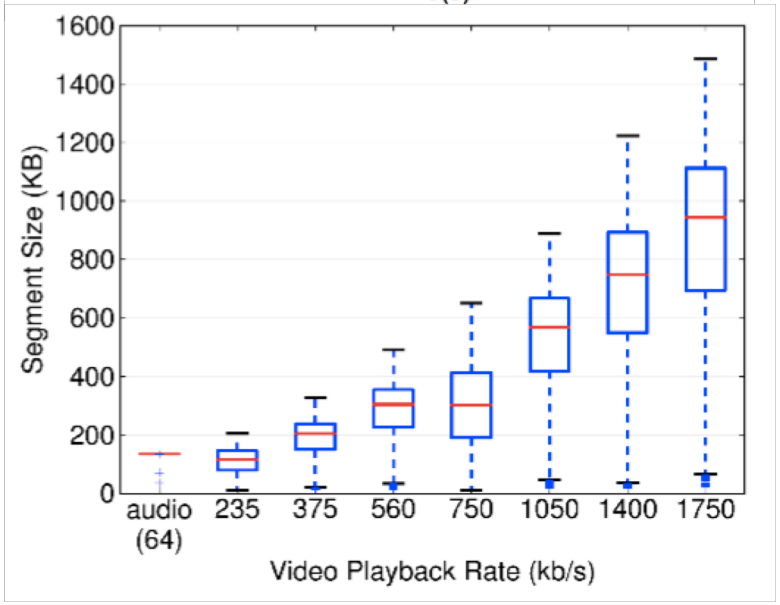
Request for a 5 times larger segment shows that congestion window opens

Throughput is low

Other Important Factors

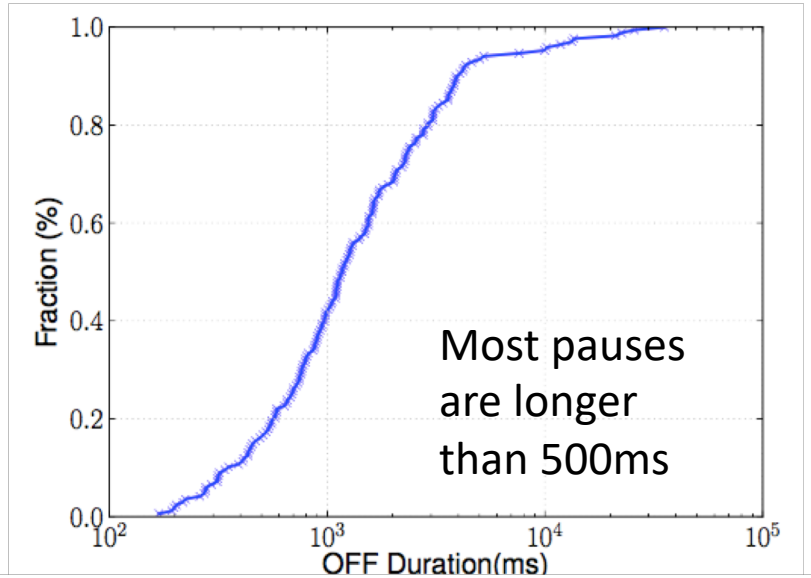
- Client choice of rate
  - Conservative
- Segment size
  - Decreases when rate decreases
  - Leads to even lower throughput at lower rate
- All factors together cause downward spiral

