CSC 450 - COMPUTER NETWORKS

Design Philosophy &
Applications

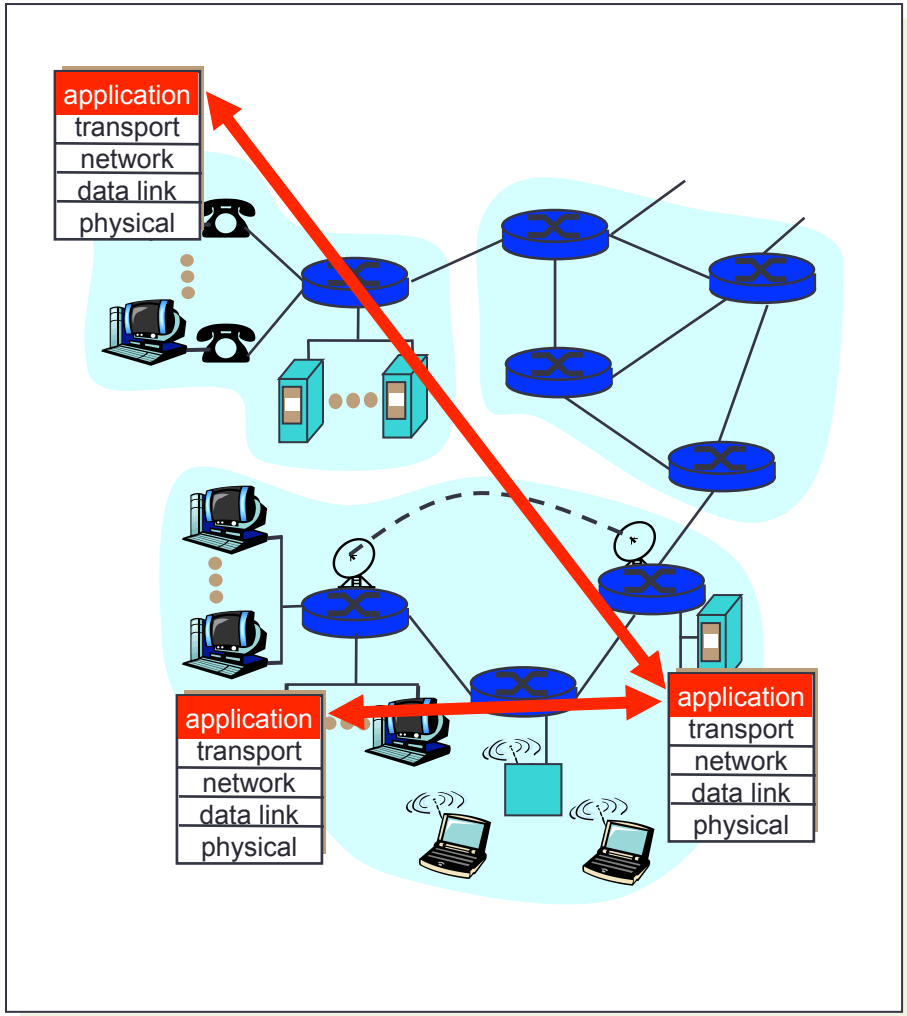
Part 2

Recap

- Application Protocol: HTTP
- Parallel vs Persistent connections (pipelines)

