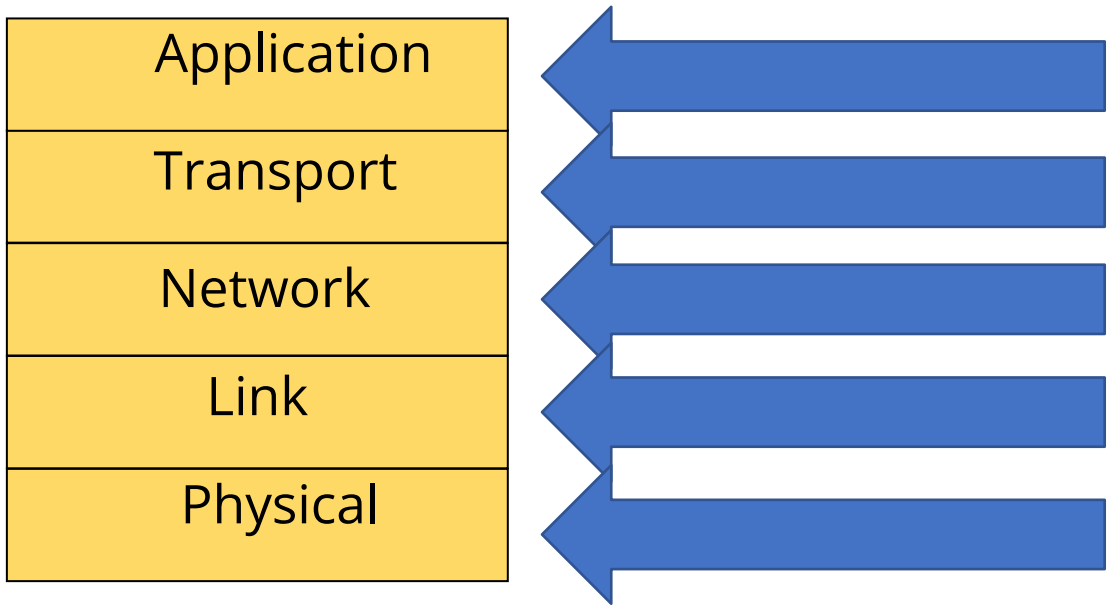
Where we are in the Course

Today: Ownership, and Perf!

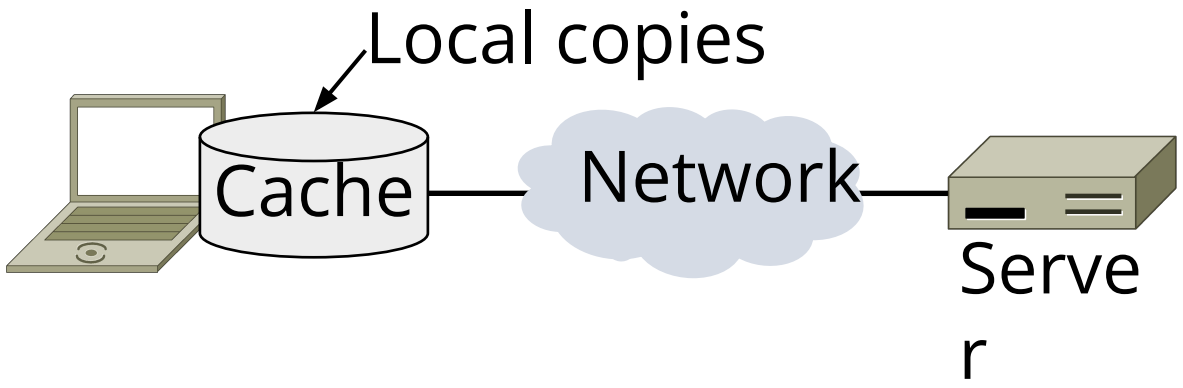


Web Caching/CDNs/Edge

(In the context of security + perf)

Web Caching

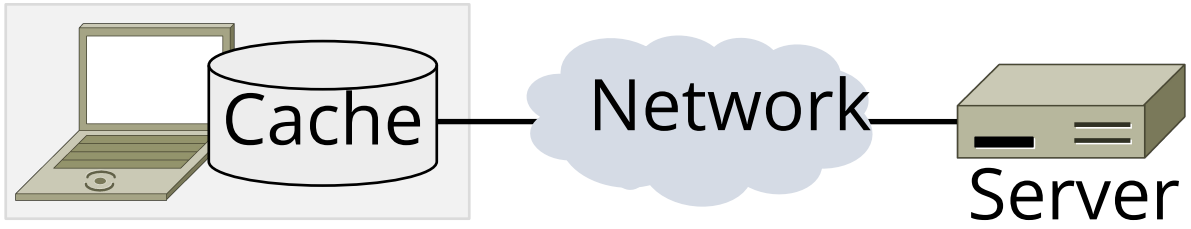
- Users often revisit web pages
  - Big win from reusing local copy!
  - This is caching



- Key question:
  - When is it OK to reuse local copy?

