

# Computer Networks

## COL 334/672

Application Layer: video streaming

Tarun Mangla

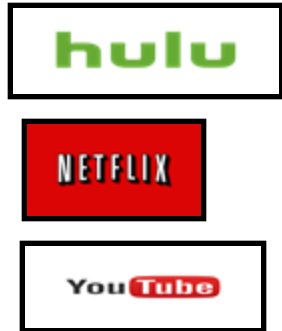
*Slides adapted from KR*

Sem 1, 2024-25

# HTTP Adaptive Streaming (HAS)

Manifest file

HTTP GET manifest



HTTP GET

→ "Seg1@360p"

720p	1	2	3	4
480p	1	2	3	4
360p	1	2	3	4

startup phase

steady-state

(ON-OFF)  
falloff

Time

▪ "intelligence" at client: client determines

- *when* to request chunk (so that buffer starvation, or overflow does not occur)
- *what encoding rate* to request (higher quality when more bandwidth available)

**Bitrate adaptation**

# Designing Bitrate Adaptation Algorithm

- **Design goal:** Maximize application performance
- **Q:** What does application performance depend on in adaptive streaming?
  - Video stalls
  - Video quality
  - Video smoothness

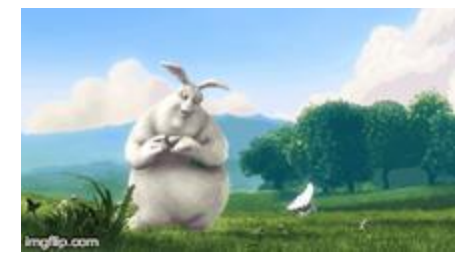
Minimize **stall duration**



Maximize **average bitrate**

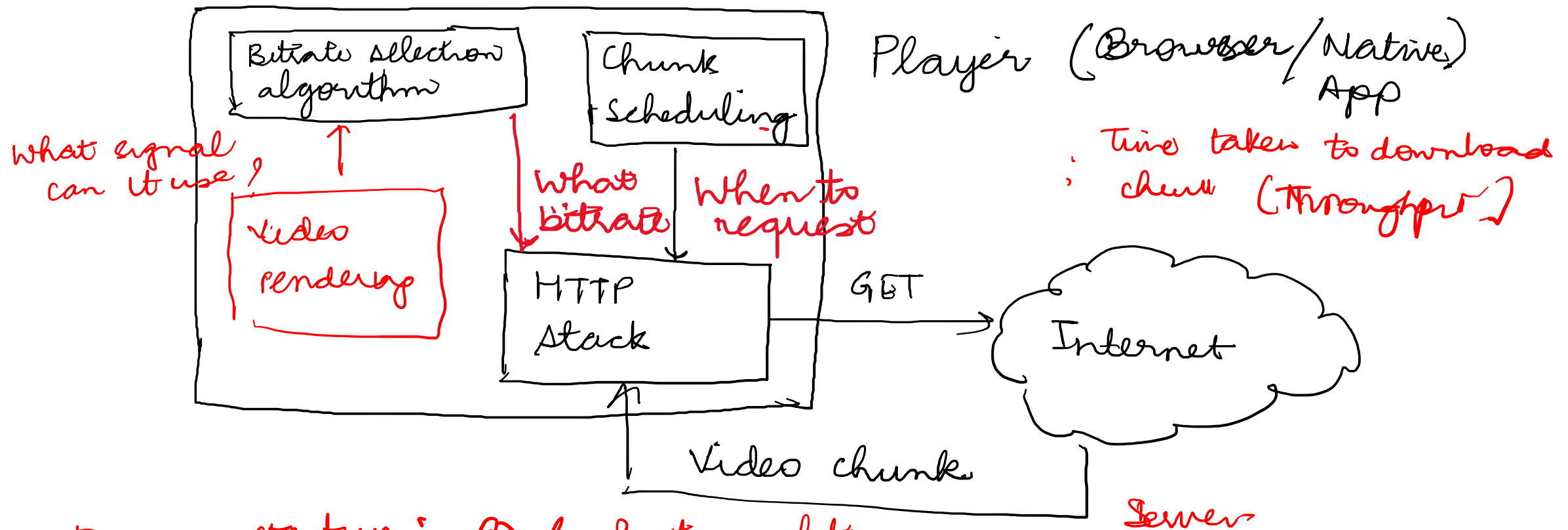


Minimize **bitrate switches**



# Bitrate Adaptation

260p 28 28 → Key frame / Independent frame  
720p 28 28



During startup: ① highest quality  
(Startup latency)

