EECS 489
Computer Networks

Fall 2019

Mosharaf Chowdhury

Material with thanks to Aditya Akella, Sugih Jamin, Philip Levis, Sylvia Ratnasamy, Peter Steenkiste, and many other colleagues.

Agenda

- Video streaming
- Datacenter applications

Recap: Web components

- Infrastructure:
  - Clients
  - Servers (DNS, CDN, Datacenters)
- Content:
  - URL: naming content
  - HTML: formatting content
- Protocol for exchanging information: HTTP

Recap: HTTP

- Client-server architecture
  - Server is “always on” and “well known”
  - Clients initiate contact to server
- Synchronous request/reply protocol
  - Runs over TCP, Port 80
- Stateless
- ASCII format
  - Before HTTP/2

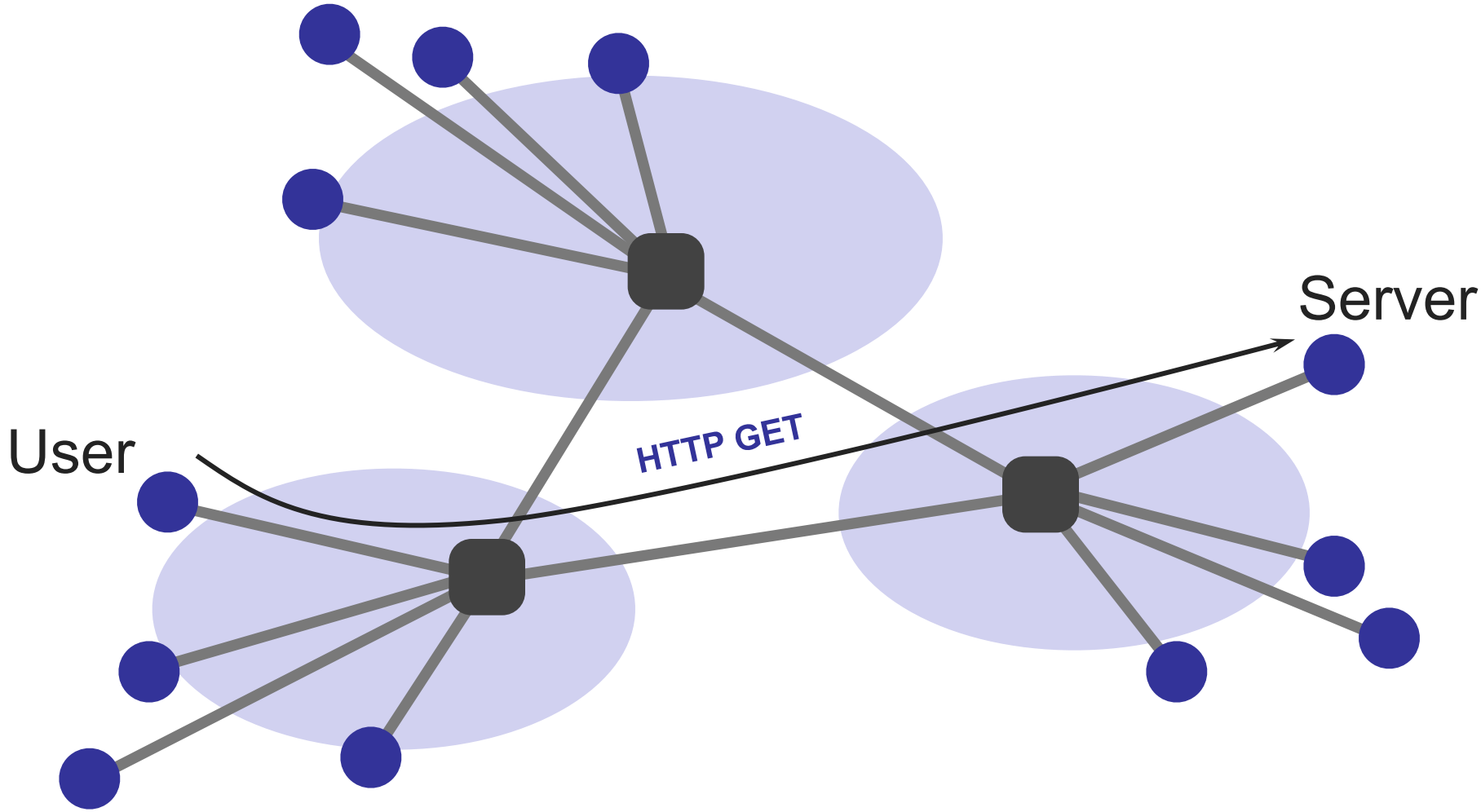
Recap: CDN

- Caching and replication as a service
- Large-scale distributed storage infrastructure (usually) administered by one entity
  - e.g., Akamai has servers in 20,000+ locations
- Combination of caching and replication
  - Pull: Direct result of clients' requests (caching)
  - Push: Expectation of high access rate (replication)

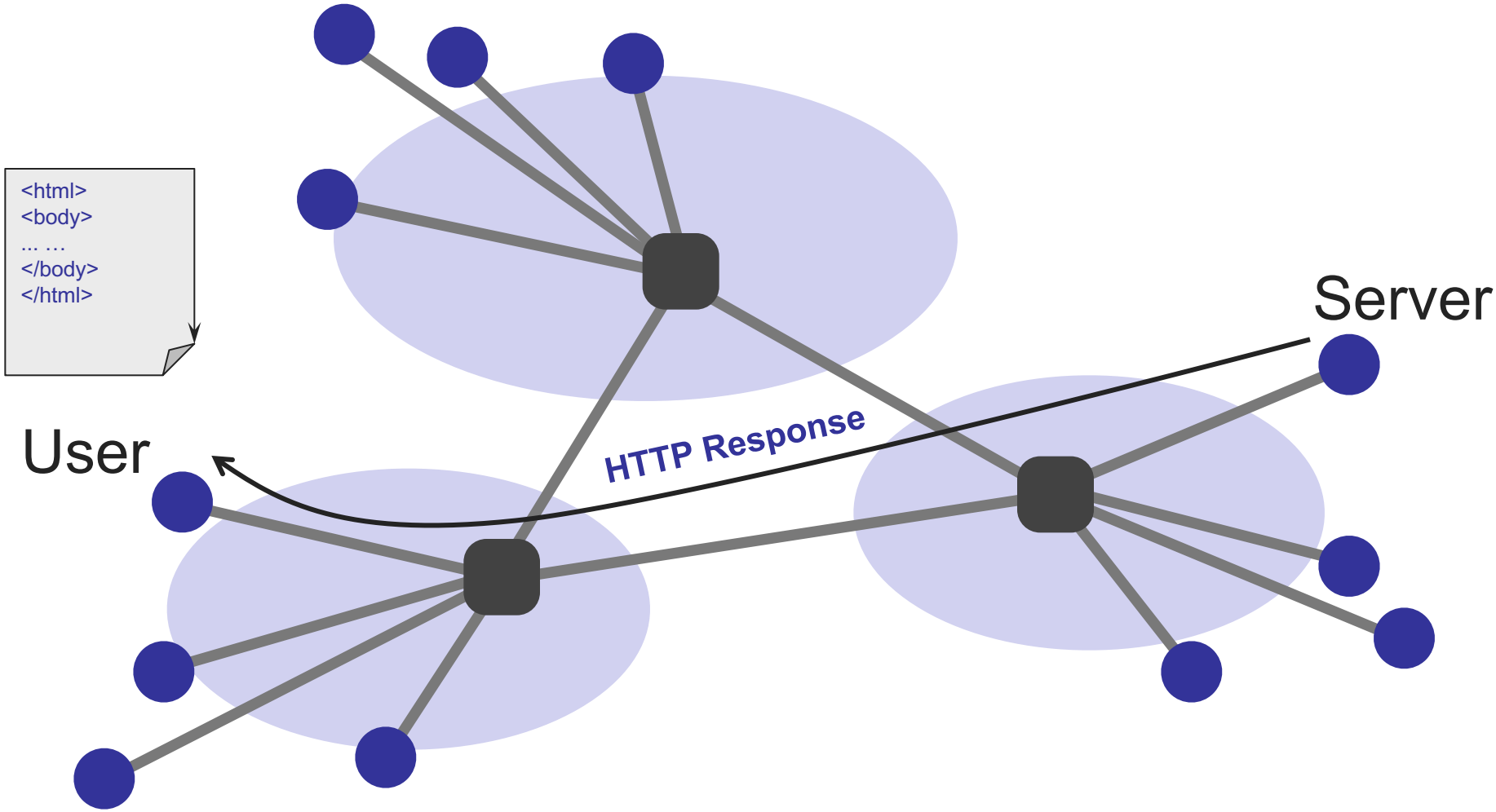
Recap: DNS

- Resolve names to addresses
  - Uniqueness: no naming conflicts
  - Scalable: many names and frequent updates
  - Distributed, autonomous administration
    - »Ability to update my own (machines') names
    - »Don't have to track everybody's updates
  - Highly available
  - Lookups are fast
- Level of indirection