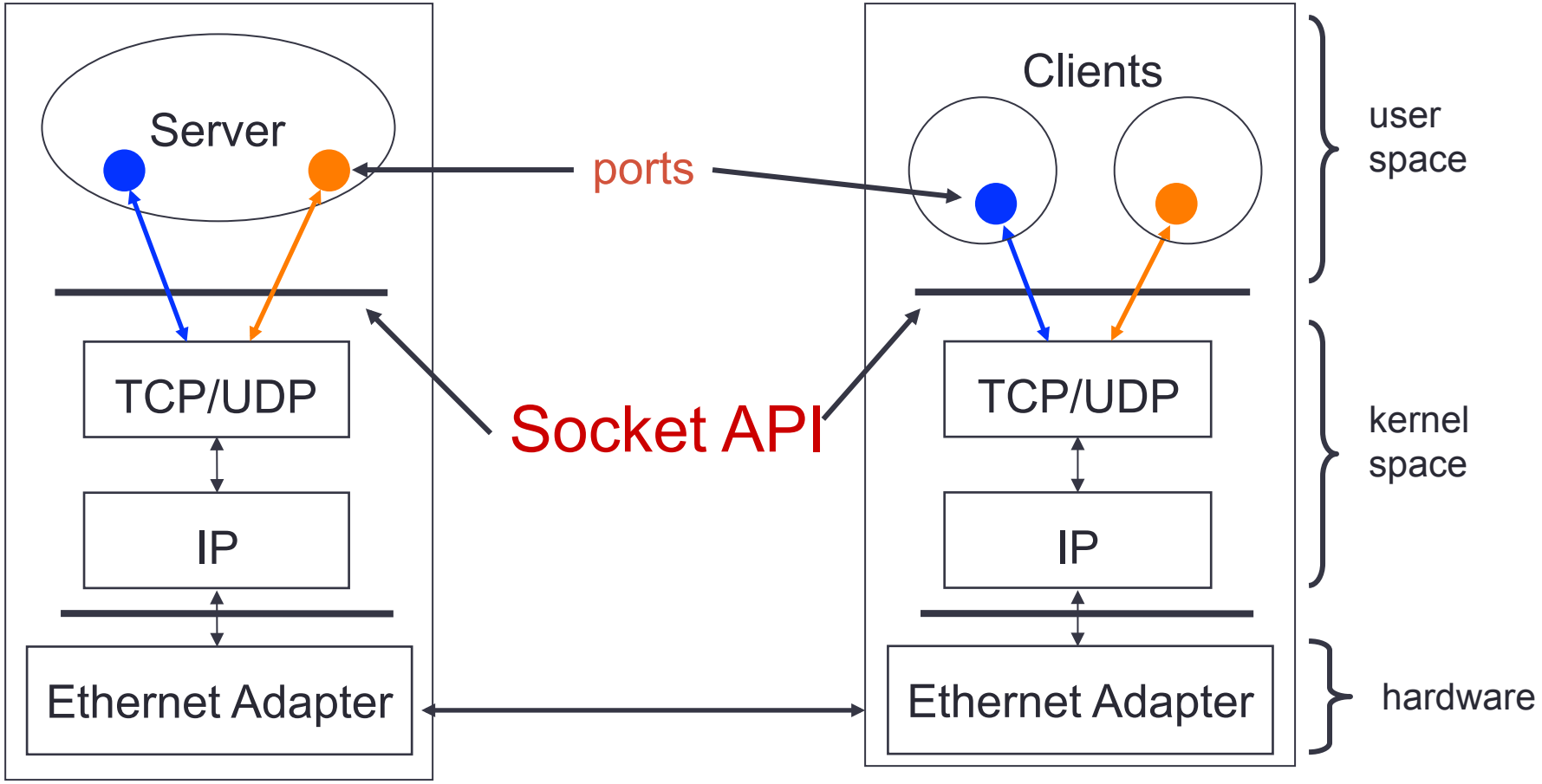
Applications and Application-Layer Protocols

- Application: communicating, distributed processes
  - Running in network hosts in "user space"
  - Exchange messages to implement app
  - e.g., email, file transfer, the Web
- Application-layer protocols
  - One "piece" of an app
  - Define messages exchanged by apps and actions taken

