Special Topics: Machine Learning (ML) for Networking

COL867

Holi, 2025

Application Performance Monitoring

Tarun Mangla

Case Studies: ML for Specific Network Learning Tasks

- Application Classification
- **Application Performance Monitoring**
- Resource Allocation
- Security

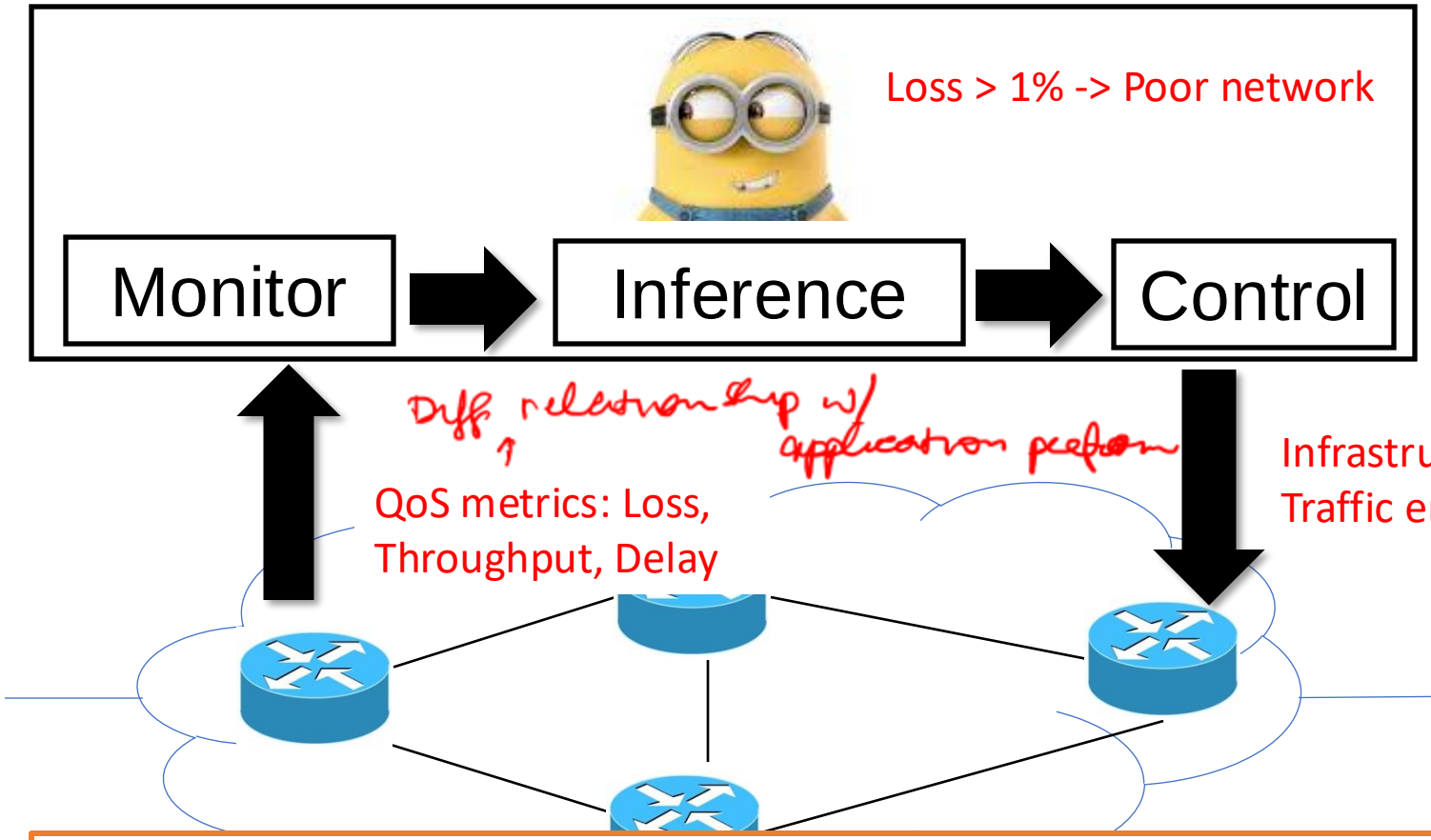
Agenda

- What is application performance monitoring
- Why is it important?
- What are the challenges?
- Performance monitoring for a specific application: video streaming
  - Modeling-based methods
  - ML-based methods

[Elastic applications
↓
Adapt to the n/w conditions

Traditional Network Management

.
.
.