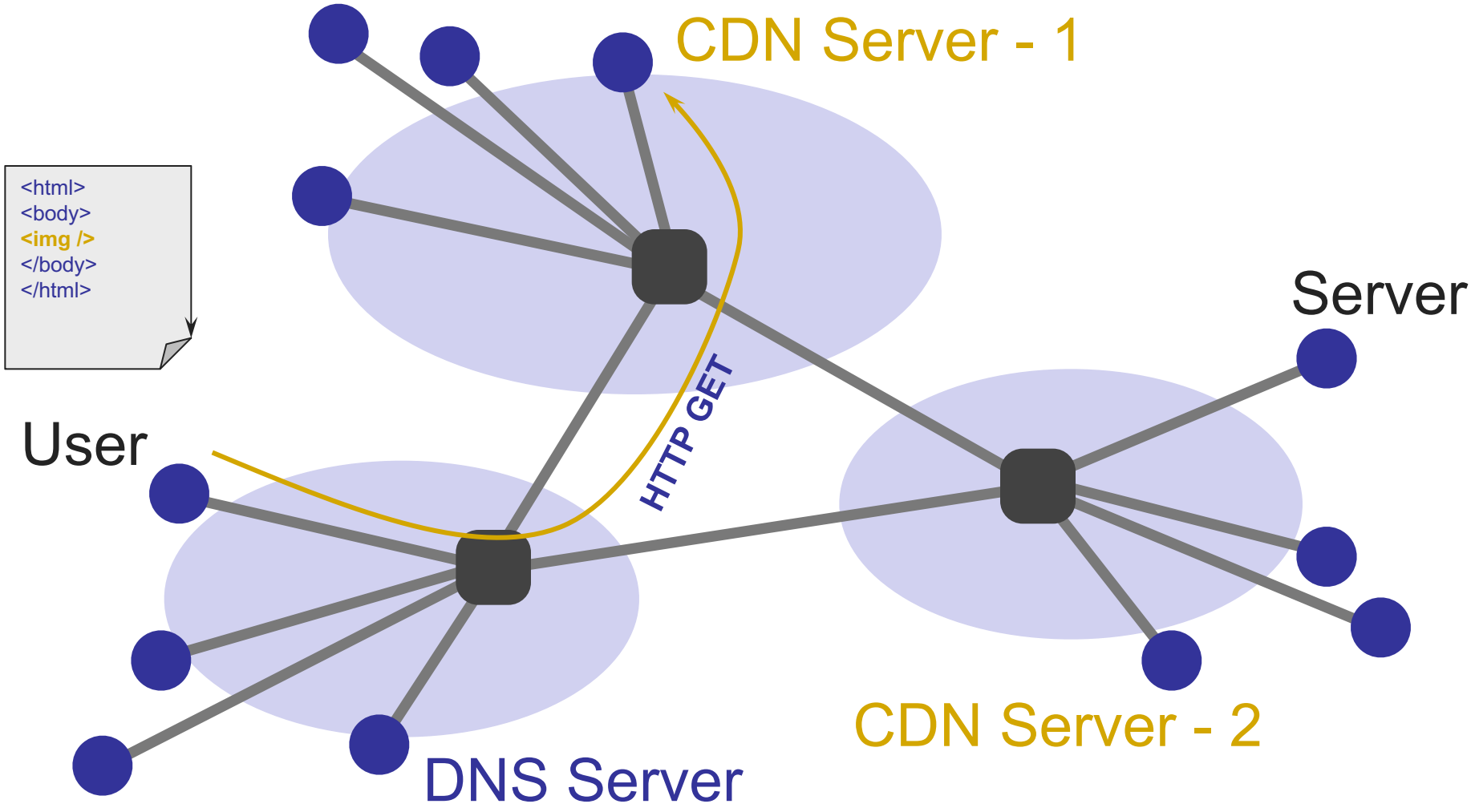
Putting it all together



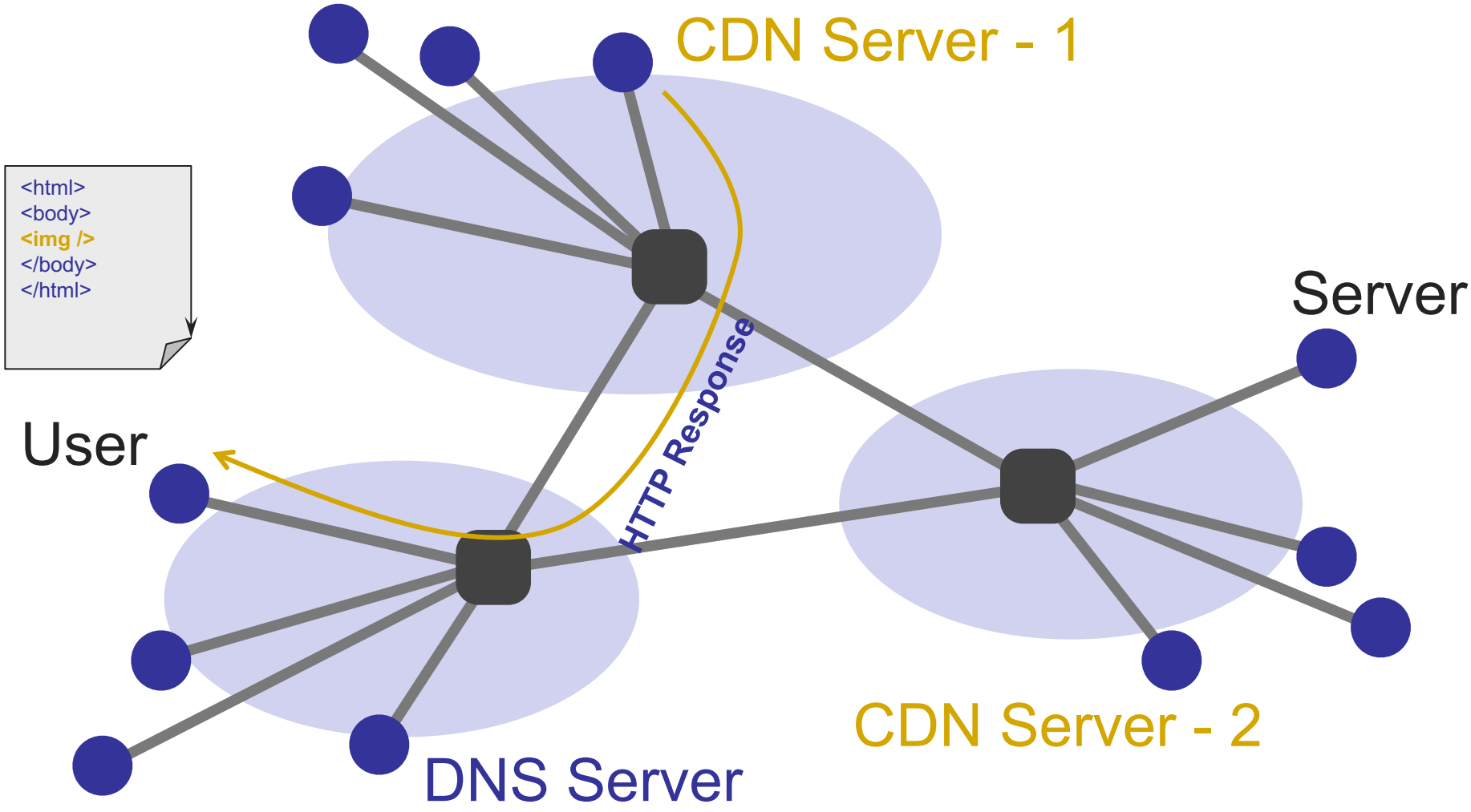
Putting it all together



Putting it all together



Putting it all together



How is video different?