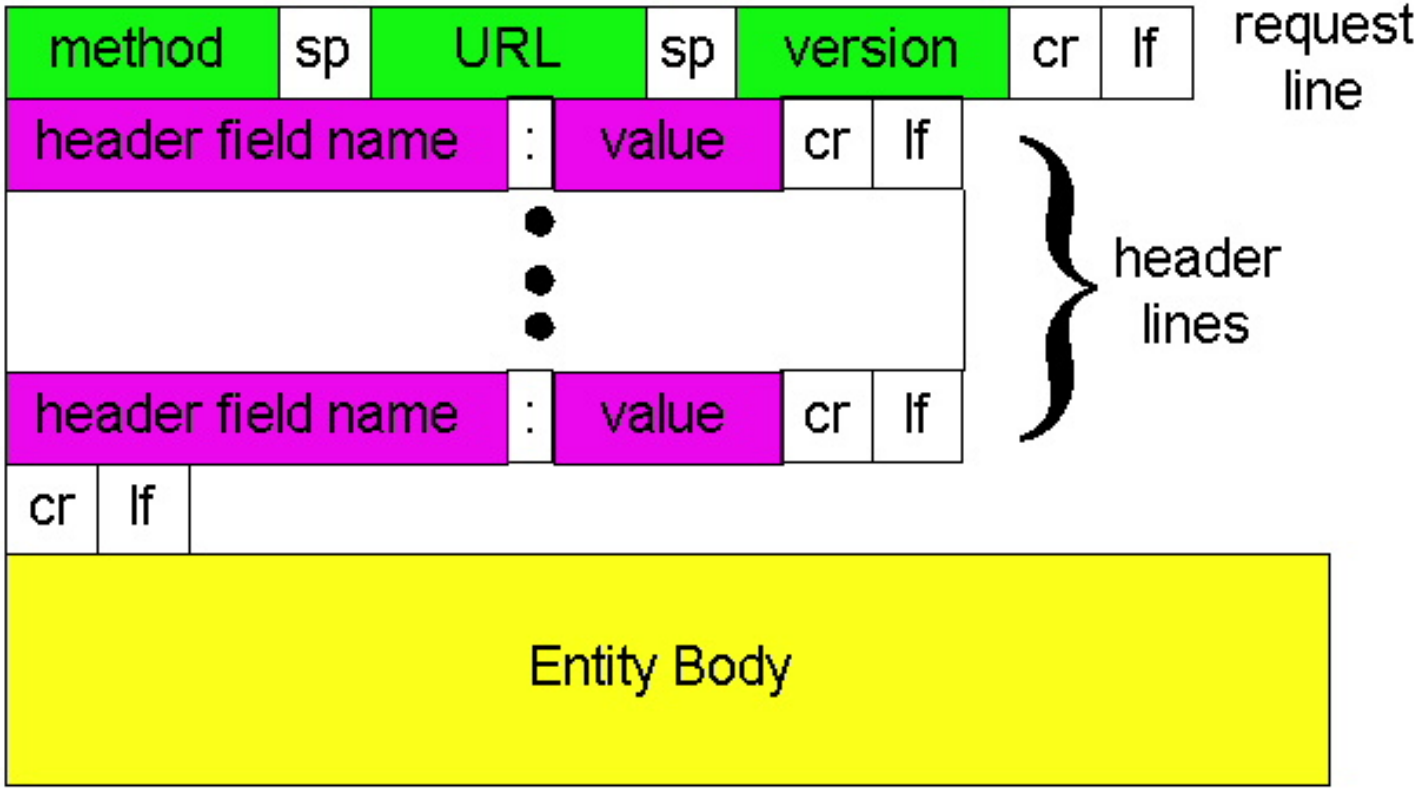


Server and Client

Server and Client exchange messages over the network through a common Socket API

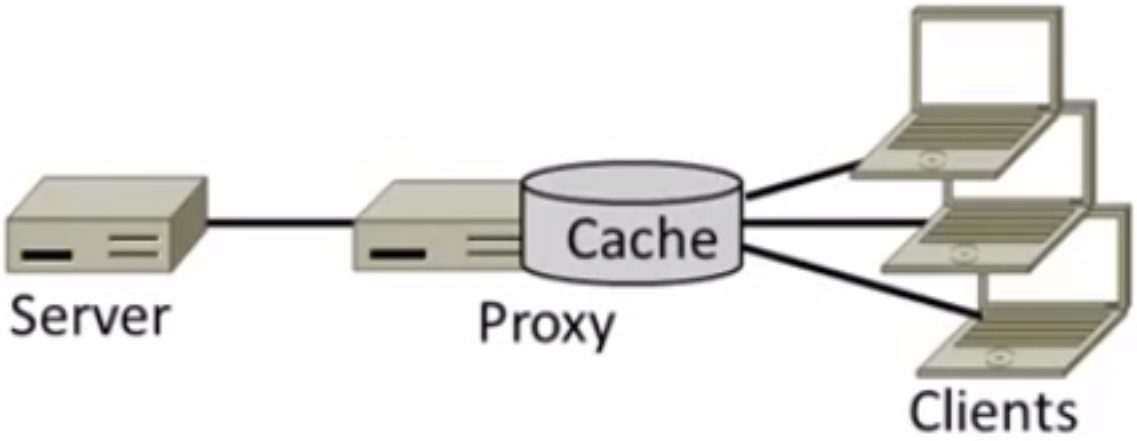


HTTP Request



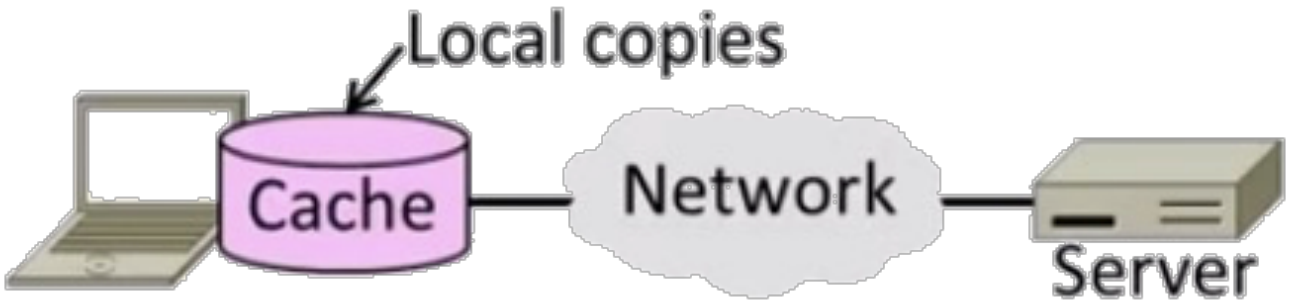