4Mbps 720p  
2Mbps 720p  
1Mbps 720p  
3Mbps

■ Q: What are the signals available to the player for bitrate adaptation?

# Bitrate adaptation: Algo #1

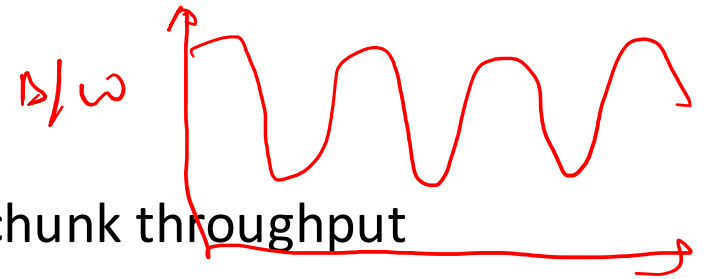
(Rate-based adaptation)

## ■ Idea:

- Estimate network bandwidth based on the past download rate.
- Download chunk at a bitrate just less than the estimated network bandwidth

## ■ Algorithm

1. Estimation: Take into account historical values, not just the last chunk throughput
2. Smoothing: Apply a smoothing filter such as average, harmonic mean or EWMA
3. Quantization: Select bitrate from the discrete set of bitrates based on estimated throughput



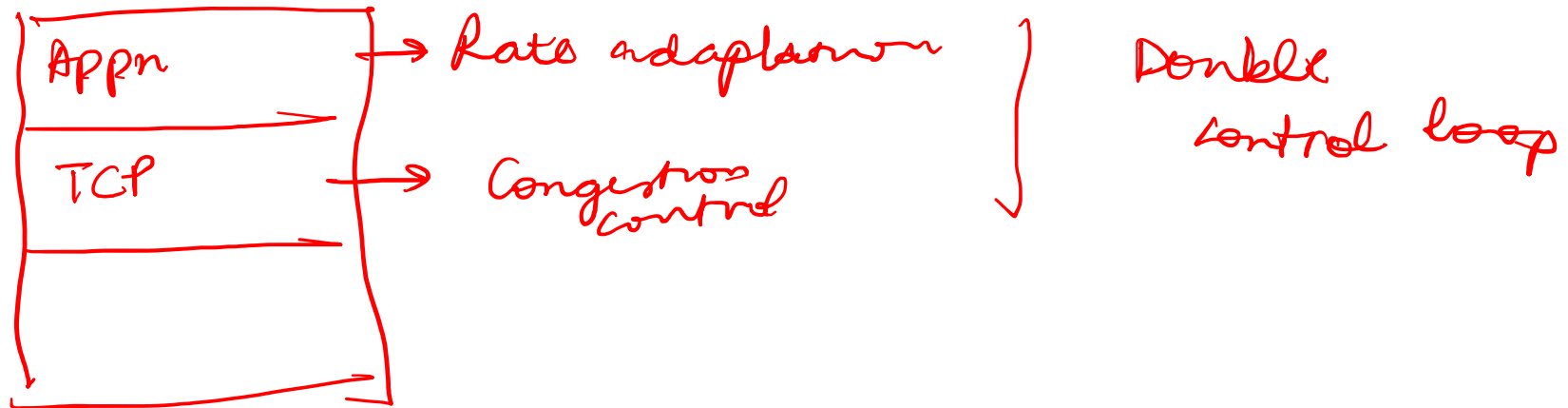
Example  
Available bitrates  
= { 200, 400, 800, ~~1600~~ }  
kbps

