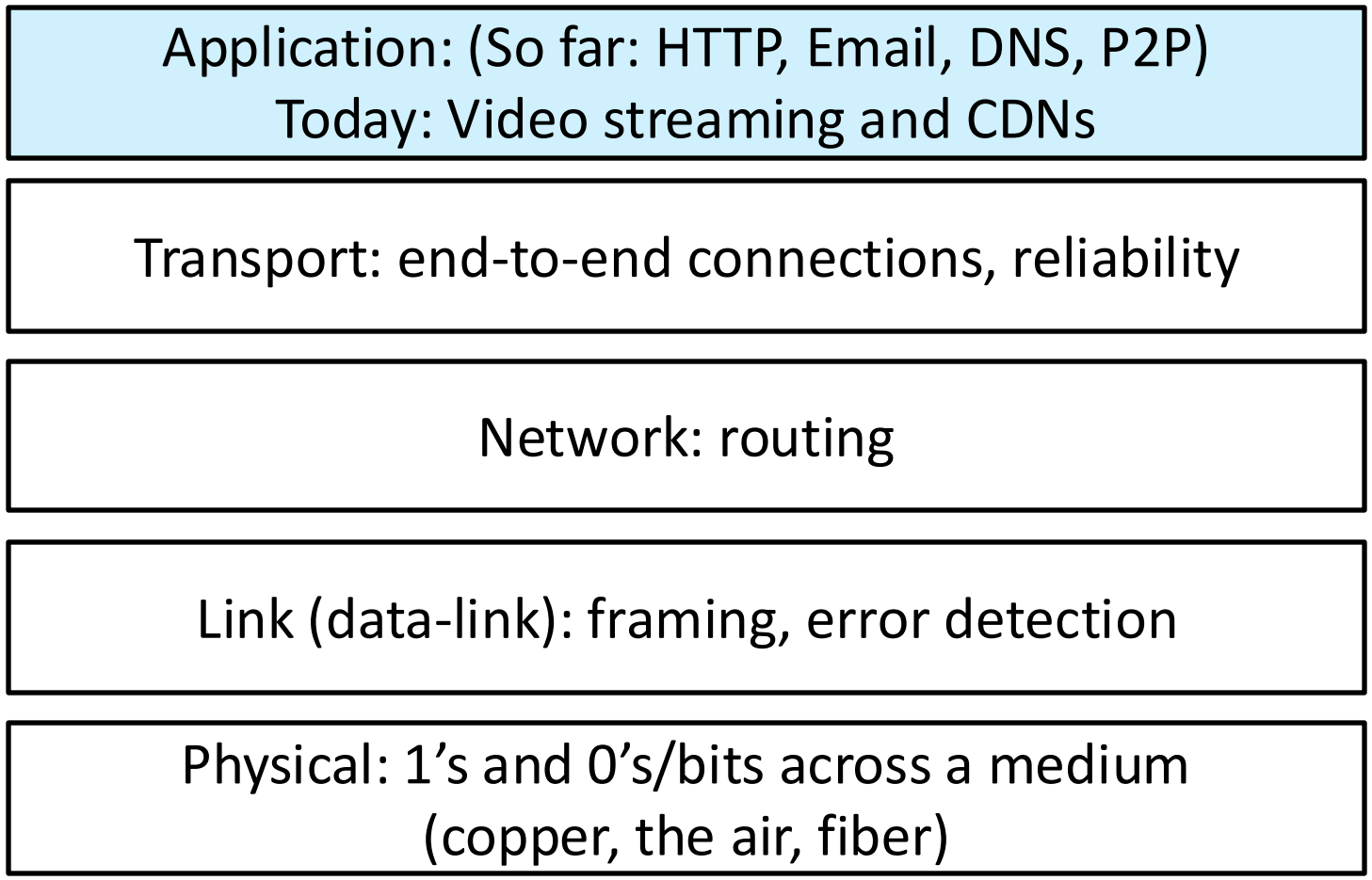
CS 43: Computer Networks

11: Video Streaming and CDNs

October 20, 2025

Slides Courtesy: Kurose & Ross, K. Webb, V. Chaganti D. Choffnes

Where we are


Application: (So far: HTTP, Email, DNS, P2P)
Today: Video streaming and CDNs
Transport: end-to-end connections, reliability
Network: routing
Link (data-link): framing, error detection
Physical: 1's and 0's/bits across a medium
(copper, the air, fiber)

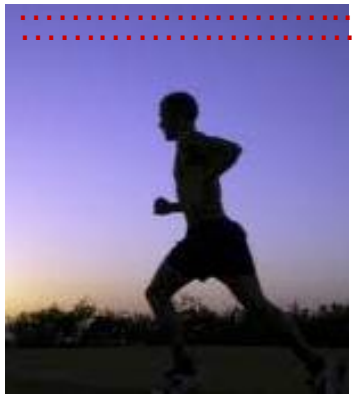
Video Streaming and CDNs: context

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

spatial coding example: instead of sending *N* values of same color (all purple), send only two values: color value (*purple*) *and number of repeated values* (N)



frame *i*

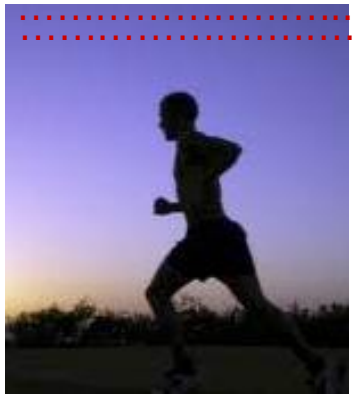
frame *i+1*

temporal coding example: instead of sending complete frame at i+1, send only differences from frame i

Multimedia: video

- CBR: (constant bit rate): video encoding rate fixed
- VBR: (variable bit rate): video encoding rate changes as amount of spatial, temporal coding changes
- examples:
  - SD: 1-3 Mbps
  - Full HD: 3-5 Mbps
  - 4K: 15-25 Mbps

spatial coding example: instead of sending *N* values of same color (all purple), send only two values: color value (*purple*) *and number of repeated values* (N)



