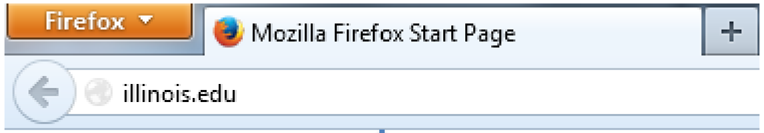
DNS and HTTP

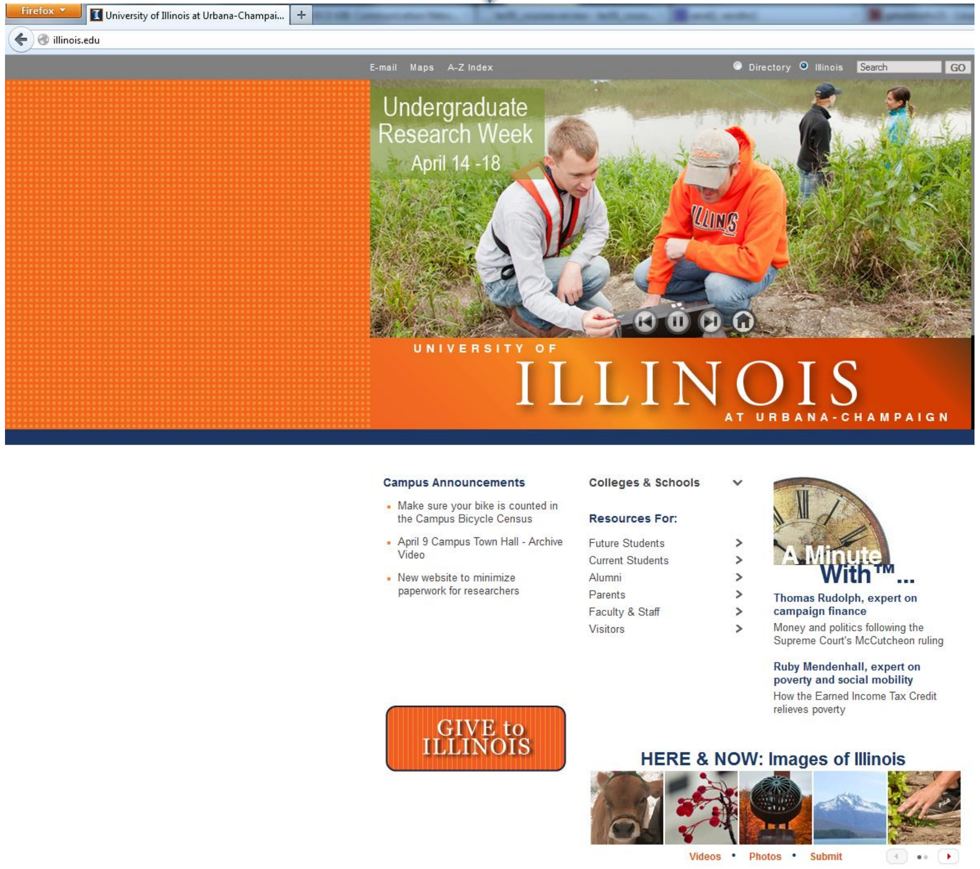
Finally, the application layer!

- We have learned about:
  - Signals being sent on wires
  - Frames carried over dumb local networks
  - Packets carried over the entire internet
  - Making communication useably reliable + efficient

WHAT’S THE POINT?

Today's Example





Undergraduate Research Week
April 14 -18

UNIVERSITY OF
ILLINOIS
AT URBANA-CHAMPAIGN

Campus Announcements

- Make sure your bike is counted in the Campus Bicycle Census
- April 9 Campus Town Hall - Archive Video
- New website to minimize paperwork for researchers

Colleges & Schools

Resources For:

Future Students
Current Students
Alumni
Parents
Faculty & Staff
Visitors

A Minute With™...