A better approach: Monitor ~~QoS metrics~~ application performance metrics

Traditional Network Management

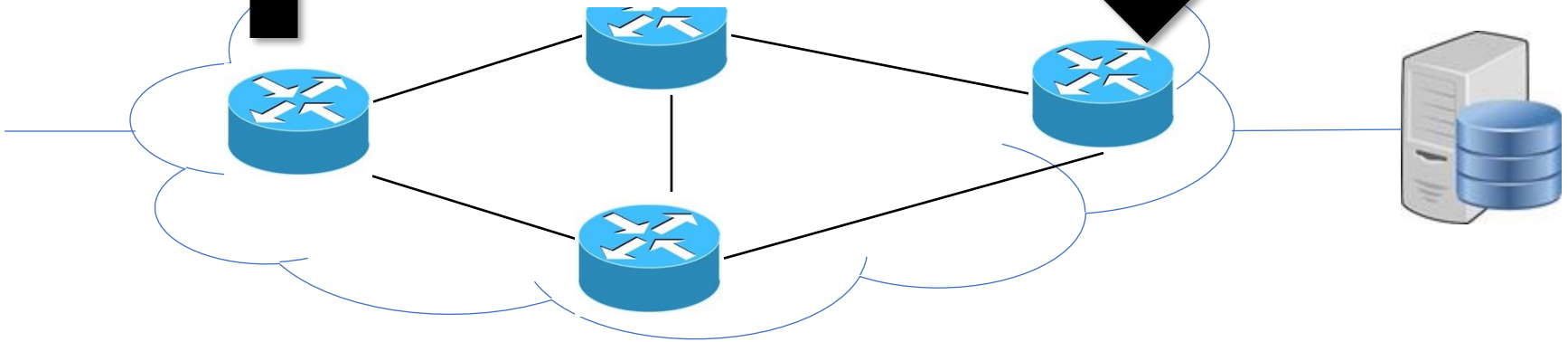
.
.
.

Application performance below threshold -> Poor network

Monitor → Inference → Control

Application performance metrics (e.g., re-buffering)

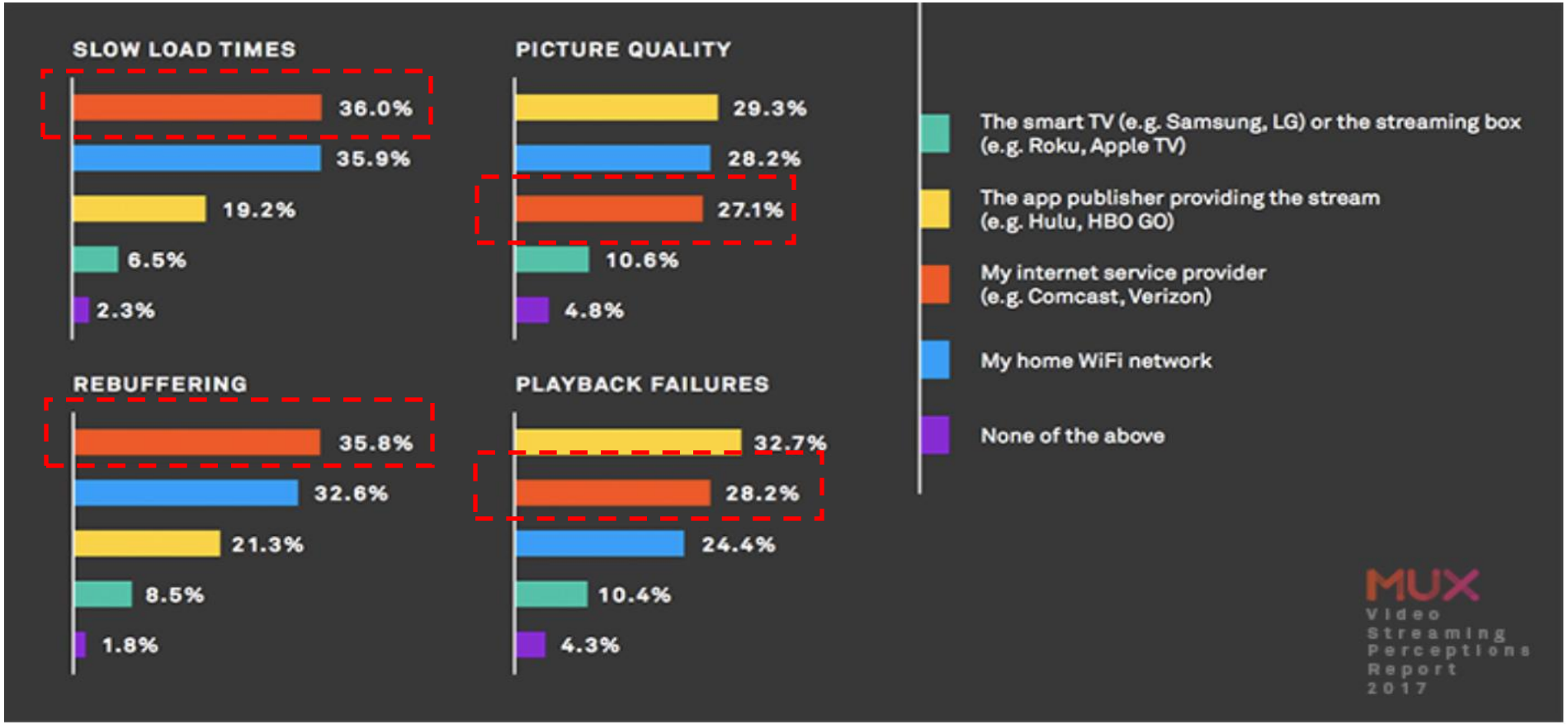
Infrastructure update, Traffic engineering etc.