HTTP caching and proxies

- Enabling content reuse



Motivation: Web Caching

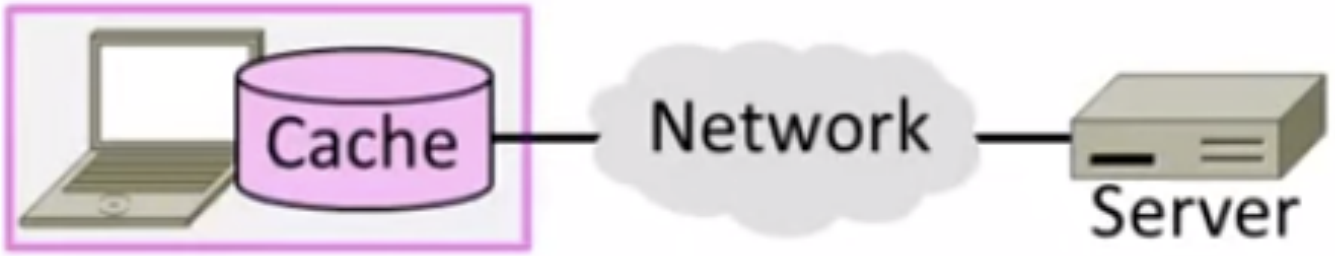
- Users often revisit web pages
  - Big win from reusing local copy!
  - This is caching



