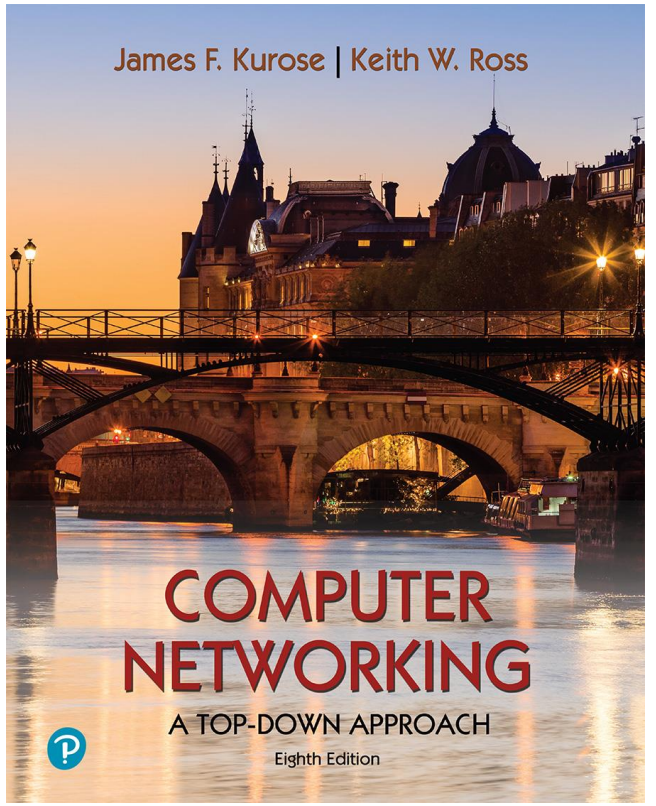
Chapter 2
Application Layer

Yaxiong Xie

Department of Computer Science and Engineering
University at Buffalo, SUNY



Computer Networking: A Top-Down Approach
8th edition n
Jim Kurose, Keith Ross
Pearson, 2020

Adapted from the slides of the book's authors

Application Layer: Overview

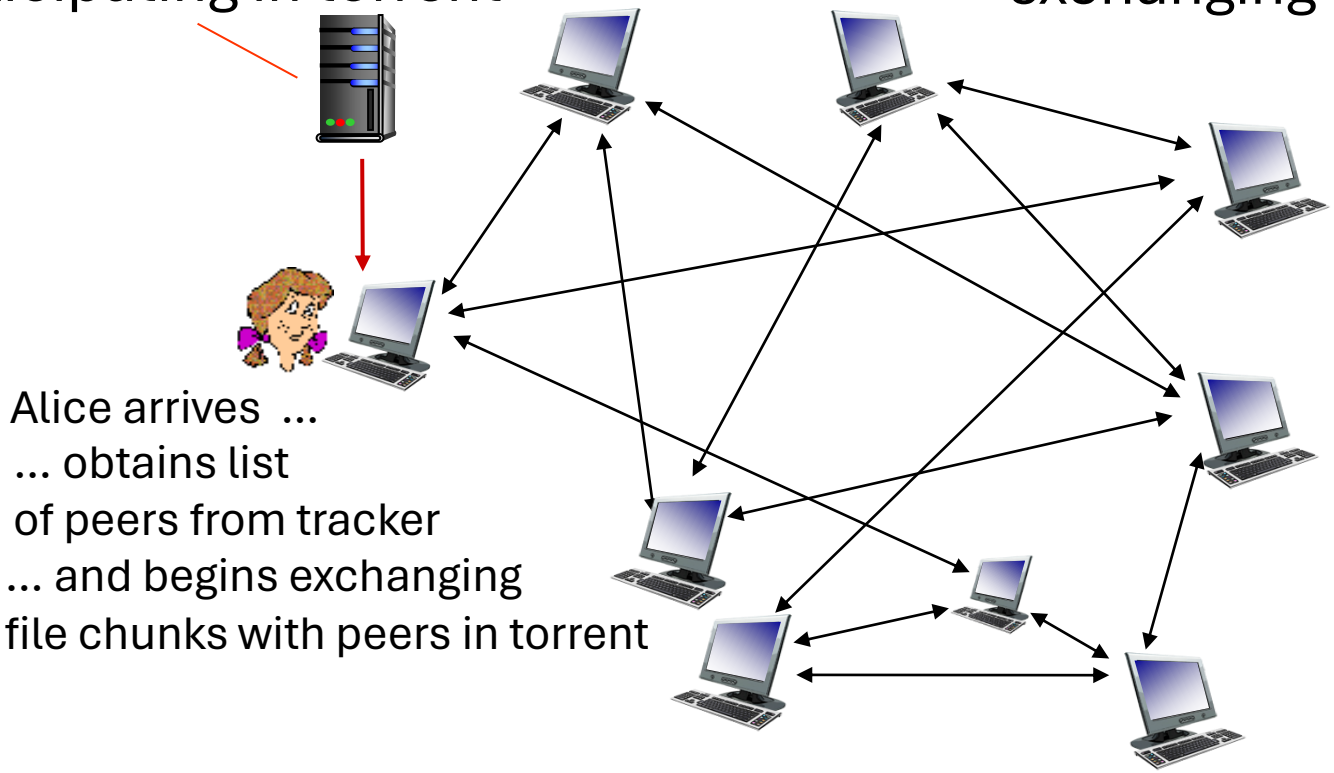
- Principles of network applications
- socket programming with UDP and TCP
- Web and HTTP
- E-mail, SMTP, IMAP
- The Domain Name System DNS
- P2P applications
- video streaming and content distribution networks

P2P file distribution: BitTorrent

- file divided into 256Kb chunks
- peers in torrent send/receive file chunks

tracker: tracks peers participating in torrent

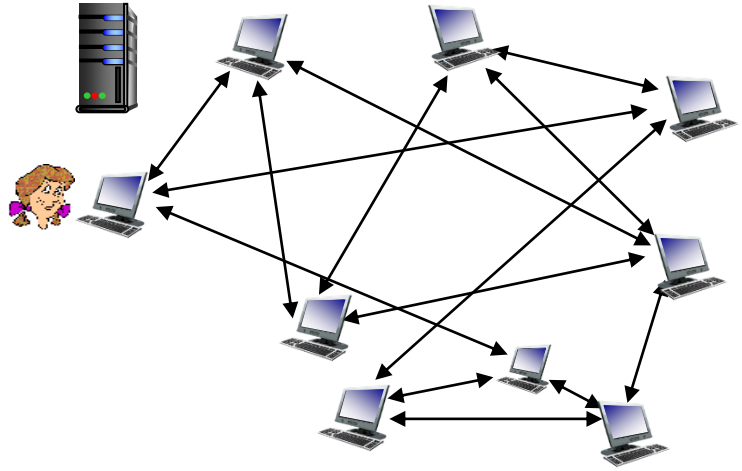
torrent: group of peers exchanging chunks of a file



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

P2P file distribution: BitTorrent

- peer joining torrent:
  - has no chunks, but will accumulate them over time from other peers
  - registers with tracker to get list of peers, connects to subset of peers ("neighbors")



- while downloading, peer uploads chunks to other peers
- peer may change peers with whom it exchanges chunks
- *churn:* peers may come and go
- once peer has entire file, it may (selfishly) leave or (altruistically) remain in torrent

BitTorrent: requesting, sending file chunks

Requesting chunks:

- at any given time, different peers have different subsets of file chunks
- periodically, Alice asks each peer for list of chunks that they have
- Alice requests missing chunks from peers, rarest first

Sending chunks: tit-for-tat

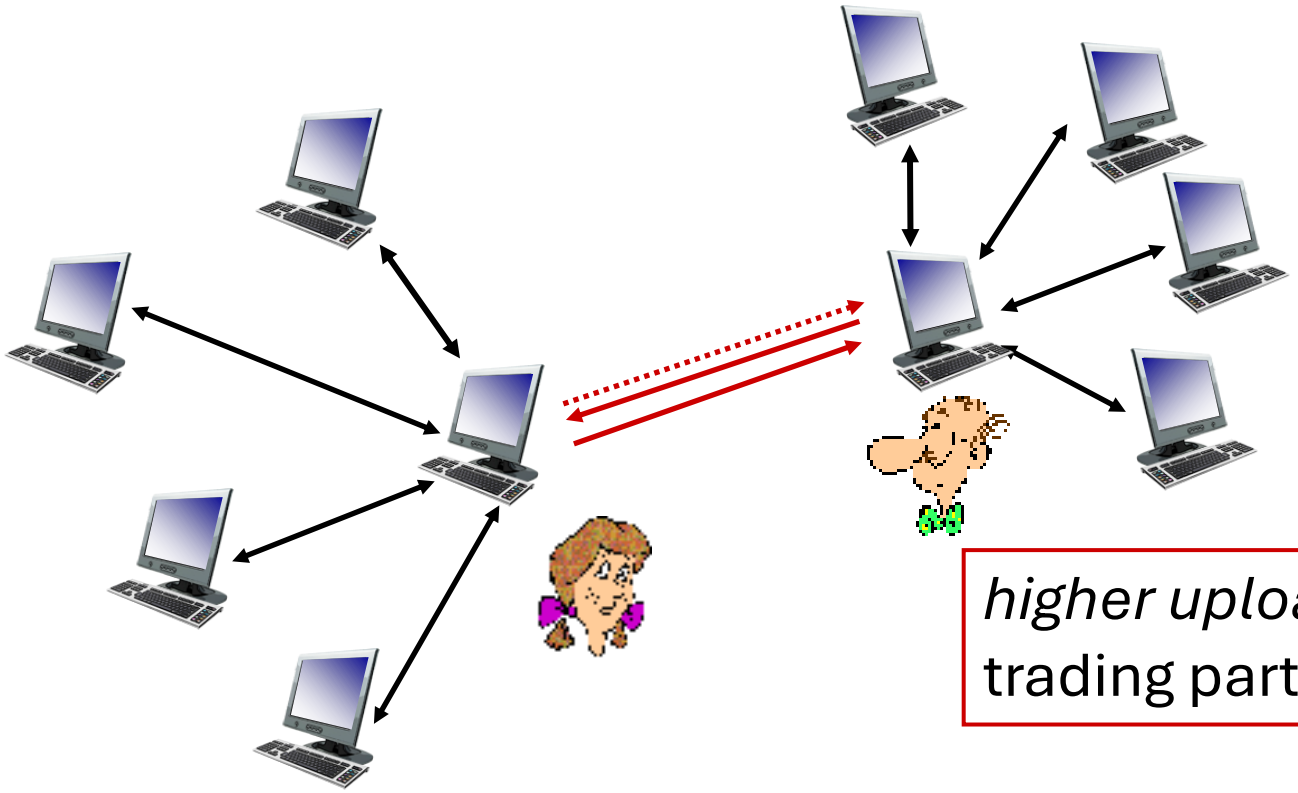
- Alice sends chunks to those four peers currently sending her chunks *at highest rate*
  - other peers are choked by Alice (do not receive chunks from her)
  - re-evaluate top 4 every 10 secs
- every 30 secs: randomly select another peer, starts sending chunks
  - "optimistically unchoke" this peer
  - newly chosen peer may join top 4

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 !

Application layer: overview

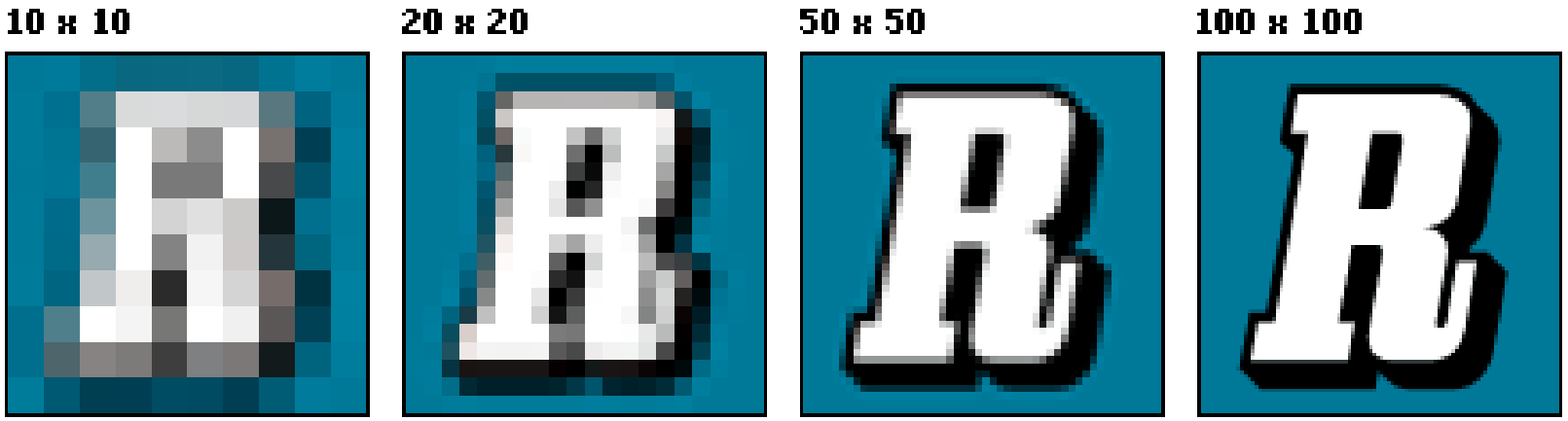
- Principles of network applications
- socket programming with UDP and TCP
- Web and HTTP
- E-mail, SMTP, IMAP
- The Domain Name System DNS
- P2P applications
- video streaming and content distribution networks

Video Streaming and CDNs: context

- stream video traffic: major consumer of Internet bandwidth
  - Netflix, YouTube, Amazon Prime: 80% of residential ISP traffic (2020)
- *challenge:* scale - how to reach ~1B users?
- *challenge:* heterogeneity
  - different users have different capabilities (e.g., wired versus mobile; bandwidth rich versus bandwidth poor)
- *solution:* distributed, application-level infrastructure

Multimedia: video

- video: sequence of images displayed at constant rate
  - e.g., 24 images/sec
- digital image: array of pixels
  - each pixel represented by bits



8000x5000

Multimedia: video coding

- Why we need video coding?