- Often too large to send in one GET
- Doesn't even make sense even if its possible
  - Users may skip forward! ⇒ save bandwidth wastage
  - Users connection quality may change (e.g., switching from WiFi to LTE) ⇒ lower resolution to save bandwidth
- Our focus is on stored video (i.e., not live)

Why video is important?

- Dominates the Internet traffic landscape
  - About every 3 of 5 bytes in 2019!
- Residential ISP traffic share of video content providers
  - Netflix: 37%
  - YouTube: 16%
  - Overall: ~70%

The video medium

- Video is a sequence of images/frames displayed at a constant rate (moving pictures)
- Digital image is an array of pixels, each pixel represented by bits
- Examples:
  - Single frame image encoding: 1024x1024 pixels, 24 bits/pixel ⇒ 3 MB/image
  - Movies: 24 frames/sec ⇒ 72 MB/sec
  - TV: 30 frames/sec ⇒ 90 MB/sec

The video medium (cont'd)

- Compression is key
  - Lots of algorithms to compress
- The same video can be (and typically is) compressed to multiple quality levels
  - E.g., 480p, 720p, 1080p, 4K
- Why multiple resolutions?
  - Adapt to conditions

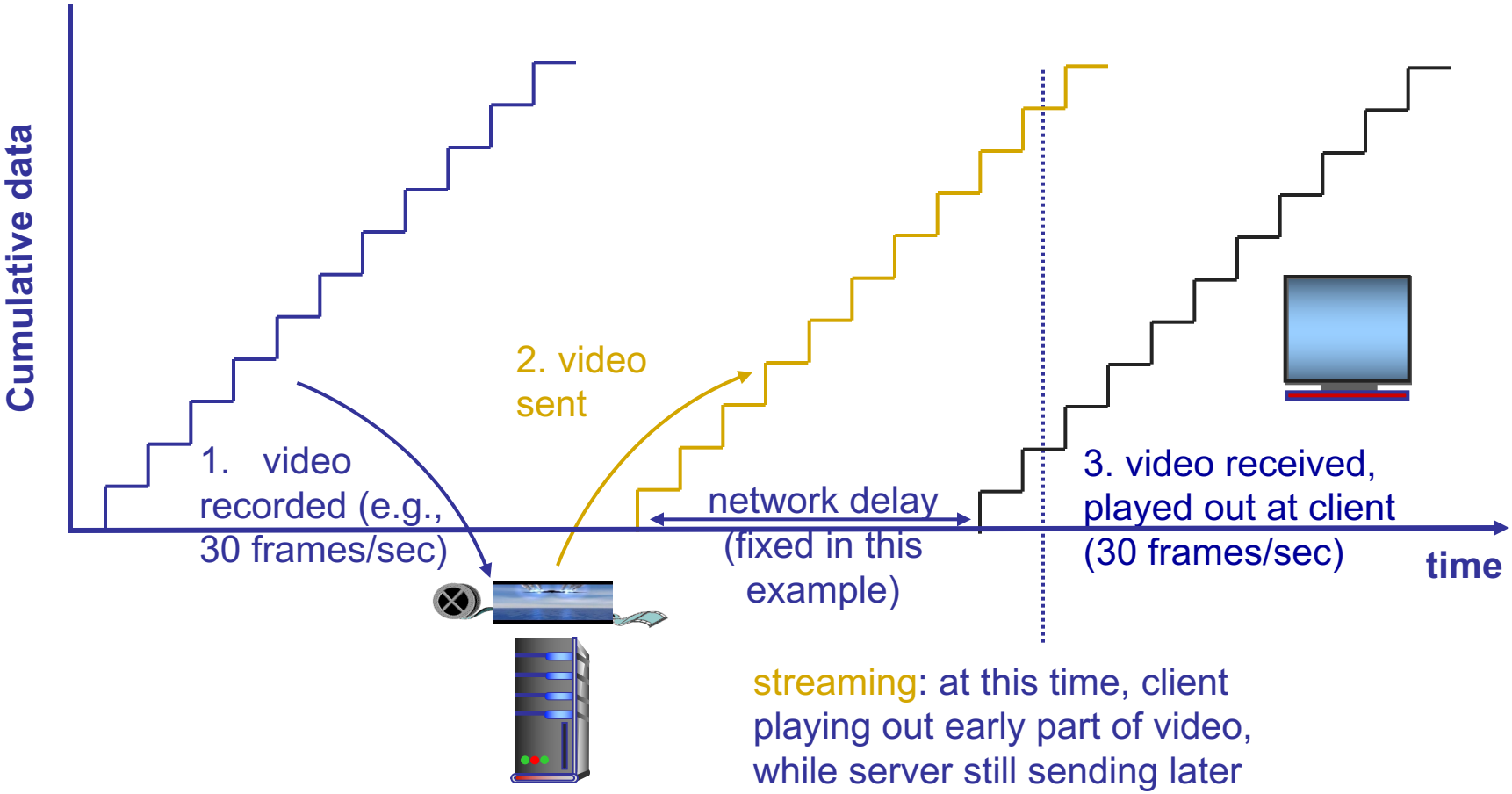
How do we serve video?

- It's in the name!
  - Video streaming

HTTP streaming

- Video is stored at an HTTP server with a URL
- Clients send a GET request for the URL
- Server sends the video file as a stream
- Client first buffers for a while. Why?
  - To minimize interruptions later
- Once the buffer reaches a threshold
  - The video plays in the foreground
  - More frames are downloaded in the background

