Computer Networks
COL 334/672

Application Layer: DNS and P2P

Tarun Mangla

Slides adapted from KR

Sem 1, 2024-25

Quiz

Password: dns

Recap: Application Layer

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

Video Streaming

- Stream stored video traffic
  - E.g., Netflix, YouTube, Hotstar

- Killer application over the Internet.

- Being able to stream video to large-scale users (say ~1B users) has been a North Star for Internet stakeholders
  - Handle client and network heterogeneity
  - Scale: Numbers and geography

Video Streaming

- stream stored video traffic: current killer
  - E.g.
- *chall*
  - Ho
  - diff
    cap
    ba
    poor)
- Enabling large-scale video streaming has been the North Star for Internet stakeholders

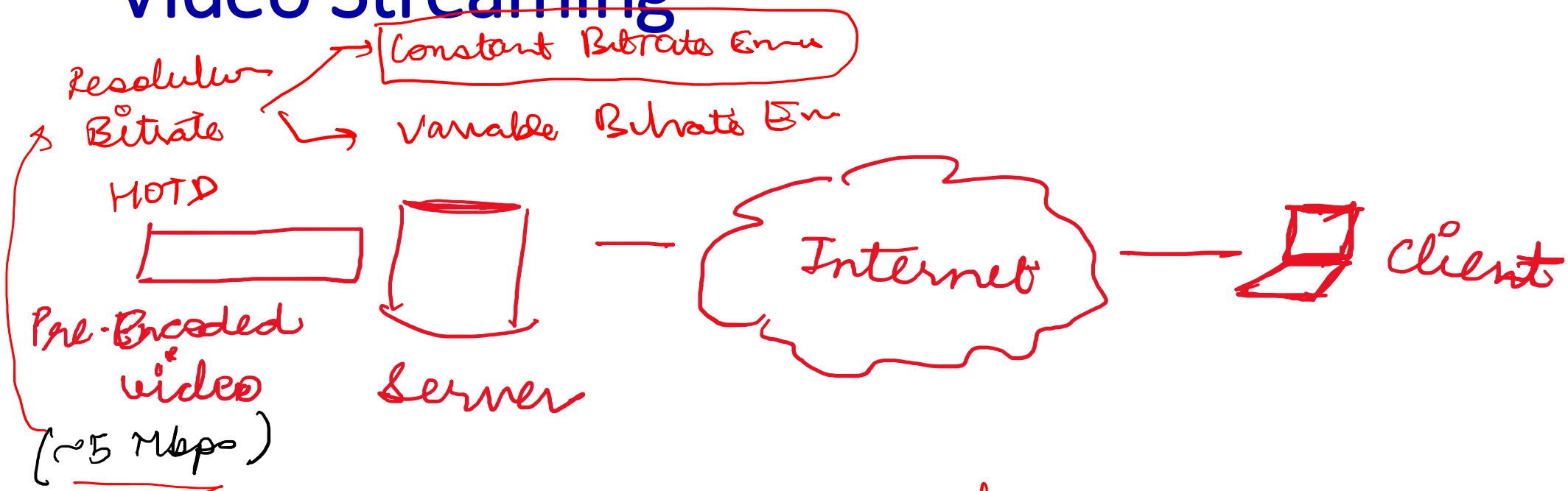
Biggest Cricket World Cup ever smashes Broadcast and Digital records

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:

Learning Goals

- What is video streaming?
- What are the protocols used for streaming video over the Internet

Video Streaming



→ latency-sensitive to some extent

Streaming: Play video as it is being downloaded and don't save it

Q: What is the application performance metric for streaming video?

Designing a video streaming system

(Reliable delivery)

Design Goals:

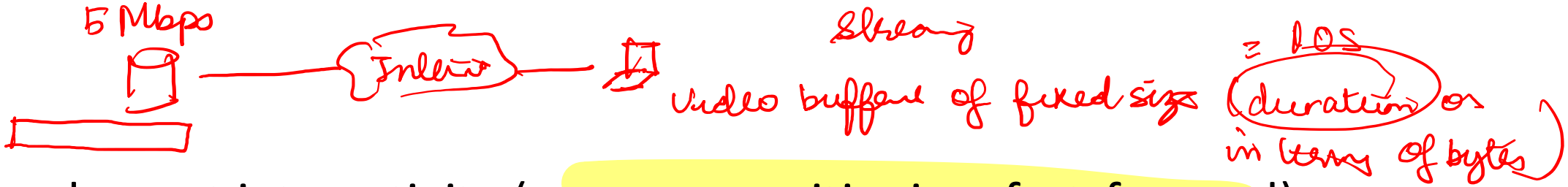
- No skipping of video content due to packet loss → All of the content to be delivered over the N/w
- Continuous playout of content, i.e., avoid video freezing (Application performance requirement)
- Interactivity: Pause, repositioning, fast-forwarding
- Scale to millions of users
- Client heterogeneity: different device types and network conditions