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

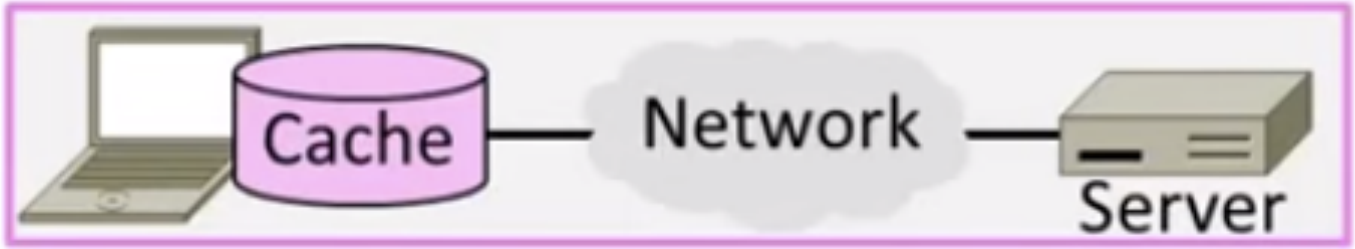
Web Caching

- 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

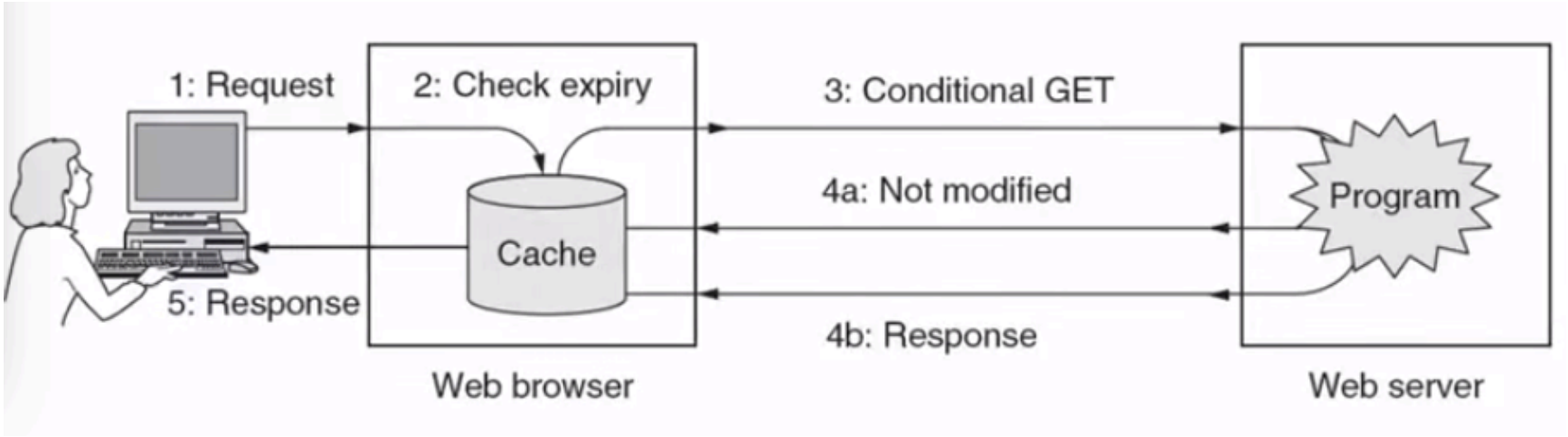


Web Caching

- 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" header from server
  - Content is available after 1 RTT



