Computer Networks
COL 334/672

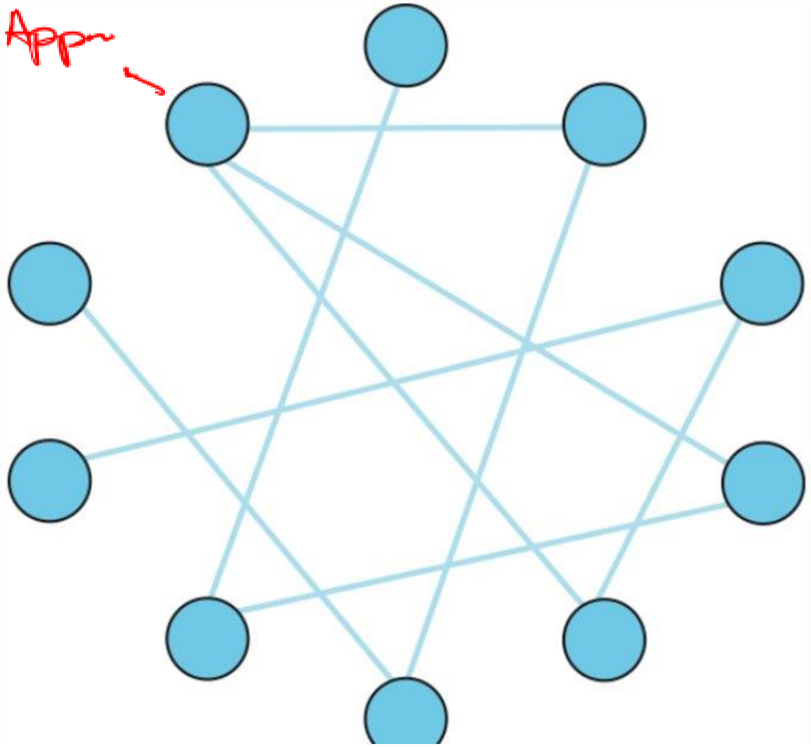
Application Layer: P2P and Video Streaming

Slides adapted from KR

Sem 1, 2025-26

Recap

- Peer to peer for content distribution
- Why: self-scalable network
- Two interesting questions:
  - How to find content? [DHT: Distributed Hash Table]
  - How to download the content?

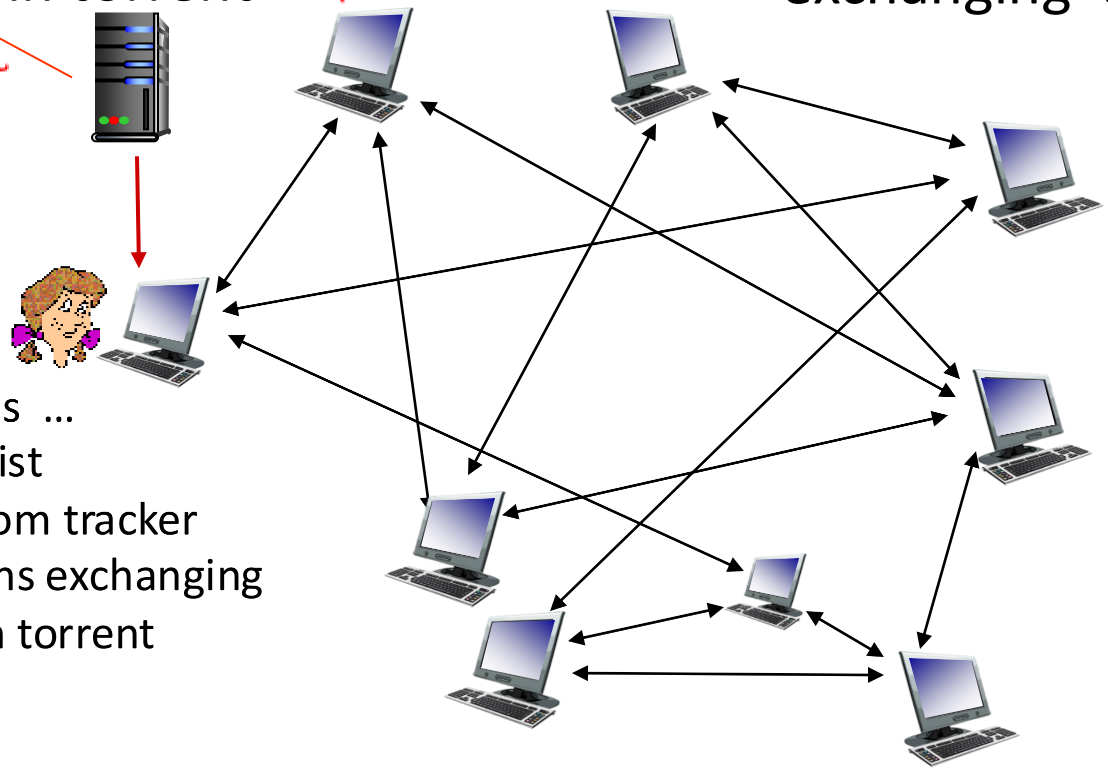


P2P file distribution: BitTorrent

tracker: tracks peers participating in torrent

↳ centralized server
mode in
↳ DHT

torrent: group of peers exchanging a particular file



Alice arrives …
… obtains list
of peers from tracker
… and begins exchanging
file peers in torrent

BitTorrent: requesting, sending file chunks

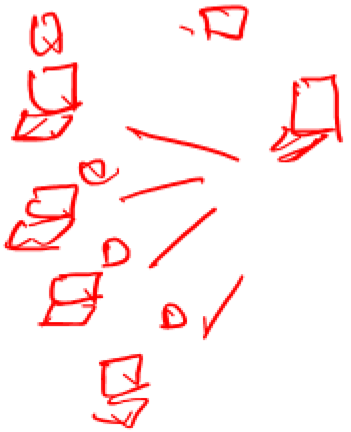
② Whom to send?

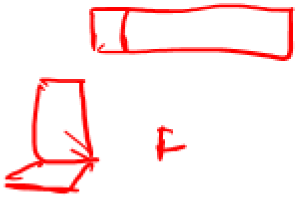
① Which chunks to request?

- at any given time, different peers may have different subsets of file chunks
- periodically, Alice asks each peer for list of chunks that they have
- Which chunk should Alice download first?