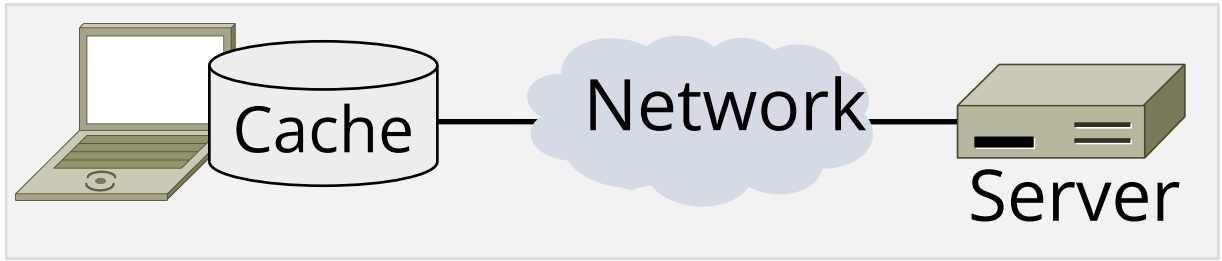
Web Caching (2)

- Locally determine copy is still valid
  - Based on expiry information such as “Expires” header from server
  - Or use a heuristic to guess (cacheable, freshly valid, not modified recently)
- Content is then available right away
  - No request needed!



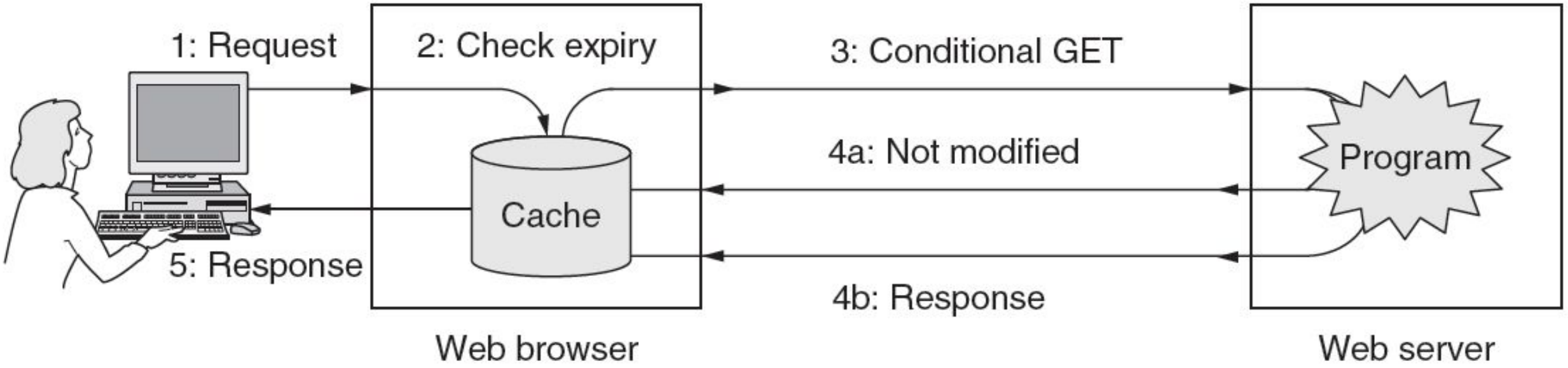
Web Caching (3)

- Revalidate copy with remote server
  - Based on timestamp of copy such as “Last-Modified” header from server
  - Or based on content of copy such as “Etag” server header
  - Content is available after 1 RTT



Web Caching (4)

- Putting the pieces together:

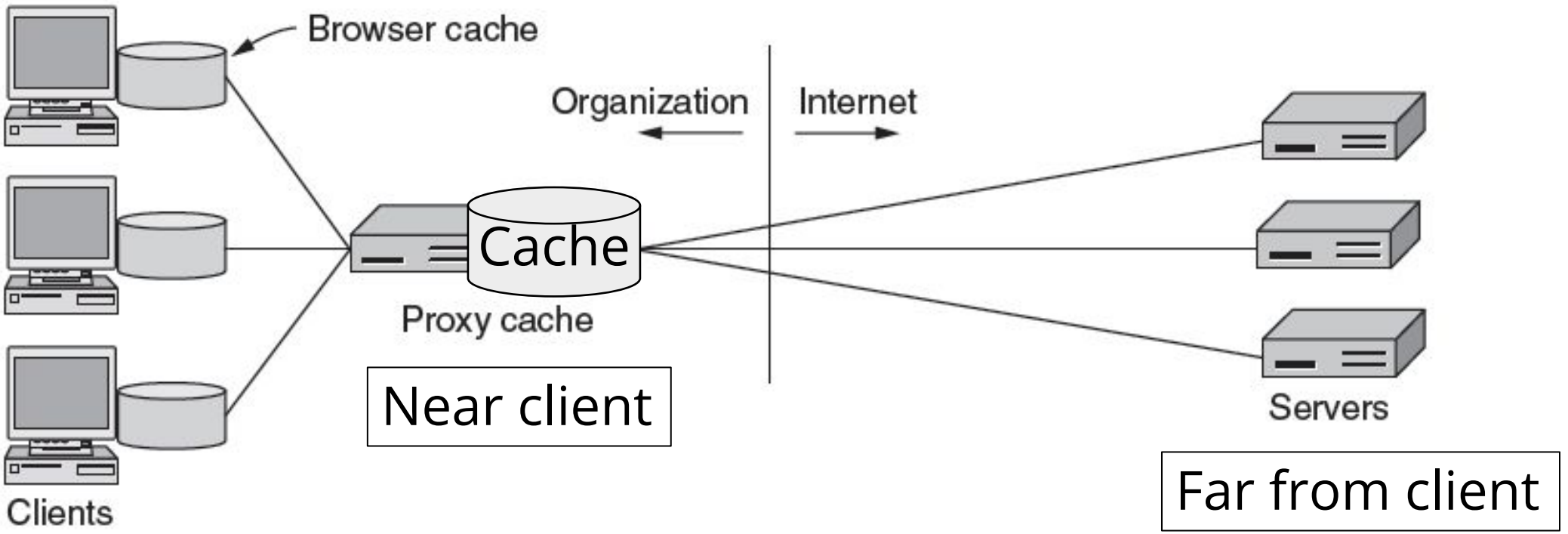


Web Proxies

- Place intermediary between pool of clients and external web servers
  - Benefits for clients include caching and security checking
  - Organizational access policies too!
- Proxy caching
  - Clients benefit from larger, shared cache

Web Proxies (2)

- Clients contact proxy; proxy contacts server



Big Question…

- How does a proxy cache interact with end-to-end HTTPS?
- Does it work at all? Why or why not?

Proxy caching largely dead for general users in 2025 : (

In corporate/controlled environments, can install root certificates on all host machines to allow a cache to intercept all HTTPS traffic!
Serious security downsides…

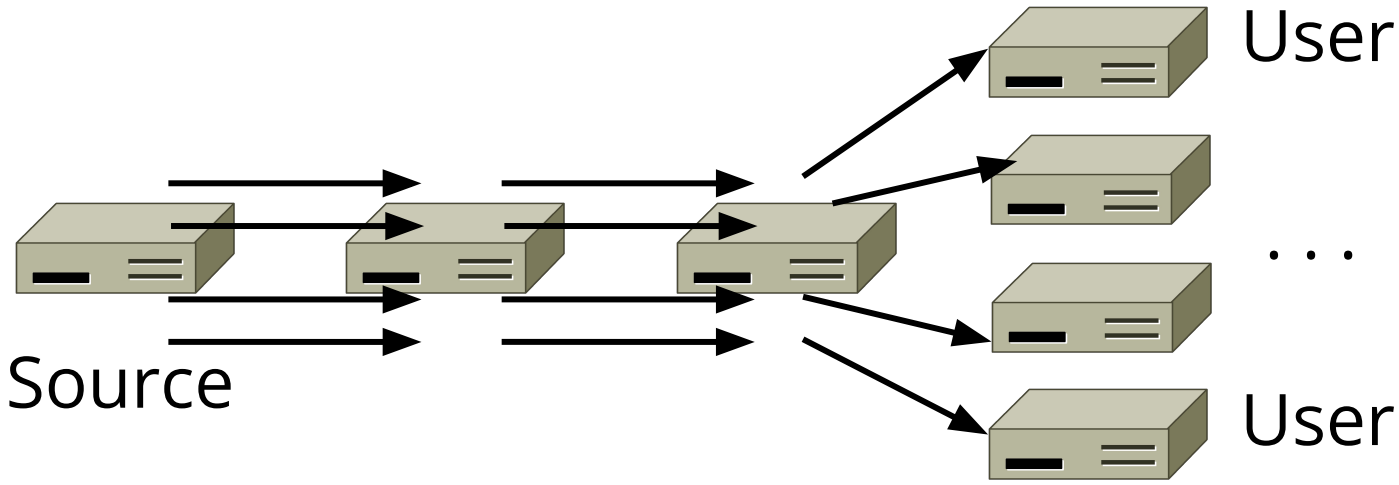
Users can also opt-in to a cache (no longer a passive middlebox), but lose the e2e benefits of https