HTTP streaming

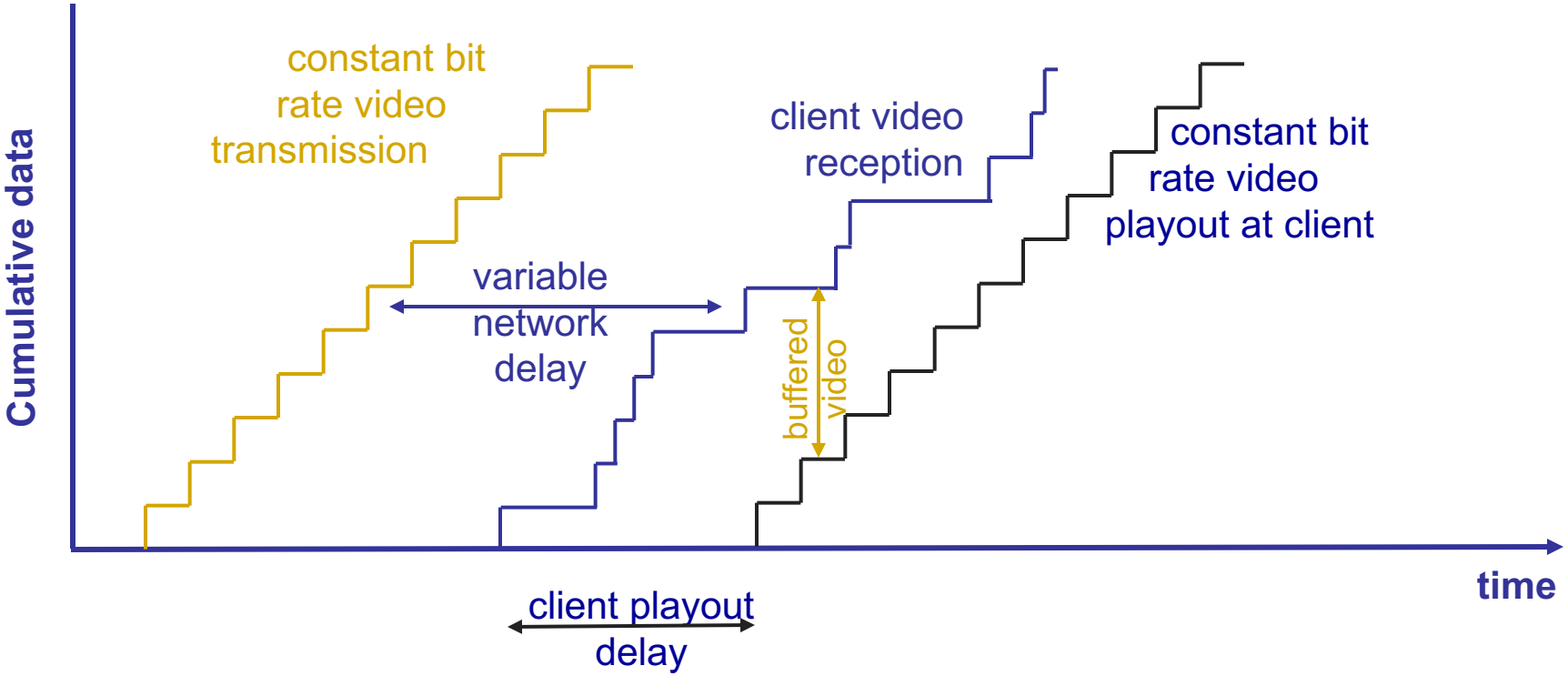


streaming: at this time, client playing out early part of video, while server still sending later part of video

Challenges

- Absorb network delay variations
- Handle user interactions
  - Jump forward, fast-forward, rewind, pause
- Handle packet loss, retransmission etc.

HTTP streaming: Revisited



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

Issues with HTTP streaming

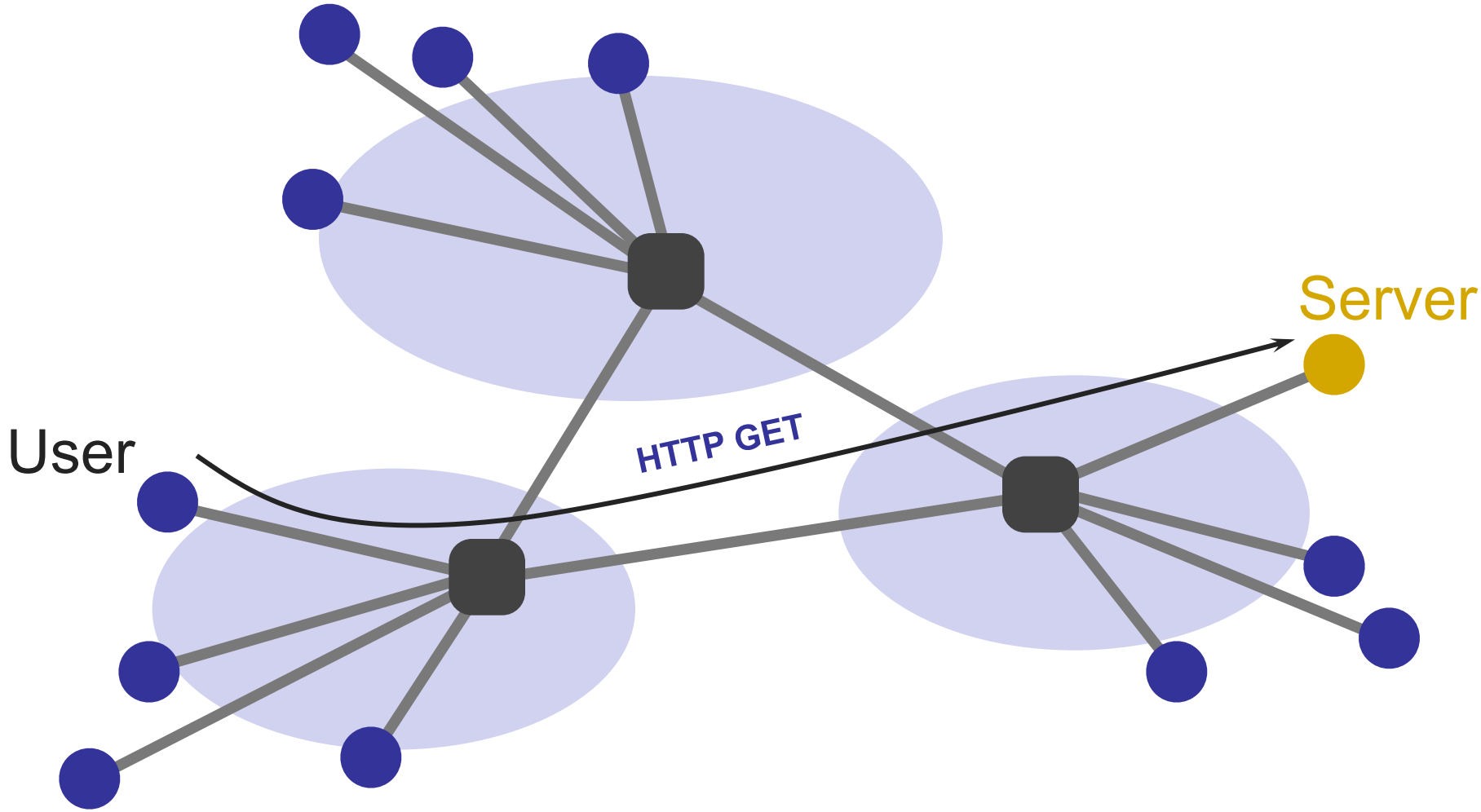
- Same bitrate for all clients
  - Clients can have very different network conditions
  - Clients network conditions can change over time
- Cannot dynamically adapt to conditions