↳ download chunks from the closest neighbor: performance-wise

↳ Rarest first → keeps the torrent chunk distribution uniform





BitTorrent: requesting, sending file chunks

→ FCFS [what i

→ send chunks to peers that have send chunks to you (tit for tat)
↳ Bootstrapping
↳ Randomly select a new peer

Sending chunks: whom to send chunks?

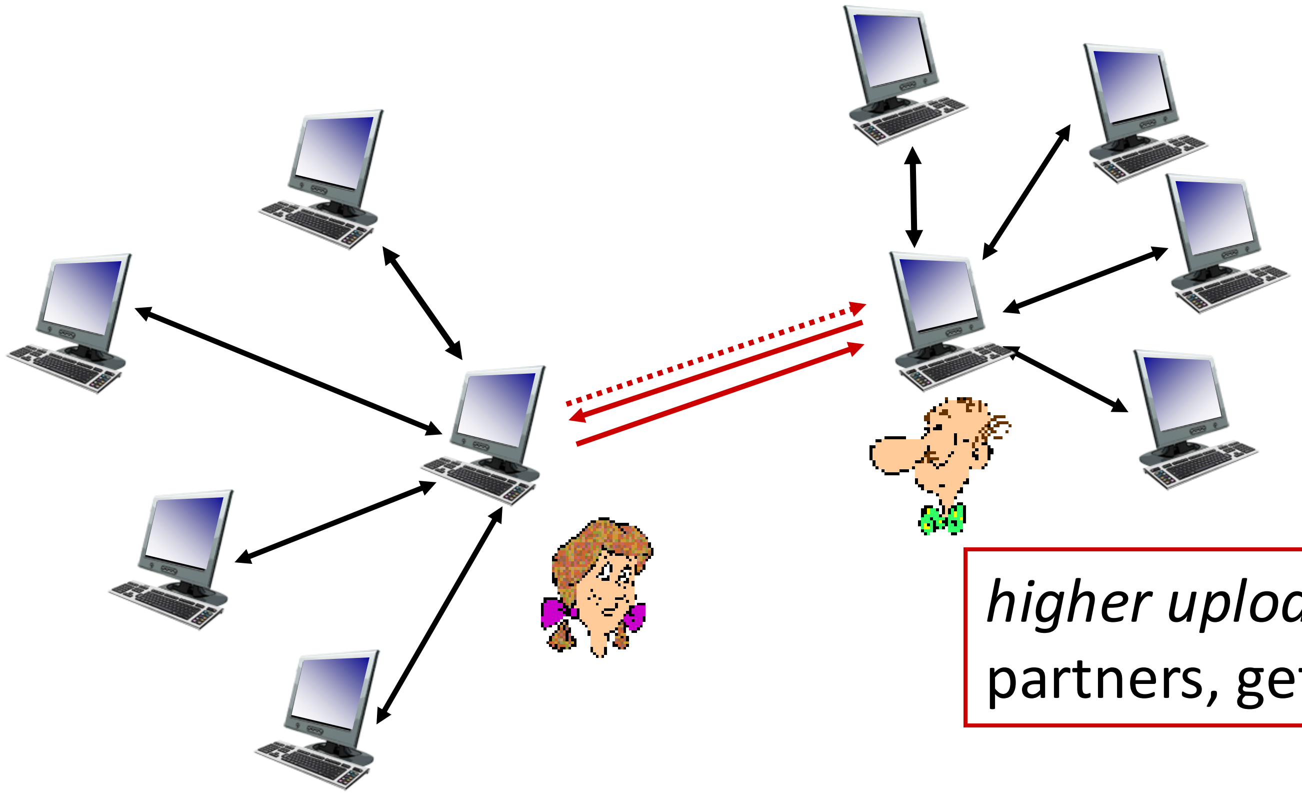
- Uses tit for tat
- sends chunks to those four peers currently sending chunks *at highest rate*
  - other peers are choked by Alice (do not receive chunks from her)
  - re-evaluate top 4 every10 secs
- every 30 secs: randomly select another peer, starts sending chunks

BitTorrent: tit-for-tat

(1) Alice "optimistically unchokes" Bob

(2) Alice becomes one of Bob's top-four providers; Bob reciprocates

(3) Bob becomes one of Alice's top-four providers



higher upload rate: find better trading partners, get file faster !

P2P Today

- Content Distribution Networks (CDNs)
  - Reliability
  - Ownership of data
- Open source data sharing
- Real-time communication
- Cryptocurrency

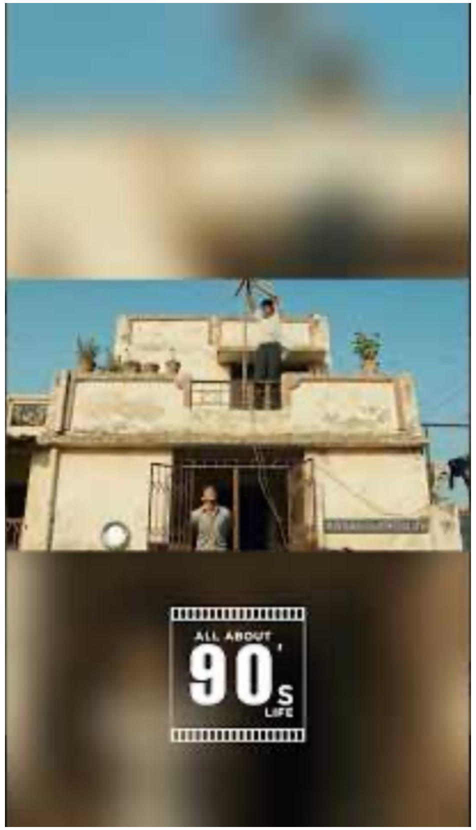
Recap: Application Layer

- HTTP
- Email
- DNS
- P2P
- **Video streaming**

What is Video Streaming?

[OTT video]

- Real-time delivery of video data over the Internet so playback begins before the entire file is downloaded
- Different from broadcast video



History of Video Streaming over the Internet


Cult Film Is a First On Internet

Reduced, decolored and sometimes silent. But still historic.

May 2, 1993: Due to bandwidth limitations, it is broadcast at 2 frames per second rather than the standard 24 frames per second. It was watched by a number of people at computer laboratories.