Thomas Rudolph, expert on campaign finance
Money and politics following the Supreme Court's McCutcheon ruling

Ruby Mendenhall, expert on poverty and social mobility
How the Earned Income Tax Credit relieves poverty

GIVE to ILLINOIS

HERE & NOW: Images of Illinois

Videos • Photos • Submit

Hypertext Transfer Protocol

To understand HTTP, let’s first look at HTML.

```
<html><head>

<title>Matthew Caesar</title>
<script text="text/javascript" src="jquery.min.js"></script>
<script text="text/javascript" src="data.gatherer.js"></script>
</head>
<body>
<img alt="" src="matt.jpg" style="width: 188px;" align="left"
hspace="20" vspace="20">
<h2 align="left">Matthew Caesar</h2>
Assistant Professor <br>
…
```

HTTP Performance

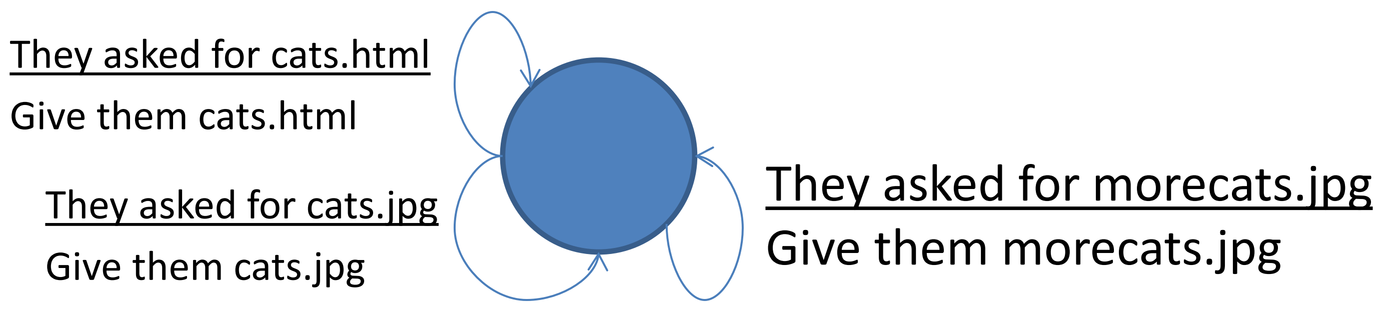
- What determines page load time?
  - Download time (large objects)
  - Latency (small objects)
    - Complex HTML structure
    - Early HTTP’s poor optimization
- How can we optimize these?

HTTP Optimizations

- Saving download time
  - Caching
    - If-Modified-Since
  - Caching proxies
- Saving round trips
  - Parallel connections
    - Supposed to be max 2 ☹
  - Reusing TCP connections (“Persistent TCP”)
  - SPDY: parallel HTTP without the parallel TCP

Statelessness, and the hacks to undo it

A vanilla HTTP server's FSM:



- Sounds like your Amazon shopping cart, right?
- The work-arounds
  - Javascript and pals, backend databases
  - Cookies
  - URL Parameters
    - Filtering http://www.newegg.com/Internal-SSDs/SubCategory/ID-636
    - http://www.newegg.com/Product/ProductList.aspx?Submit=ENE&N=100008120%204027%204017&IsNodeId=1&name=%24100%20-%20%24200
- Our needs are clearly beyond HTTP's original intent. Time to move on? (see "Embassies: Radically Refactoring the Web", best paper NSDI 2013)

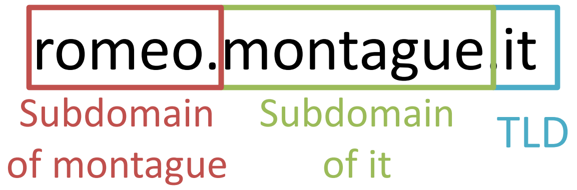
DNS – A simple goal

www.cs.illinois.edu ➔ 128.174.252.83

What’s in a domain name?