Network Management Control Loop

Why would operators care about application performance?

Consider video streaming application ..

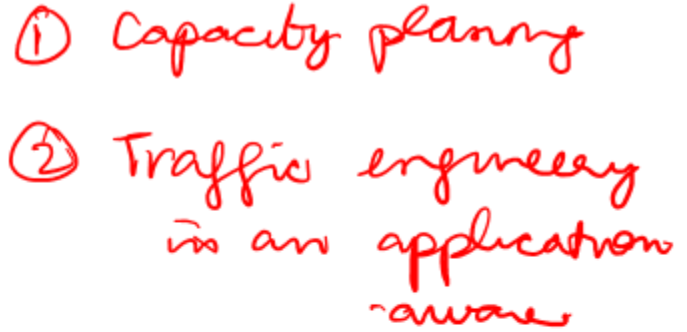


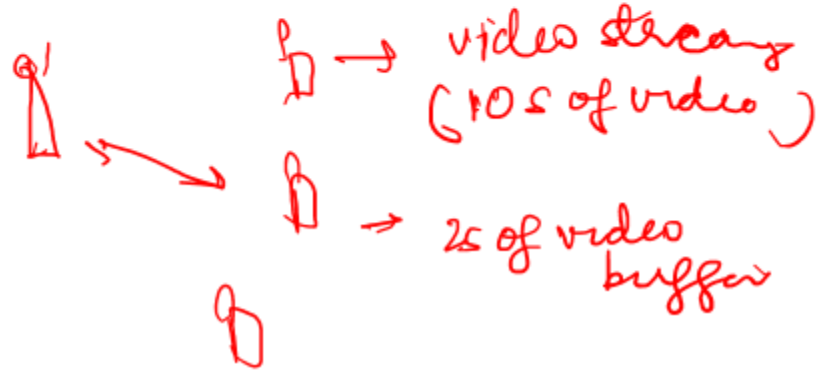
ISPs get blamed for application performance issues!

What kind of control application monitoring enables?

- Long-term capacity planning

- Real-time resource provisioning
  - Throttle a flow in an application-aware manner





How would an ISP monitor application performance?

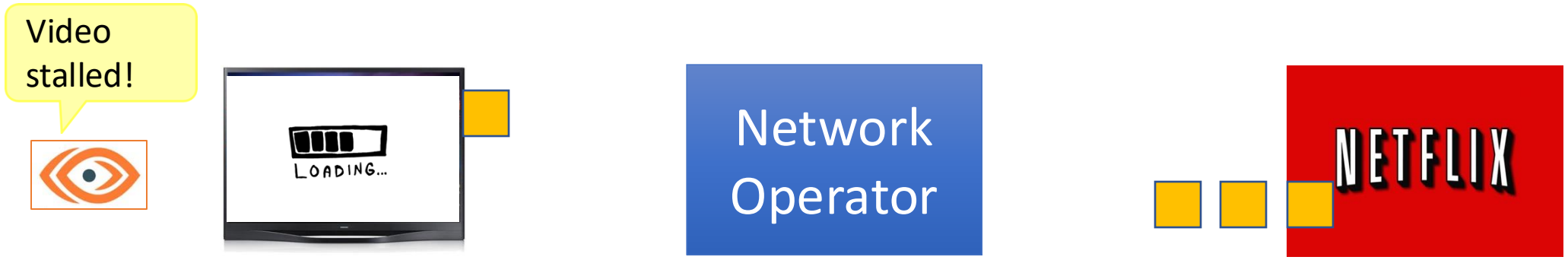
→ Video streaming → Re-buffering
→ Video quality