Biggest Cricket World Cup ever smashes Broadcast and Digital records

2023 headline

The decision to make coverage freely available for mobile users via Disney+ Hotstar in India led to a whopping 295 million LIVE Tournament viewers tuning in. Across the event, there were five world records broken on Disney+ Hotstar for digital peak concurrency, with the final attracting cricket's highest concurrent audience ever, having already made history at four other 2023 World Cup matches, demonstrating the appeal of cricket and the excitement the ODI format continues to offer:

How to design a video streaming system

Types of video streaming

① . On-demand video streaming [Pre-recorded content is streamed over the Internet]

② Live-streaming [Cricket worldcup]

③ Video conferencing (Real-time interactive video)

Totally different

[Team call
Whatsapp call]

④ Augmented reality (AR)

Virtual reality (VR)

How is video on demand (VoD) encoded?

Video Technology

- Video is a sequence of pictures → 30fps

HD video

1MB

30 frames

240Mbps

GOP

Group of picture

↳ I-frame : Independent frame

↳ Dependent frame : depends on an I-frame
(P or B-frames)

compress video
- → lossless
- → lossy

① Temporal redundancy

② Spatial Redundancy

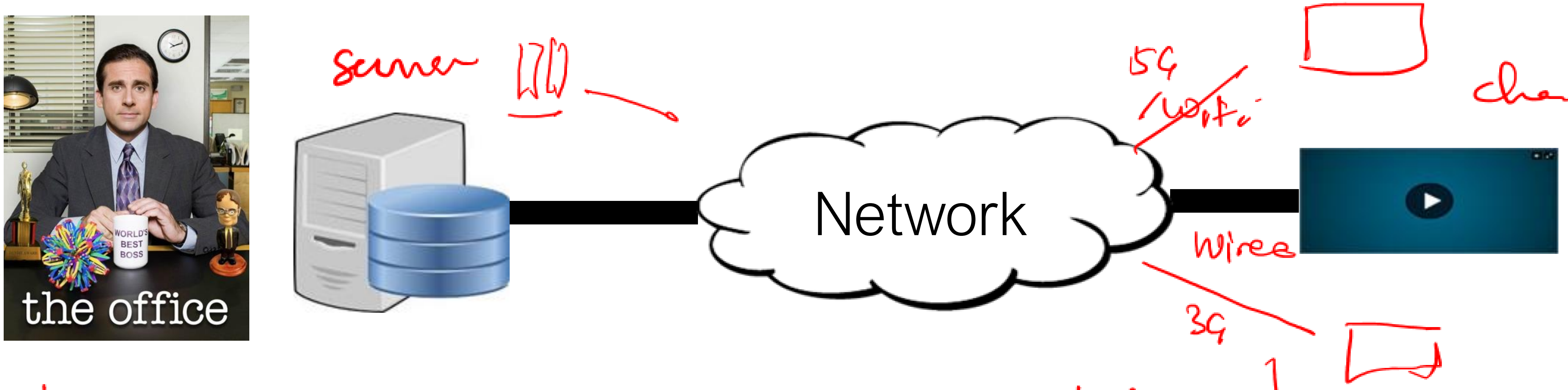
mechanism for compression

FACT : Compression is computationally expensive

What are the design goals for video on demand (VoD)?

System

Video File



HD

1. Play content without freezing (No rebuffering)
2. Download is sequential
3. Scalable system
4. Reliability is needed
5. Handle heterogeneity
6. Interactivity (Play / Pause / Rewind / FF)

Designing a video streaming system

Design Constraints:

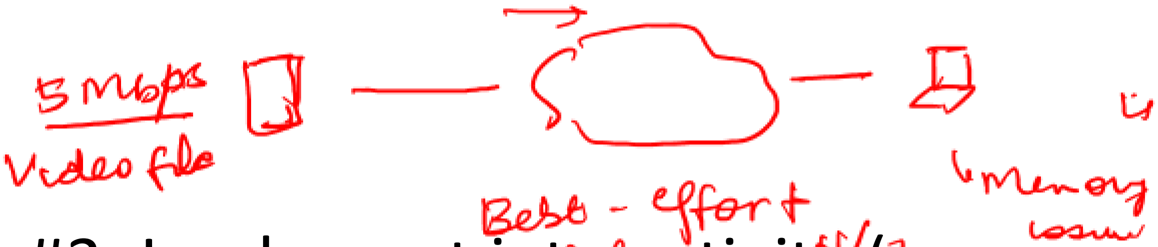
- No skipping of video content due to packet loss
- Continuous playout of content, i.e., avoid video freezing
- Interactivity: Pause, repositioning, fast-forwarding
- Scale to millions of users
- Client heterogeneity: different device types and network conditions

Achieving Design Goals

#1: No skipping of video content due to packet loss (Reliability)

↳ Just use TCP → [Congestion control]

#2: Continuous playout of content, i.e., avoid video freezing



RTT & B/w are different

↳ Download at higher rate

Buffer (fixed size)

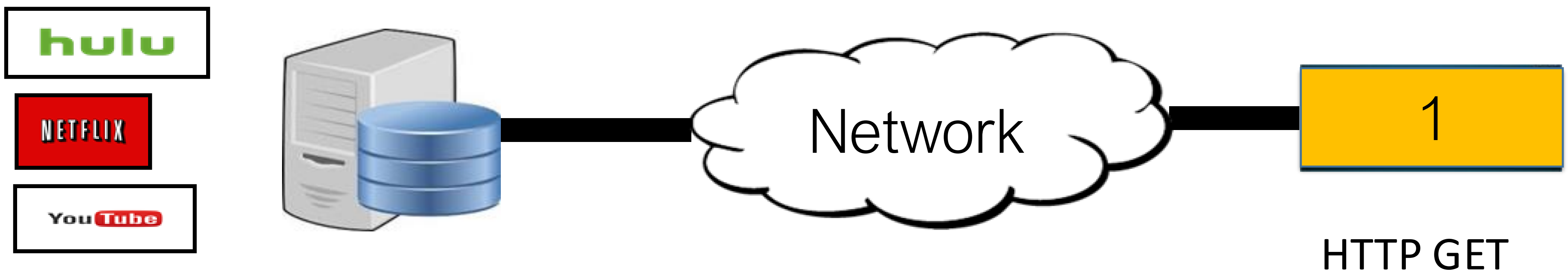
100 MB / 10 second

#3: Implement interactivity (pause, repositioning, fast-forward)