$T$   
 $T \approx \frac{\max R_i}{(1+\alpha)^i}$  Throughput  
 $\{ 700, 900, 1100, 1300 \}$   
conservative

# Issue with Rate-based Adaptation

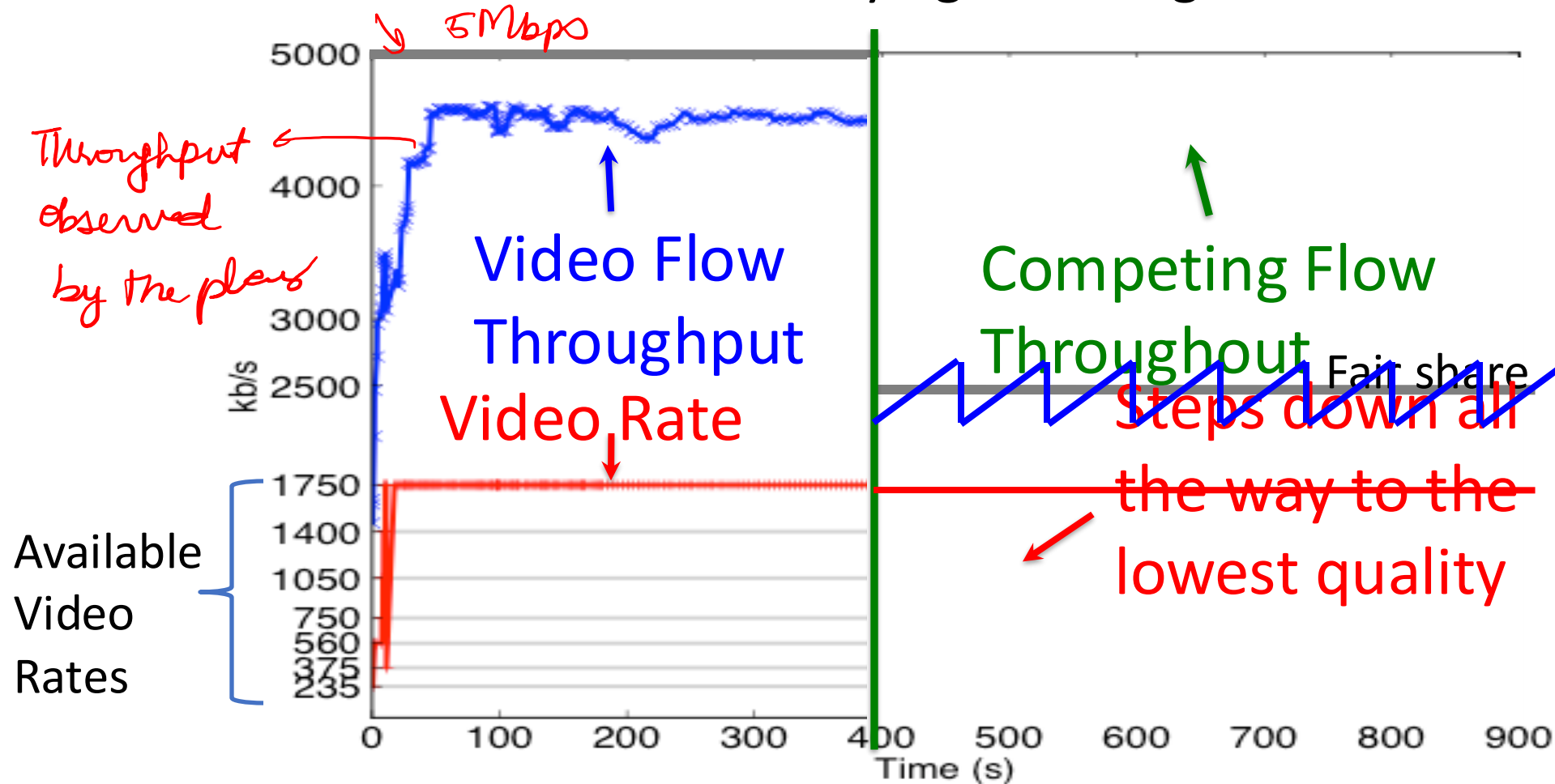
① Fluctuation :