Content Delivery Networks + Edge

- As the web took off in the 90s, traffic volumes grew and grew. This:
  1. Concentrated load on popular servers
  2. Led to congested networks and need to provision more bandwidth
  3. Gave a poor user experience
- Idea:
  - Place popular content near clients
  - Helps with all three issues above

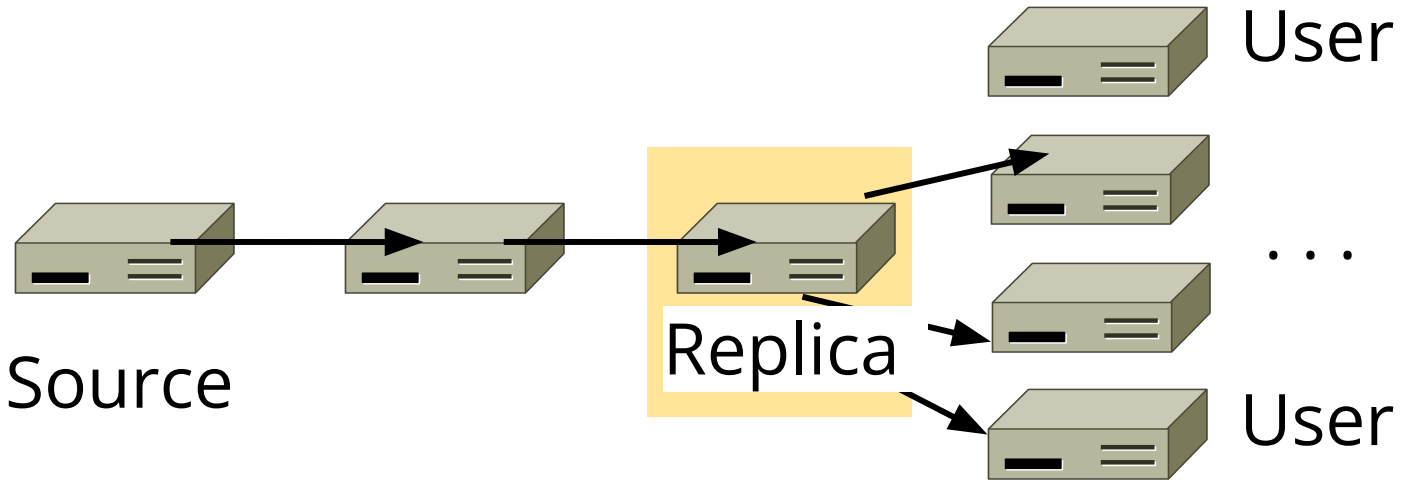
Before CDNs

- Sending content from the source to 4 users takes 4 x 3 = 12 “network hops” in the example