DASH : Dynamic Adaptive Streaming over HTTP

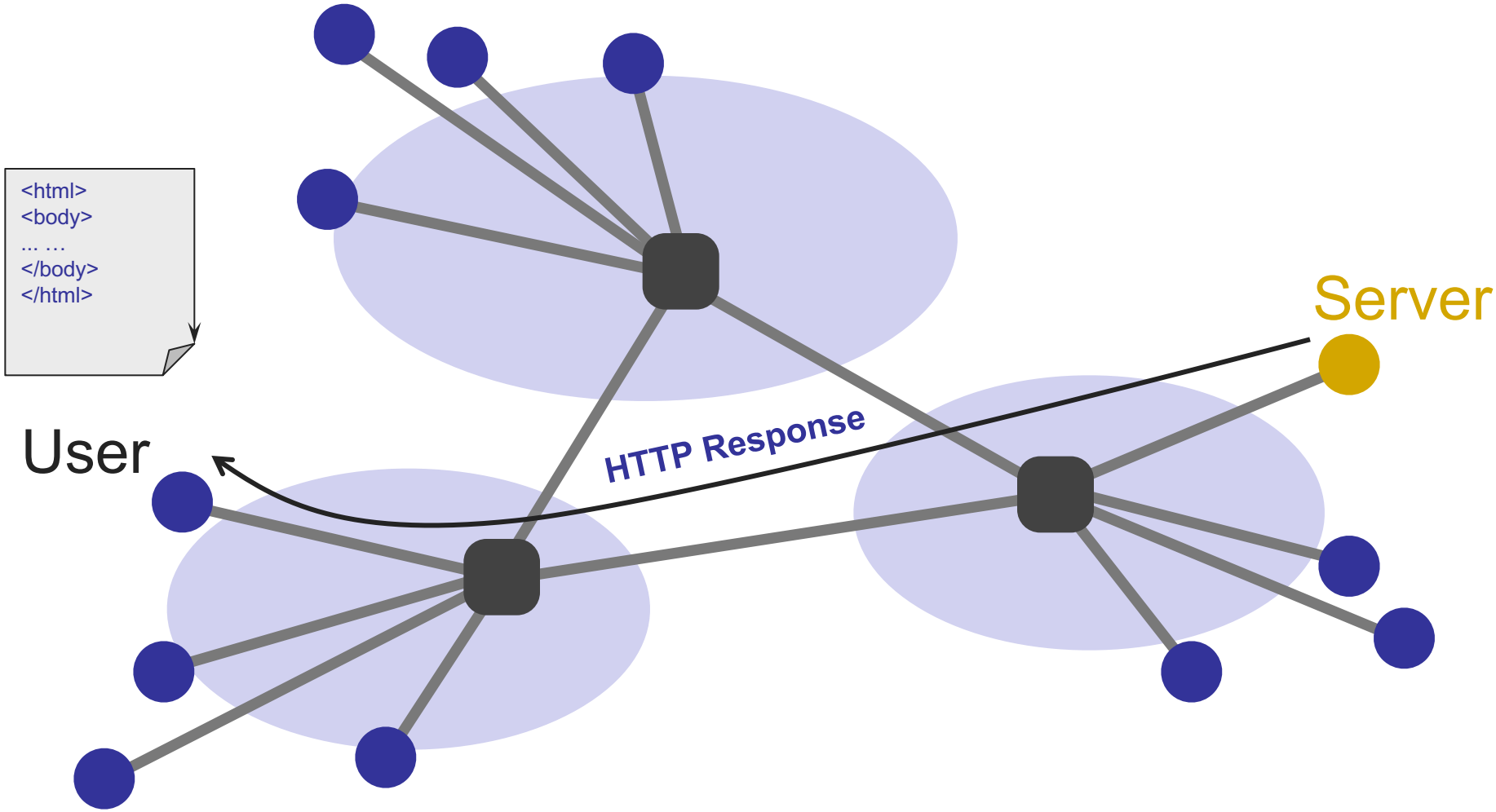
- Keep multiple resolutions of the same video
  - Stored in a manifest file in the HTTP server
- Client asks for the manifest file first to learn about the options
- Asks for chunks at a time and measures available bandwidth while they are downloaded
  - Low bandwidth ⇒ switch to lower bitrate
  - High bandwidth ⇒ switch to higher bitrate

5-MINUTE BREAK!

Who's serving Web services?

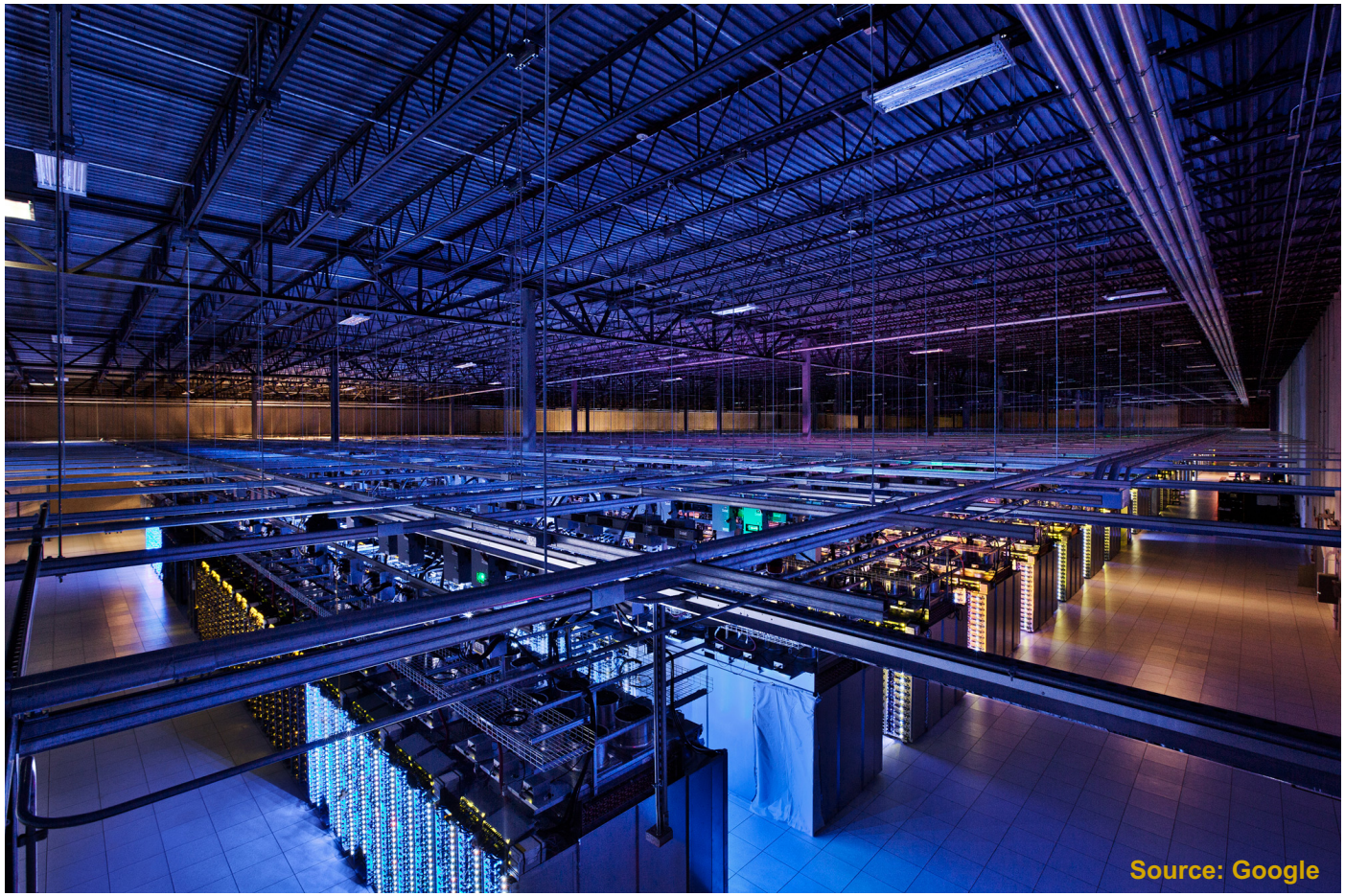


Who's serving Web services?