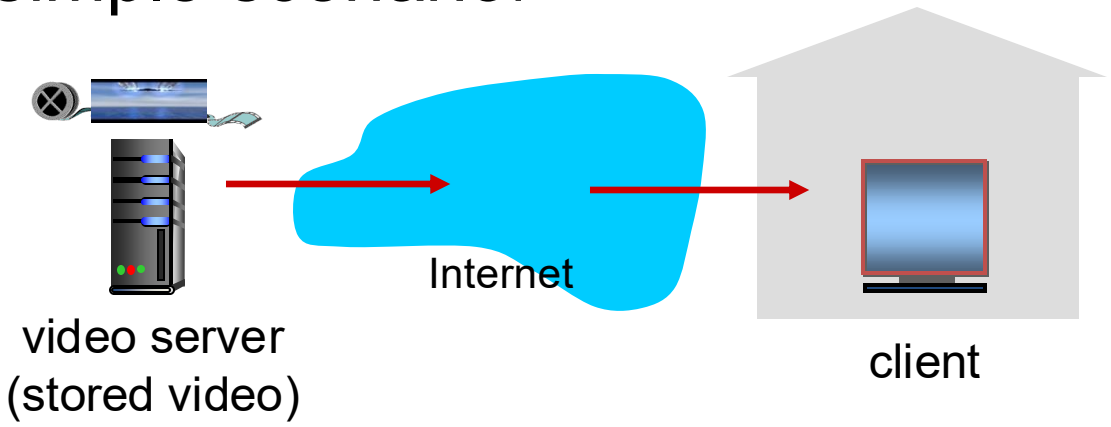
frame *i*

temporal coding example: instead of sending complete frame at i+1, send only differences from frame i

frame *i+1*

Streaming stored video

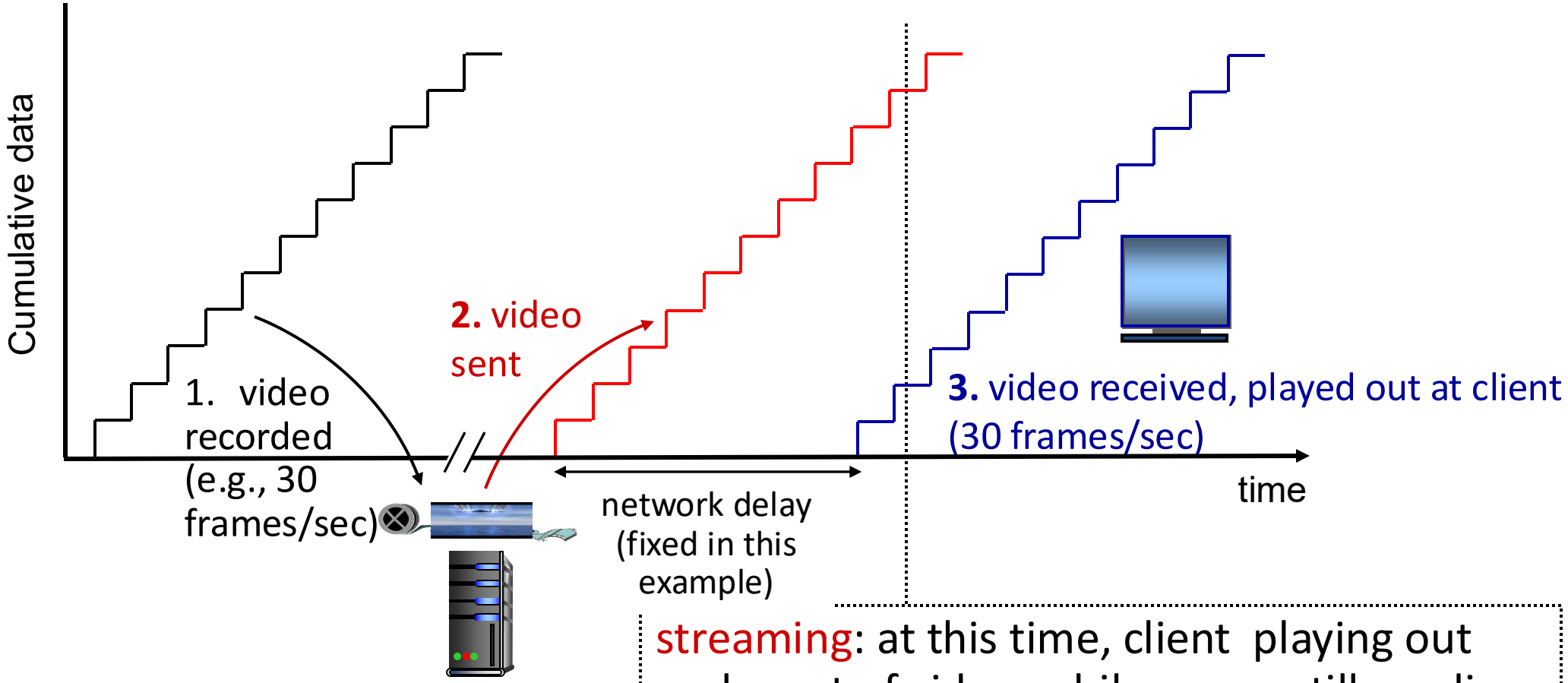
simple scenario:



Main challenges:

- server-to-client bandwidth will *vary* over time, with changing network congestion levels (in house, access network, network core, video server)
- packet loss, delay due to congestion will delay playout, or result in poor video quality

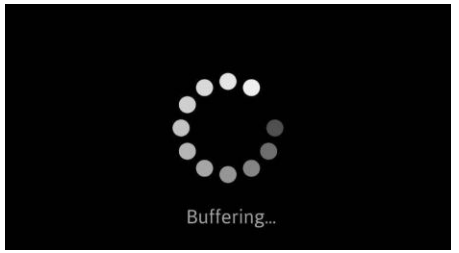
Streaming stored video



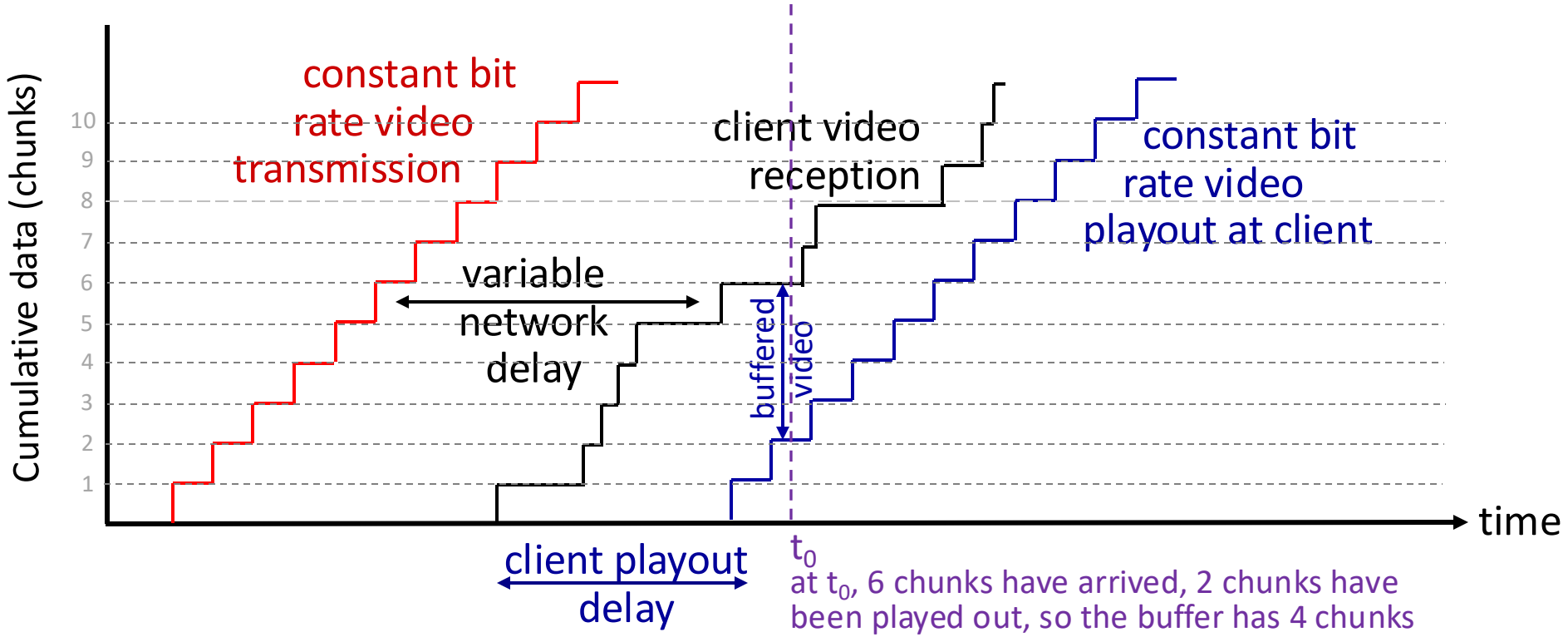
streaming: at this time, client playing out early part of video, while server still sending later part of video