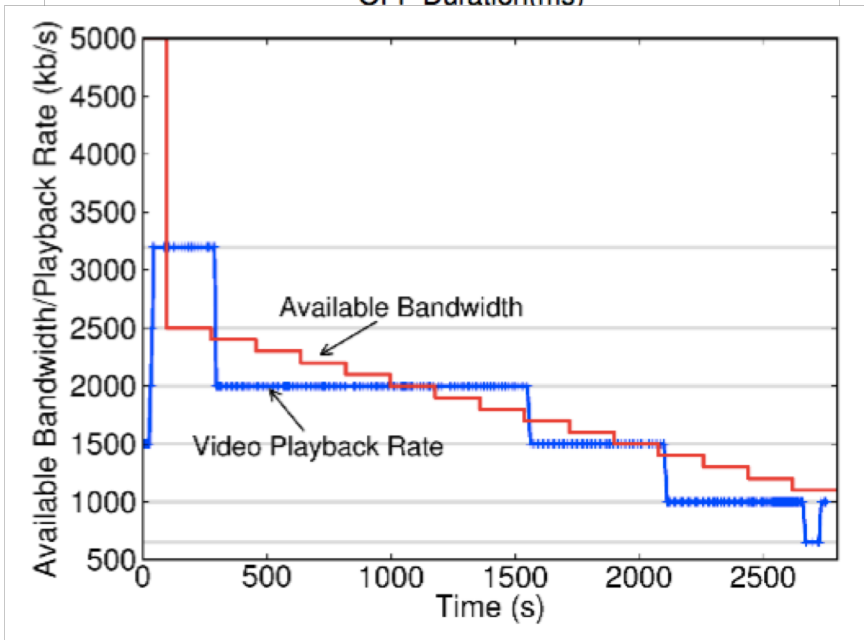




Observing Service B

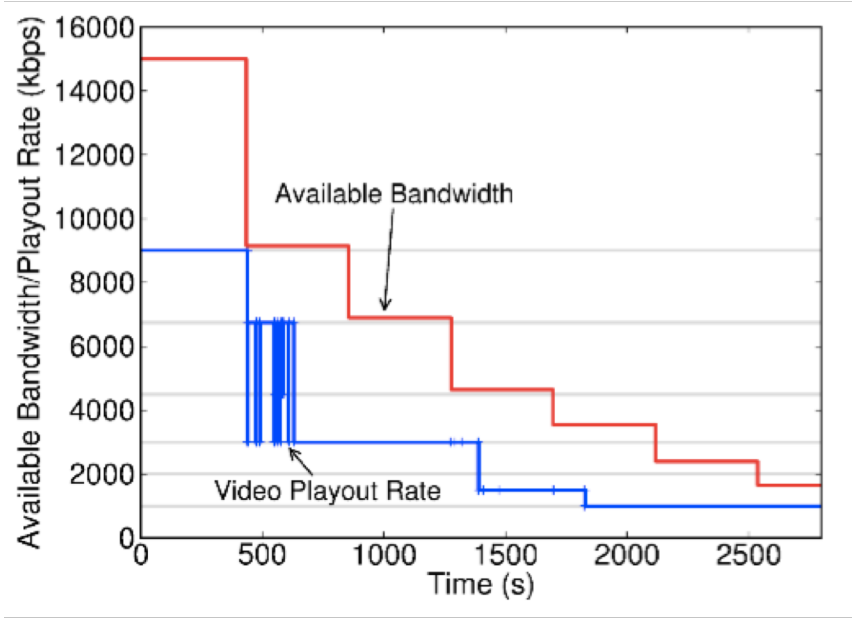
- Playback buffer full -> Stop removing data from TCP buffer
- TCP buffer full -> Zero window advertisements
- On-off behaviour
- Little data (<800kbps) downloaded between off phases
- Low TCP throughput limits chosen video rate