Moving average or smoothing can help

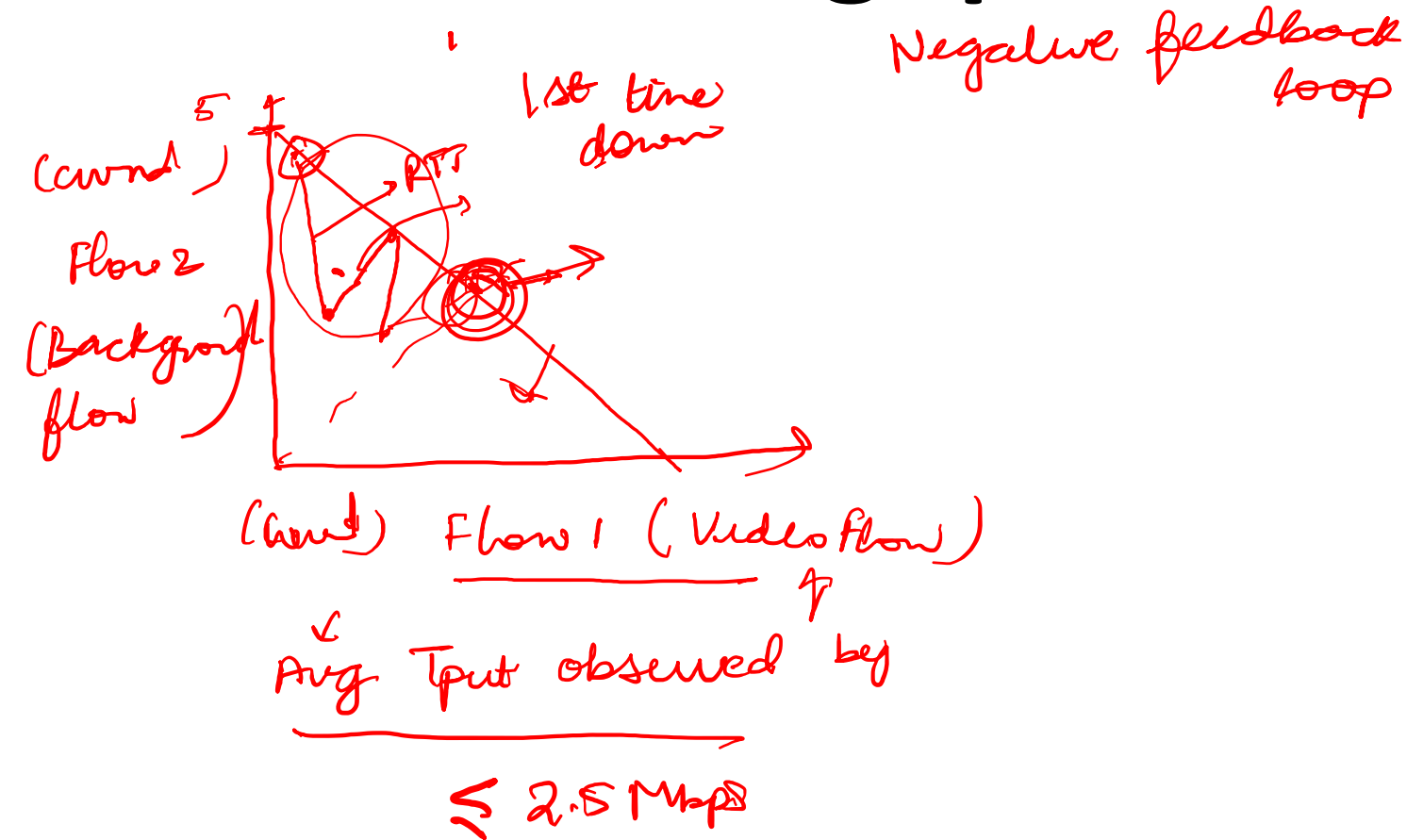


# Issue with Rate-based Adaptation

- Poor interaction with the underlying TCP congestion control



# TCP Throughput of the Video Flow



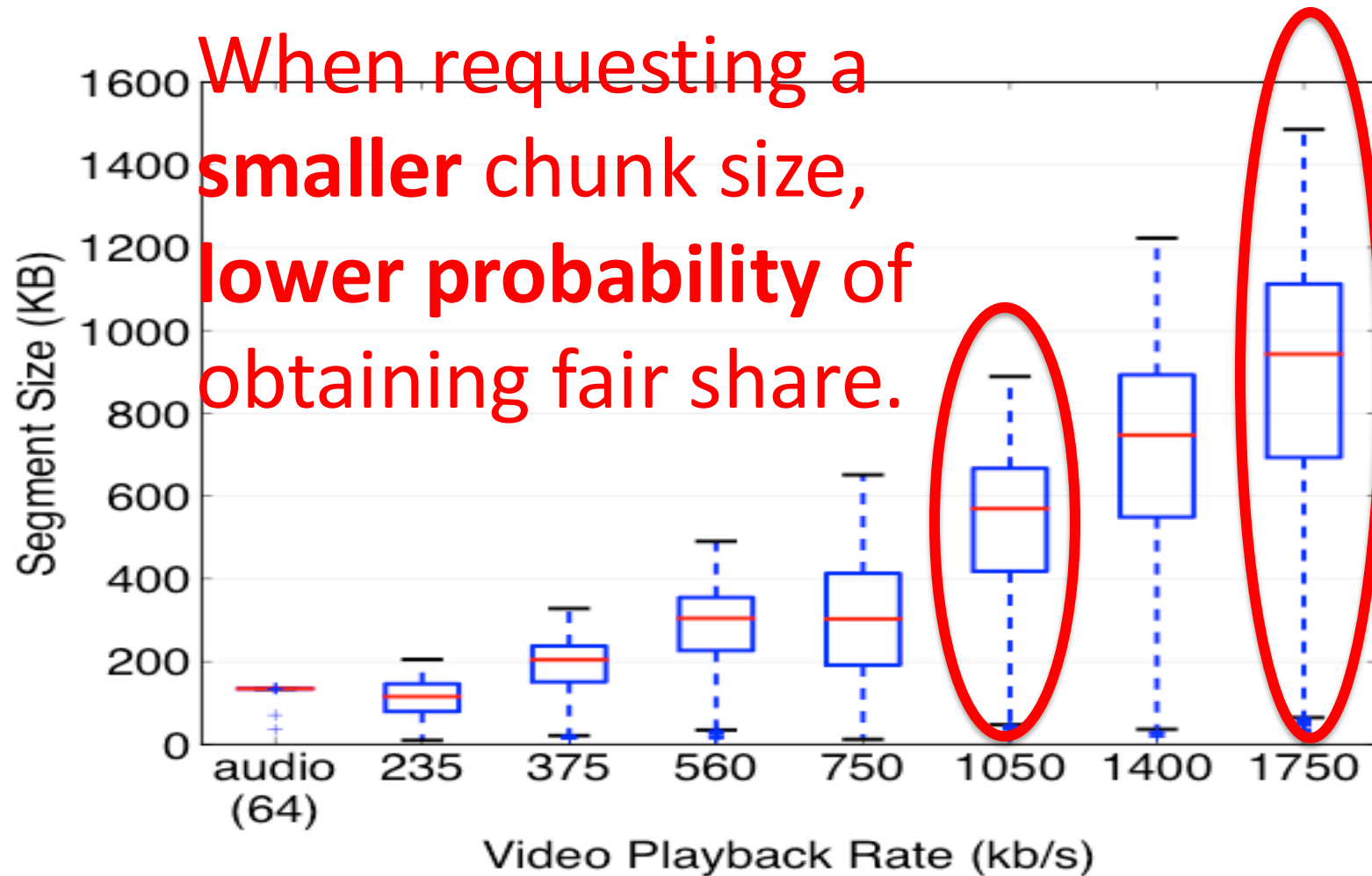
- TCP sender resets its congestion window during OFF period
- Throughput will be affected especially with a competing flow
- Experience packet loss during slow start
- 50% of the segments get < 1.8Mb/s *chunks*