Who's serving Web services? Datacenters

Who's serving Web services? Datacenters



Who's serving Web services? Datacenters

Datacenter networks

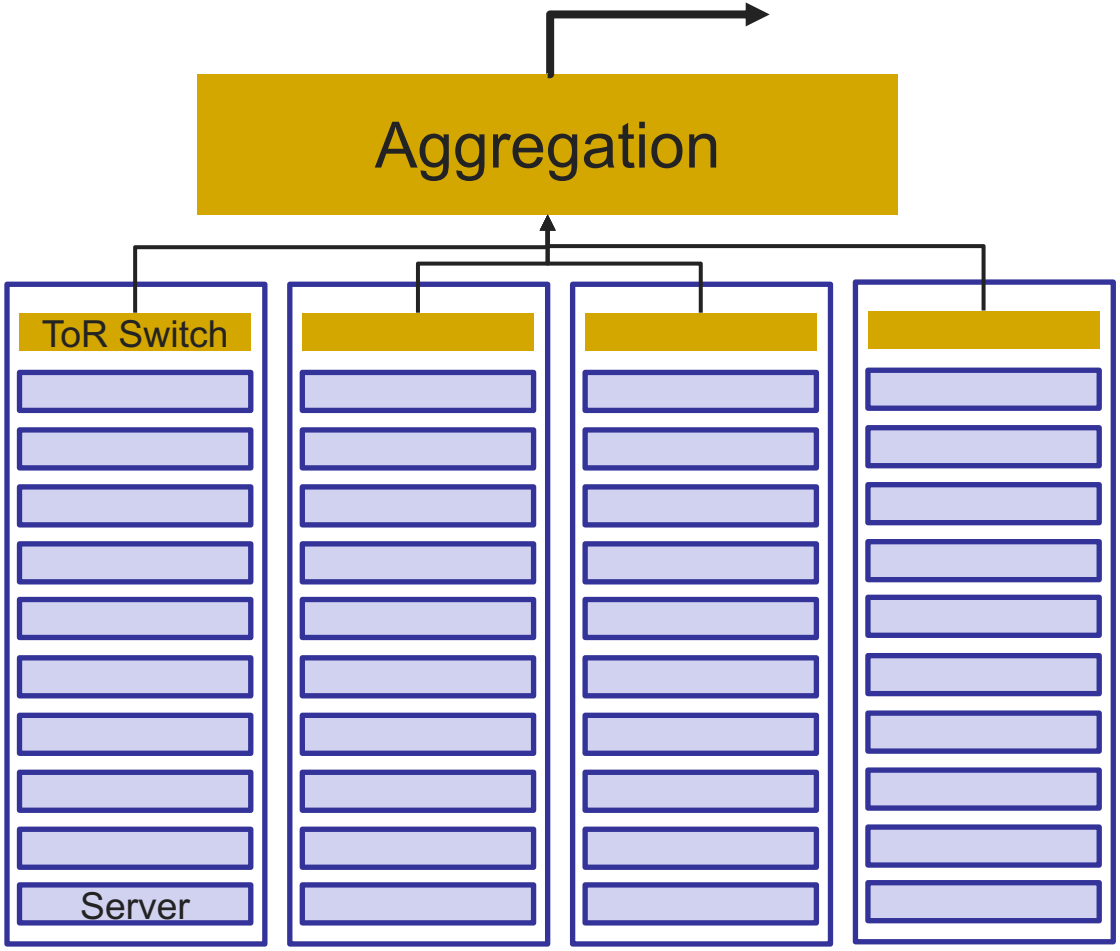
- Tens to hundreds of thousands of hosts, often closely coupled, in close proximity:
  - E-commerce (e.g., Amazon)
  - Content servers (e.g., Netflix, YouTube, Apple, Microsoft)
  - Search engines, data mining (e.g., Google)
  - Social networks (e.g., Facebook, Twitter, Instagram)
- Challenges:
  - Multiple applications, each serving massive numbers of clients

How big exactly?

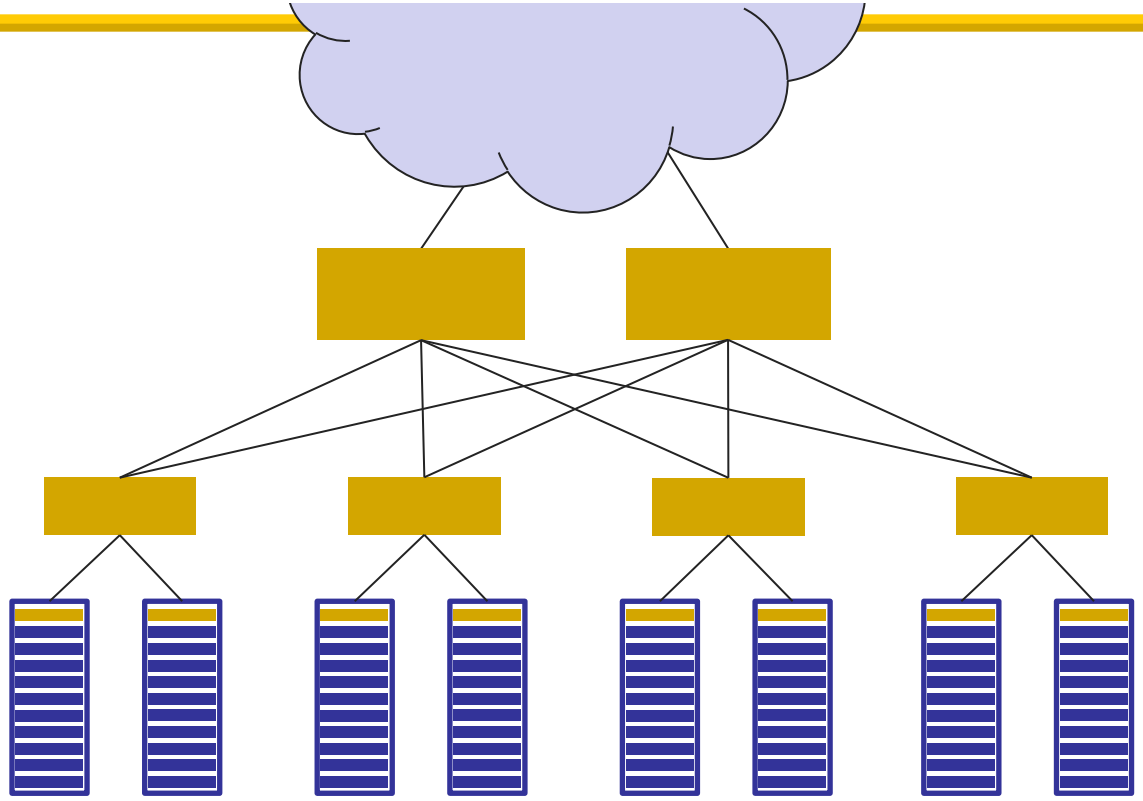
- 1M servers/site [Microsoft/Amazon/Google]
- > $1B to build one site [Facebook]
- >$20M/month/site operational costs [MS'09]
- Data center hardware spending will grow to $177 billion in 2017. [Gartner report]

- But only O(10-100) sites

Datacenter networks



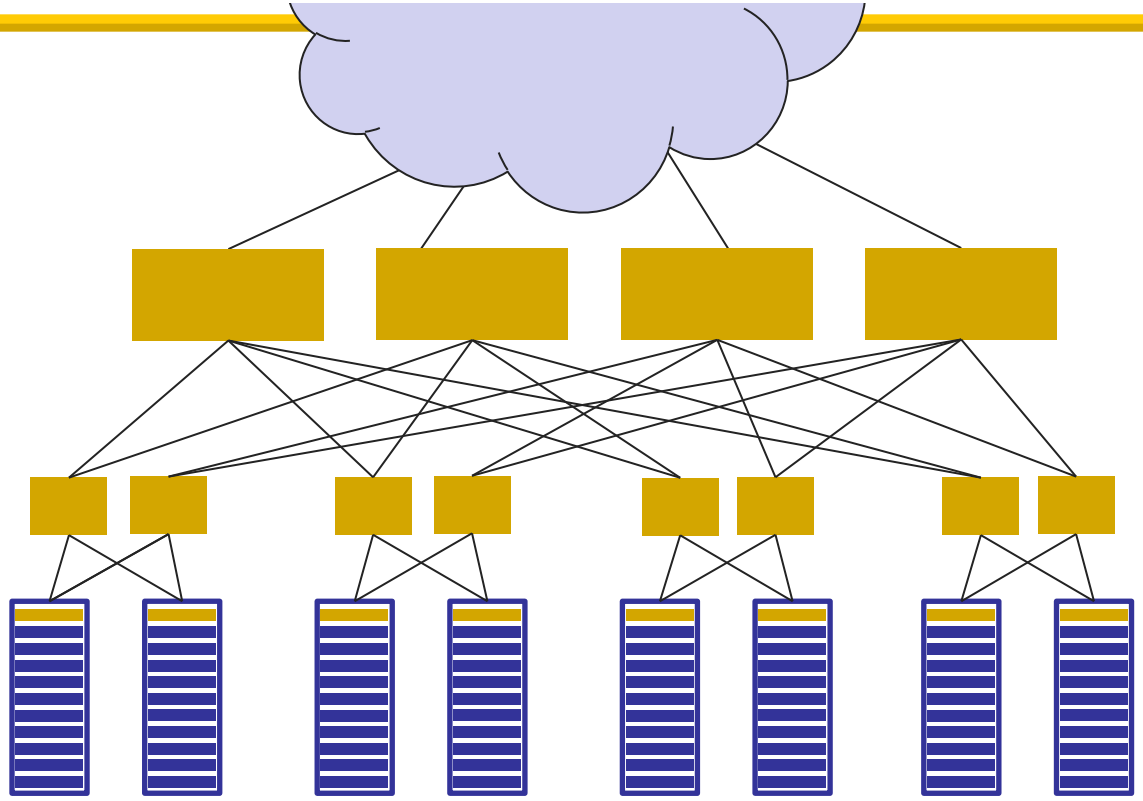
Datacenter networks (Cont.)