① Analyze the N/w traffic

② Feedback from users / app developers (collaboration b/w end-host & n/w)

Monitoring application performance is challenging for operators

- App developers can monitor performance through in-app SDKs
- But network operators do not have access to end points
- Only have access to network data

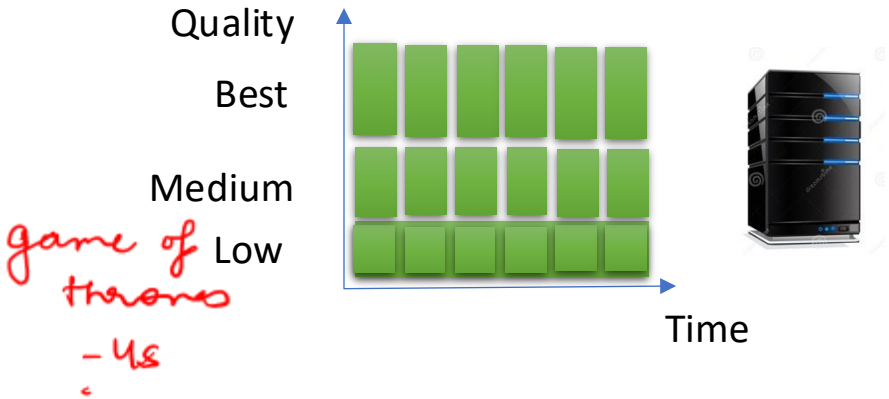


Need to infer application performance from network measurement data

Agenda

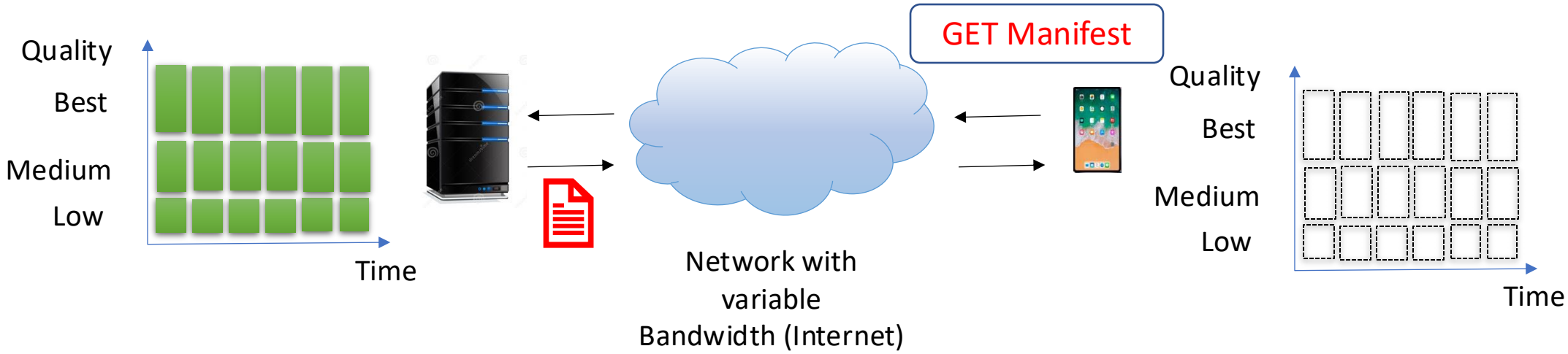
- What is application performance prediction
- Challenges
- **Video streaming performance prediction (or QoE estimation)**
  - Modeling-based method
  - ML-based method

HTTP Adaptive Streaming (HAS)