8 MByte = 64 Mbit

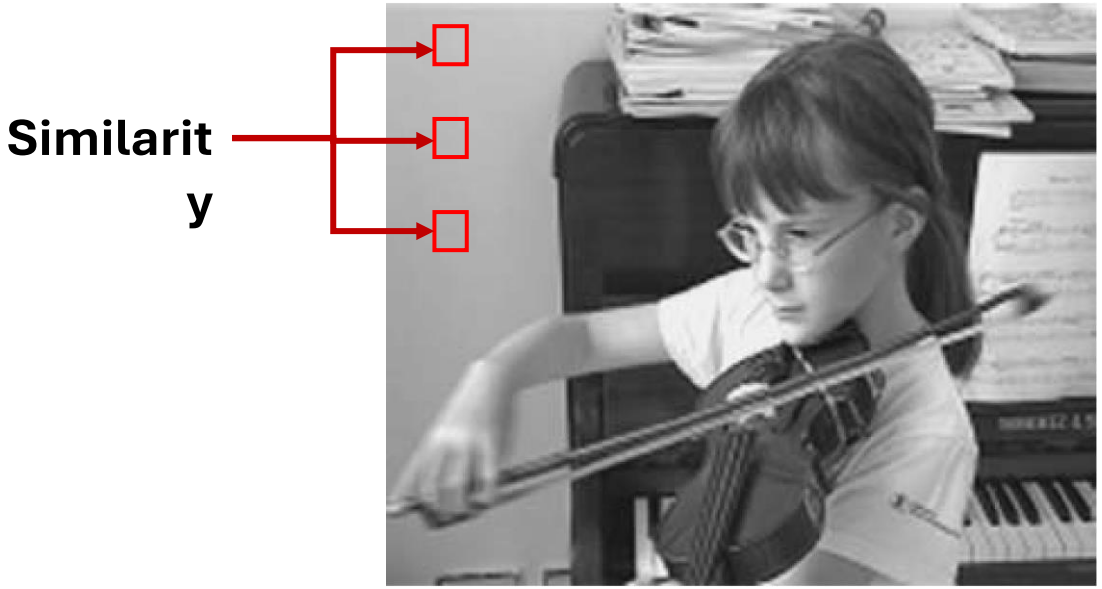
Example: a video with 30 fps

30 images per second

Data Rate: $30 \times 60Mbit = 1800Mbps = 1.8Gbps$

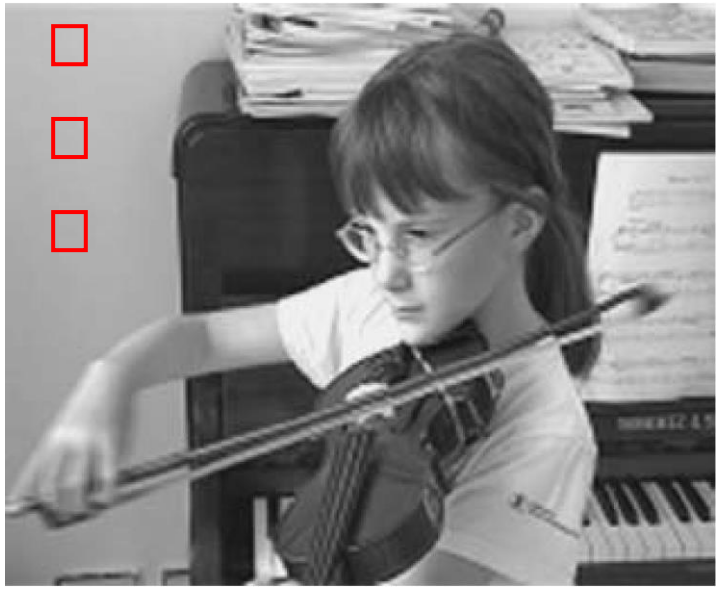
Multimedia: video coding

- coding: leveraging redundancy *within* and *between* images to decrease # bits used to encode image
  - spatial (within image)



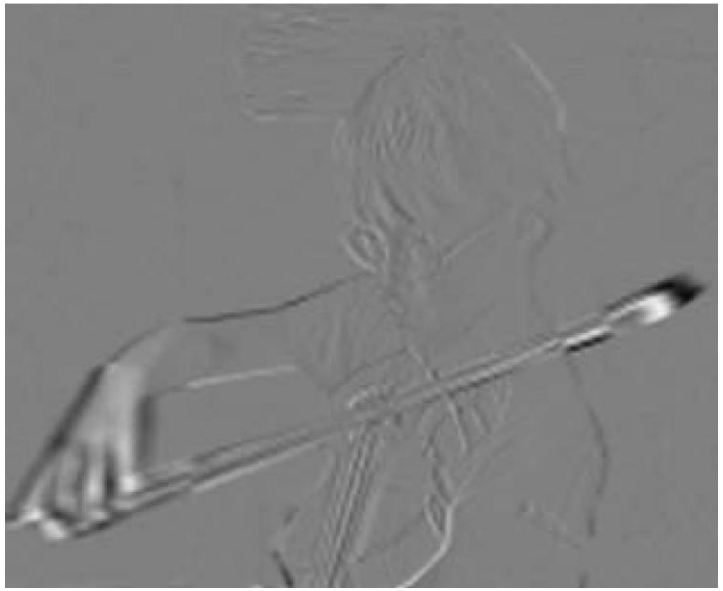
Multimedia: video coding

- coding: leveraging redundancy *within* and *between* images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)



Frame i

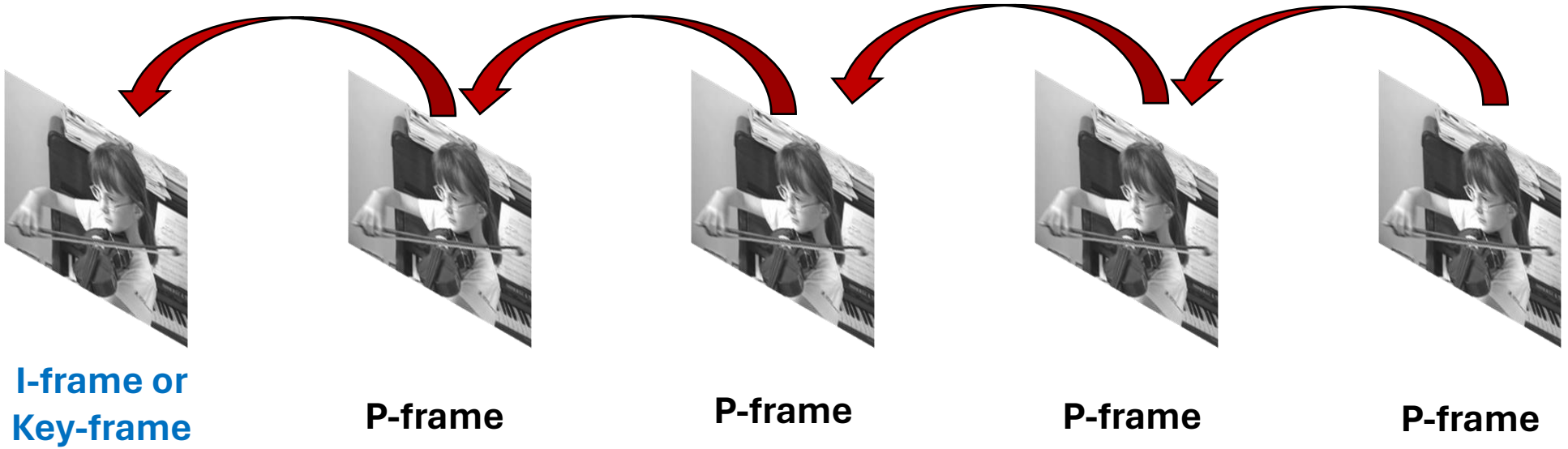
Frame i+1



Difference between frames

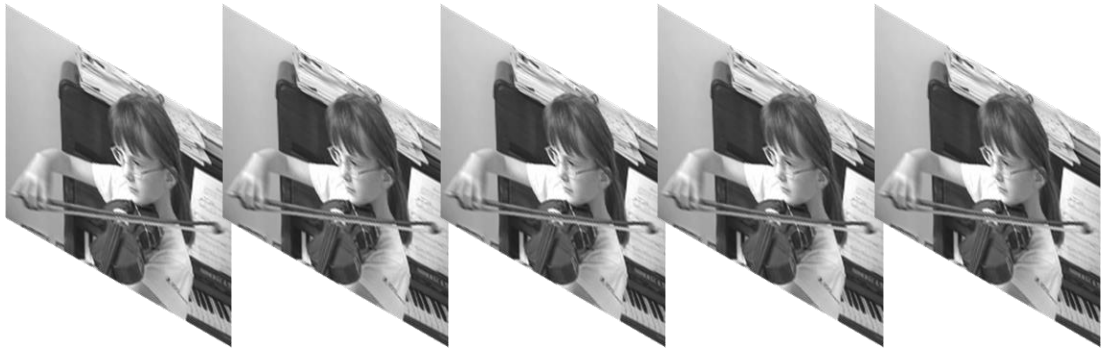
Multimedia: video coding

- coding: leveraging redundancy *within* and *between* images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)

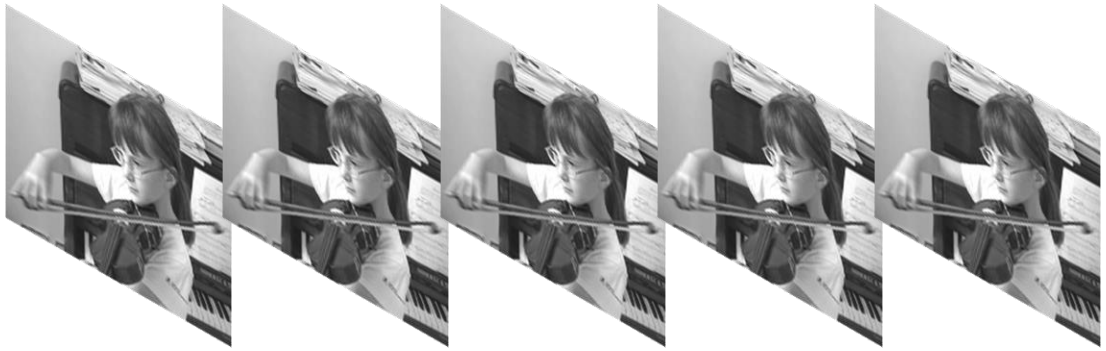


Multimedia: video coding

- coding: leveraging redundancy *within* and *between* images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)