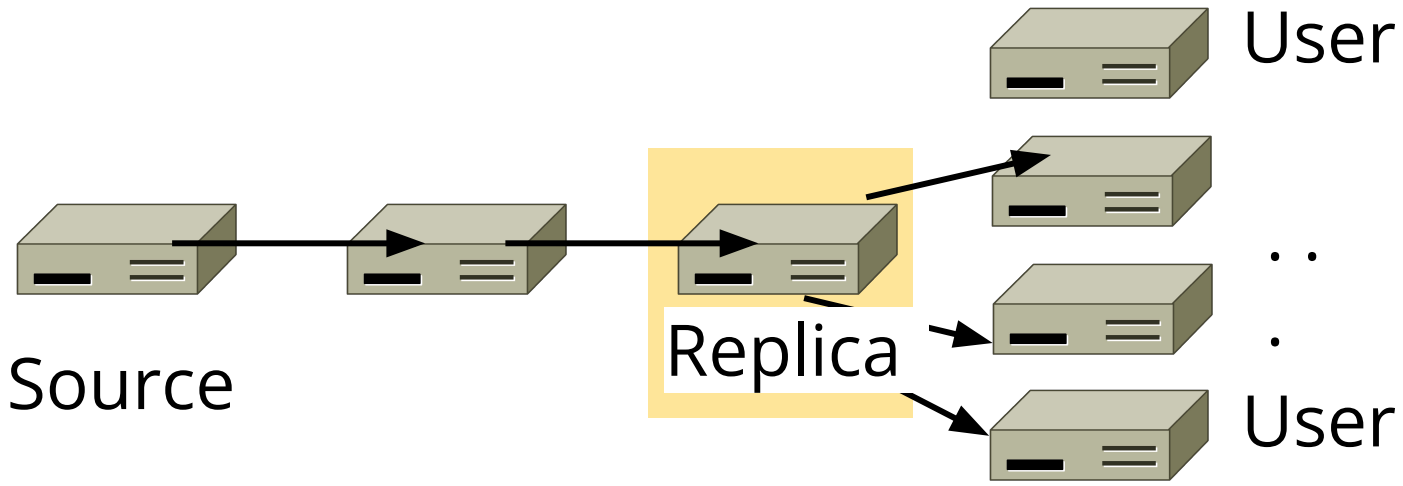
After CDNs

- Sending content via replicas takes only 4 + 2 = 6 “network hops”



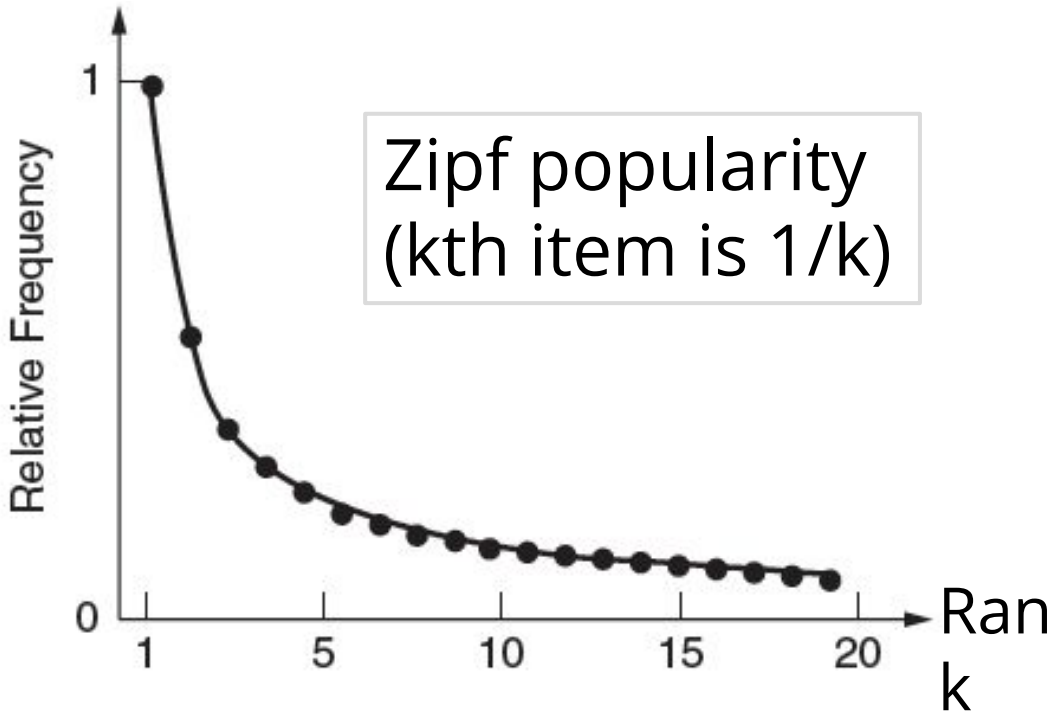
After CDNs (2)

- Benefits assuming popular content:
  - Reduces server, network load
  - Improves user experience



Popularity of Content

- Zipf's Law: few popular items, many unpopular ones…
  - "Zipfian distribution"
  - both matter!



