INFM 603: Information Technology and Organizational Context

Session 11: HTML and CSS
(And Networking, Computing Tradeoffs)

Jimmy Lin
The iSchool
University of Maryland

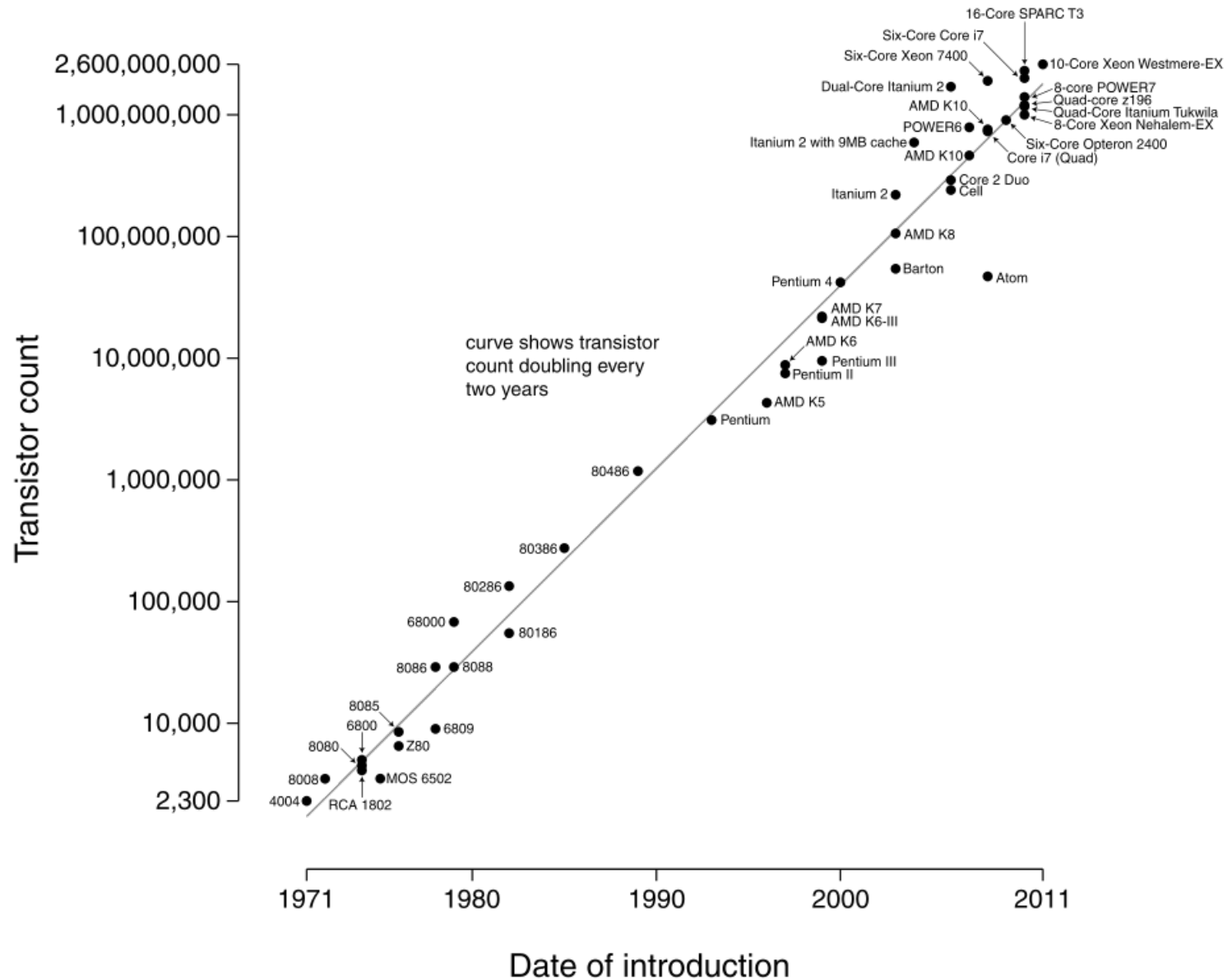Thursday, September 13, 2012

Ways to characterize computing

- How big?

- How fast?

- How reliable?

Computing is fundamentally about tradeoffs!

Example 1: Multi-Core

Microprocessor Transistor Counts 1971-2011 & Moore's Law

Transistor count vs **Date of introduction**

Y-axis (Transistor count): 2,300; 10,000; 100,000; 1,000,000; 10,000,000; 100,000,000; 1,000,000,000; 2,600,000,000

X-axis (Date of introduction): 1971; 1980; 1990; 2000; 2011

curve shows transistor count doubling every two years

Data points labeled: 16-Core SPARC T3, Six-Core Core i7, Six-Core Xeon 7400, 10-Core Xeon Westmere-EX, Dual-Core Itanium 2, 8-core POWER7, AMD K10, Quad-core z196, Quad-Core Itanium Tukwila, POWER6, 8-Core Xeon Nehalem-EX, Itanium 2 with 9MB cache, Six-Core Opteron 2400, AMD K10, Core i7 (Quad), Core 2 Duo, Cell, Itanium 2, AMD K8, Barton, Atom, Pentium 4, AMD K7, AMD K6-III, AMD K6, Pentium III, Pentium II, AMD K5, Pentium, 80486, 80386, 80286, 68000, 80186, 8086, 8088, 8085, 6800, 6809, 8080, Z80, 8008, MOS 6502, 4004, RCA 1802

About This Mac

OS X

Version 10.8.1

Software Update...