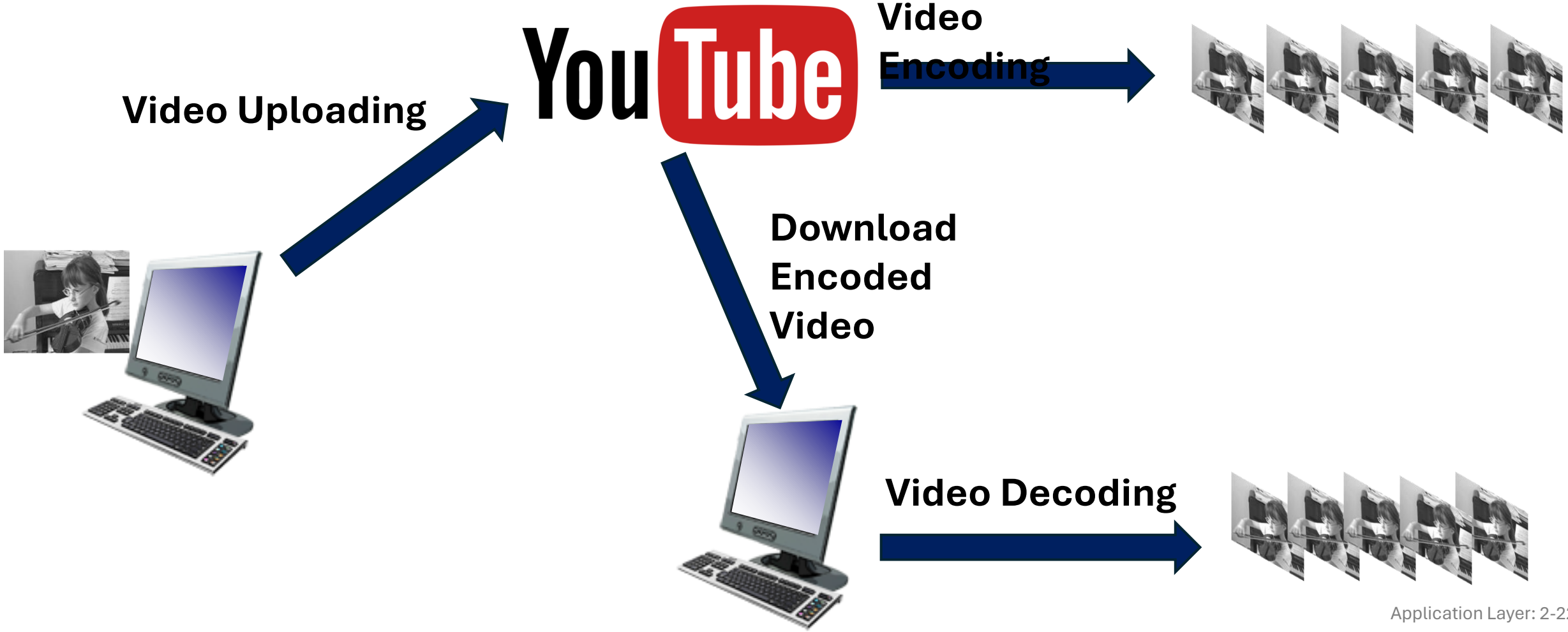
I-frame or Key-frame P-frame P-frame P-frame P-frame



I-frame or Key-frame P-frame P-frame P-frame P-frame

Video Streaming Applications

Video Streaming Applications: Encoding



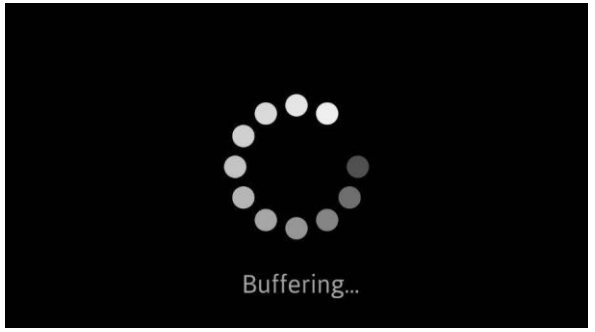
Video Streaming Applications: QoE (Quality of User Experience)