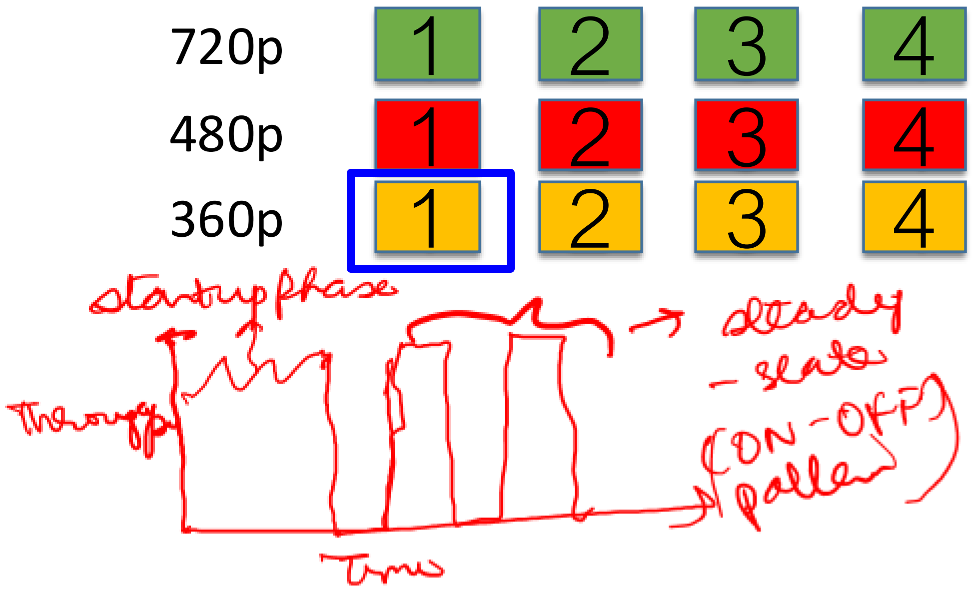
HTTP Adaptive Streaming (HAS)

Manifest file

HTTP GET manifest



HTTP GET
“Seg1@360p”



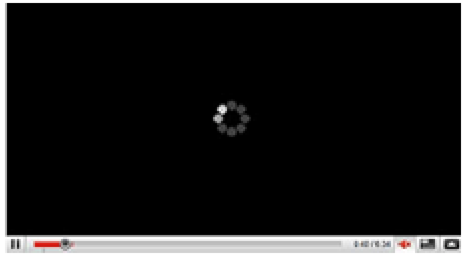
- *“intelligence”* at client: client determines
  - *when* to request chunk (so that buffer starvation, or overflow does not occur)
  - *what encoding rate* to request (higher quality when more bandwidth available)

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

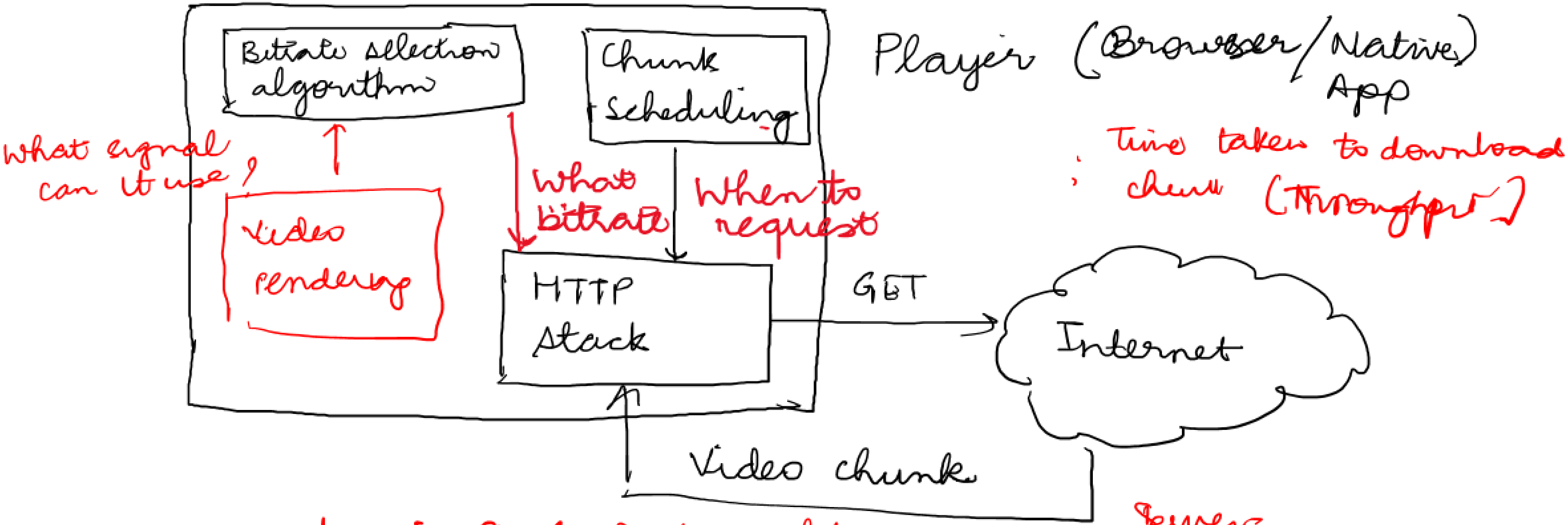


Maximize average bitrate

Minimize bitrate switches

Bitrate Adaptation

360p 2s 2s → Key frame / Independent frame
720p 2s 2s



During startup: ① highest quality
(Startup latency)

Server
4Mbps 720p
2Mbps 360p
1Mbps
3Mbps

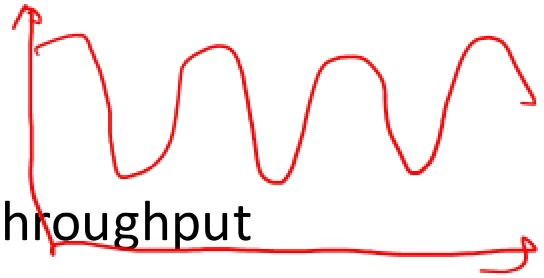
- **Q**: What are the signals available to the player for bitrate adaptation?