Observing Service C

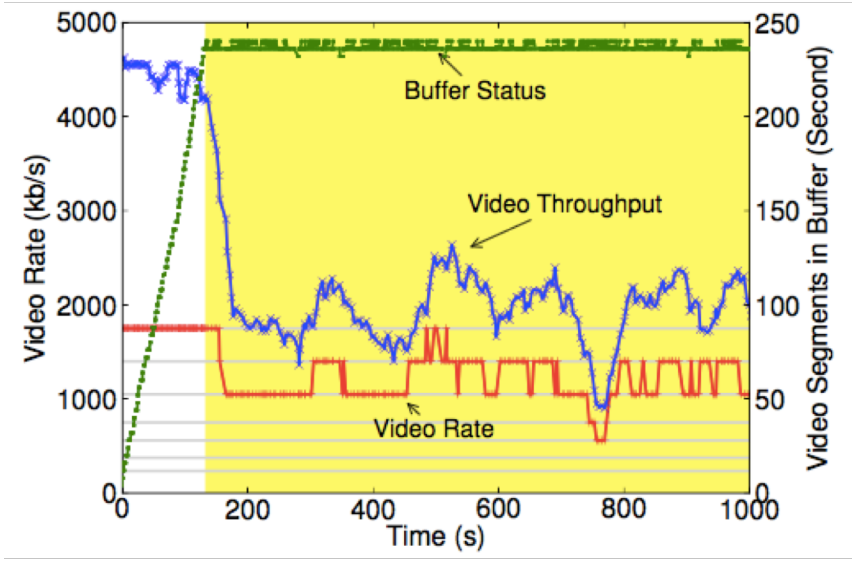
- Whole file download causes new TCP connection on rate change
- TCP connection opening causes low throughput
- Client goes back to lower rates



Reverse Engineering Rate Adaptation Algorithm

- Inferred from measurements
  - Use average of last 10 measured L/T
  - Add conservative factor ~40%
- Effect of changing different parameters
  - Conservative factor
  - Bandwith estimate
  - Bigger segments
    - Improve bandwidth estimates
    - Reduce impact of TCP dynamics

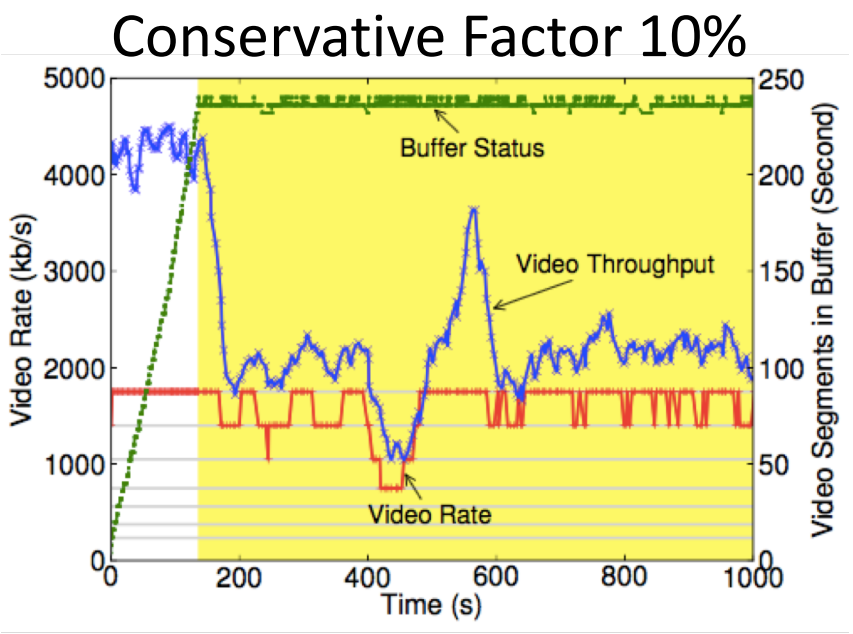
Baseline



Rate Adaptation Algorithm

- Inferred from measurements
  - Use average of last 10 measured L/T
  - Add conservative factor ~40%