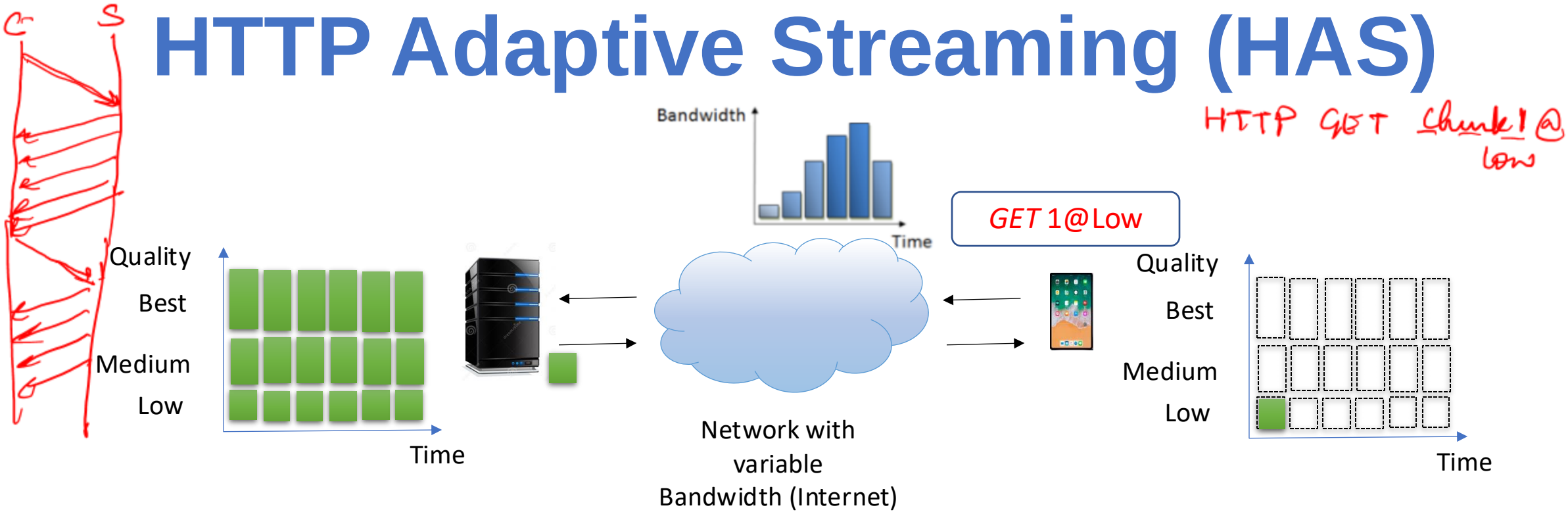
- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server

HTTP Adaptive Streaming (HAS)



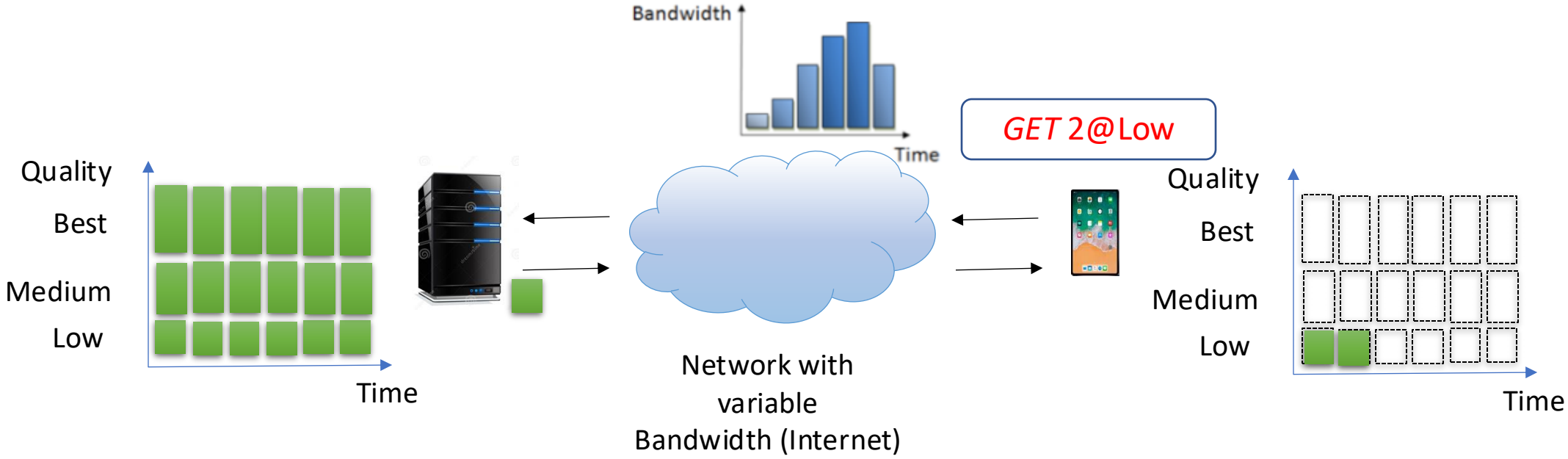
- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server
- The player first downloads a manifest file containing information about video chunks

HTTP Adaptive Streaming (HAS)