Streaming stored video: challenges

- continuous playout constraint: during client video playout, playout timing must match original timing
  - … but network delays are variable (jitter), so will need client-side buffer to match continuous playout constraint
- other challenges:
  - client interactivity: pause, fast-forward, rewind, jump through video
  - video packets may be lost, retransmitted



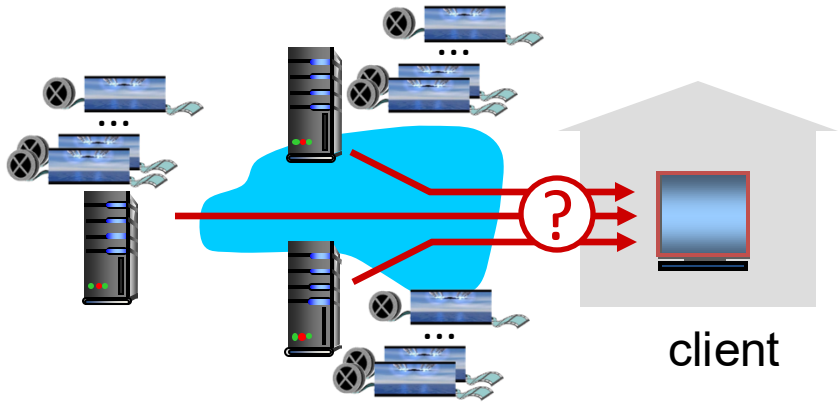
Streaming stored video: playout buffering



- *client-side buffering and playout delay:* compensate for network-added delay, delay jitter

Streaming multimedia: DASH

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



server:

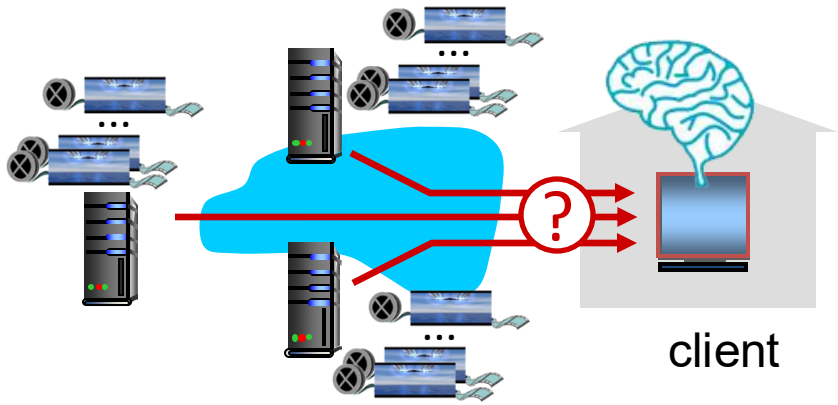
- divides video file into multiple chunks
- each chunk encoded at multiple different rates
- different rate encodings stored in different files
- files replicated in various CDN nodes
- *manifest file:* provides URLs for different chunks

client:

- periodically estimates server-to-client bandwidth
- consulting manifest, requests one chunk at a time
  - GET requests
  - 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

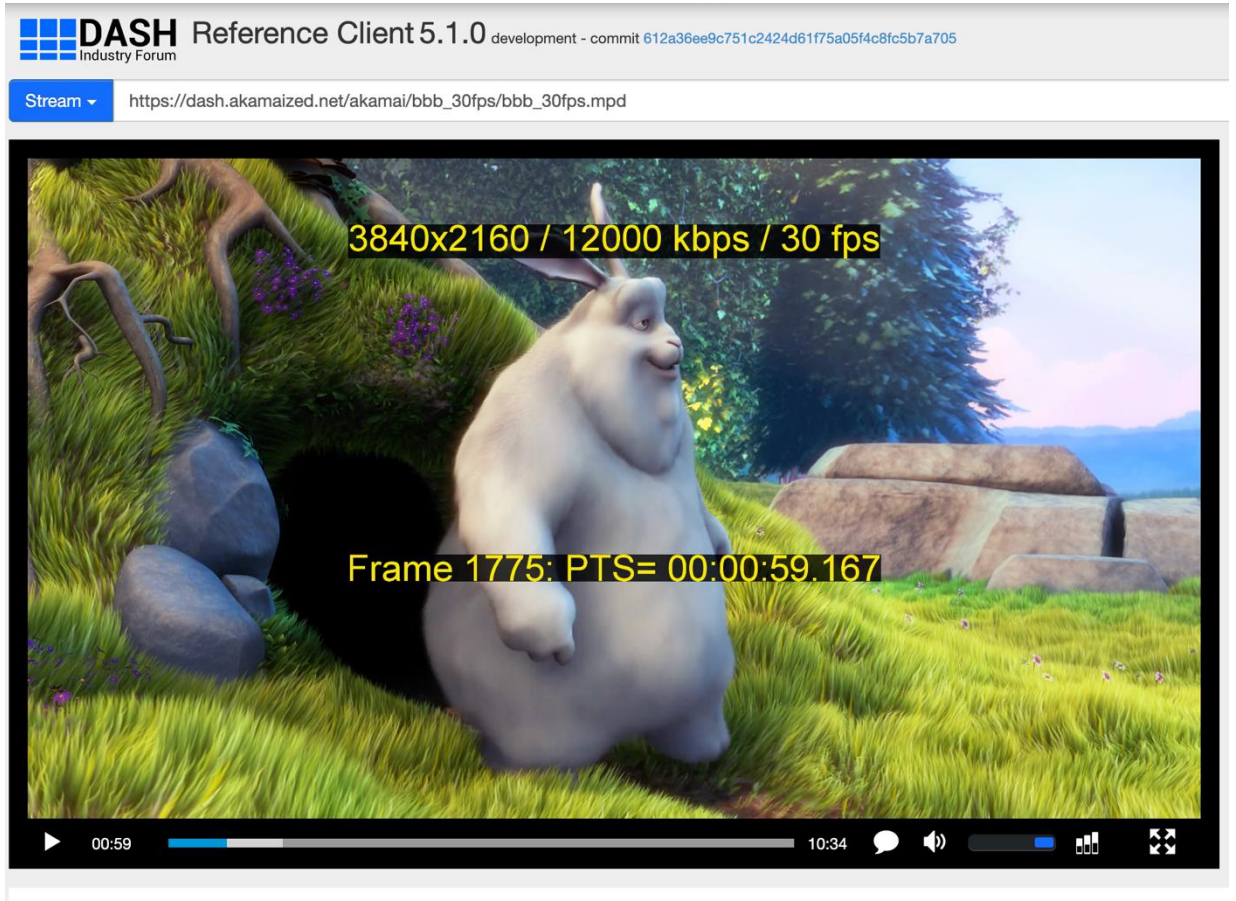
- *"intelligence"* at client: client determines
  - *when* to request chunk (so that buffer starvation, or overflow does not occur)
  - *what encoding rate* to request (higher quality when more bandwidth available)
  - *where* to request chunk (can request from URL server that is "close" to client or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