Constant duration  $\rightarrow L$

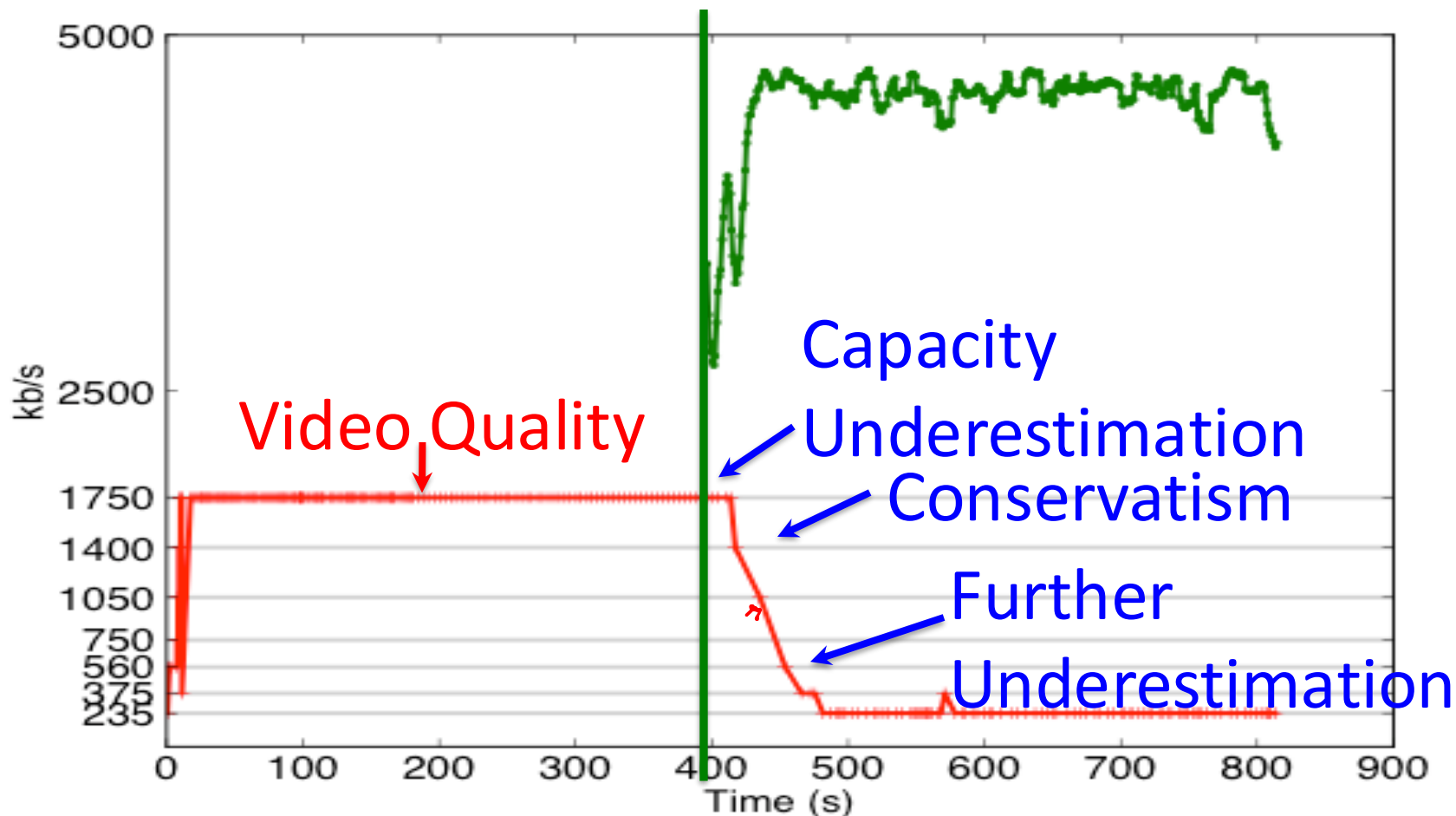
File Size  $\approx L \times R$

# Smaller Chunk Size for Lower