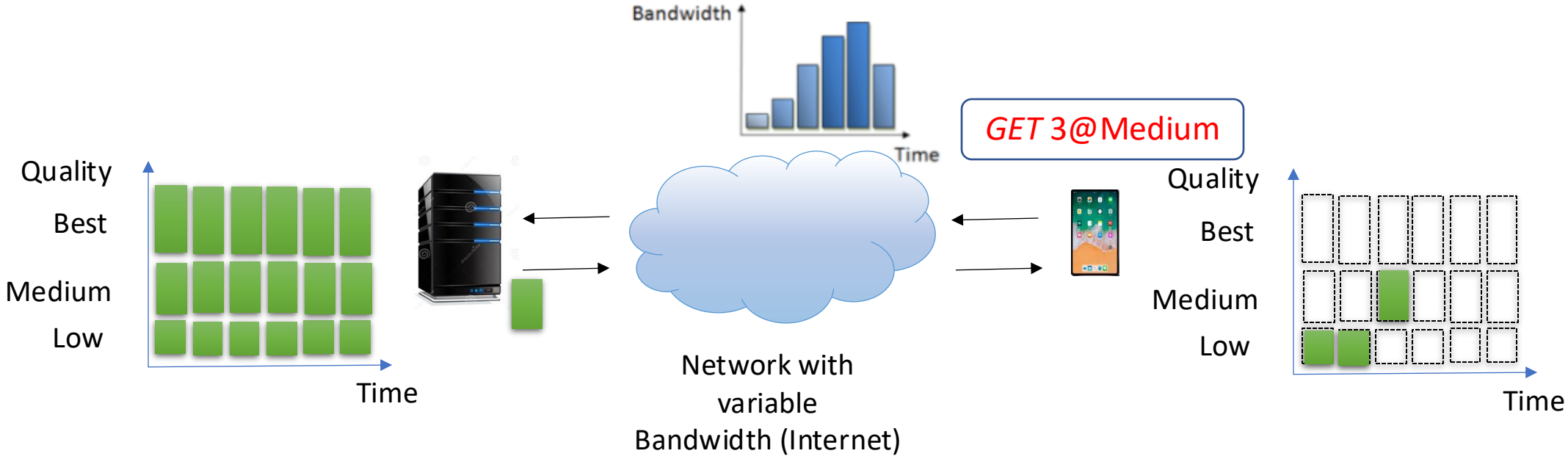
- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server
- The player first downloads a manifest file containing information about video chunks
- Client sends HTTP GET requests for chunks of the quality that matches network conditions

HTTP Adaptive Streaming (HAS)



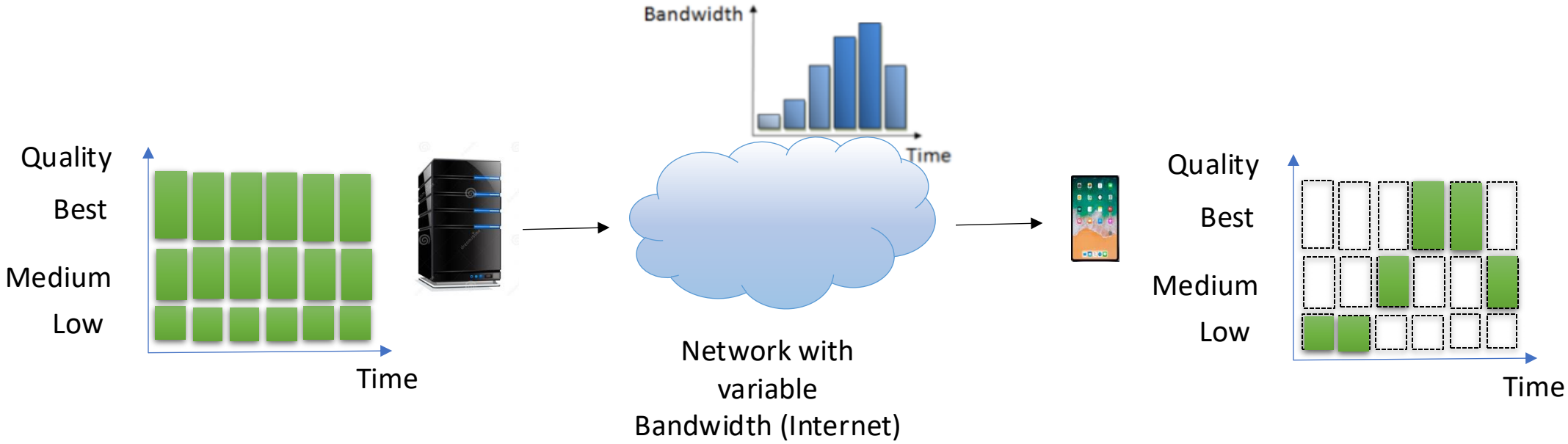
- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server
- The player first downloads a manifest file containing information about video chunks
- Client sends HTTP GET requests for chunks of the quality that matches network conditions