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

B/W



- 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

<u>Example</u>

Available bitrates

{200, 400, 800, 1600}

Kbps

T

max R_i α

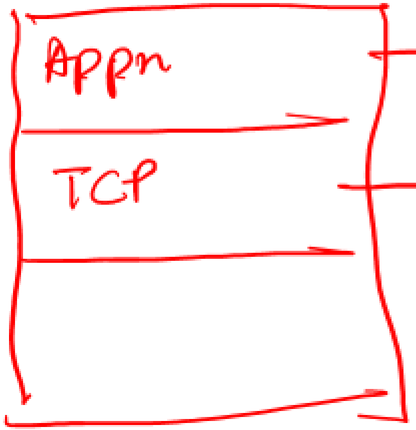
T ≥ (Chunk) Throughput

{700, conservative, 1000, 700}

Issue with Rate-based Adaptation

① Fluctuation :

moving average or smoothing can help



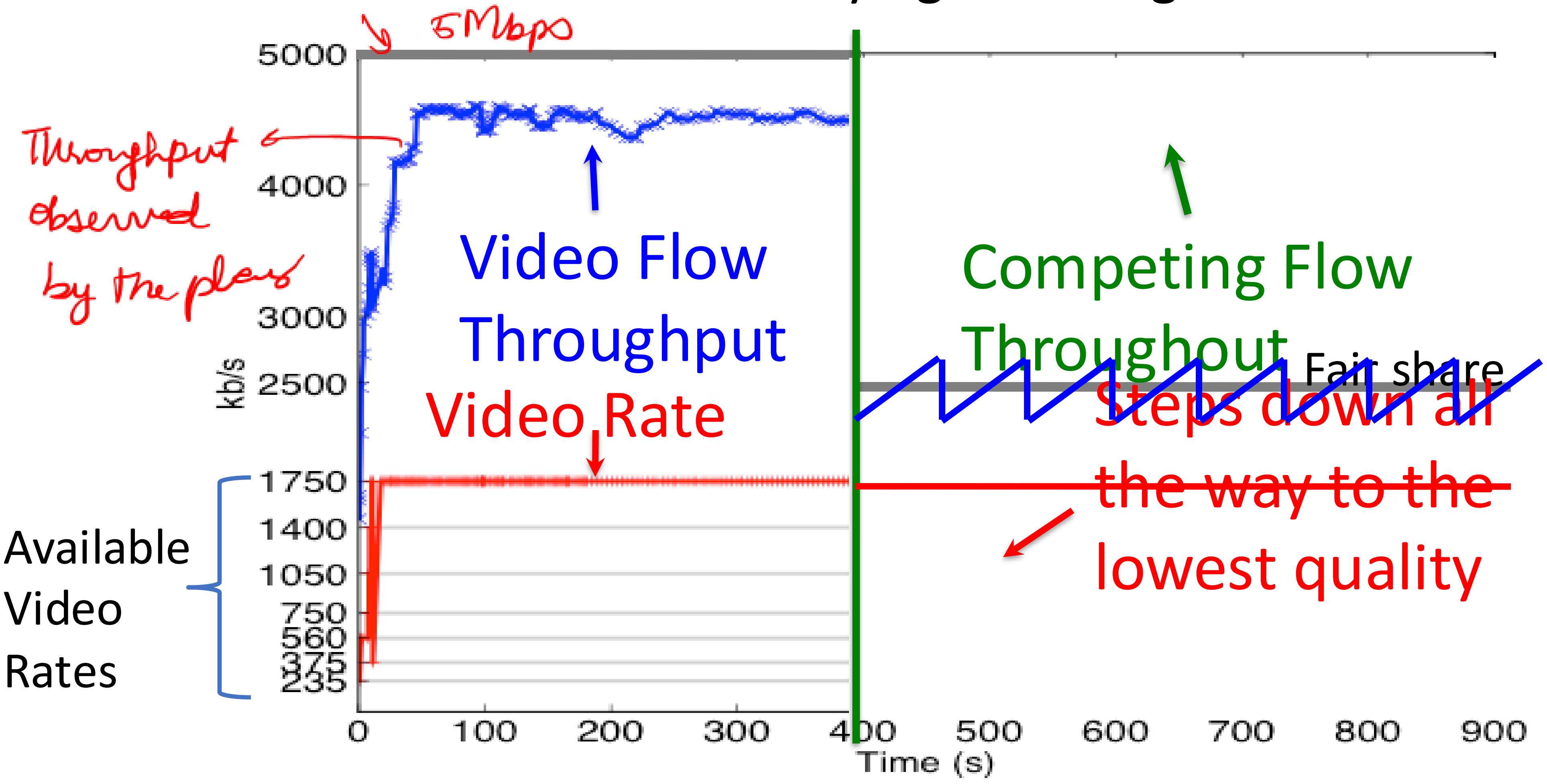
→ Rate adaptation

→ Congestion control

Double control loop

Issue with Rate-based Adaptation

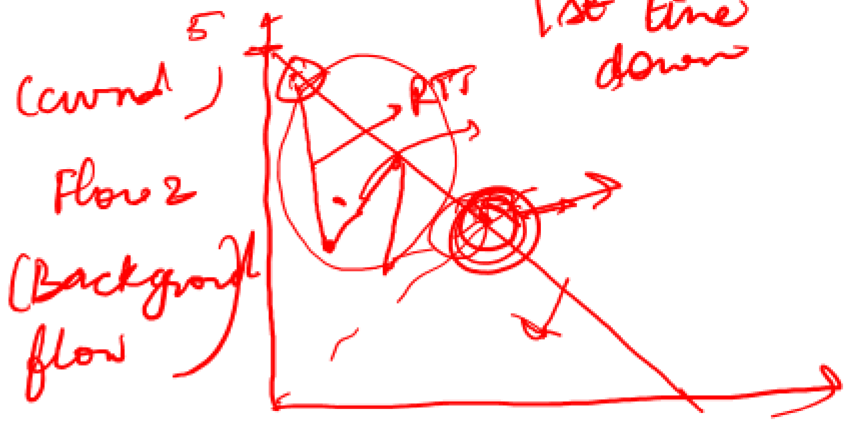
- Poor interaction with the underlying TCP congestion control



Source: Confused, Timid and Unstable: Picking a Video Rate is Hard

TCP Throughput of the Video Flow

Negative feedback loop



(cwnd) Flow 1 (Video Flow)