Video Rate



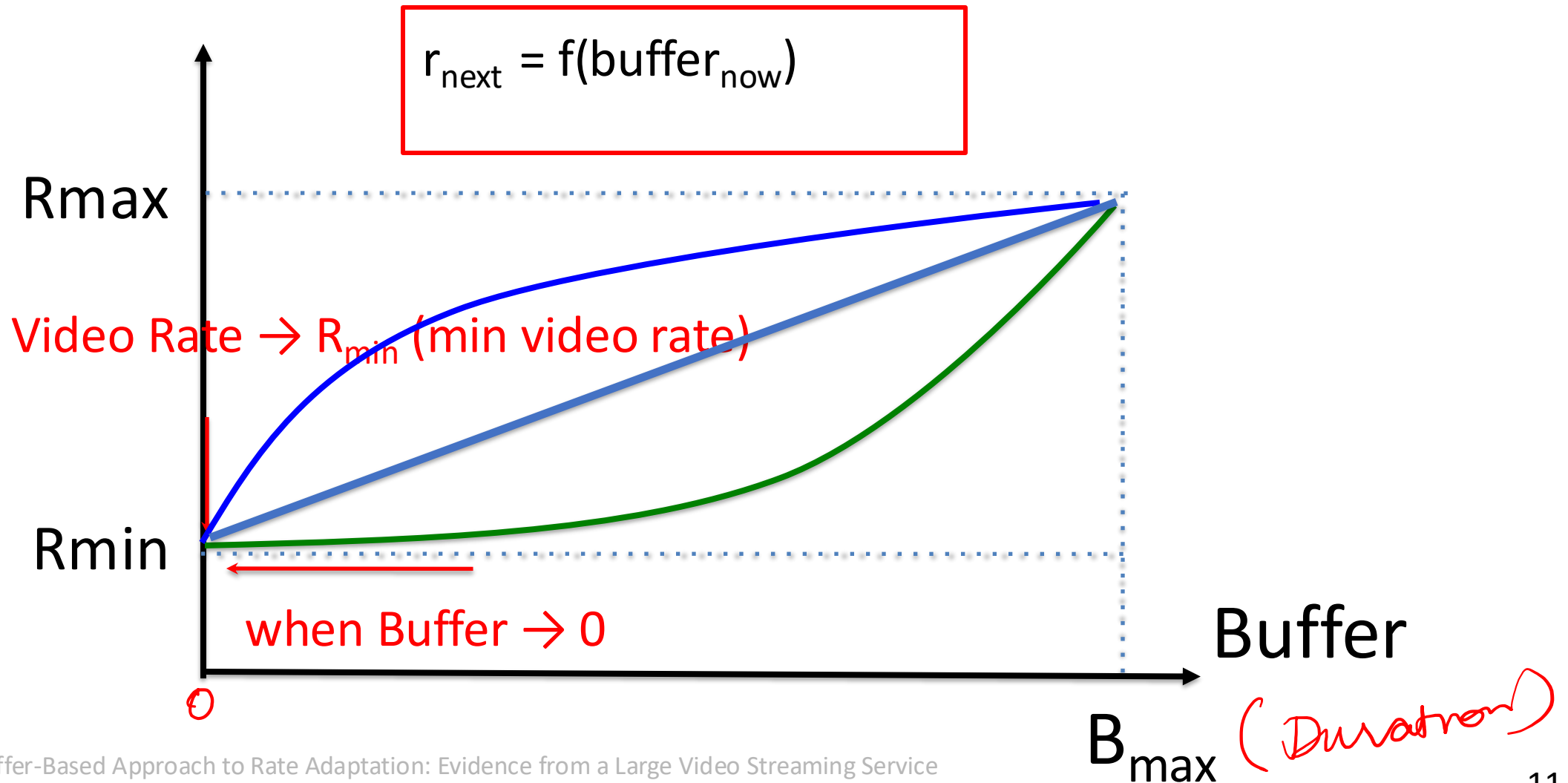
Buffer

# The Complete Story



Being conservative can trigger a vicious cycle!