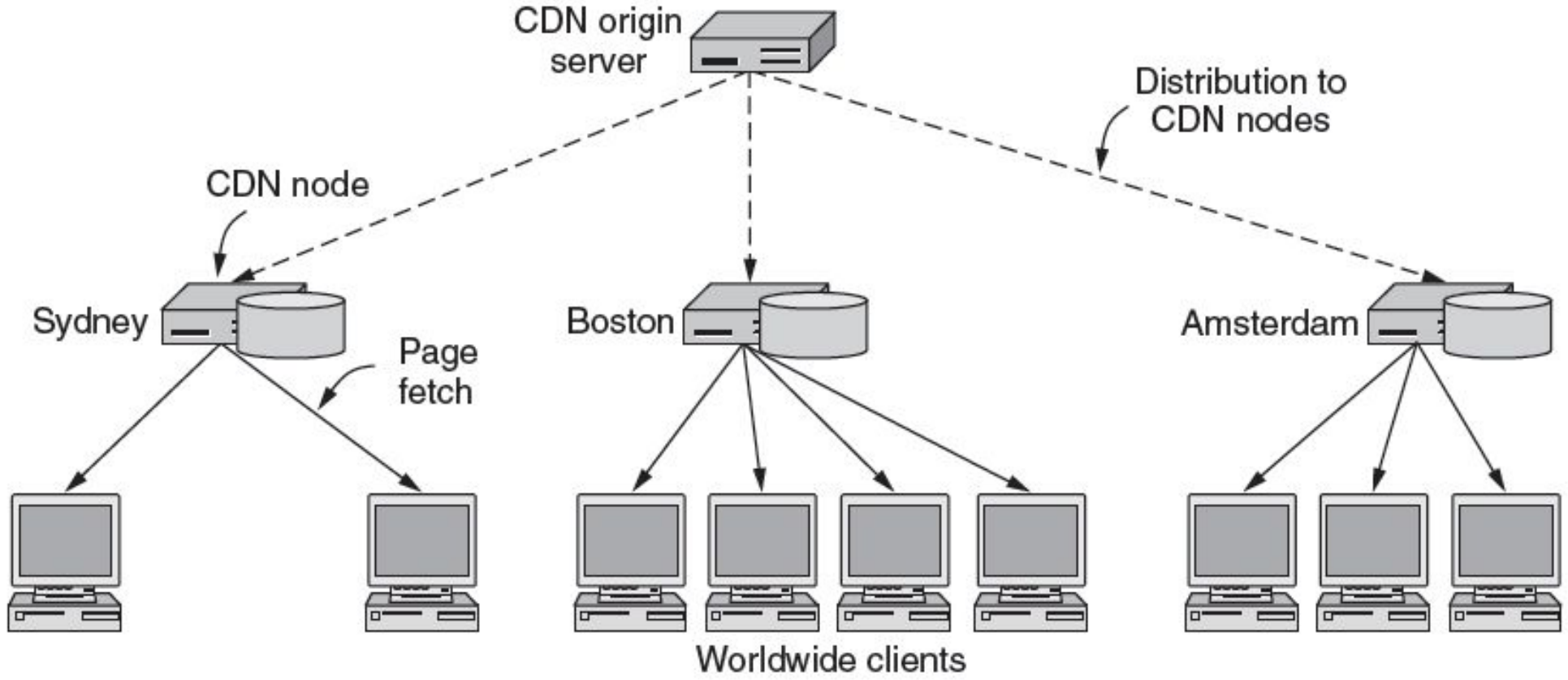
George Zipf (1902-1950)

Source: Wikipedia

How to place content near clients?