www.cs.illinois.edu

mail.google.com

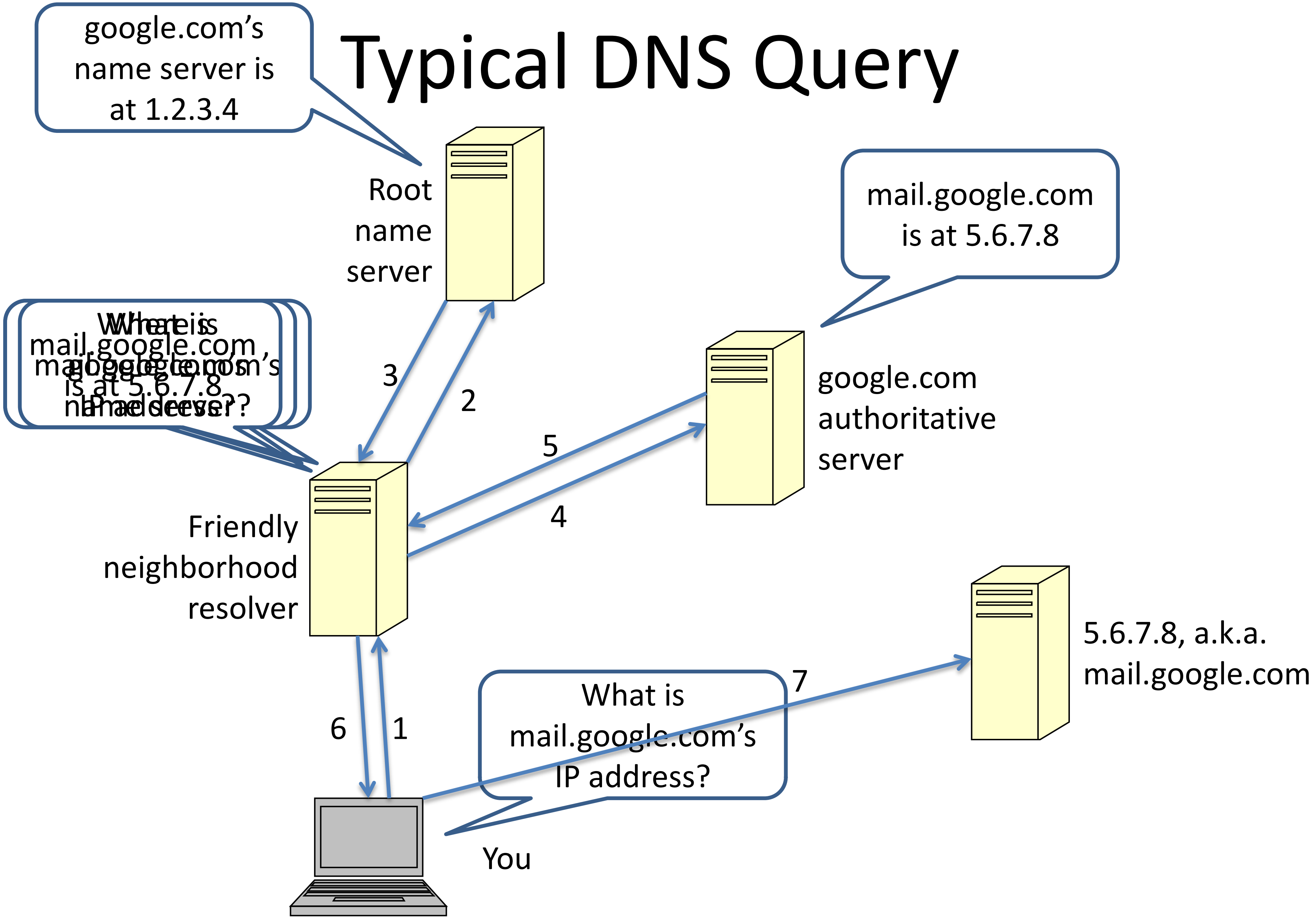


- Hierarchical names
- Hierarchical ownership (what makes it not flat)
  - Right to decide what IP a name resolves to
  - Right to delegate subdomains
  - Responsibility to help with resolution
    - Return IP address
    - Return next name server

DNS Roles

- Root name servers
  - Responsible for all the TLDs
  - Knows the addresses of every domain’s *name servers*
- Authoritative name servers
  - Responsible for a domain (google.com)
  - For all subdomains, it knows either
    - an IP address
    - the subdomain’s name server
- Recursive resolver
  - Handles lookups for many end hosts
  - Caches IP addresses and name server addresses
- End host
  - Talks to a resolver
  - Caches IP addresses

Typical DNS Query


google.com's name server is at 1.2.3.4
Root name server
mail.google.com is at 5.6.7.8
mail.google.com,
is at 5.6.7.8
3
2
google.com authoritative server
5
4
Friendly neighborhood resolver
5.6.7.8, a.k.a. mail.google.com
7
What is mail.google.com's IP address?
6
1
You

Web Browser Summary

- Browser resolves domain name -> IP address
  - Contacts a DNS resolver…
  - Who contacts possibly multiple other servers…
  - Caching
- Browser retrieves page from server
  - HTTP GET
  - Caching, or else server replies
  - POST methods
- HTTP Optimizations:
  - parallel connections
  - persistent HTTP
  - SPDY

DNS – Main concepts

- Domains
  - Top Level Domains (com, edu, uk, mil, gov, …)
  - Subdomains (com → example.com → www.example.com)
- Name servers
  - Authoritative (tells you the IP for example.com)
  - Root (tells you where example.com’s name server is)
  - Iterative vs. recursive
- Caching
  - Resolver and host cache end-host IP addresses
  - Resolver caches name server IP addresses
  - Entries expire after a TTL