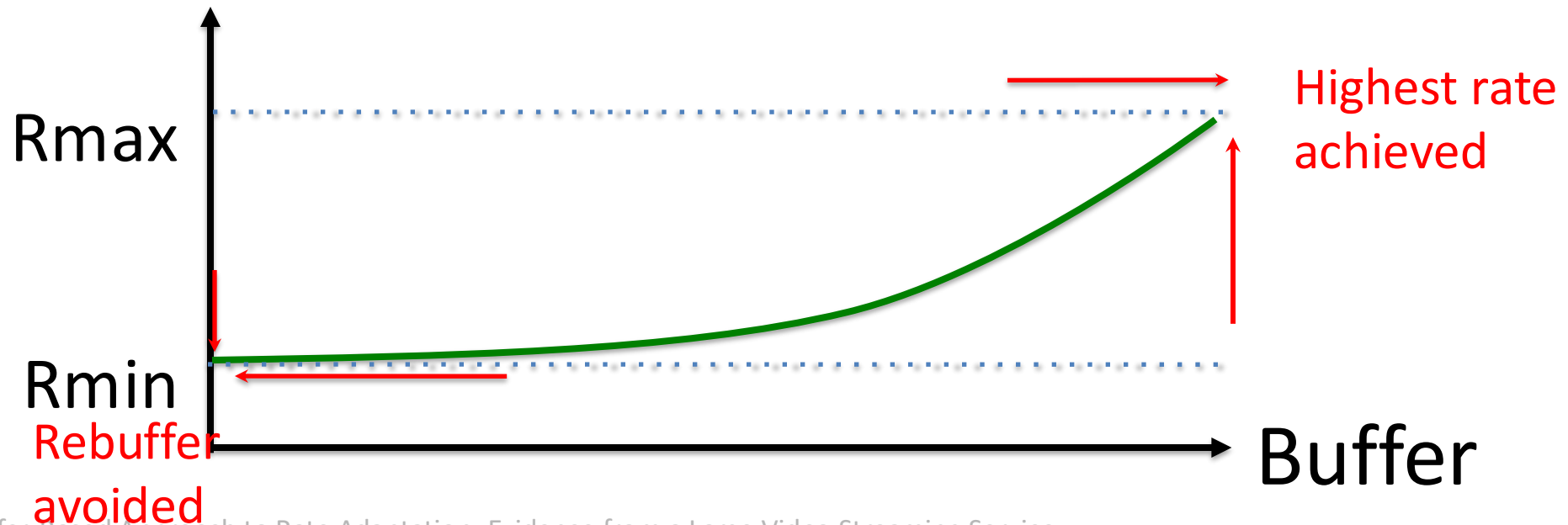
# Buffer-based adaptation: Algorithm Sketch



Source: A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service

# Advantages of buffer-based adaptation

- Utilize the full capacity of the link
  - Avoid on-off behavior as long as the video quality is less than maximum
  - Request the highest video rate before the buffer is full
- Avoid “unnecessary” re-buffering
  - Reduce the bitrate as the buffer occupancy decreases



# Buffer-based adaptation: Algorithm

① Betragskurve

Live stream : Buffer size ( $\leq 10s$ )



# Summary

- HTTP-based adaptive streaming (HAS) used for delivering Internet video
- Bitrate adaptation is important to ensure a high Quality of Experience (QoE)
- Various bitrate adaptation algorithms have been proposed
  - Rate-based: Rely on past observed throughput
  - Buffer-based: Rely on current buffer occupancy
- • Other methods: Control theory approach, machine learning
- Open problems: Bitrate adaptation, encoding, storage, server selection ...