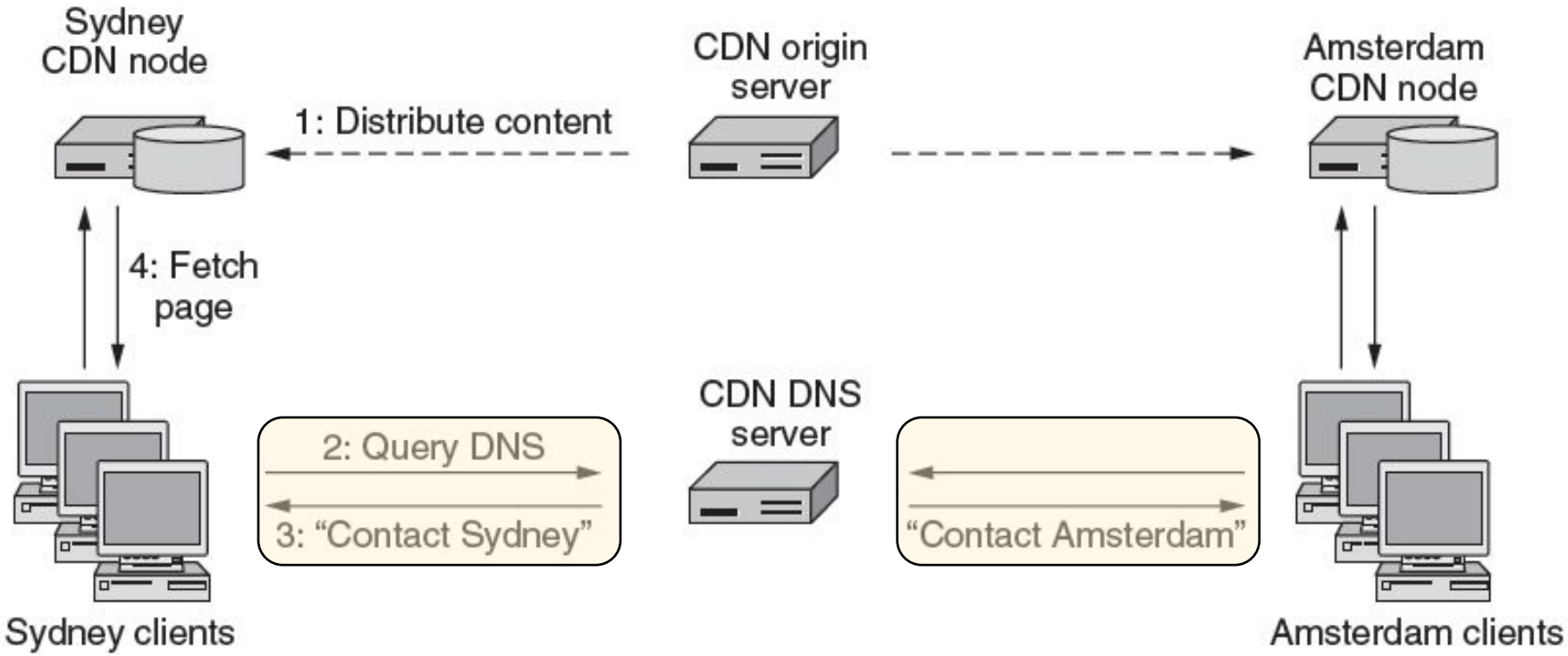
- Use browser and proxy caches
  - Helps, but limited to one client or clients in one organization
- Want to place replicas across the Internet for use by *all nearby clients*
  - Often done by clever use of DNS

Content Delivery Network



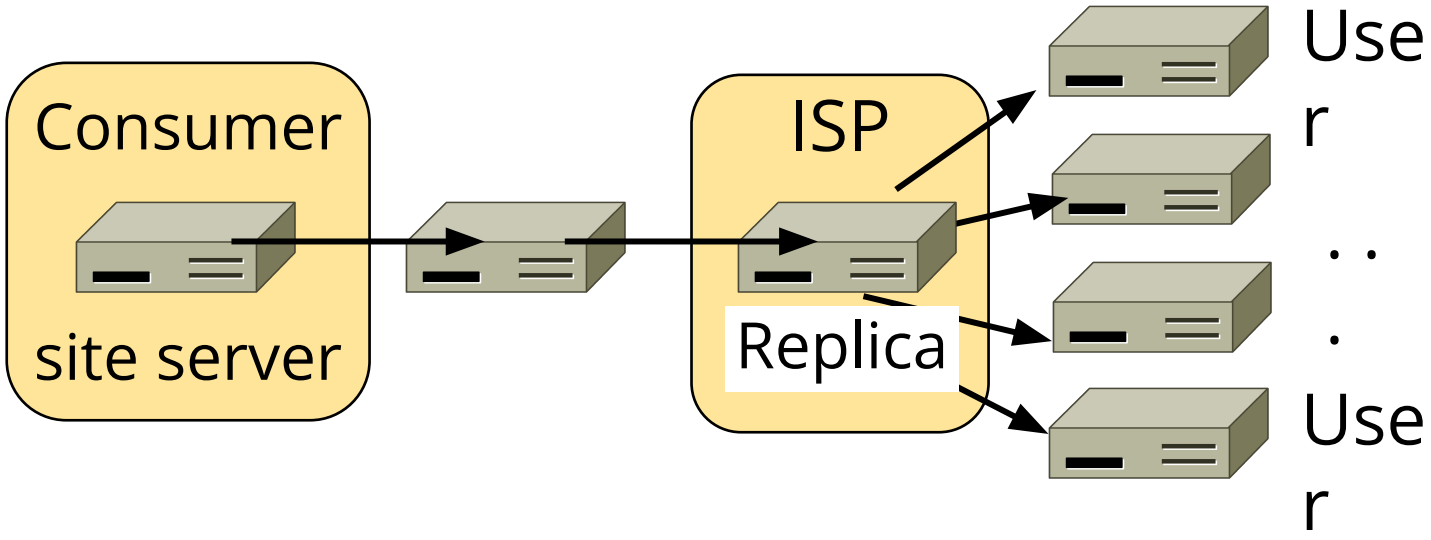
Content Delivery Network (2)