Achieving Design Goals

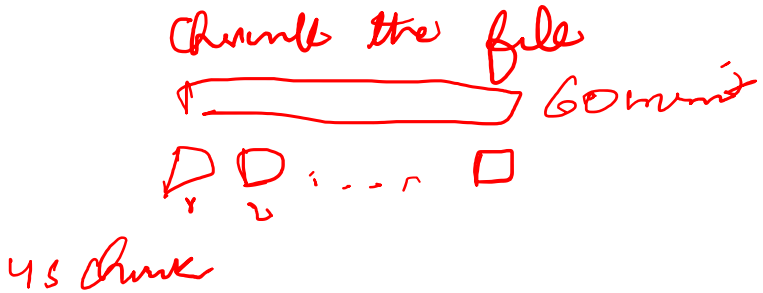
#1: No skipping of video content due to packet loss

Reliable transport (TCP)

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



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



45 chunk

How to achieve scale?

RTSP servers → FLASH

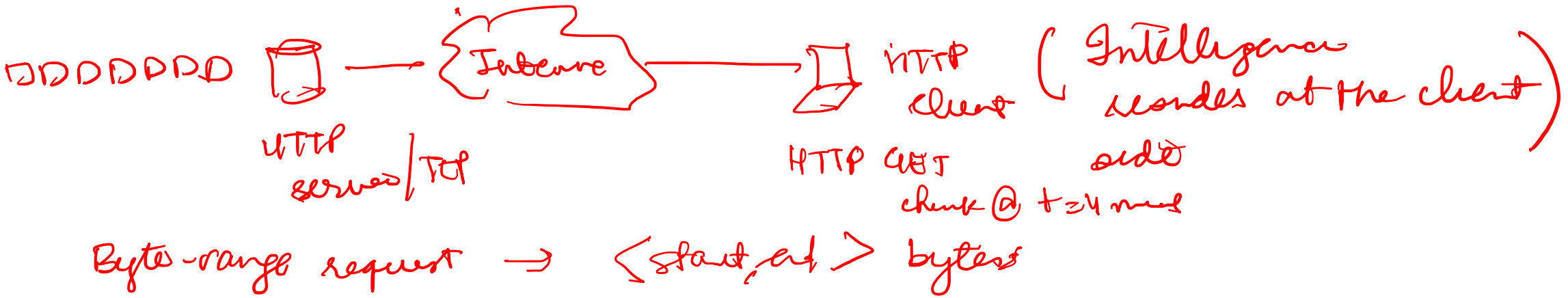
Browser → HTTP

CDNs

Web → HTTP

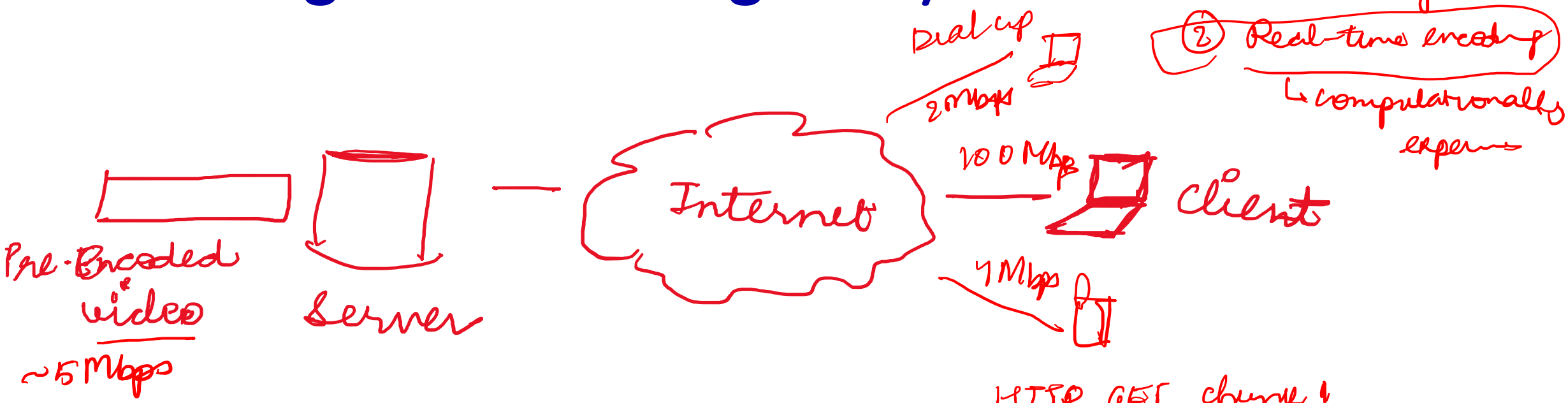
- Need geographically distributed video servers (special servers?)
  - ↳ Reuse existing infrastructure
  - ↳ More firewall friendly