Avg Tput observed by

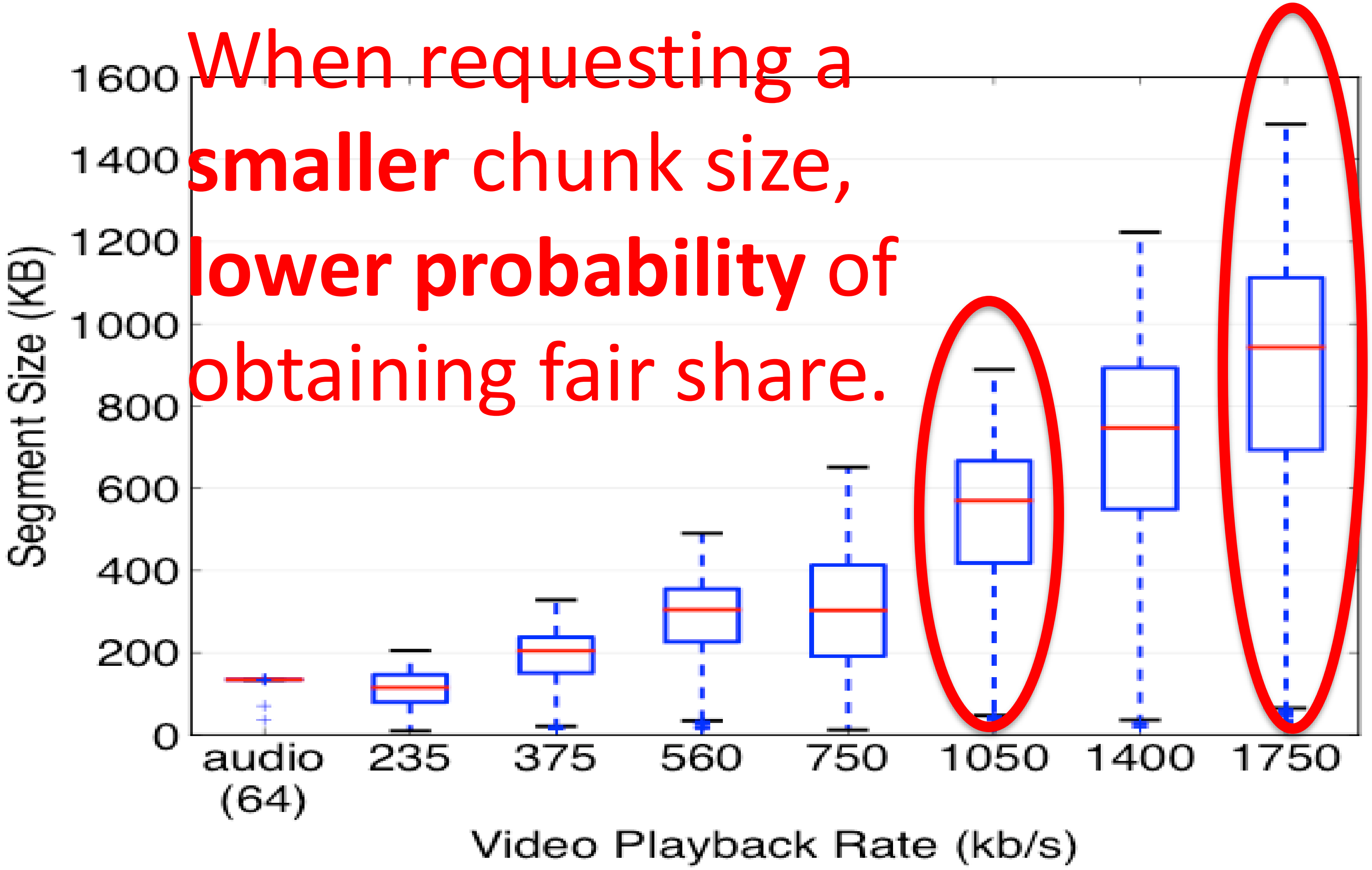
≤ 2.5 Mbps

- 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~~ chunks get < 1.8Mb/s

Constant duration = L   File size = L × R

Smaller Chunk Size for Lower Video Rate

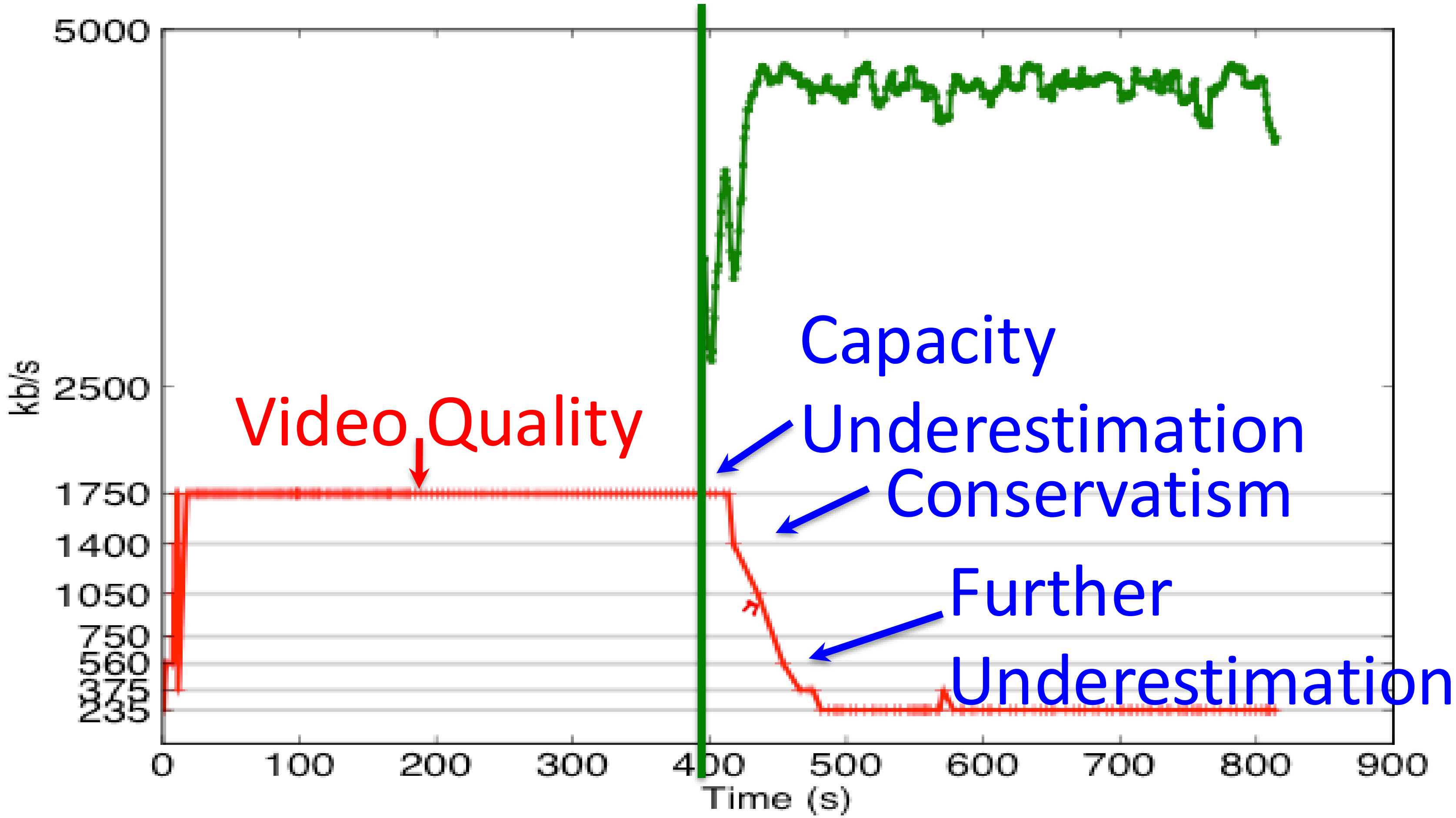
When requesting a **smaller** chunk size, **lower probability** of obtaining fair share.