- DNS gives different answers to clients
  - Tell each client the nearest replica (map client IP)



Business Model

- Clever model pioneered by Akamai
  - Placing site replica at an ISP is win-win
  - Improves site experience and reduces ISP bandwidth usage



Edge Compute

- A generalization of the CDN big-idea
- Move storage *and/or compute* closer to the end user
  - Within the ISP
  - Or even closer (at the cell tower, or in each neighborhood)
- Improves latency b/c of physics and fewer network hops
- Increases complexity significantly
  - Operating + maintaining many machines across environments!
  - Need to connect users to the correct edge resource dynamically

CDN & Edge: Issues

- Security
  - What about private information?
  - How to cache/forward encrypted content?
    - Basically can't! Big players just share/ship keys.
- Net neutrality
  - internet.org, FreeBasics --> Basically CDNs
    - But for reasons of *price*, not efficiency
  - Who decides who gets to place CDNs?
    - What if Netflix wants to give you edge hardware for free, and you refuse?
      - (Actually happened during peering controversy in early 2010s)

What makes CDNs work when Proxy Caches don't?

- Both are possible due to the design *affordances* of the network, the Internet

affordances

Plural form of **affordance**

noun

1. A **potential action** that is made possible by a given object or **environment**; especially, one that is made easily discoverable.

Who decides what the network can and cannot do?

Many stakeholders…

- Users
- Regulators
- Standards Bodies
- Equipment makers
- Network Operators

Users

- Vote with their wallets, eyeballs, and actions
- Collectively have the most power, individually the least

Regulators

Make policy around spectrum, rules of operation

- ITU
  - International body organized by the UN, frequency plans, high-level coordination for international comms
- FCC (USA)
  - Regulates communications, devices, content, “FCC Approval”
- FTC (USA)
  - Regulates large industries, monopolies, fair competition
- Cities + States
  - Local regulations around physical access, RF energy, beauty, etc.

Standards Bodies

- Make technical specs, technical decisions
  - IETF:
    - “Internet Engineering Task Force”, usually L3+, global
    - Open to anyone! You can go!
  - IEEE
    - Usually L1 and L2: “WiFi”, “Ethernet”
    - Also open, higher barrier to entry
  - 3GPP
    - Cellphones, L1, L2, some application, cross-references IEEE and IETF
    - No individual participation, equipment vendors + operators only
  - Many others…

Making decisions at the IETF

- Open participation, so can be tough to make progress
- Strive for “rough consensus and running code”
  - https://en.wikipedia.org/wiki/Rough_consensus
  - *“Note that 51% of the working group does not qualify as "rough consensus" and 99% is better than rough. It is up to the Chair to determine if rough consensus has been reached”*
  - *“supporting percentage is less important for determining "rough consensus" than ensuring opposing views are addressed”*
  - Humming used to decide big issues… accepting QUIC, IPv6, etc…
  - Two implementations built through “hackathons”

IETF Hum Demo

Equipment Makers

- Build to standard, or don't implement everything!
- Often concerns around patents since this is a technical and fast-moving field
- Choice to be more open or more proprietary
  - Aside: Open Source software licensing for network code is complicated… see "Affero" versions of the GPL
  - Can implement to standards to allow interoperability even if proprietary implementation
  - "Plug Fests" to test out compatibility, organized by IEEE, 3GPP, and others

Network + Service Operators

- Decide what to build were
- Decide how much to charge for service
- Decide the *structure* of the service
  - Symmetric, asymmetric, prepaid, postpaid, etc.
- Need to meet regulations
- Rely on standards to purchase compatible equipment

You!

- As someone sitting in this room…
  - You're uniquely on a path where you could end up in any (or maybe even all) of these roles!
  - Important to keep in mind the implications of your technical work on the real-world outcomes of the people and planet around you!

To close out the quarter, a brief moment of reflection

- Take 3 minutes, individually, and think through what your big takeaways are for this quarter:
  - What nuggets of information do you want to remember 2 weeks from now? 6 months from now? 6 years from now?
- Now form small groups, and share the takeaways you are comfortable to share with your peers…
  - Any good ones others came up with you want to add?

🎉Thanks for a great quarter🙏

See you all next Thursday, or at office hours before!

If you enjoyed this class, check out the datacenter systems class which also heavily involves networking, or distributed systems, to build systems on top of networks!