Video Start Time

The time between your click of the video and the playing of the video

Video Quality

Rebuffering Event

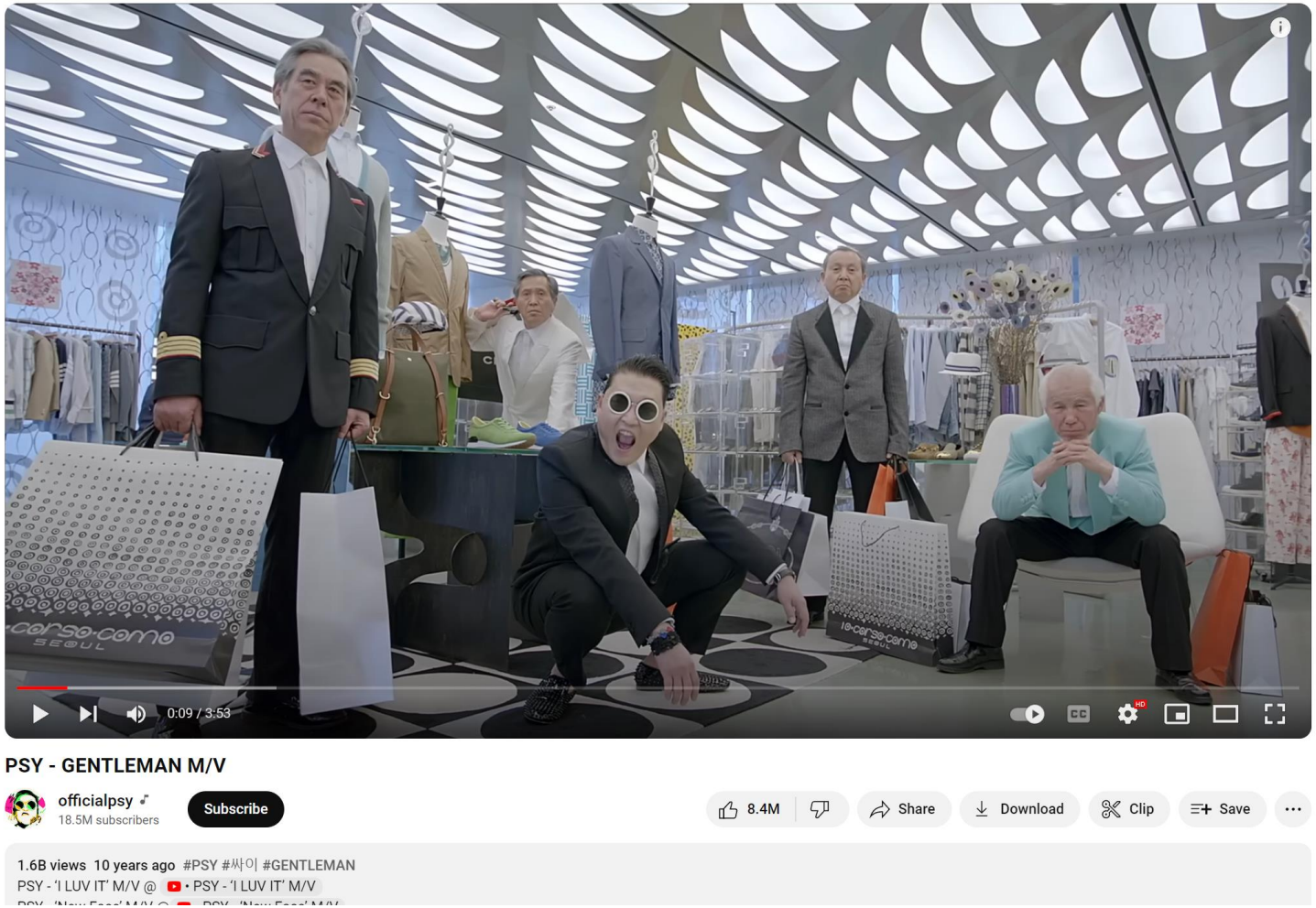


Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP

Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP



Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP

PSY - GENTLEMAN M/V

officialpsy
18.5M subscribers
Subscribe

8.4M Share Download Clip Save

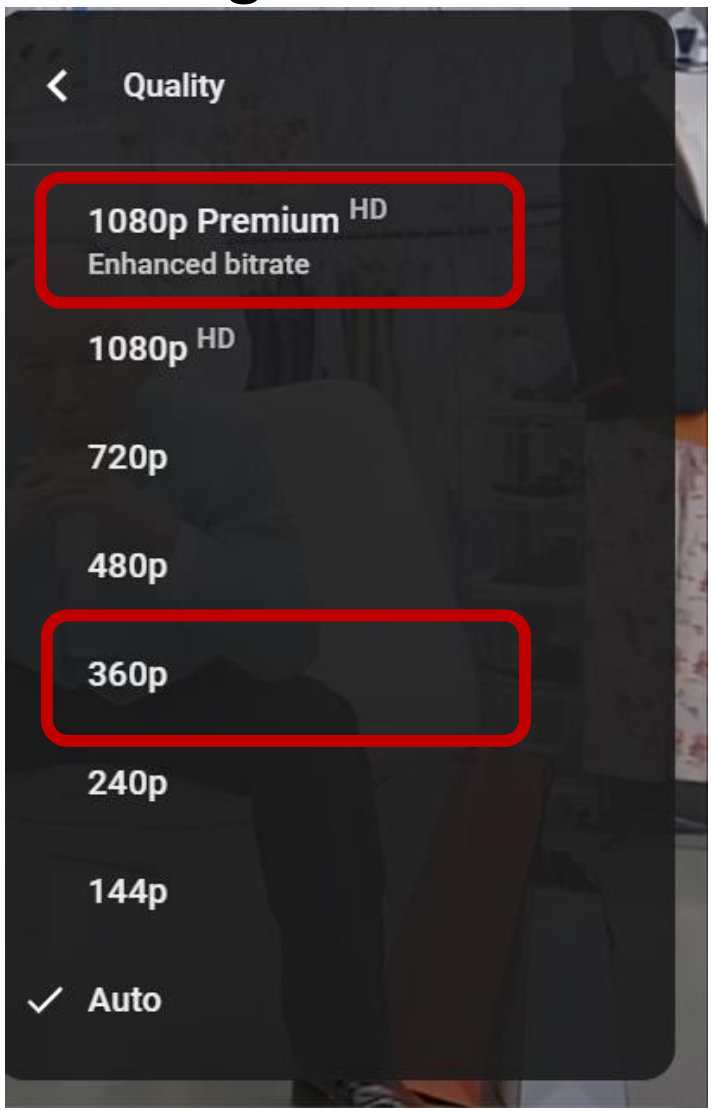
Streaming multimedia: DASH

*D*ynamic, *A*daptive *S*treaming over *H*TTP

PSY - GENTLEMAN M/V

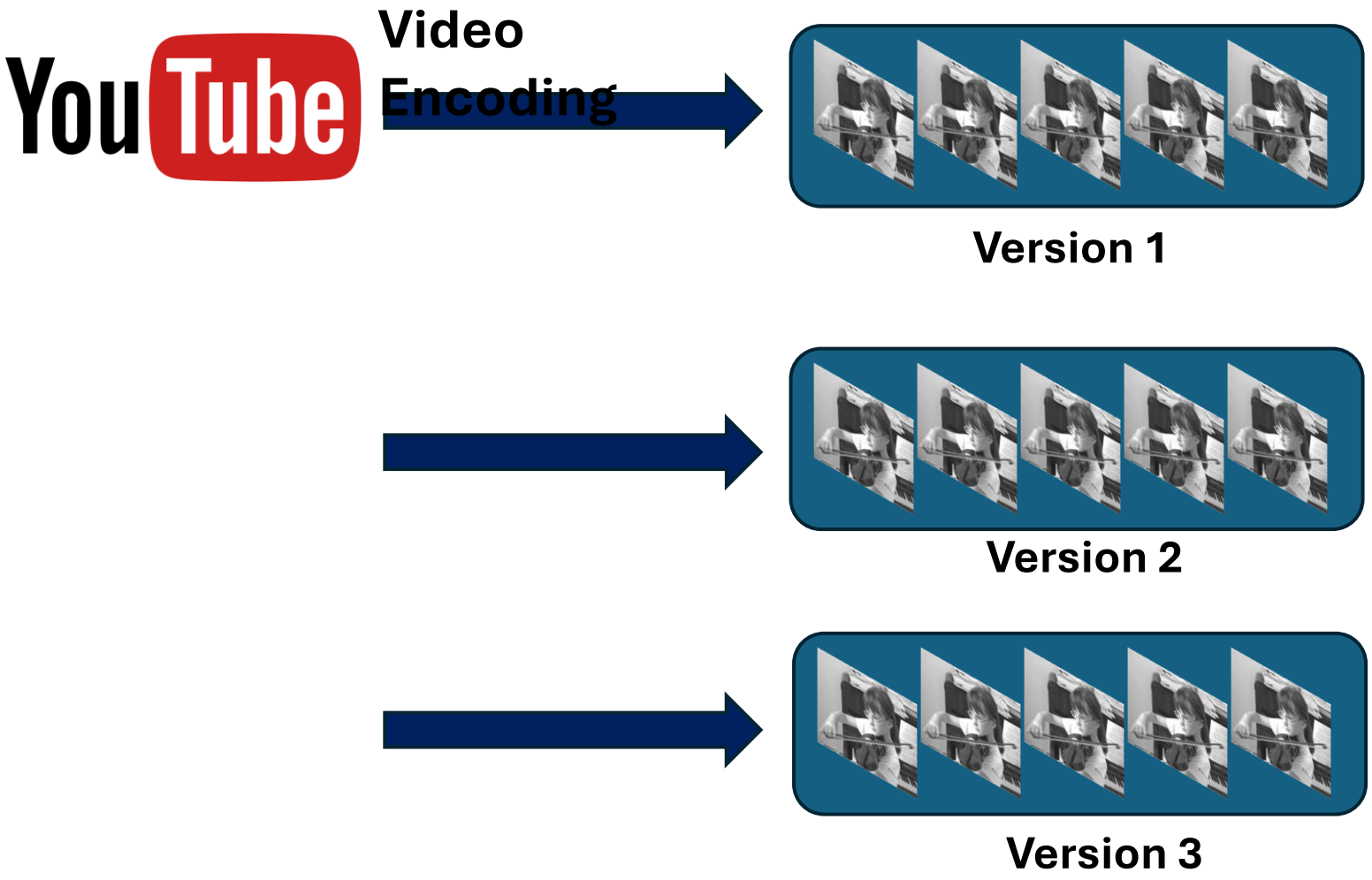
officialpsy
18.5M subscribers
Subscribe

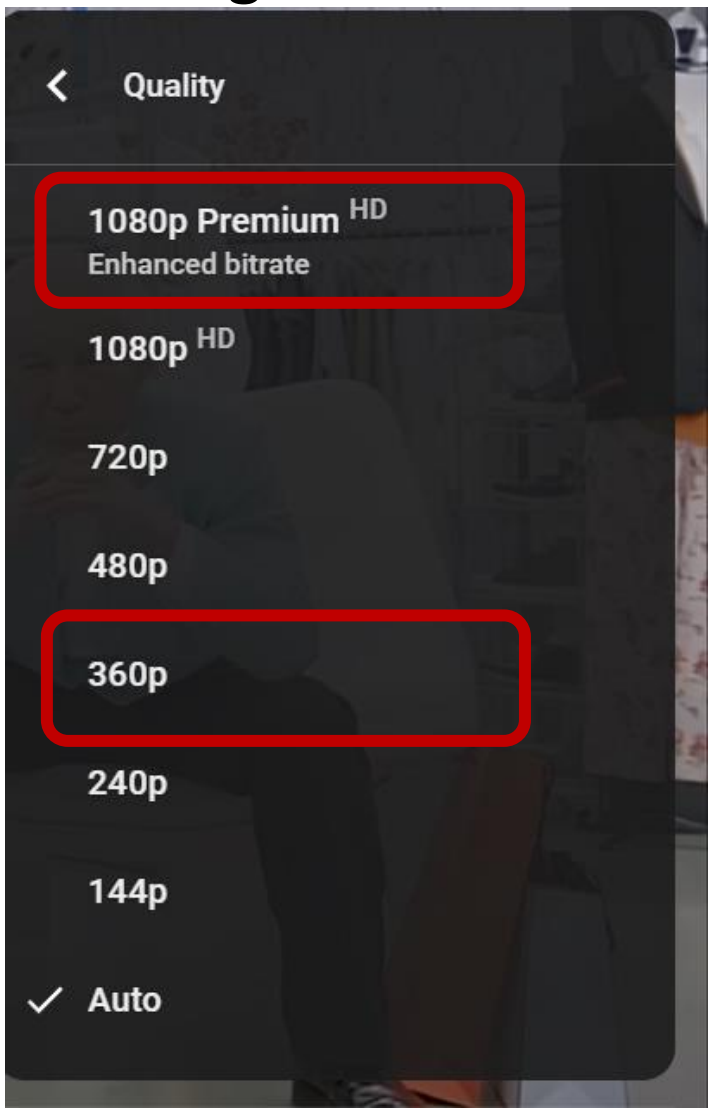
8.4M Share Download Clip Save



Streaming multimedia: DASH

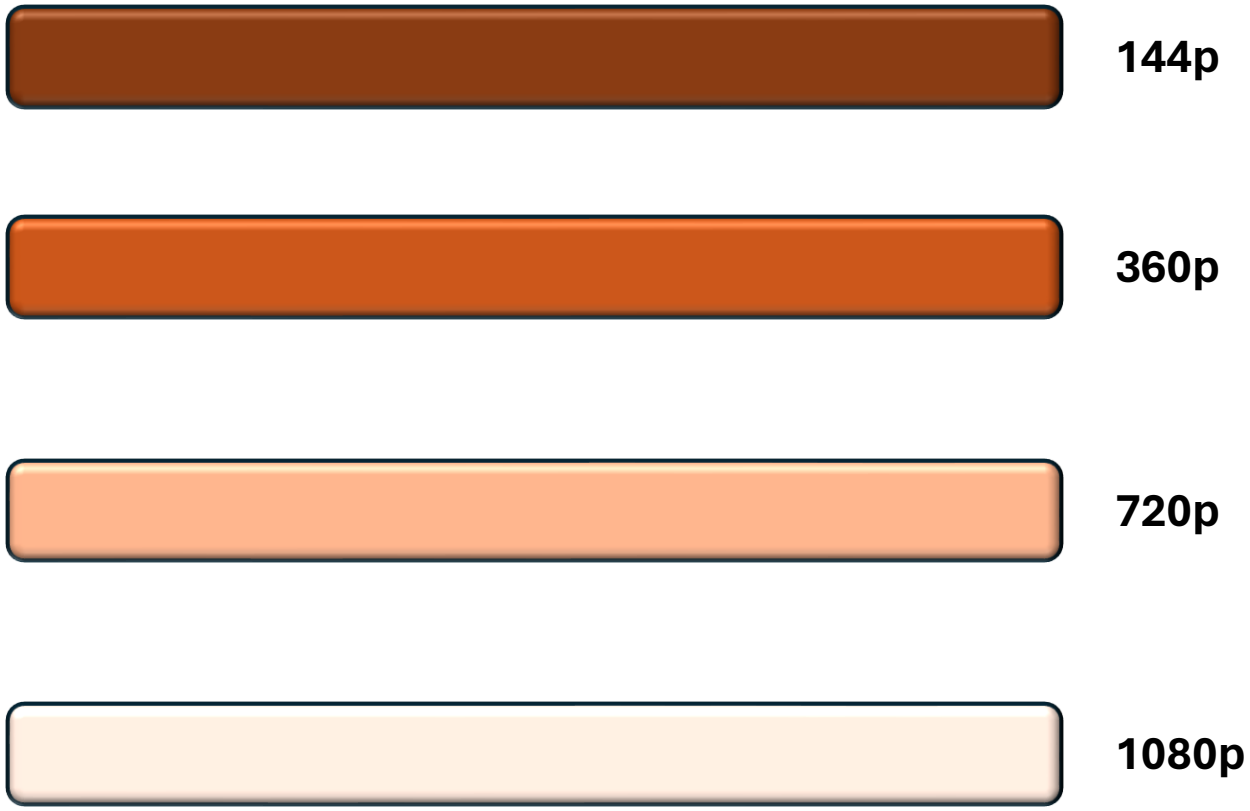
*D*ynamic, *A*daptive *S*treaming over *H*TTP

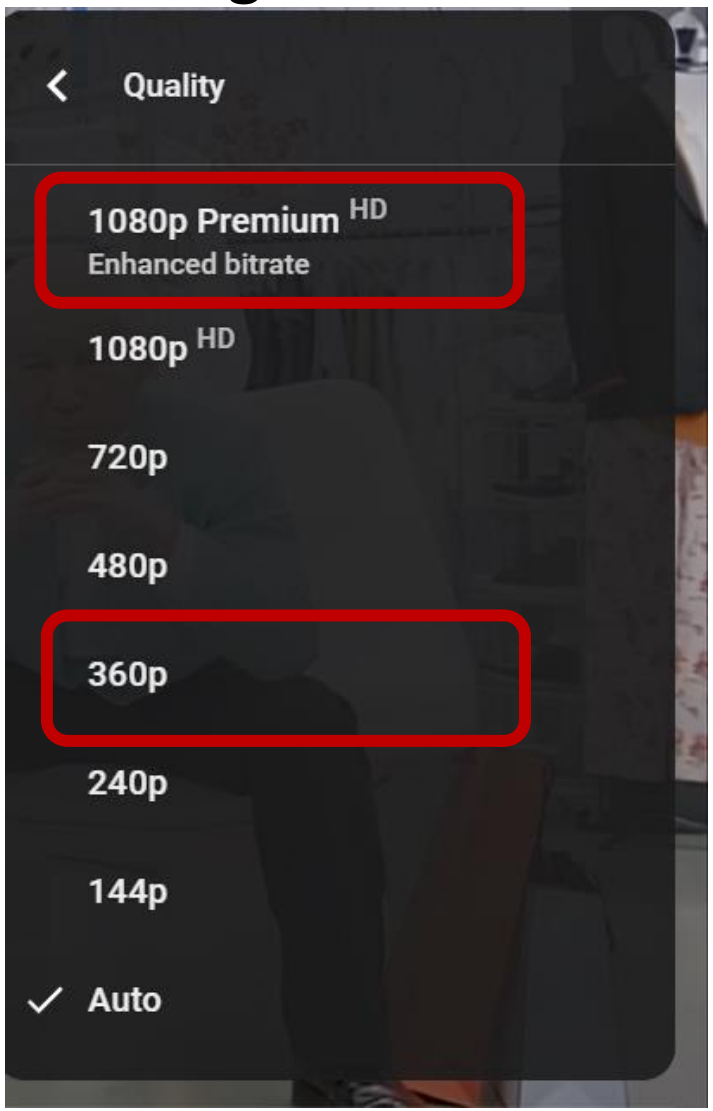




Streaming multimedia: DASH

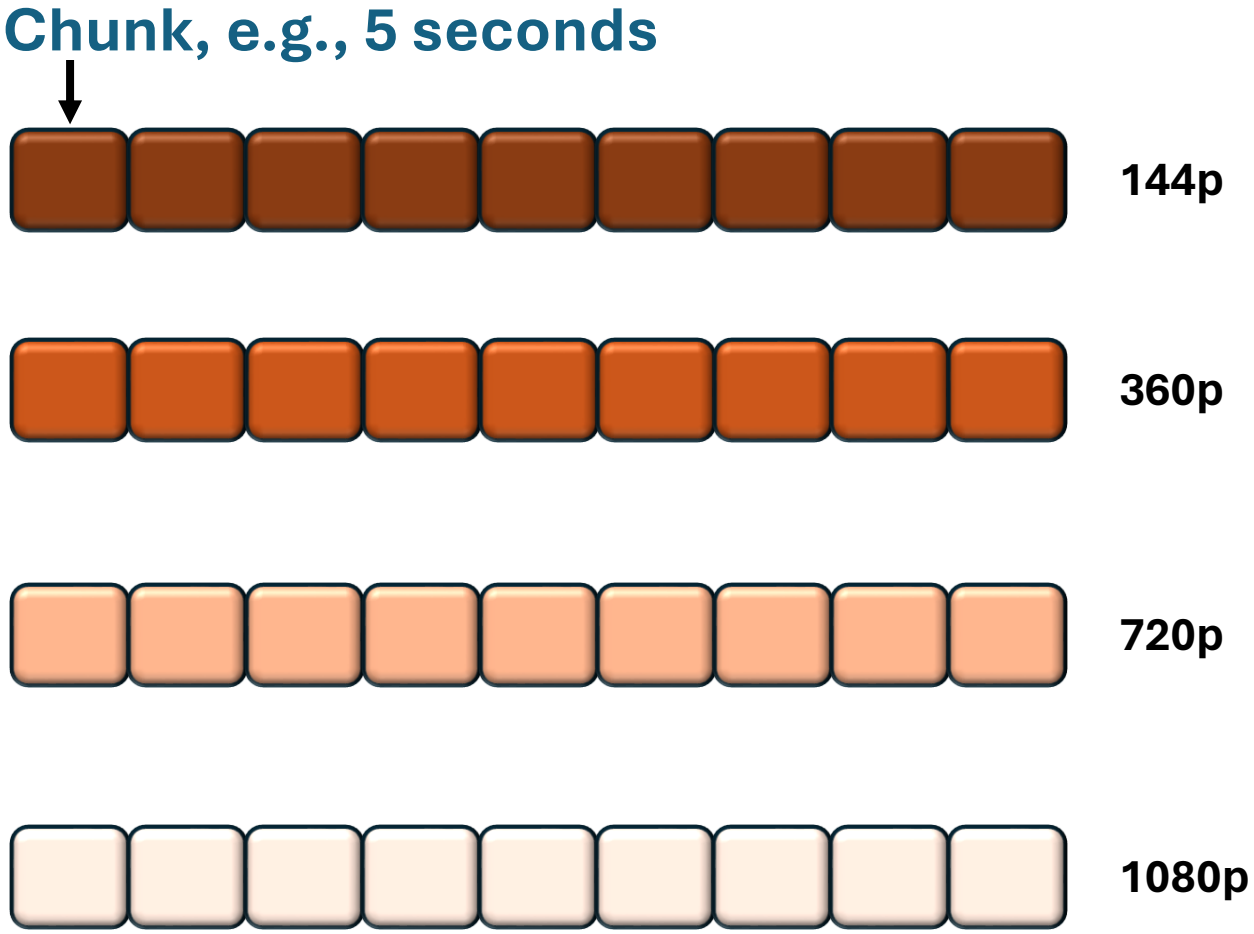
*D*ynamic, *A*daptive *S*treaming over *H*TTP