Putting the pieces together

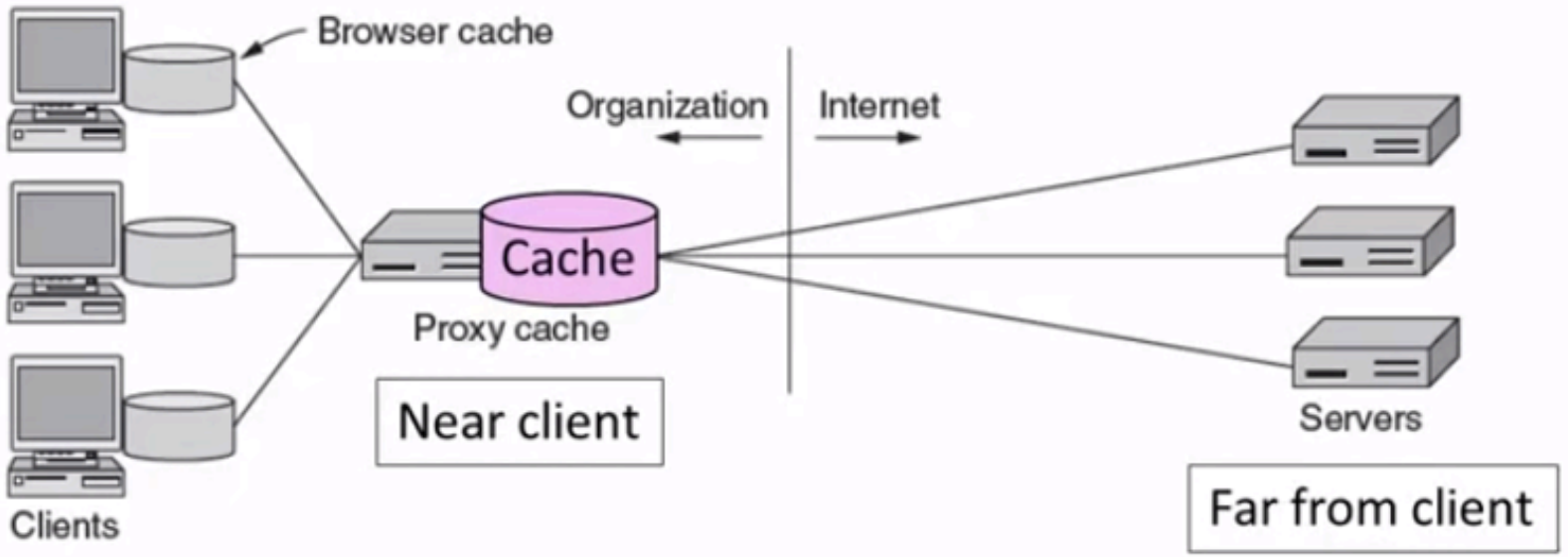


Web Proxies

- Place intermediary between pool of clients and external web servers
  - Benefits for clients include greater caching and security checking
  - Organizational access policies too!
- Proxy caching
  - Clients benefit from larger, shared cache
  - Benefits limited by secure / dynamic content, as well as “long tail”