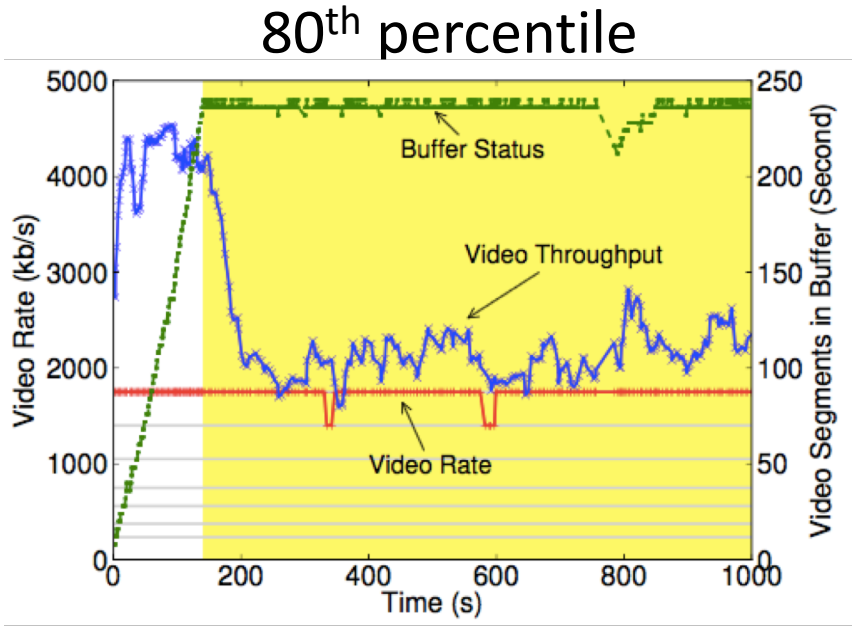
- Effect of changing different parameters
  - Conservative factor = 10%
  - Bandwith estimate
  - Bigger segments
    - Improve bandwidth estimates
    - Reduce impact of TCP dynamics



Rate Adaptation Algorithm

- Inferred from measurements
  - Use average of last 10 measured L/T
  - Add conservative factor ~40%

- Effect of changing different parameters
  - Conservative factor
  - Bandwith estimate using percentiles
  - Bigger segments
    - Improve bandwidth estimates
    - Reduce impact of TCP dynamics

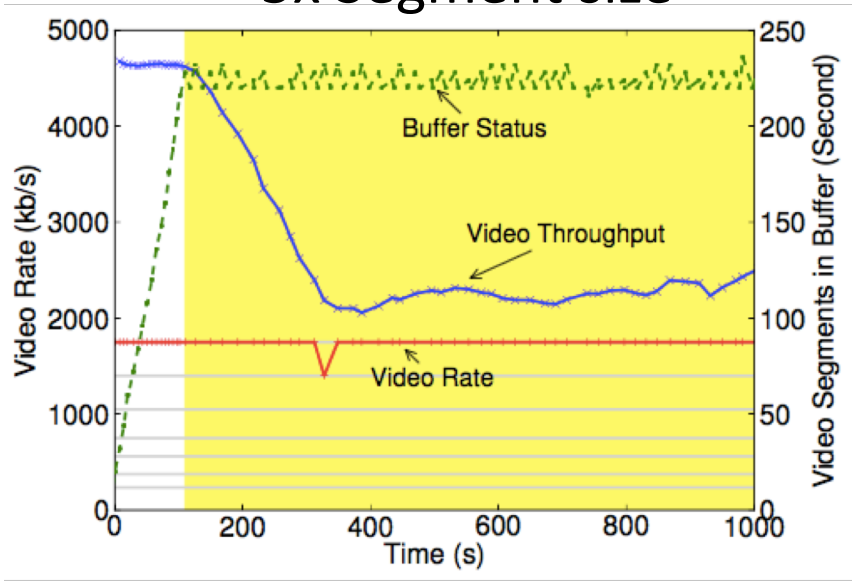


Rate Adaptation Algorithm

- Inferred from measurements
  - Use average of last 10 measured L/T
  - Add conservative factor ~40%

- Effect of changing different parameters
  - Conservative factor
  - Bandwith estimate
  - Bigger segments
    - Improve bandwidth estimates
    - Reduce impact of TCP dynamics

5x Segment size



Contents

- Why use HTTP to deliver video?
- Dynamic Adaptive Streaming over HTTP
- How to choose an appropriate data rate?
- How to choose a CDN cache?

Florian Wamser, Steffen Höfner, Michael Seufert, and Phuoc Tran-Gia. 2017. **Server and Content Selection for MPEG DASH Video Streaming with Client Information.** In Proceedings of the Workshop on QoE-based Analysis and Management of Data Communication Networks (Internet QoE '17). ACM, New York, NY, USA, 19-24. DOI: https://doi.org/10.1145/3098603.3098607

CONGESTION IN BEST EFFORT NETWORKS