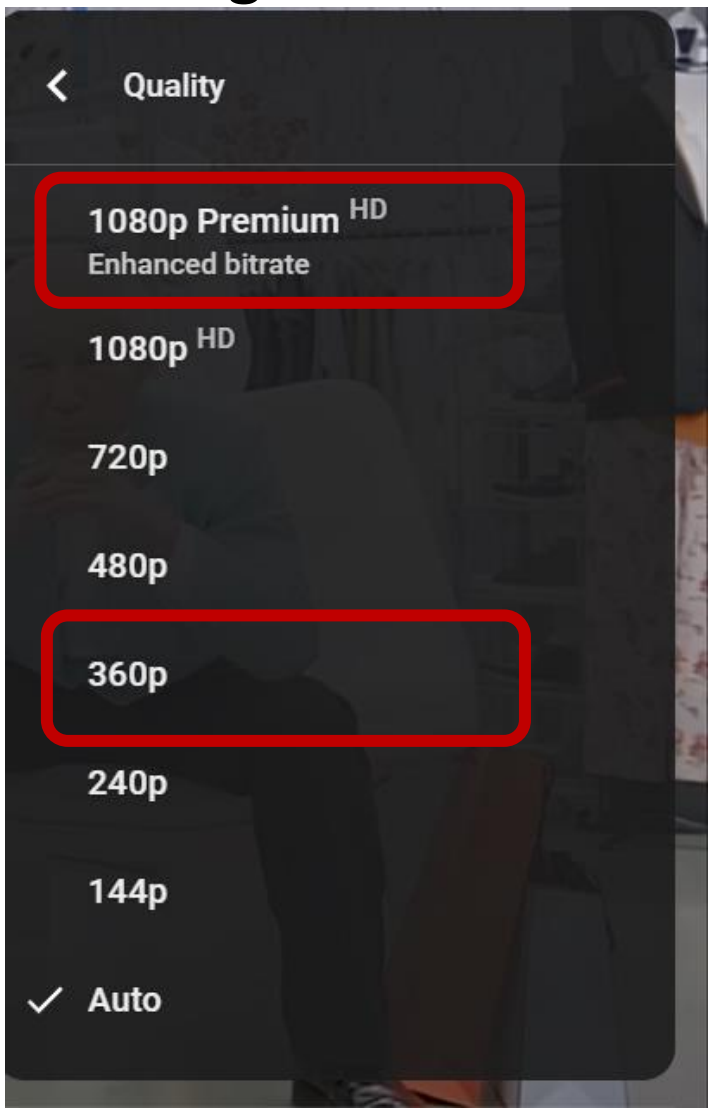




Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP



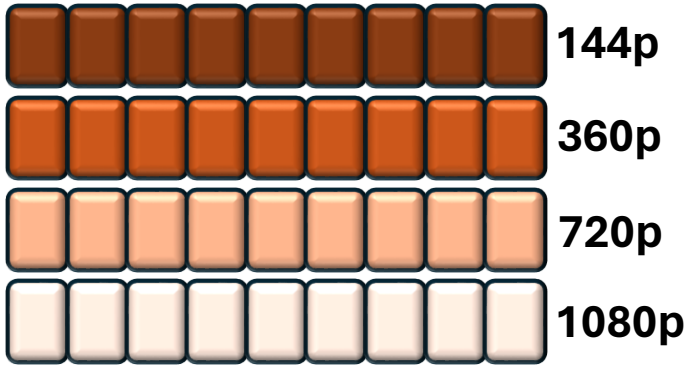


Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP

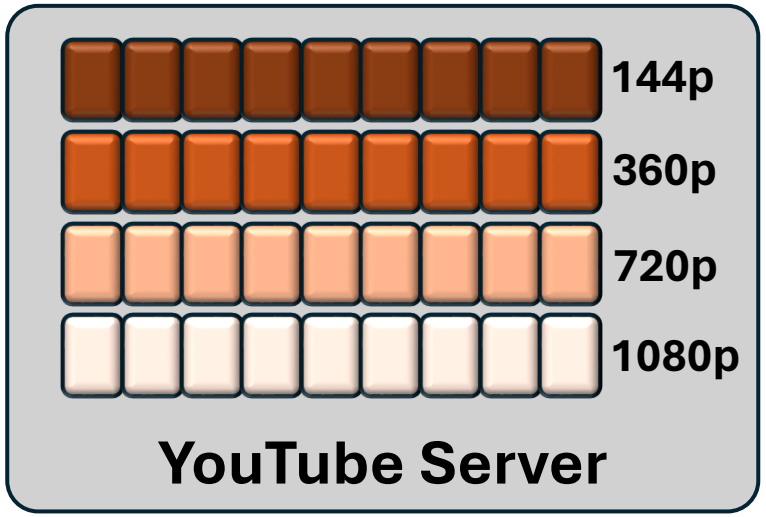
server:

- divides video file into multiple chunks
- each chunk encoded at multiple different rates
- different rate encodings stored in different files
- files replicated in various CDN nodes
- *manifest file:* provides URLs for different chunks



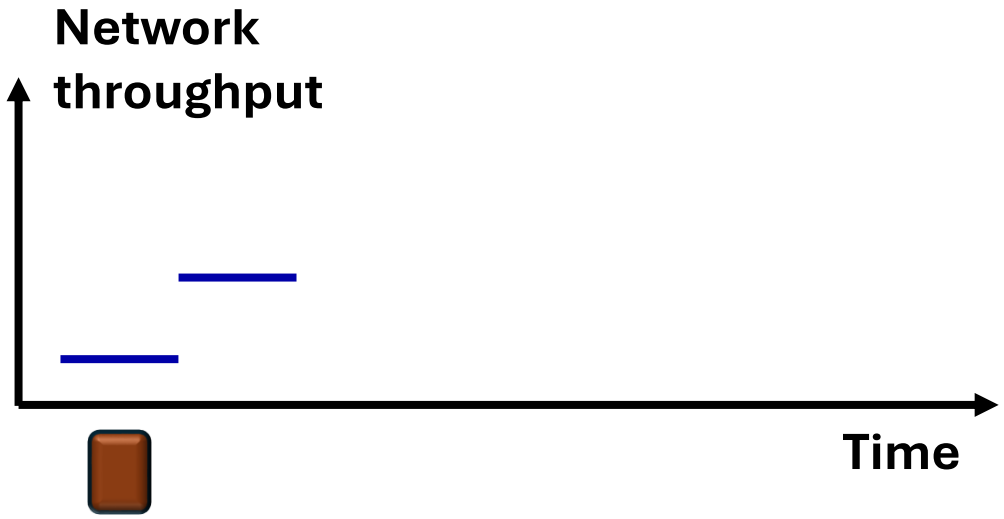
Streaming multimedia: DASH

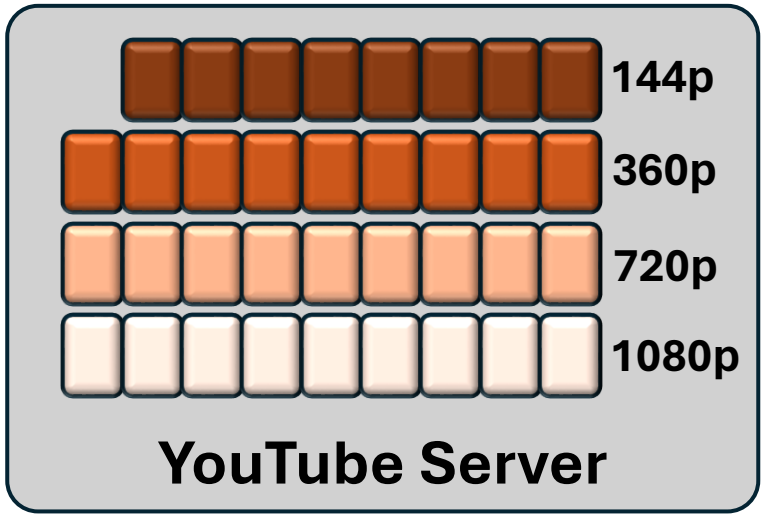
*D*ynamic, *A*daptive
*S*treaming over *H*TTP



Streaming multimedia: DASH

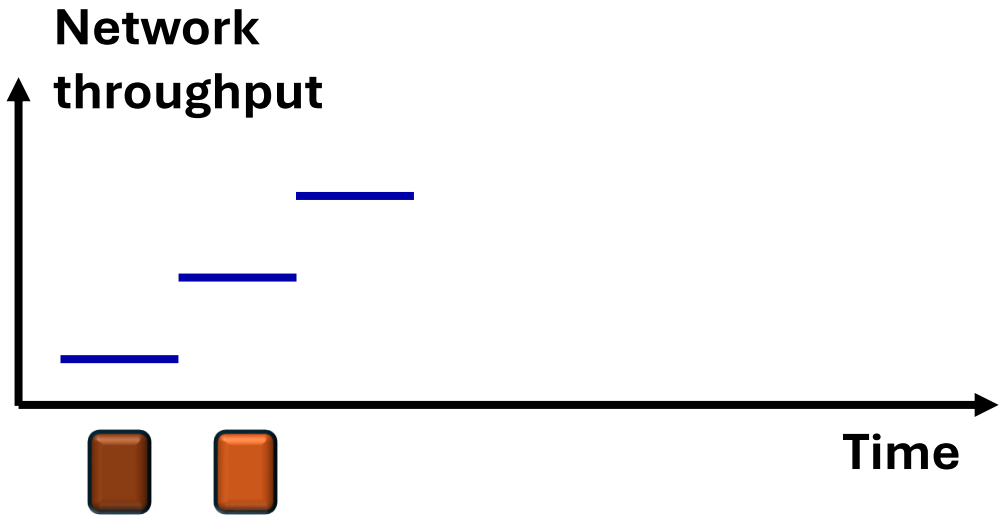
*D*ynamic, *A*daptive
*S*treaming over *H*TTP

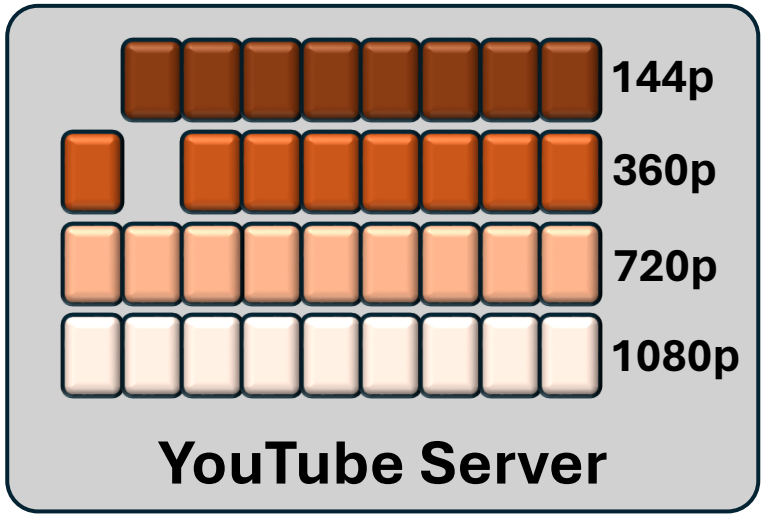




Streaming multimedia: DASH

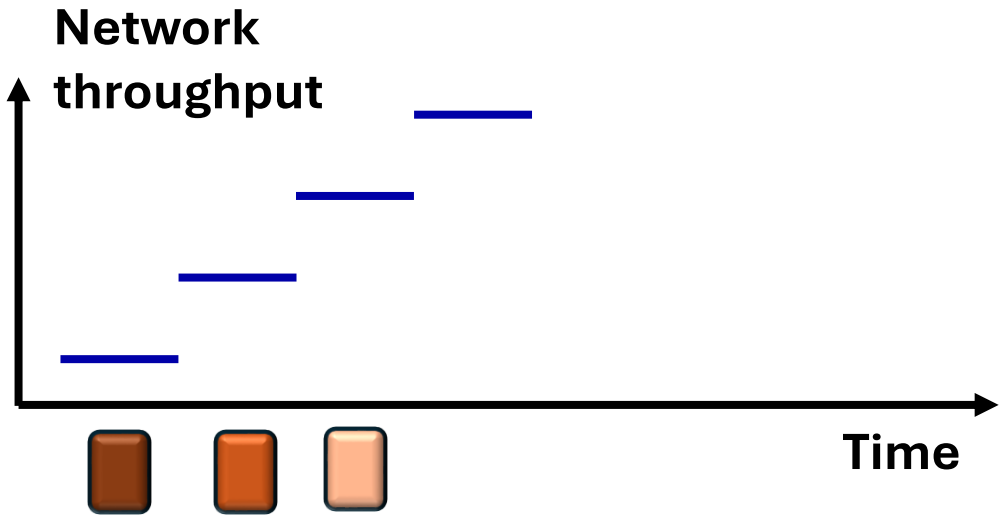
*D*ynamic, *A*daptive
*S*treaming over *H*TTP