HTTP - based already



(Intelligence resides at the client)

Handling client heterogeneity



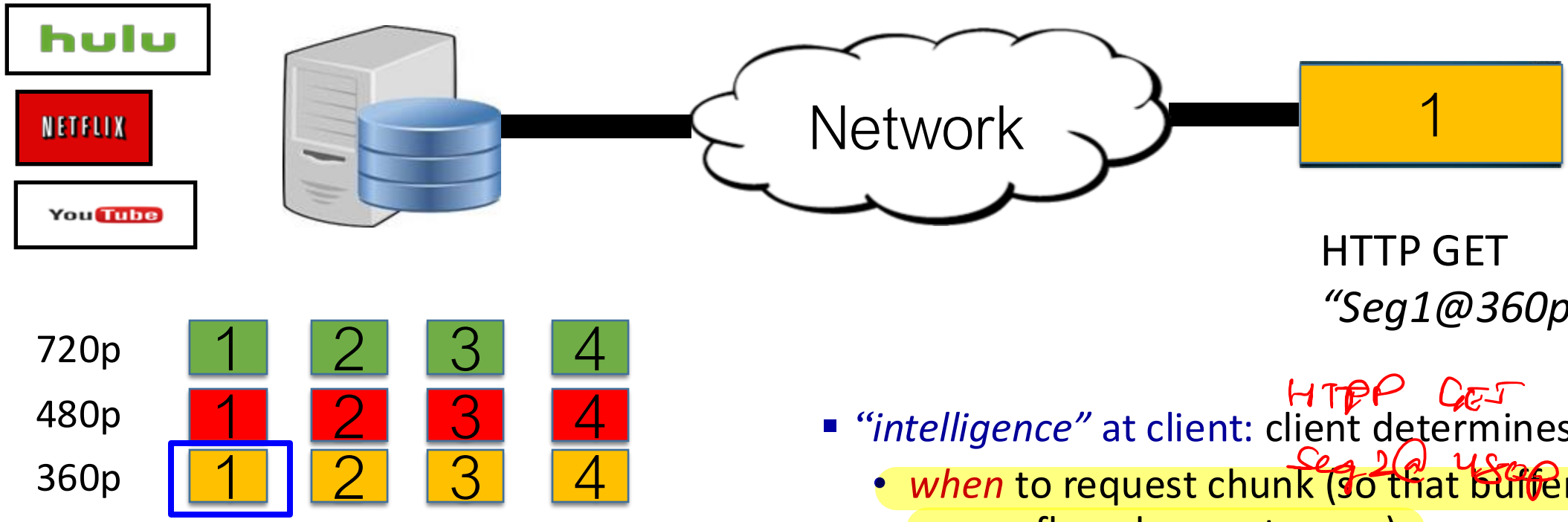
Q: What bitrate should we encode the video?

Constraints: clients with diverse network conditions and device capabilities

- Real-time encoding [not efficient]
- Pre-encode multiple versions

Dynamic Adaptive Streaming over HTTP (DASH)

HTTP Adaptive Streaming (HAS)



- *"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

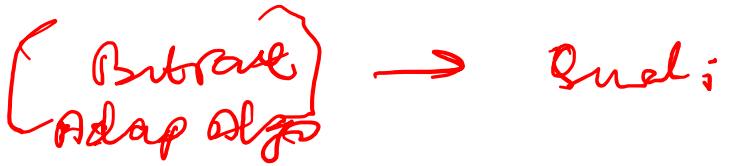
When to request a new video chunk?

- Client keeps a maximum buffer threshold, i.e., the maximum amount of downloaded but not played video
  - Either expressed as duration or number of bytes

- If the current video buffer occupancy > max buffer threshold, wait for the video buffer to deplete to less than max buffer threshold

- Once video buffer occupancy < max buffer threshold, request a new chunk

At what bitrate?

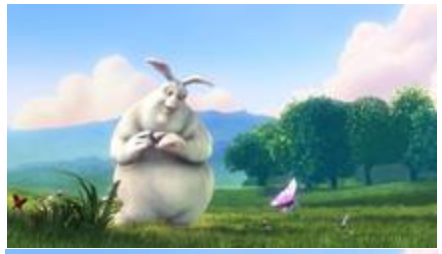


Designing Bitrate Adaptation Algorithm

- 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