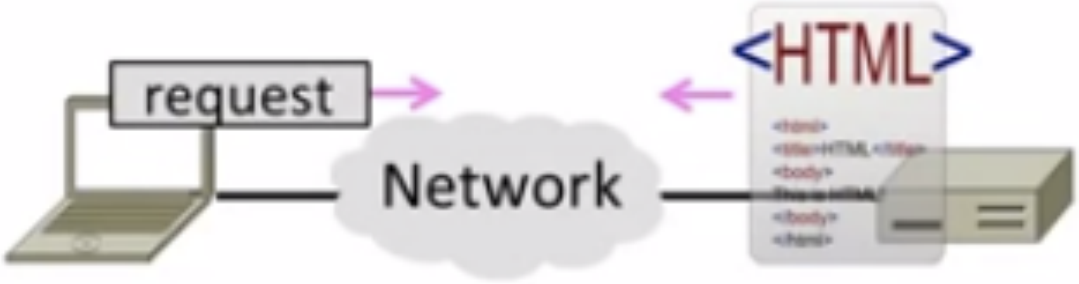
Web Proxies

- Clients contact proxy; proxy contacts server



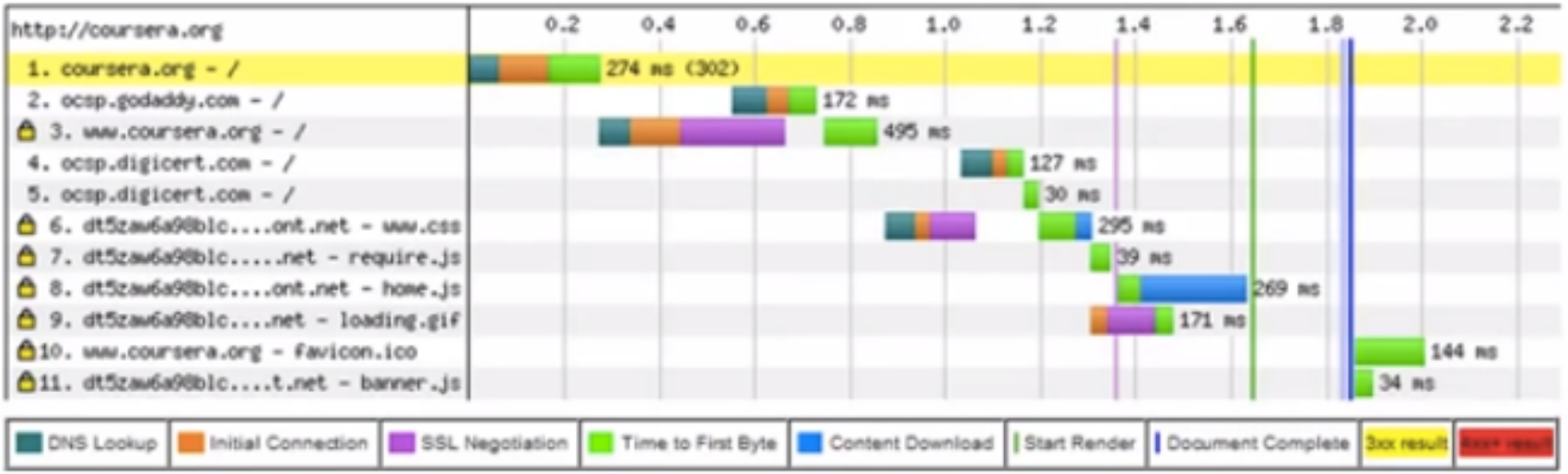
Future of HTTP

- How will we make the web faster?
- A brief look at some approaches



Modern Web Pages

- Waterfall diagram shows progression of page load



webpagetest tool for http://coursera.org (Firefox, 5/1 Mbps, from VA, 3/1/13)