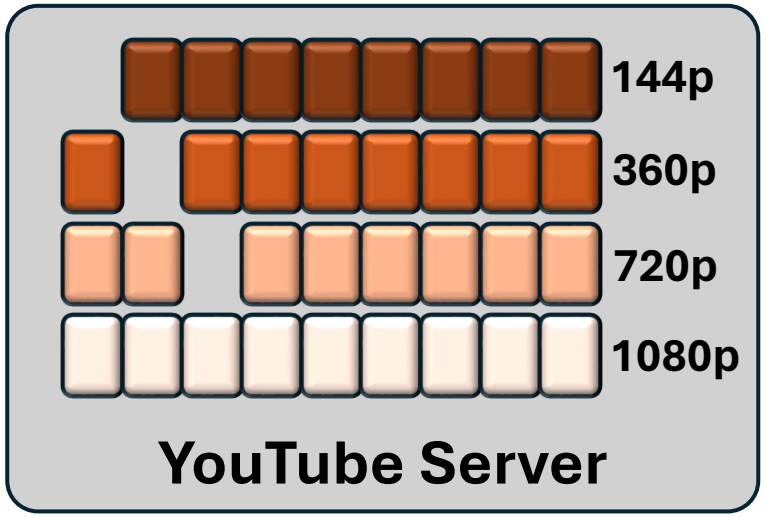




Streaming multimedia: DASH

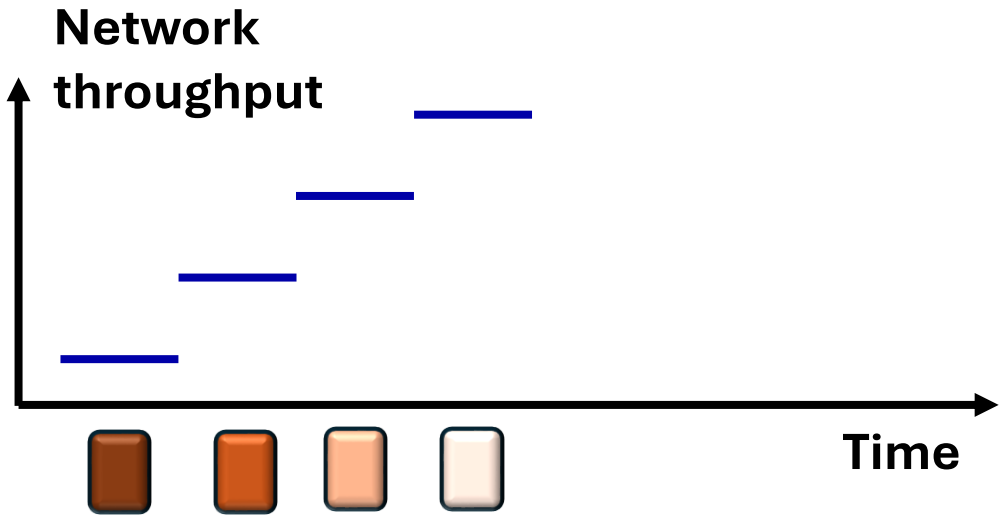
*D*ynamic, *A*daptive
*S*treaming over *H*TTP

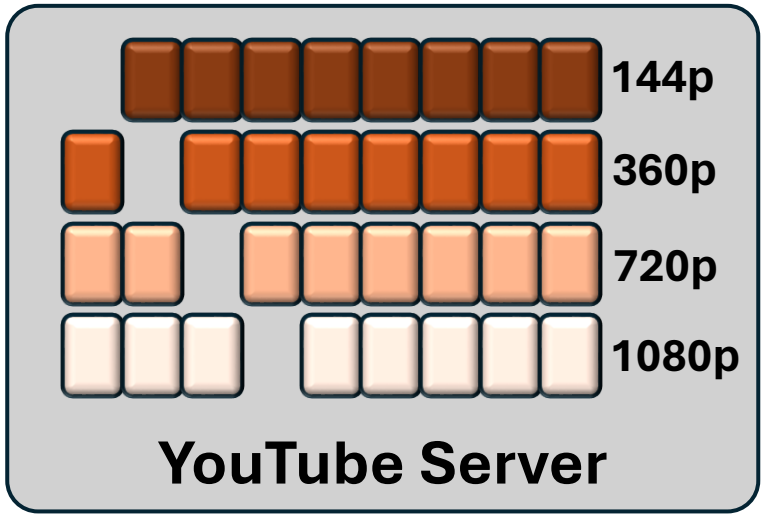




Streaming multimedia: DASH

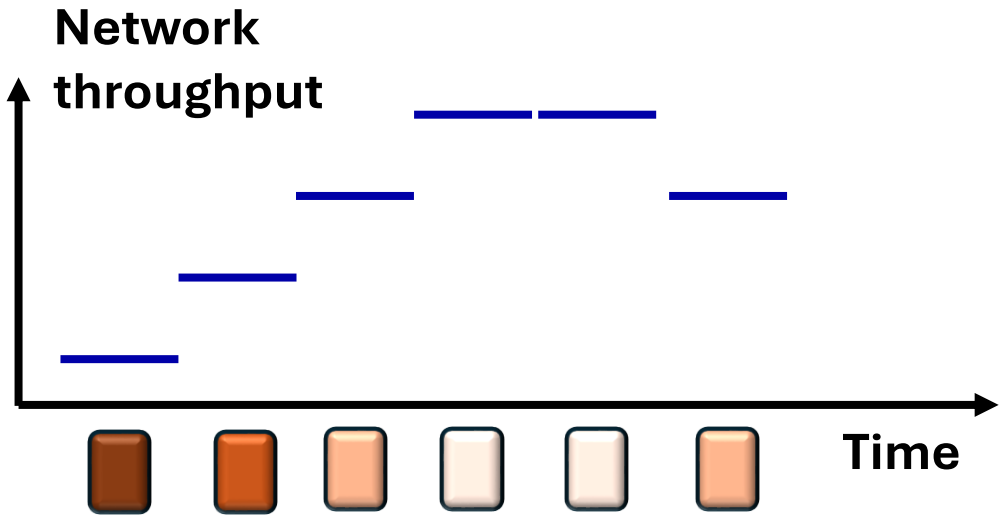
*D*ynamic, *A*daptive
*S*treaming over *H*TTP

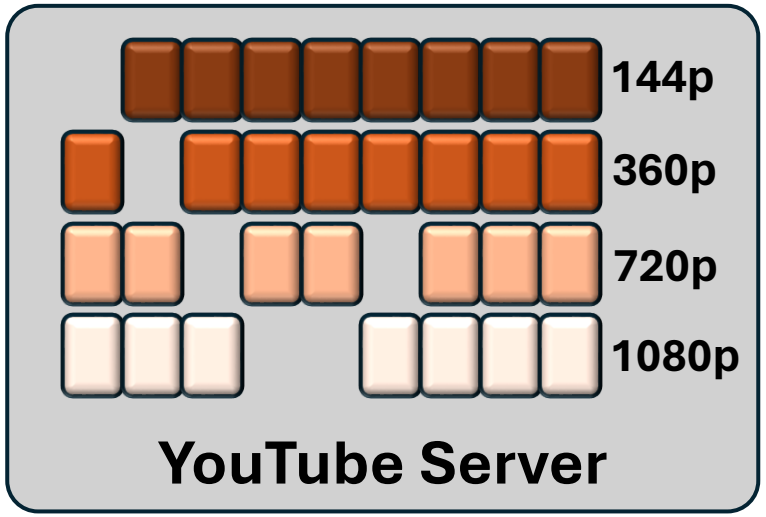




Streaming multimedia: DASH

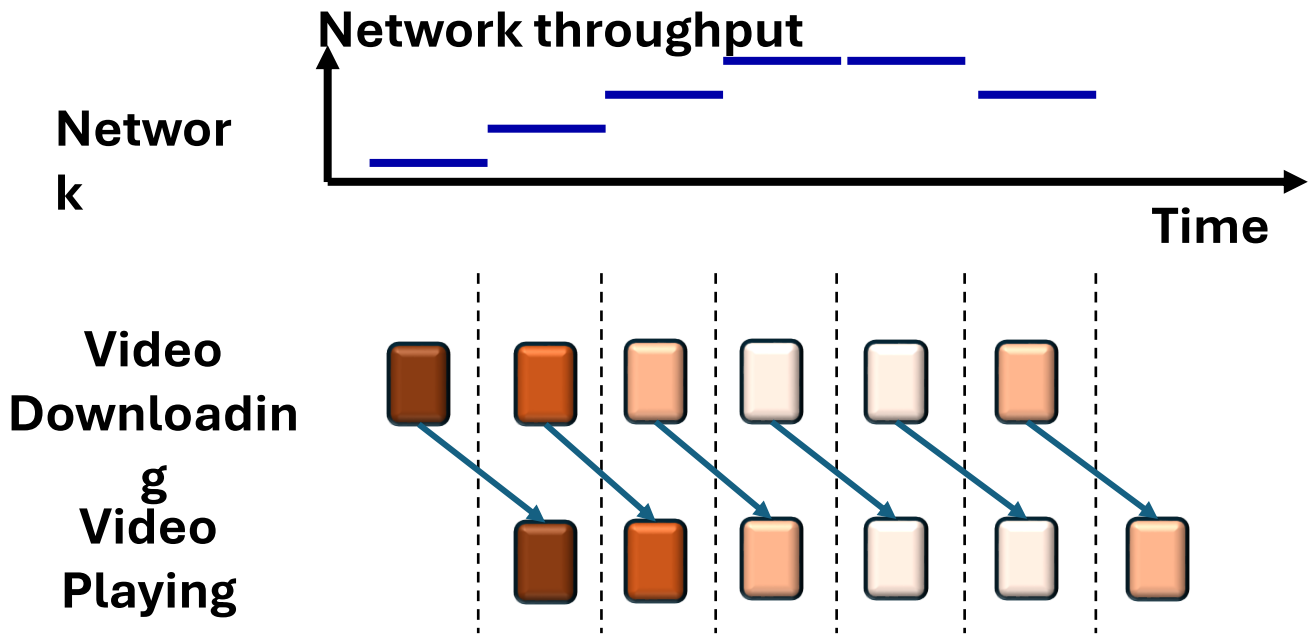
*D*ynamic, *A*daptive
*S*treaming over *H*TTP





Streaming multimedia: DASH

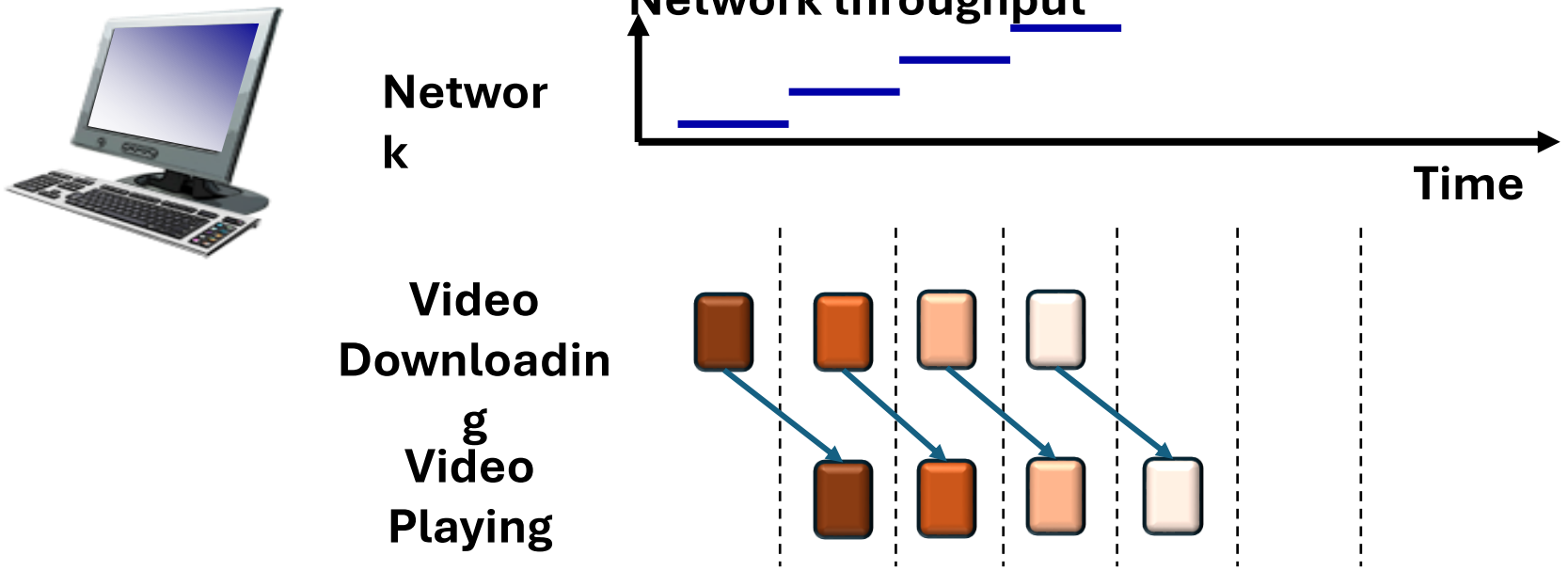
*D*ynamic, *A*daptive
*S*treaming over *H*TTP