Streaming video = encoding + DASH + playout buffering

Demo: Dash Reference Player

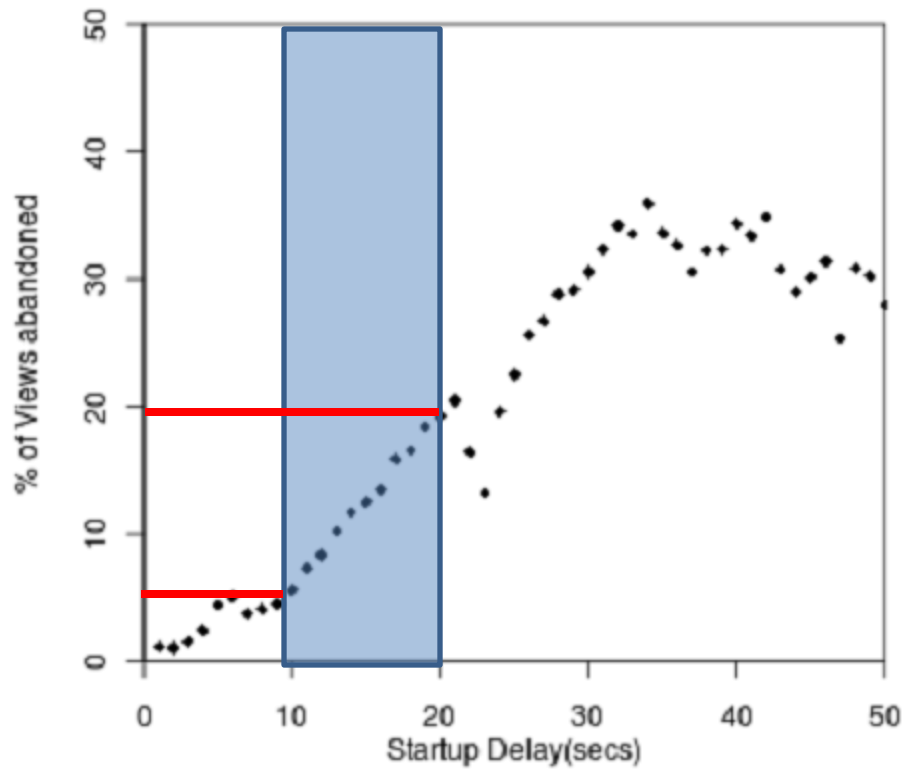
Link: [Dash Reference Player]



Speed is Important!

- Impact on user experience
  - Users navigating away from pages
  - Video startup delay

- 4x increase in abandonment with 10s increase in delay



Demo: How much faster is a website using a CDN?

- Look at ping time for a website in various countries:
  https://tools.keycdn.com/ping
  https://tools.keycdn.com/performance
  - Hosted by a CDN: nbcnews.com

- Check if a website uses a CDN
  https://www.cdnplanet.com/tools/cdnfinder

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

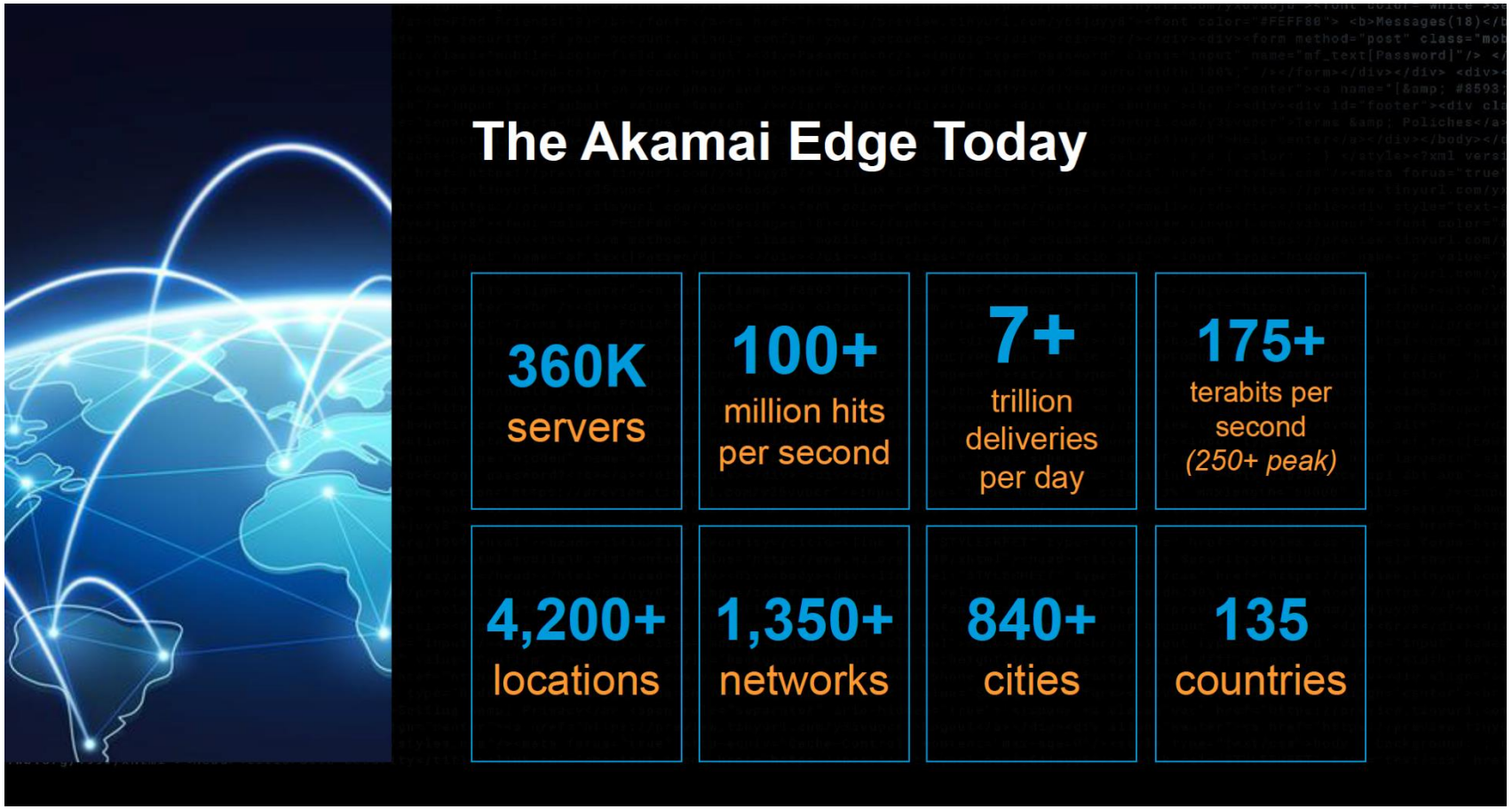
….quite simply: this solution *doesn't scale*