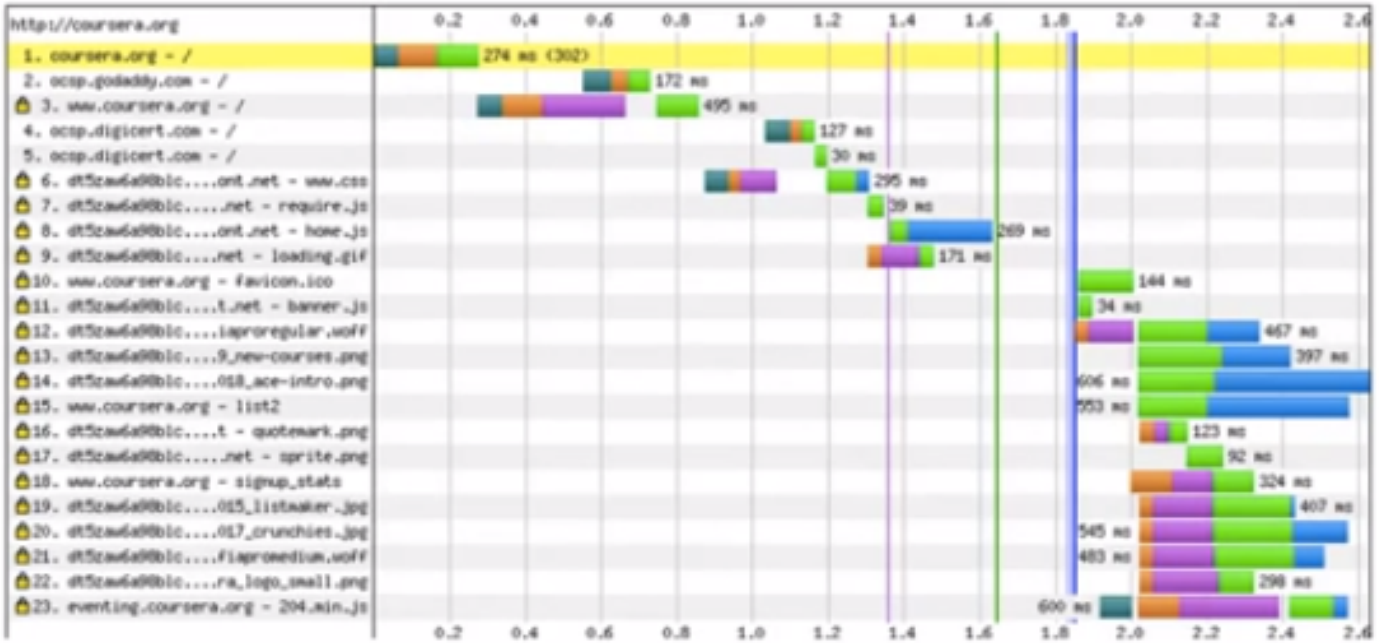
Modern Web Pages

Yikes!

-23 requests

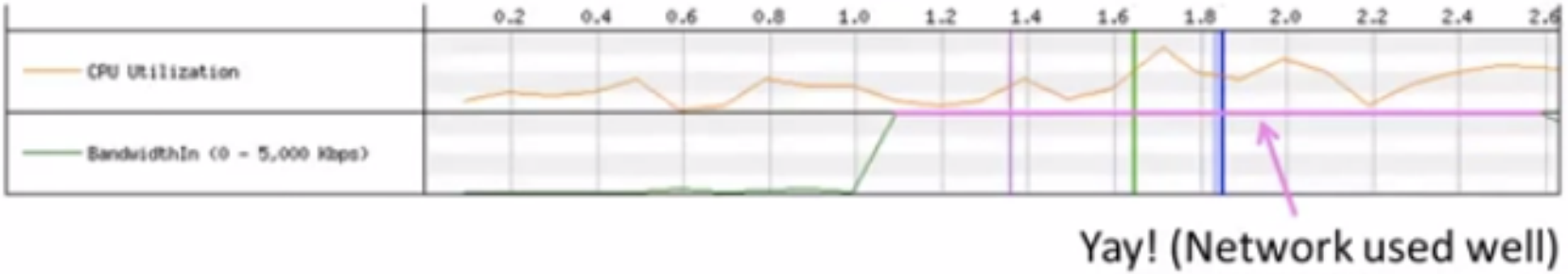
-1 Mb data

-2.6 secs



webpagetest tool for http://coursera.org (Firefox, 5/1 Mbps, from VA, 3/1/13)

Modern Web Pages



- Waterfall and PLT depends on many factors
  - Very different for different browsers
  - Very different for repeat page views
  - Depends on local computation as well as network

Recent work to reduce PLT

Pages grow ever more complex!

- Larger, more dynamic, and secure
- How will we reduce PLT?

1. Better use of the network
   - HTTP/2 effort based on SPDY
2. Better content structures
   - mod_pagespeed server extension

SPDY ("speedy")

- A set of HTTP improvements
  - Multiplexed (parallel) HTTP requests on one TCP connection
  - Client priorities for parallel requests
  - Compressed HTTP headers
  - Server push of resources
- Now being tested and improved
  - Default in Chrome, Firefox
  - Basis for an HTTP/2 effort

mod_pagespeed

- Observation:
  - The way pages are written affects how quickly they load
  - Many books on best practices for page authors and developers
- Key idea:
  - Have server re-write (compile) pages to help them load quickly!
  - mod_pagespeed is an example

mod_pagespeed

- Apache server extension
  - Software installed with web server
  - Rewrites pages “on the fly” with rules based on best practices
- Example rewrite rules:
  - Minify Javascript
  - Flatten multi-level CSS files
  - Resize images for client
  - And much more (100s of specific rules)

Email Applications

- Application Protocols
  - HTTP
  - Email
    - POP3 (Post Office Protocol Version 3)
    - IMAP (Internet Message Access Protocol)
    - SMTP (Simple Mail Transfer Protocol)