HTTP Adaptive Streaming (HAS)



- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server
- The player first downloads a manifest file containing information about video chunks
- Client sends HTTP GET requests for chunks of the quality that matches network conditions

HTTP Adaptive Streaming (HAS)



- Video is divided into chunks or segments of certain duration
- Each chunk is encoded into discrete quality levels or bitrates and stored on an HTTP server
- The player first downloads a manifest file containing information about video chunks
- Client sends HTTP GET requests for chunks of the quality that matches network conditions

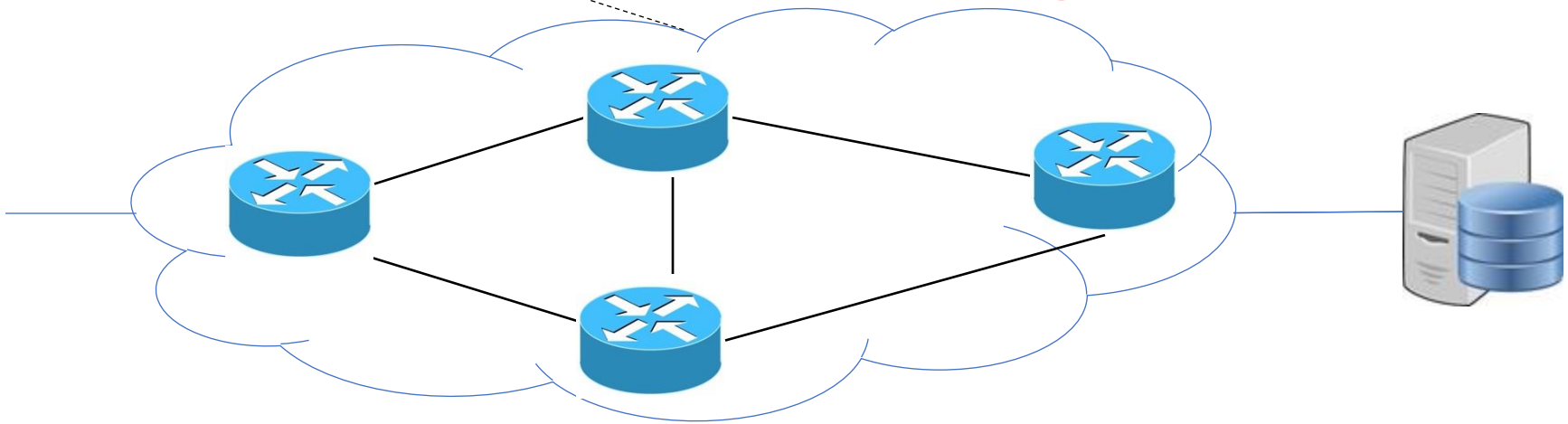
A video session can be modeled as a sequence of chunk downloads

Problem Statement

Network measurement data → Video Streaming QoE

- Re-buffering
- Video quality
- Video switches

Passive Measurement Data → Video Traffic + Session Identification → Video Performance Prediction



Can we model video performance using passive network data?

→ Log the HTTP Requests

(Unencrypted data)

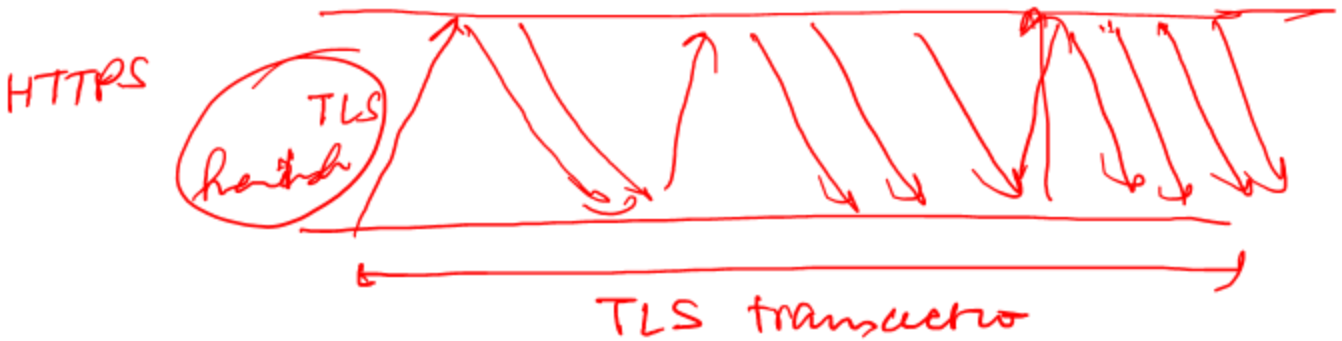
videoapp.com / chunk1 - q:high

48

Re-buffering :