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)*
- At least 70% of the world's bits are delivered by a CDN!
  - *enter deep:* push CDN servers deep into many access networks
    - "Go to the user"
    - Akamai: 240,000 servers deployed in > 120 countries (2015)
  - *bring home:* smaller number (10's) of larger clusters in POPs near access nets
    - "Bring the users to our data"
    - Cloudflare, Fastly

Akamai today:



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

More Examples of CDNs

- Cloudflare
  - Rising to be largest CDN
  - Emphasis on cybersecurity and DDoS mitigation
  - Shares data collected from CDN: https://radar.cloudflare.com/
- Fastly
- Others
  - Google, Facebook, AWS, AT&T, Level3

Cloudflare today:

The Cloudflare global network

Our vast global network, which is one of the fastest on the planet, is trusted by millions of web properties.

With direct connections to nearly every service provider and cloud provider, the Cloudflare network can reach about 95% of the world's population within approximately 50 ms.

330
cities in 125+ countries, including mainland China

13,000
networks directly connect to Cloudflare, including every major ISP, cloud provider, and enterprise

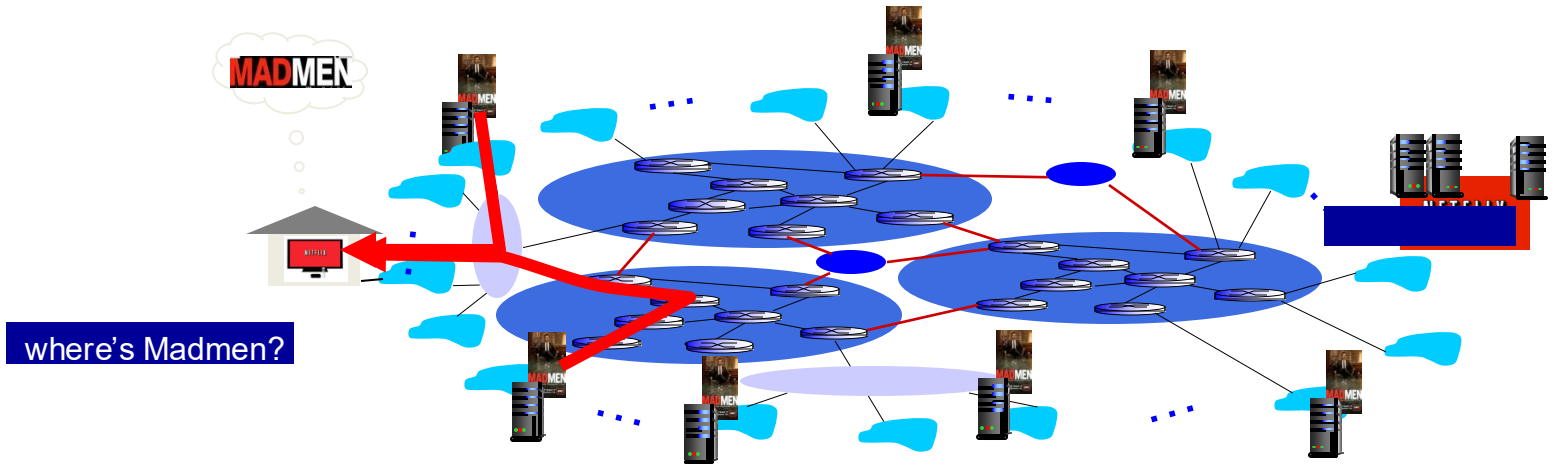
405 Tbps
global network edge capacity, consisting of transit connections, peering and private network interconnects

~50 ms
from about 95% of the world's Internet-connected population

Source: https://www.cloudflare.com/network/

Content distribution networks (CDNs)