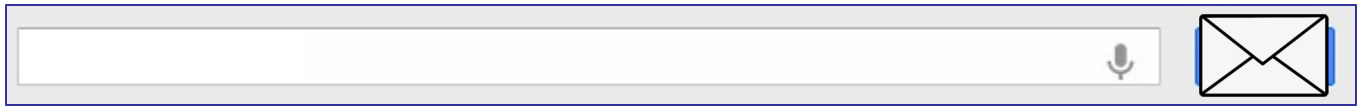
Challenges

- Not enough bandwidth
  - Oversubscription: Less bandwidth in the ToR-Agg links than all the servers bandwidth in the rack
  - Oversubscription ratio: Ratio between bandwidth underneath and bandwidth above
- Not enough paths between server pairs
  - Load balancing issues
  - Failure recovery issues

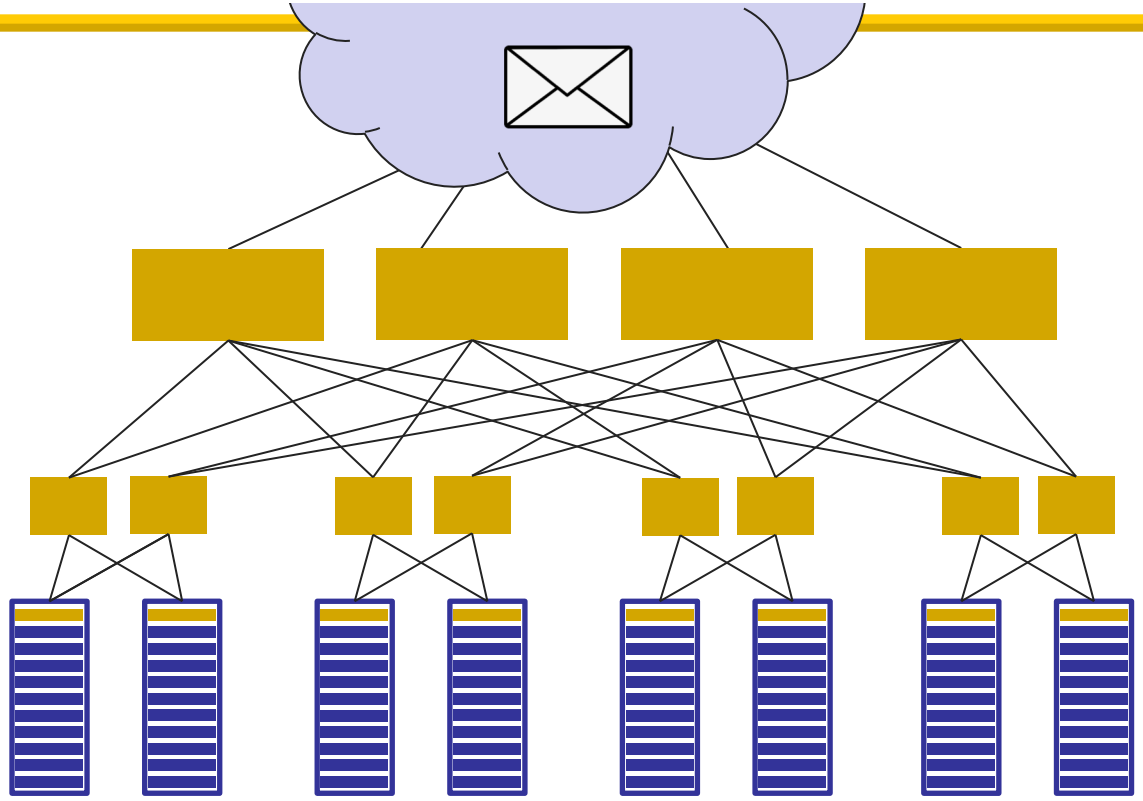
Modern datacenter networks: More bandwidth, more paths