Model buffer occupancy

Encrypted data : can you track HTTP downloads



→

HTTP : download time
, Size of the transaction

① Video quality
② Re-buffer
③ Switches

1 → high → Bitrate level
2 → low
↳ 240p

When Traffic is Unencrypted

- Deploy a web proxy in the network to monitor HTTP requests corresponding to video chunks
- For each chunk in a session:
  - Extract **chunk identifier *(i)*** and **chunk quality *(Q_i)*** from the URI
  - Obtain **response completion timestamp *(T_i)*** from proxy

/videoapp/path/**V0987654321**/**track03**/**segment101**.ts?p=324&token=**32543563654645**

Content ID Chunk quality Chunk ID Session ID

QoE metrics estimation

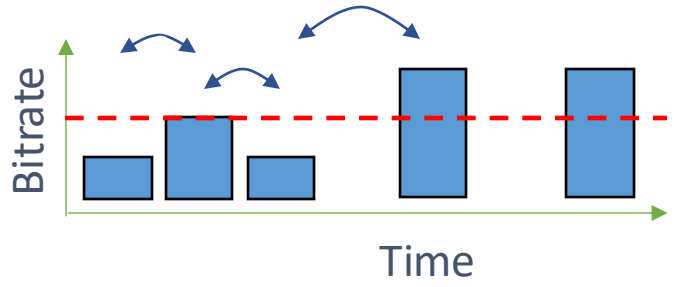
- Average bitrate: $\frac{\sum_{i=1}^{N} Q_i}{N \times L}$,

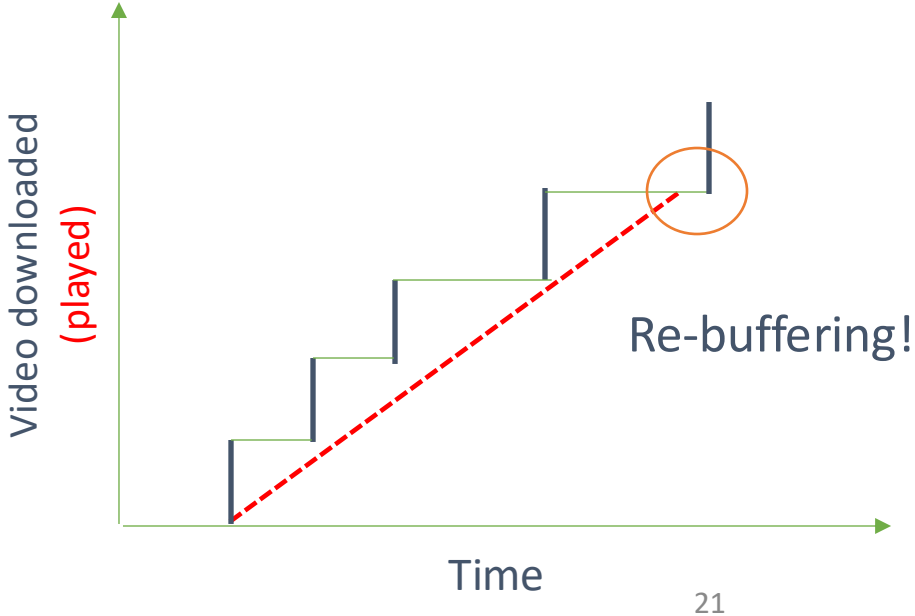
  N is the # of chunks in sessions, L is chunk duration

- Number of switches: $\sum_{i=2}^{N} I(Q_i \neq Q_{i-1})$

  I is the indicator function

- Re-buffering ratio: Model buffer occupancy by accounting downloaded video chunks and total time elapsed





Modeling Buffer Occupancy

200 kbytes
100 kbps

- Buffer filling: L seconds for every chunk download

- Buffer draining: Assuming linear playback, 1s every 1s

B_i : Buffer when chunk i was downloaded

4s chunk 2 seconds to download

What is $B_{i+1} = B_i + (4-2)$