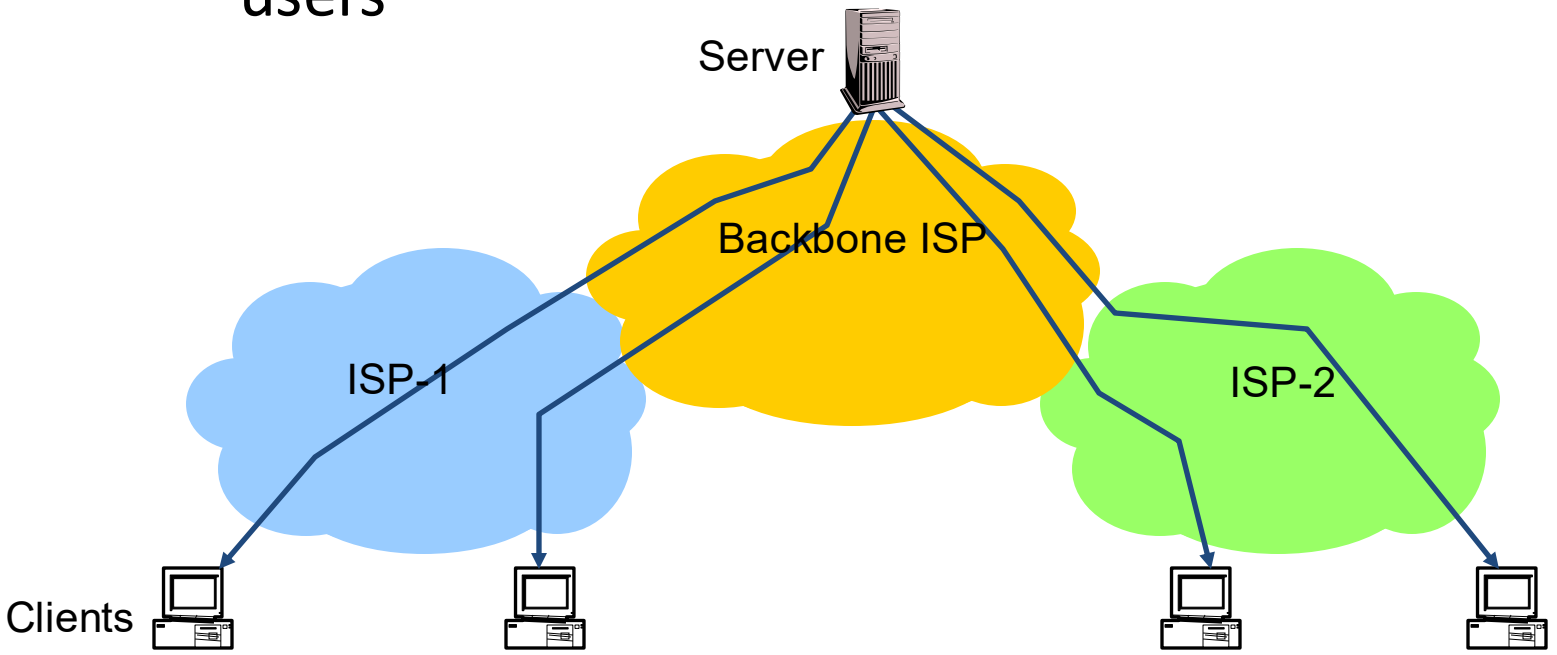
- CDN: stores copies of content (e.g. MADMEN) at CDN nodes



nearby CDN replicas

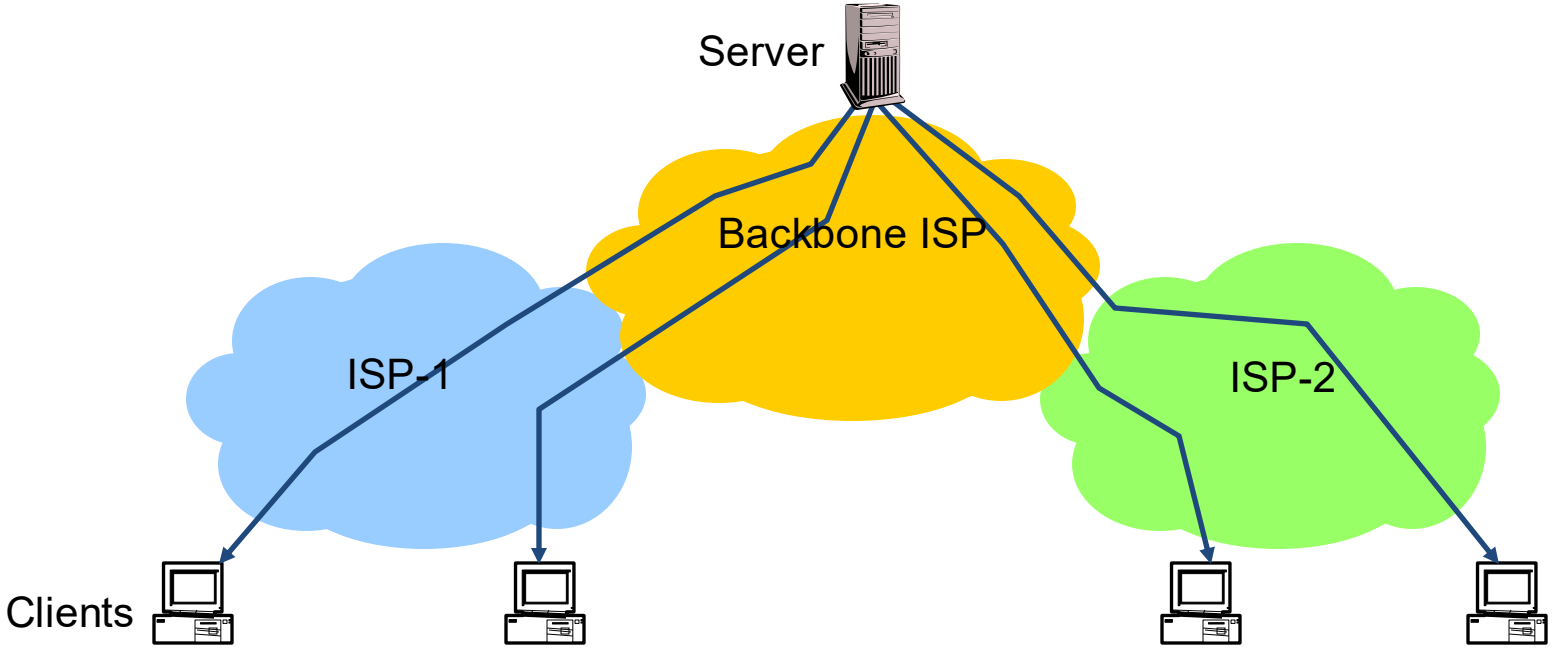
CDN caching

- Locality of reference:
  - Users tend to request the same object in succession
  - Some objects are popular: requested by many users