Source: Confused, Timid and Unstable: Picking a Video Rate is Hard

Buffer

The Complete Story

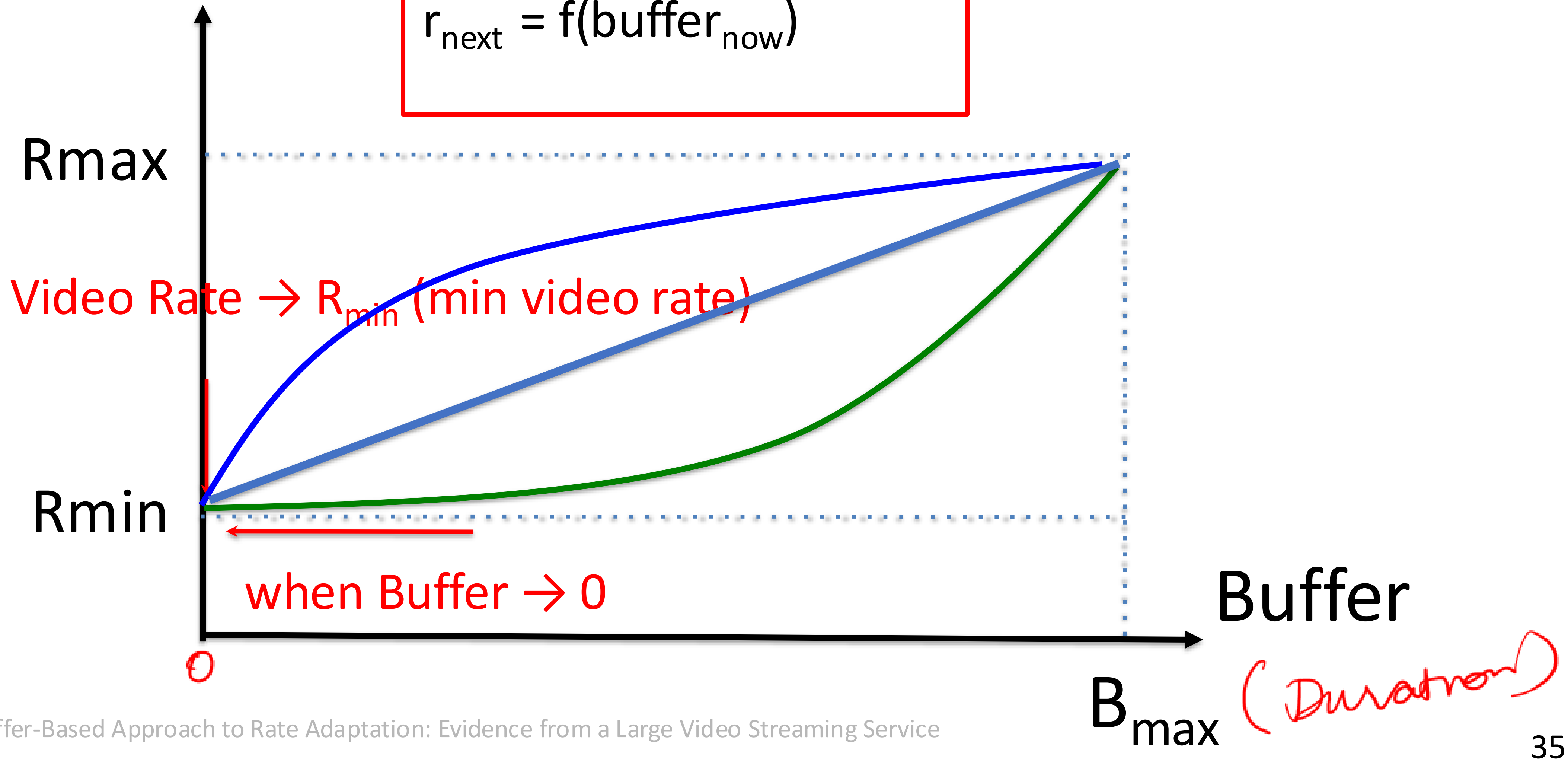


Being conservative can trigger a vicious cycle!