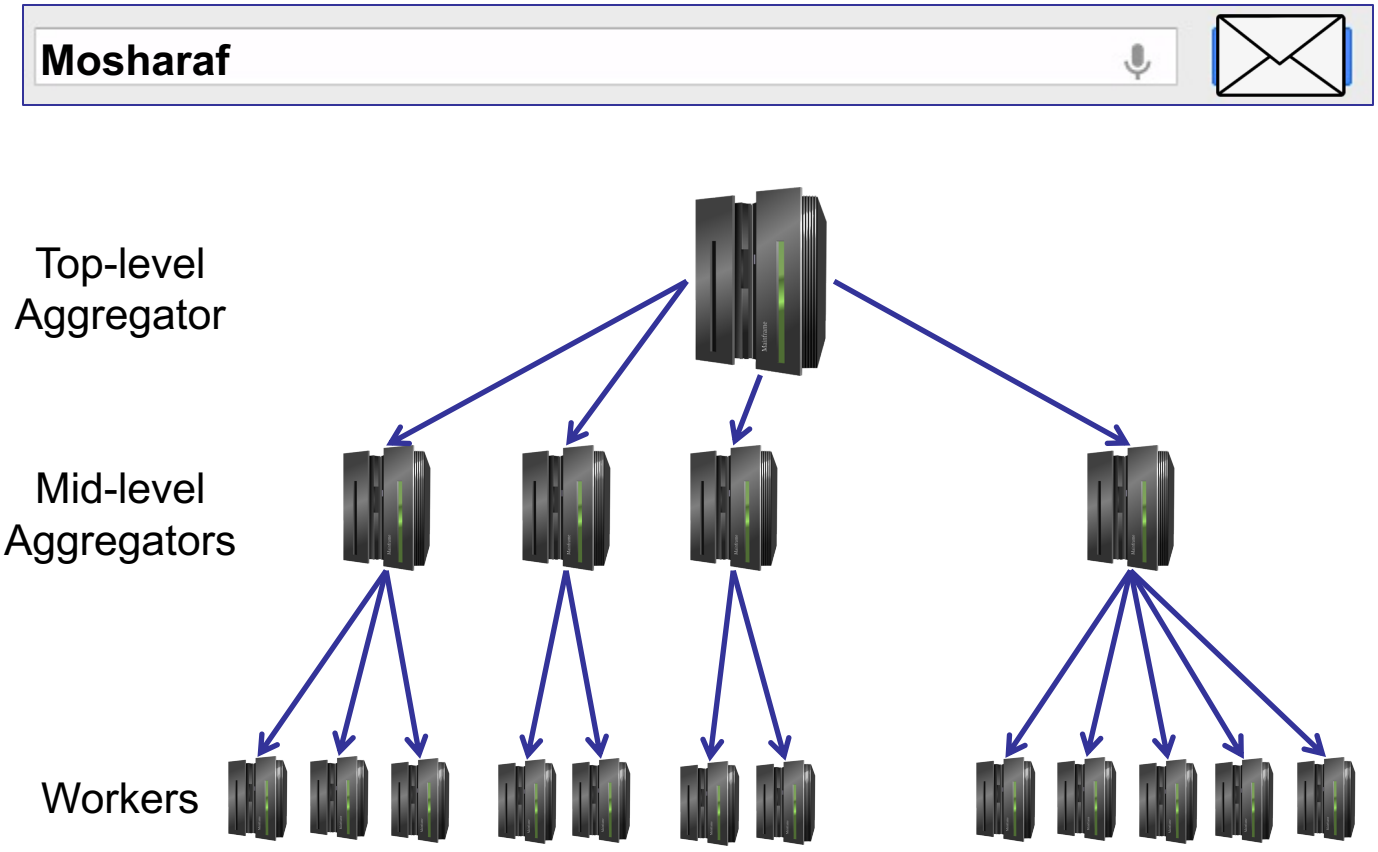
Let's take a Web service



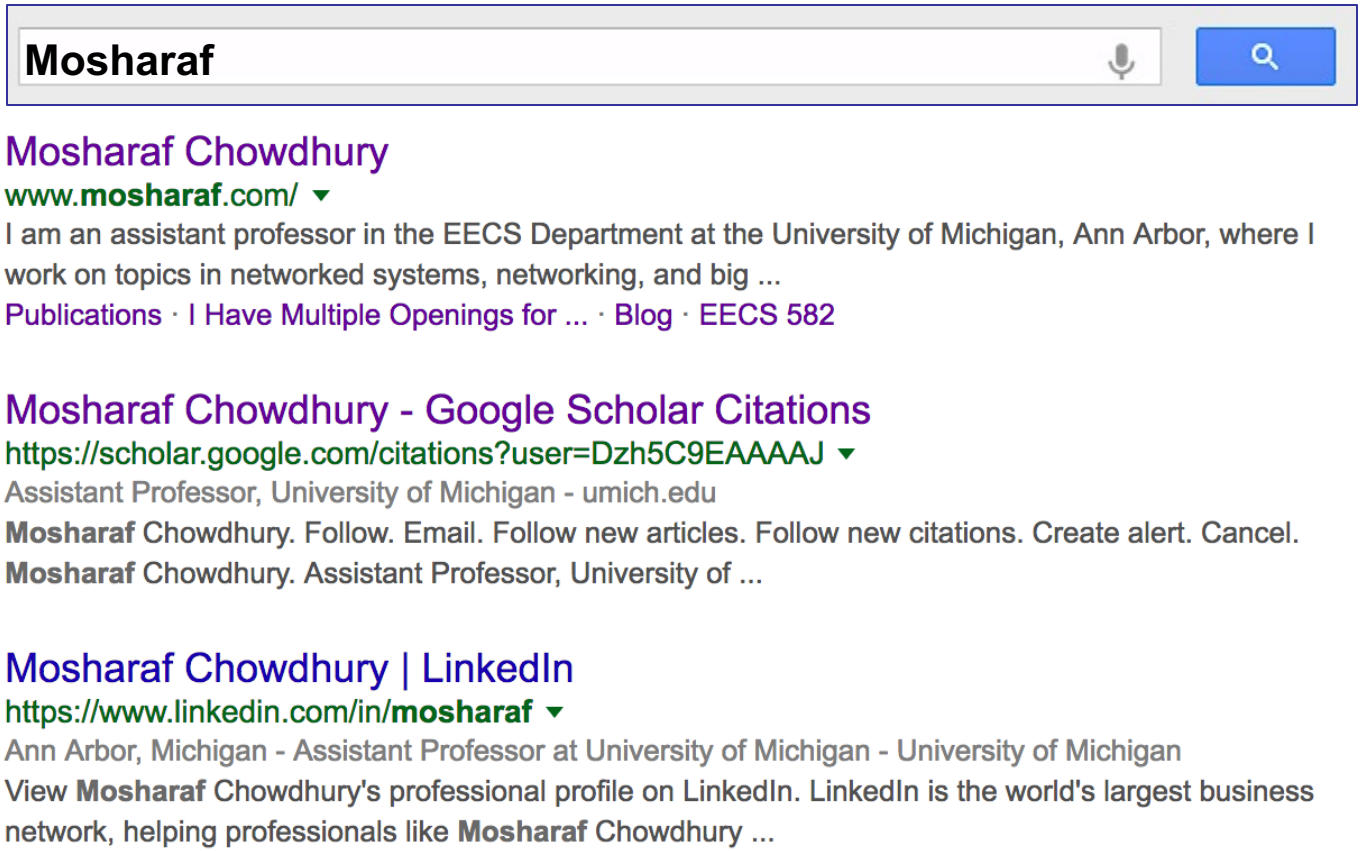
Who's serving Web services?



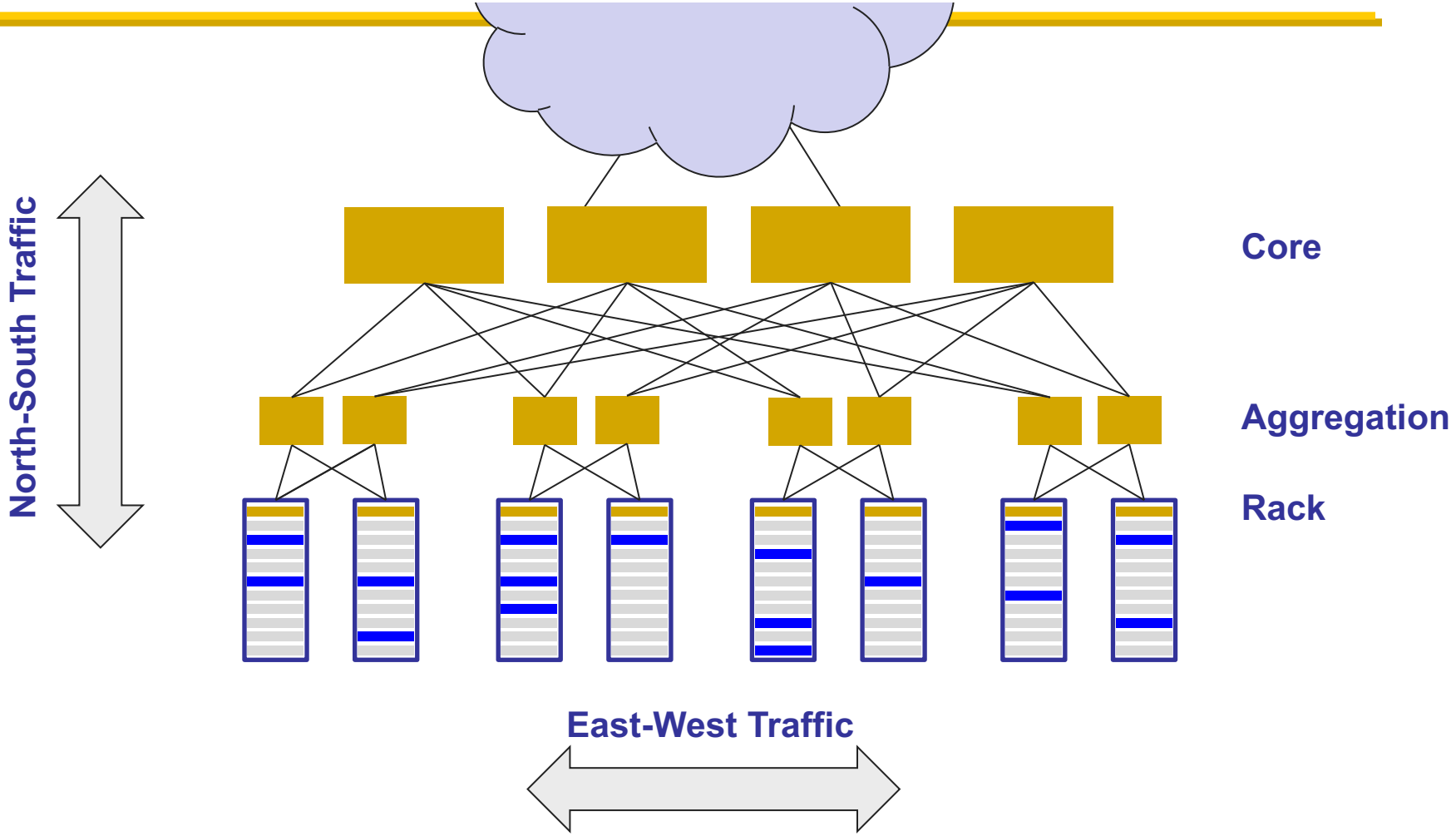
What happens inside?



Partition-Aggregate traffic



Partition-Aggregate traffic



End-to-end response time

- Less than 200 milliseconds between receiving user query in the browser and displaying the results
  - RTT = O(10) to 100 milliseconds
  - What remains?

- Next time, when the page is not loading fast enough, think about the poor servers working for you ☺

Summary

- Video streaming
  - Too large to send as a whole, so stream it
  - Dynamically adapt to the network and users
- Cloud applications
  - Forms the backend of modern web services
  - Where all the processing happens