Minimize Video Start Time

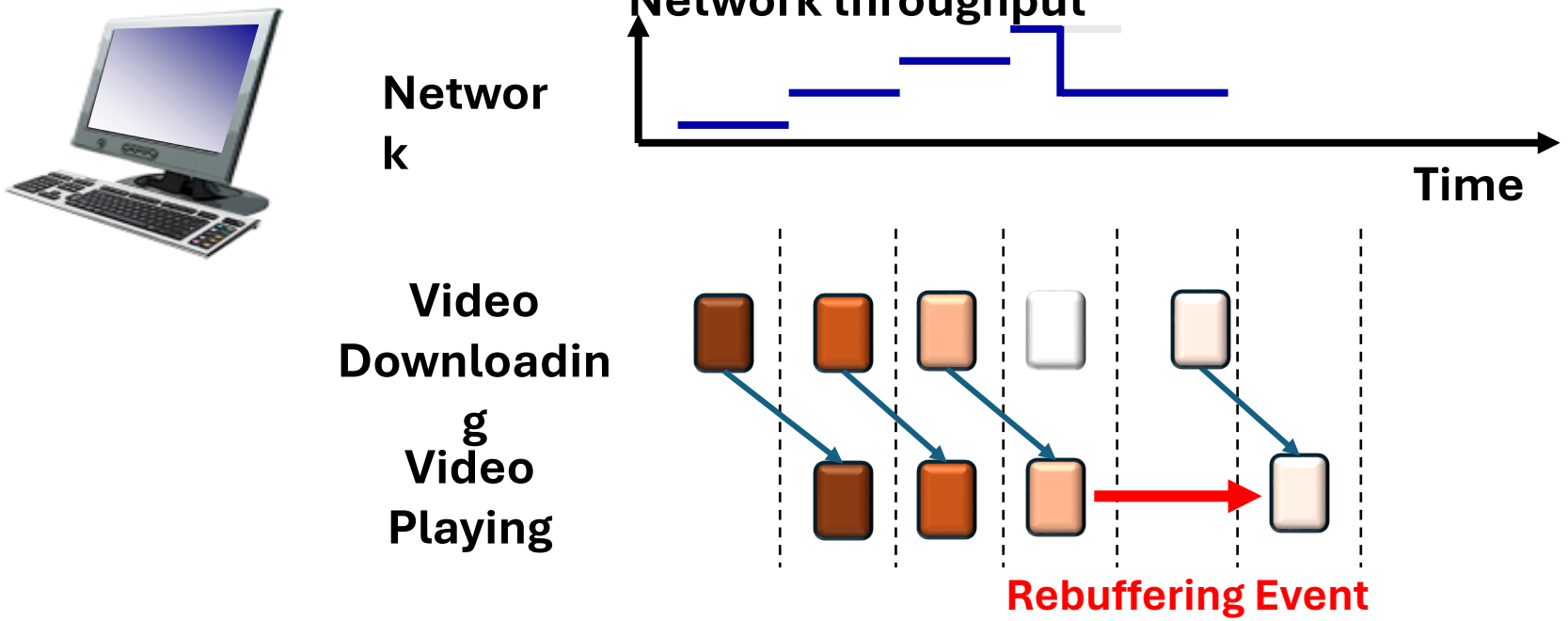
Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP



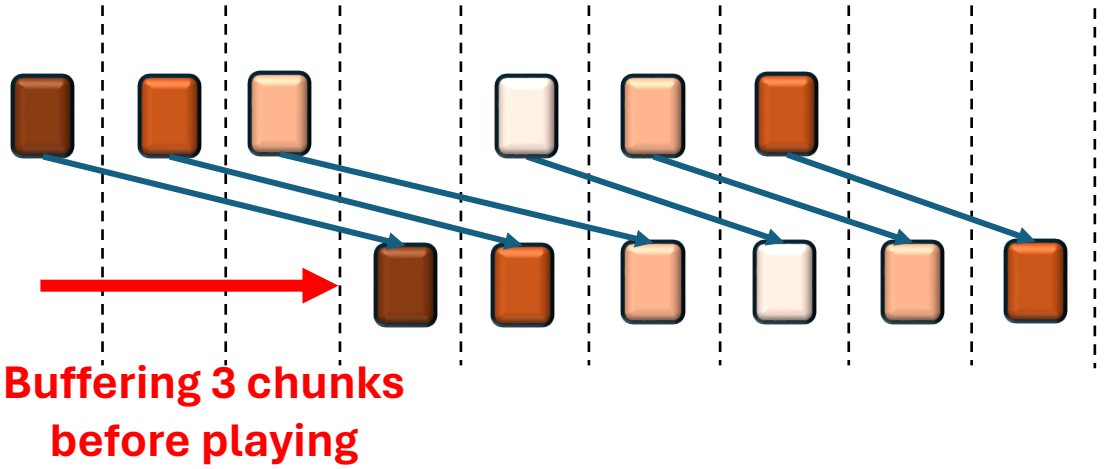
Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP



Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP

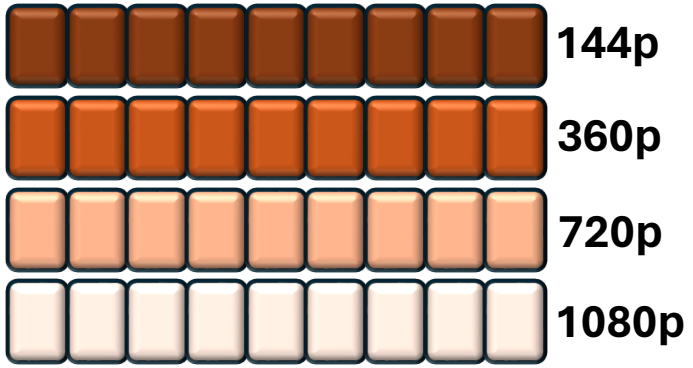


Streaming multimedia: DASH

*D*ynamic, *A*daptive
*S*treaming over *H*TTP

server:

- divides video file into multiple chunks
- each chunk encoded at multiple different rates
- different rate encodings stored in different files
- files replicated in various CDN nodes
- *manifest file:* provides URLs for different chunks

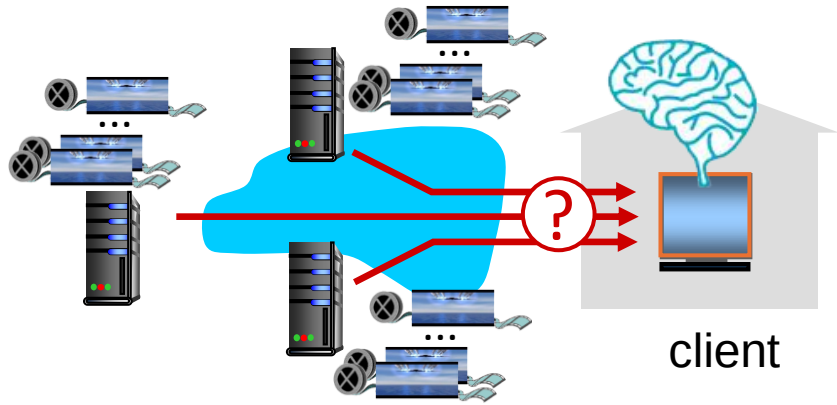


client:

- periodically estimates server-to-client bandwidth
- consulting manifest, requests one chunk at a time
  - chooses maximum coding rate sustainable given current bandwidth
  - can choose different coding rates at different points in time (depending on available bandwidth at time), and from different servers

Streaming multimedia: DASH

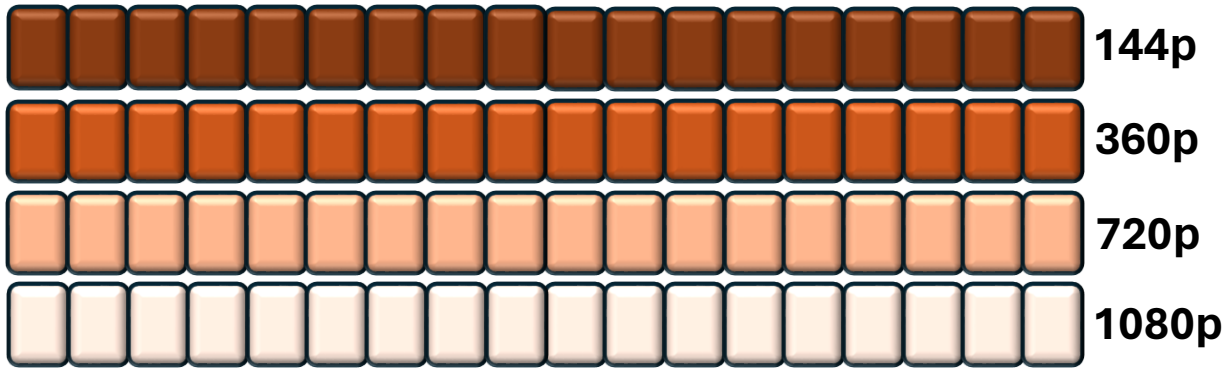
- *"intelligence"* at client: client determines
  - *when* to request chunk (so that buffer starvation, or overflow does not occur)
  - *what encoding rate* to request (higher quality when more bandwidth available)
  - *where* to request chunk (can request from URL server that is "close" to client or has high available bandwidth)



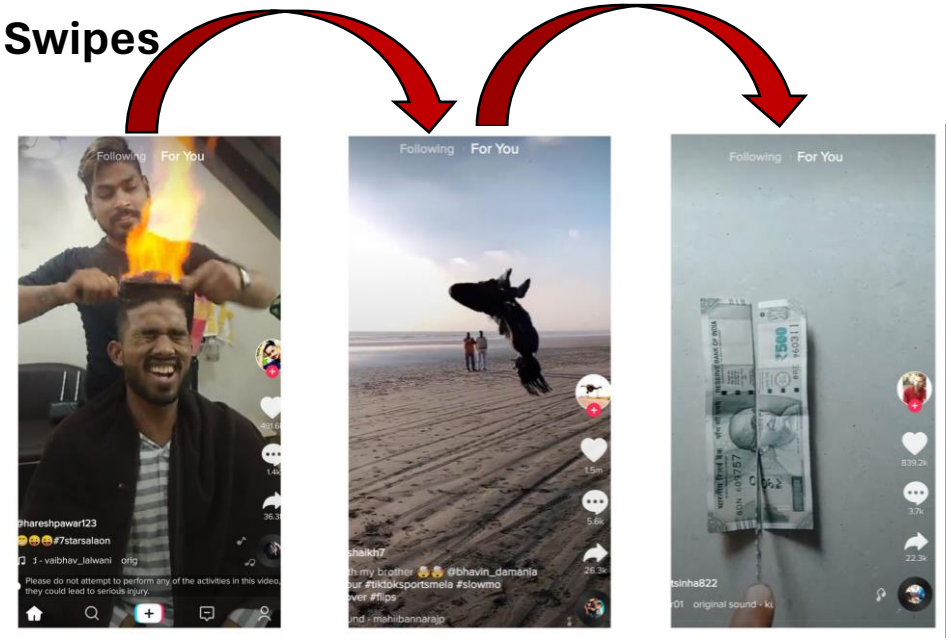
Streaming video = encoding + DASH + playout buffering