Source: Confused, Timid and Unstable: Picking a Video Rate is Hard

Buffer-based adaptation: Algorithm Sketch

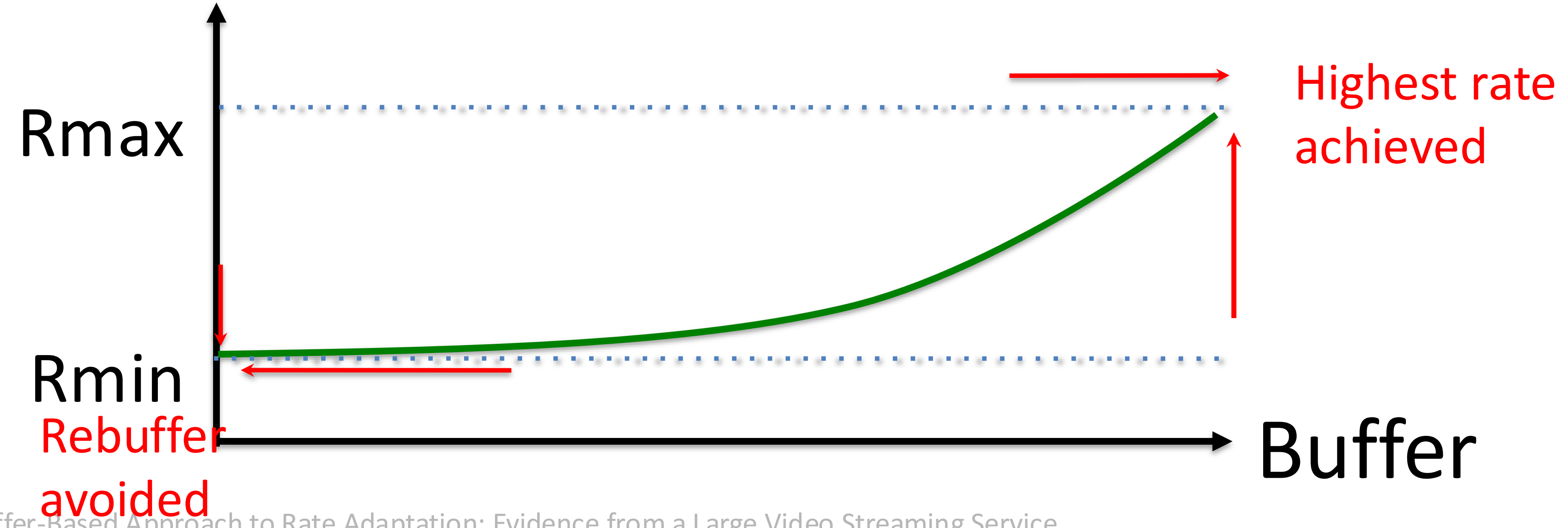
$r_{next} = f(buffer_{now})$



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



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

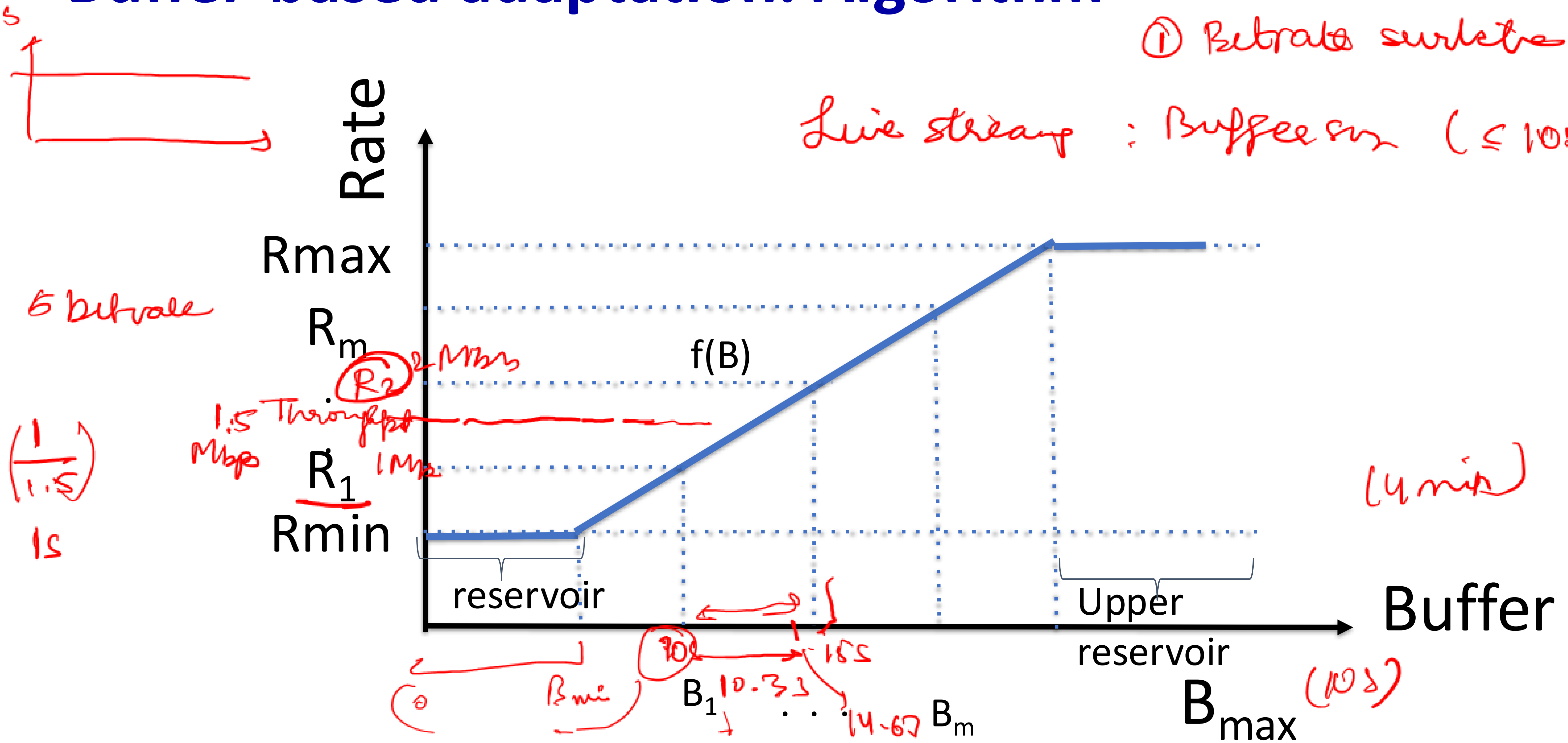
(BOLA):

Buffer-based adaptation: Algorithm

(Netflix)

① Bitrate switching

Live streaming : Buffer size (≤ 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 …