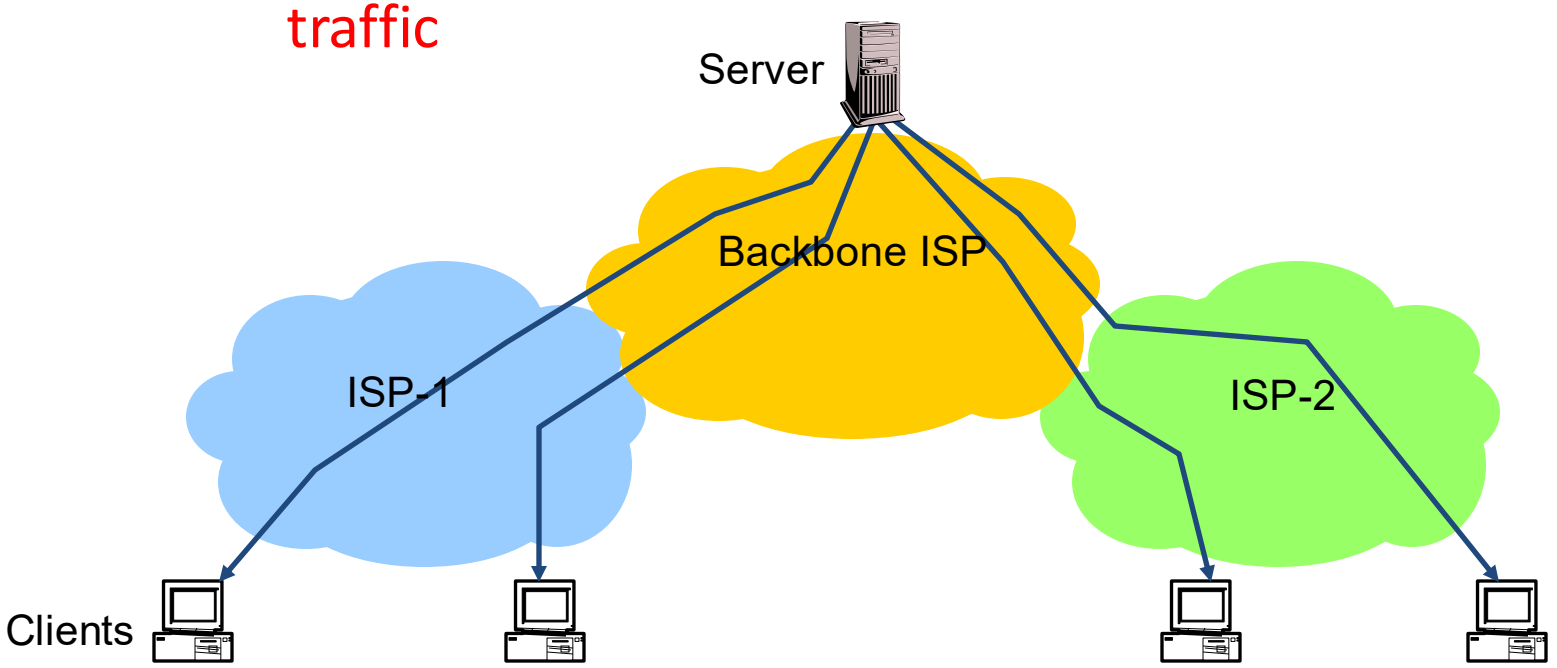
Where to cache content?

A. At the client
B. At the server (distributed server load)
C. At the Service Providers (ISPs)



Where to cache content?

A. At the client (browser) – avoid extra network transfers
B. At the server (distributed server load) – reduce load
C. At the Service Providers (ISPs) – reduce external traffic



Key Components of a CDN

- Distributed servers
  - Usually located inside of other ISPs
  - Often located in IXPs
- High-speed network connecting them
- Clients (eyeballs)
  - Can be located anywhere in the world
  - They want fast web performance
- Glue
  - Something that binds clients to “nearby” replica servers

High-Performance Content Distribution

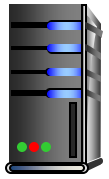
- Major challenges:
  - How do we direct the user to a nearby replica instead of the centralized source?
  - How do we determine which replica is the best to send them to?
  - Ensure that replicas are always available?

Challenge 1: Finding the CDN

- Three main options:
  - Application redirect (e.g., HTTP)
  - "Anycast" routing
  - DNS resolution (most popular in practice)

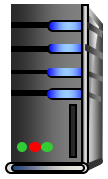
- Example: NBC News + Akamai

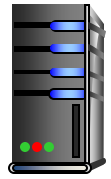
NBC News + Akamai

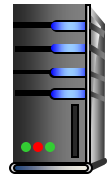


www.nbcnews.com

Request: nbcnews.com/article
Response: HTML with link
to cache.nbcnews.com media

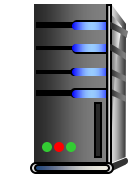




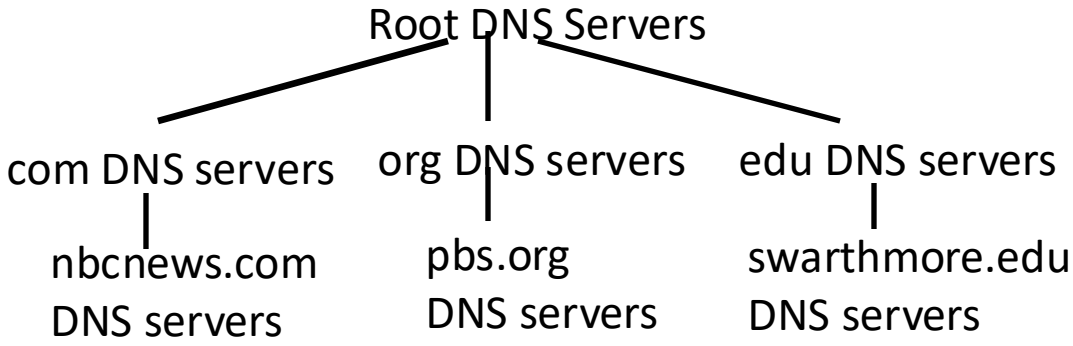


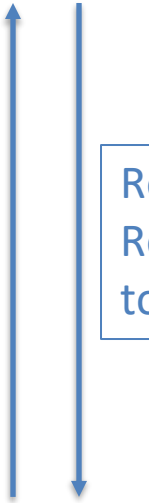
Content servers: serve media.