Video Streaming Applications

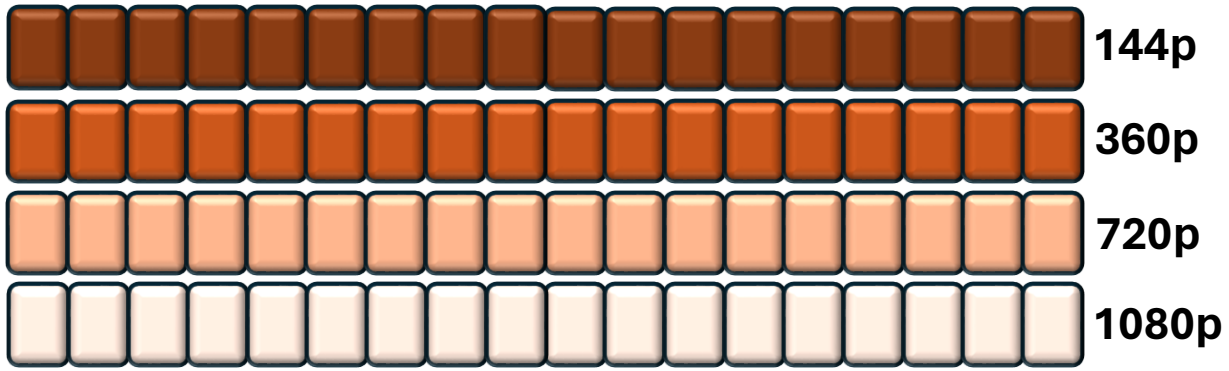
Video Streaming Applications



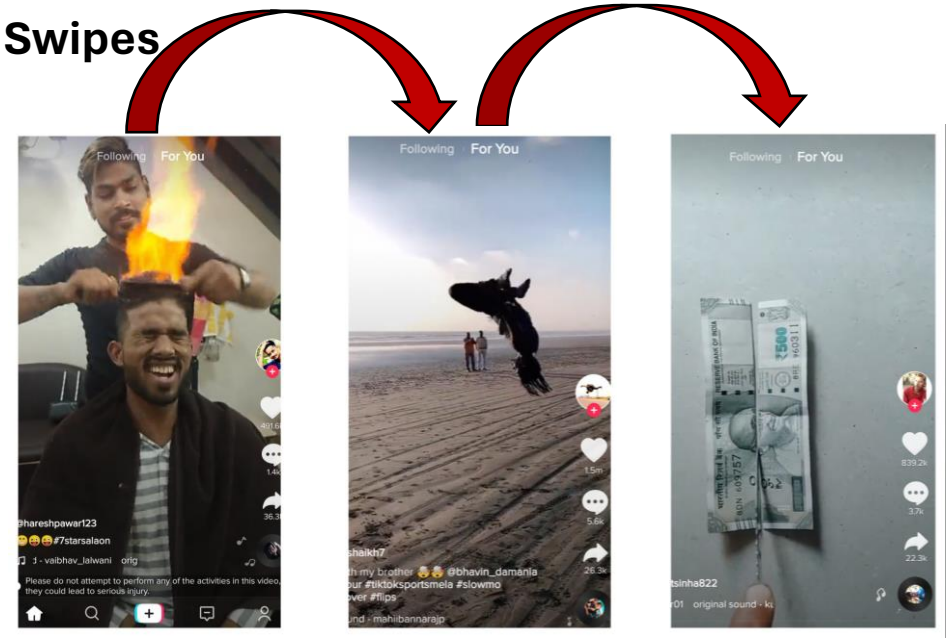
User Swipes



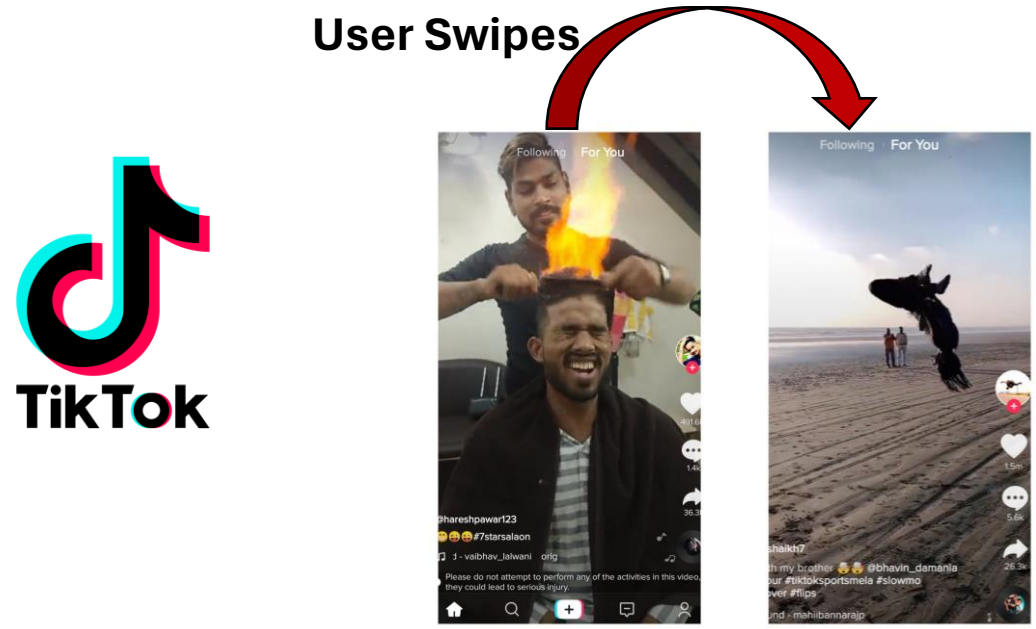
Video Streaming Applications



User Swipes

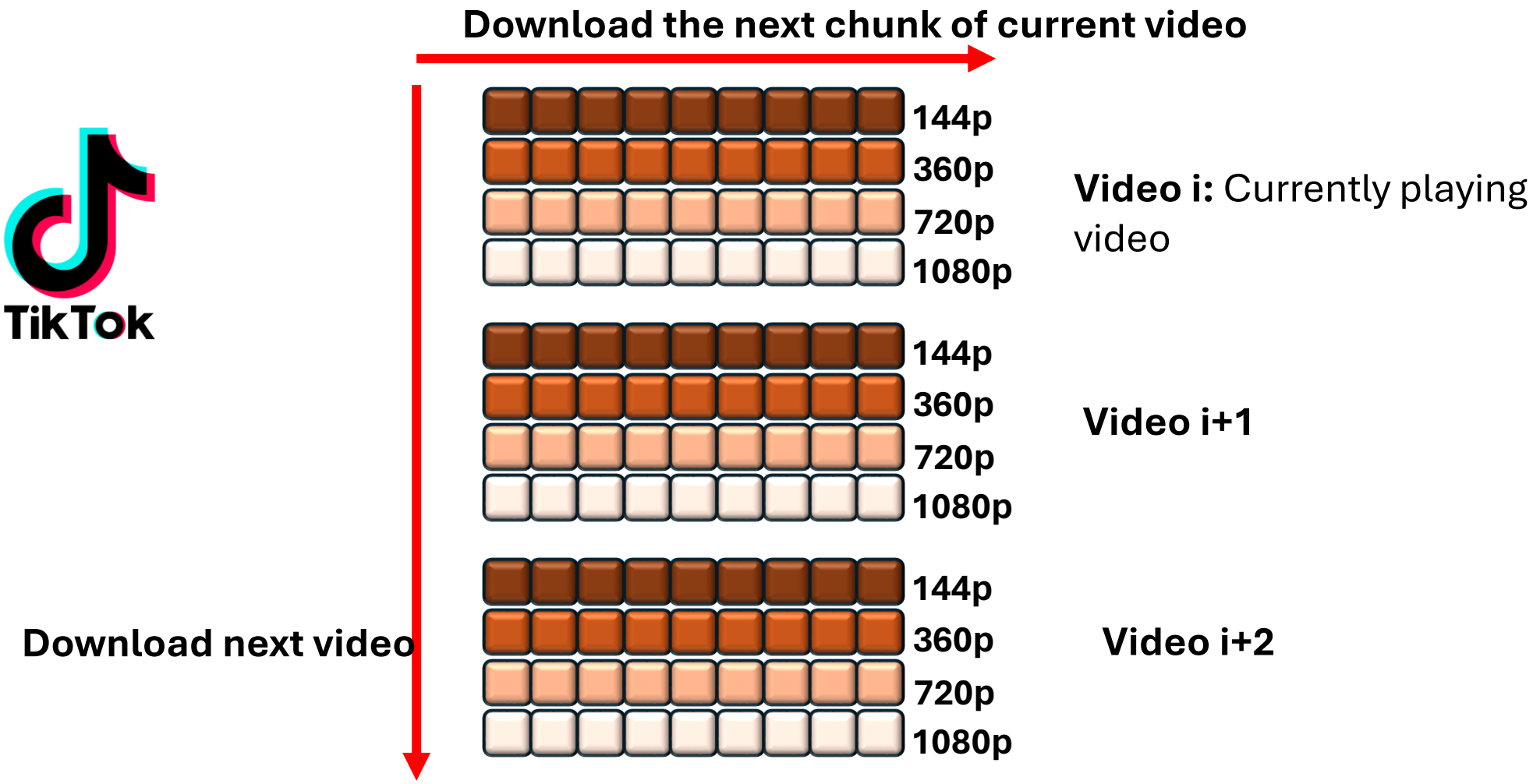


Video Streaming Applications



Is the next video ready for playing when the user swipes?

Video Streaming Applications



Content distribution networks (CDNs)

challenge: how to stream content (selected from millions of videos) to hundreds of thousands of *simultaneous* users?

- *option 1:* single, large "mega-server"
  - single point of failure
  - point of network congestion
  - long (and possibly congested) path to distant clients

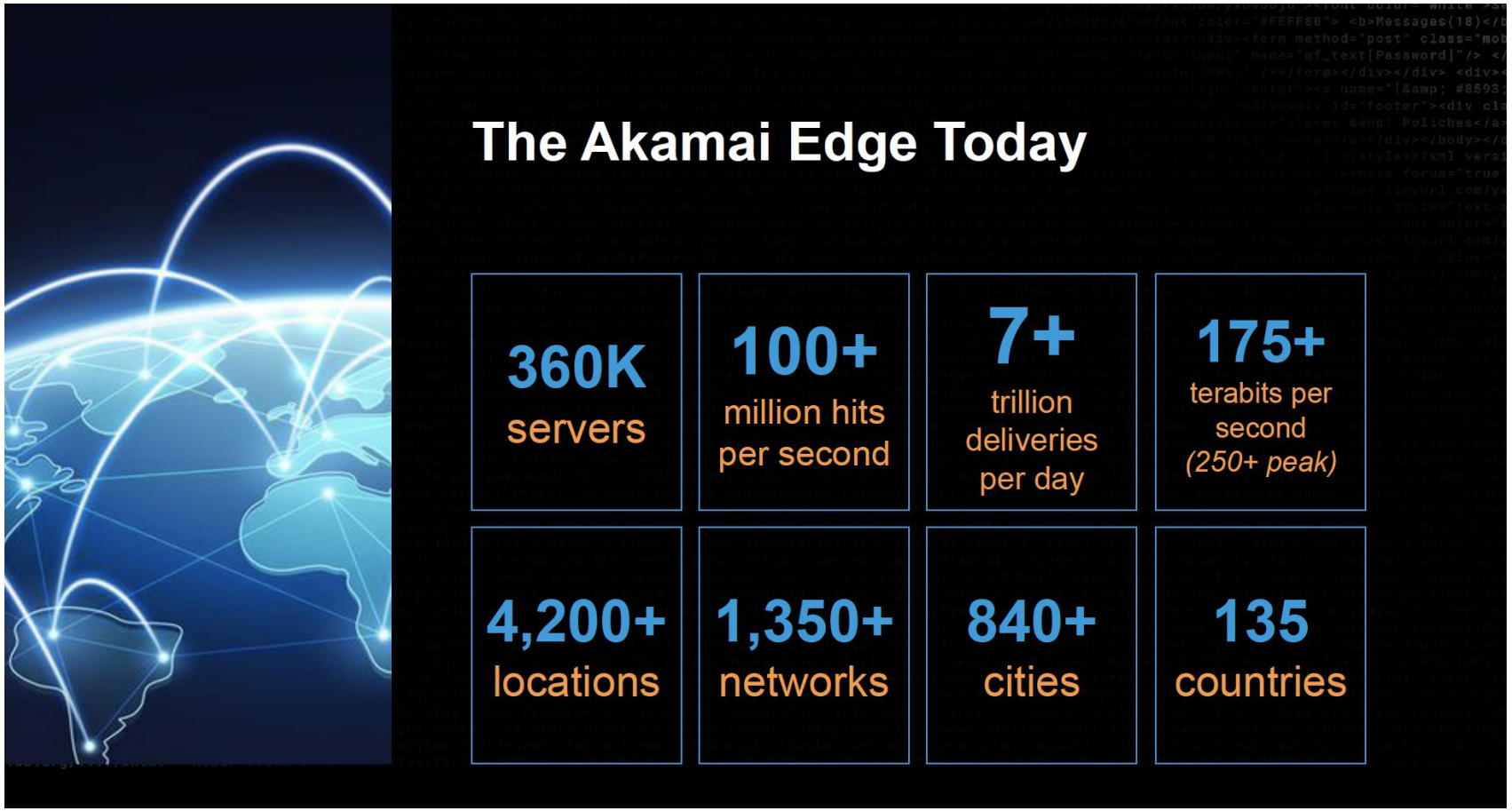
….quite simply: this solution *doesn't scale*

Content distribution networks (CDNs)

challenge: how to stream content (selected from millions of videos) to hundreds of thousands of *simultaneous* users?

- *option 2:* store/serve multiple copies of videos at multiple geographically distributed sites *(CDN)*
  - *enter deep:* push CDN servers deep into many access networks
    - close to users
    - Akamai: 240,000 servers deployed in > 120 countries (2015)
  - *bring home:* smaller number (10's) of larger clusters in POPs near access nets
    - used by Limelight

Akamai today:



Source: https://networkingchannel.eu/living-on-the-edge-for-a-quarter-century-an-akamai-retrospective-downloads/

How does Netflix work?

- Netflix: stores copies of content (e.g., MADMEN) at its (worldwide) OpenConnect CDN nodes
- subscriber requests content, service provider returns manifest
  - using manifest, client retrieves content at highest supportable rate
  - may choose different rate or copy if network path congested