NBC News + Akamai

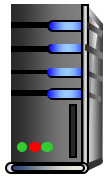


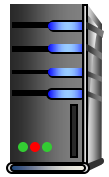
www.nbcnews.com





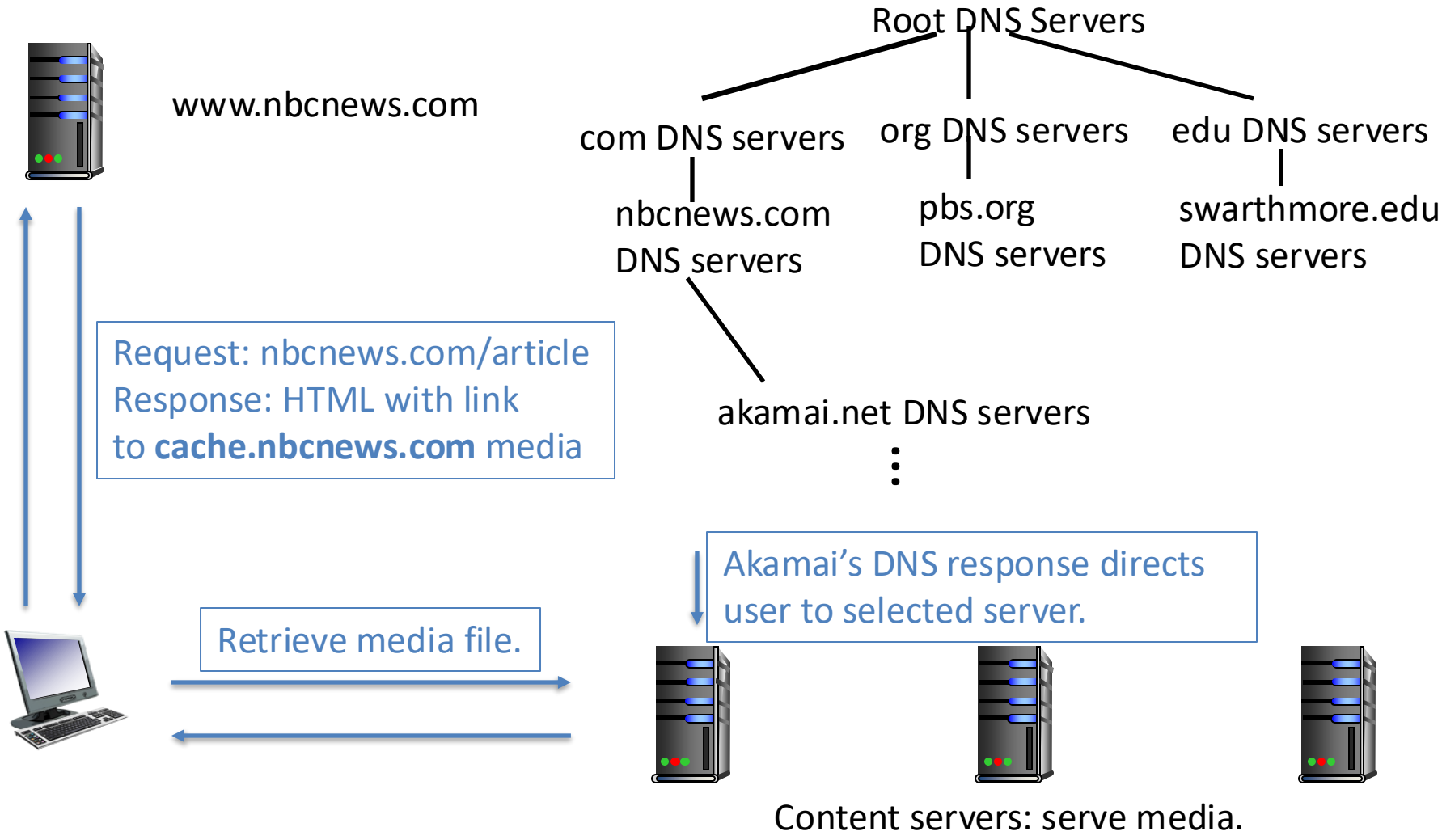
Request: nbcnews.com/article
Response: HTML with link to **cache.nbcnews.com** media



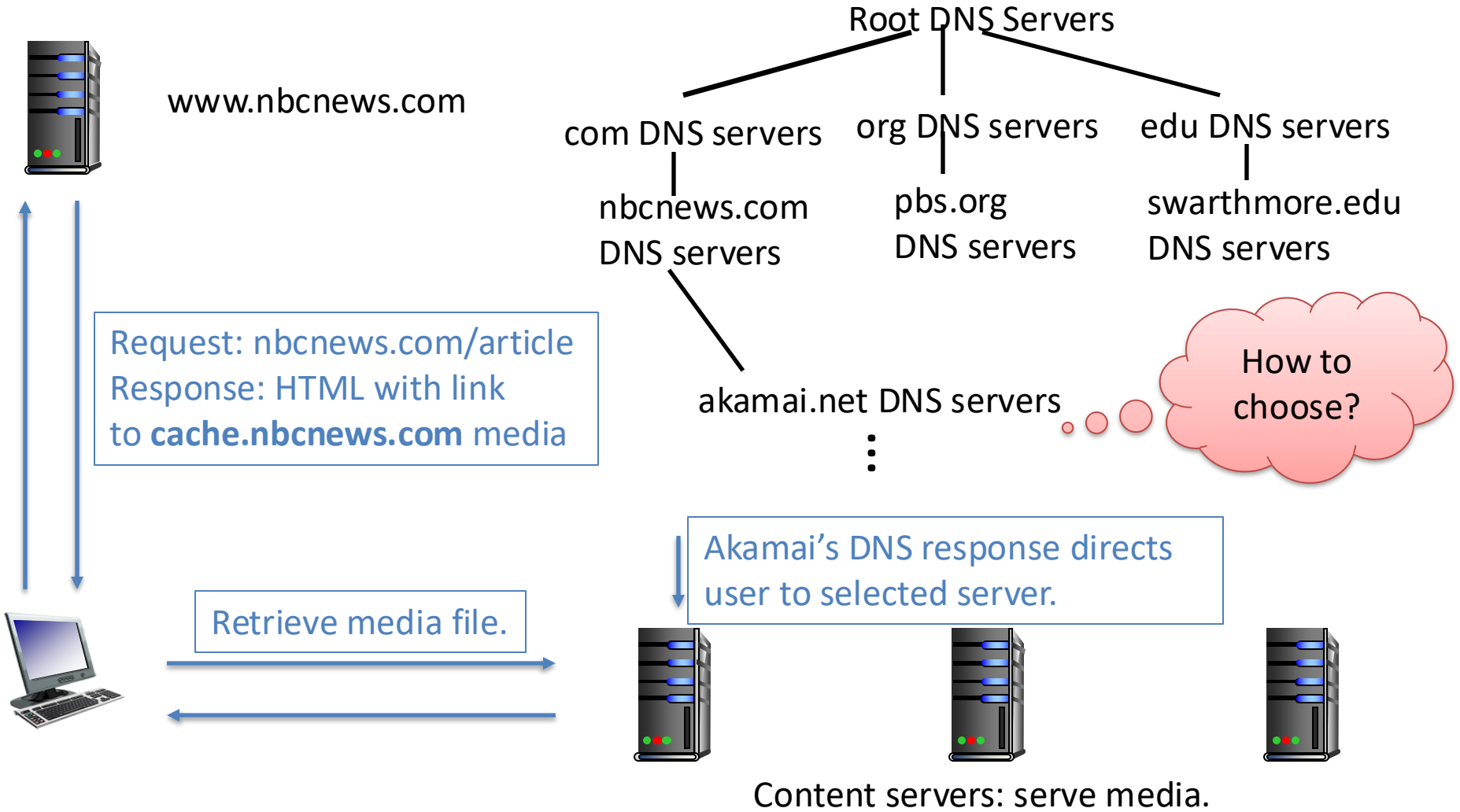


Content servers: serve media.

NBC News + Akamai



NBC News + Akamai



Which metric is most important when choosing a server? (CDN or otherwise)

A. RTT latency

B. Data transfer rate / throughput

C. Hardware ownership

D. Geographic location

E. Some other metic(s) (such as?)

This is the CDN operator's secret sauce!

Which metric is most important when choosing a server? (CDN or otherwise)

A. RTT latency

B. Data transfer rate / throughput

C. Hardware ownership

D. Geographic location

This is the CDN operator's secret sauce!

E. Some other metic(s) (such as?)

Summary

- DASH: streaming despite dynamic conditions
- CDNs: locating “good” replica for media server

Announcements

- Quiz Wednesday