Processor 2.7 GHz Intel Core i7

Memory 16 GB 1600 MHz DDR3

More Info...

TM and © 1983–2012 Apple Inc.
All Rights Reserved. License Agreement

Trends in Computing: #1
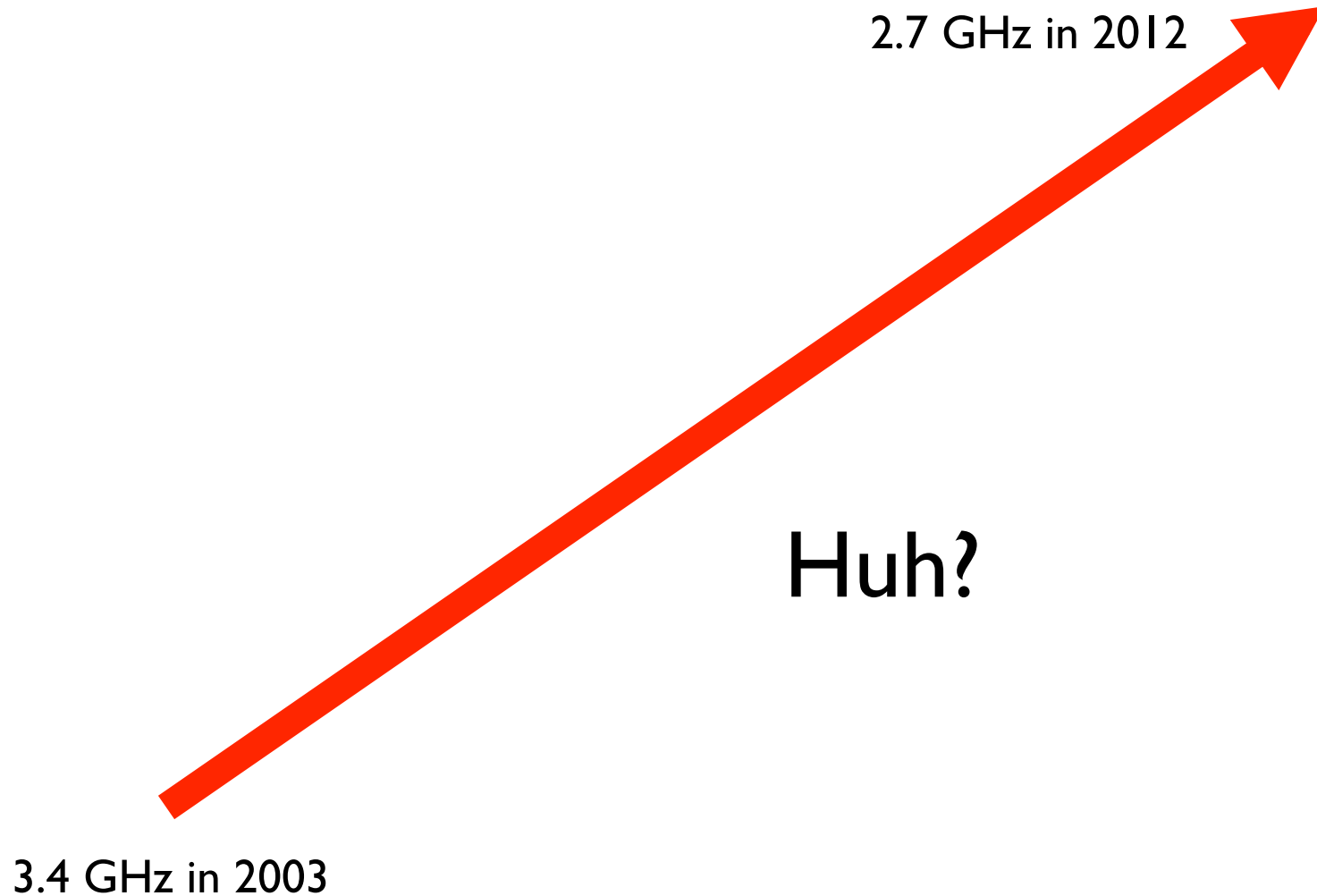
2.7 GHz in 2012

Huh?

3.4 GHz in 2003

ransistor Counts 1971-2011 & Moore's Law

16-Core SPARC T3

Six-Core Core i7

Six-Core Xeon 7400

● 10-Core Xeon Westmere-EX

Dual-Core Itanium 2 ●

8-core POWER7
Quad-core z196
Quad-Core Itanium Tukwila
8-Core Xeon Nehalem-EX

AMD K10

POWER6 ●

Itanium 2 with 9MB cache ●

AMD K10 ●

Six-Core Opteron 2400
Core i7 (Quad)

● Core 2 Duo
● Cell

Itanium 2 ●

● AMD K8

● Barton

● Atom

Pentium 4 ●

AMD K7
AMD K6-III

AMD K6

curve shows transistor
count doubling every
two years

● Pentium III
Pentium II

● AMD K5

● Pentium

80486 ●
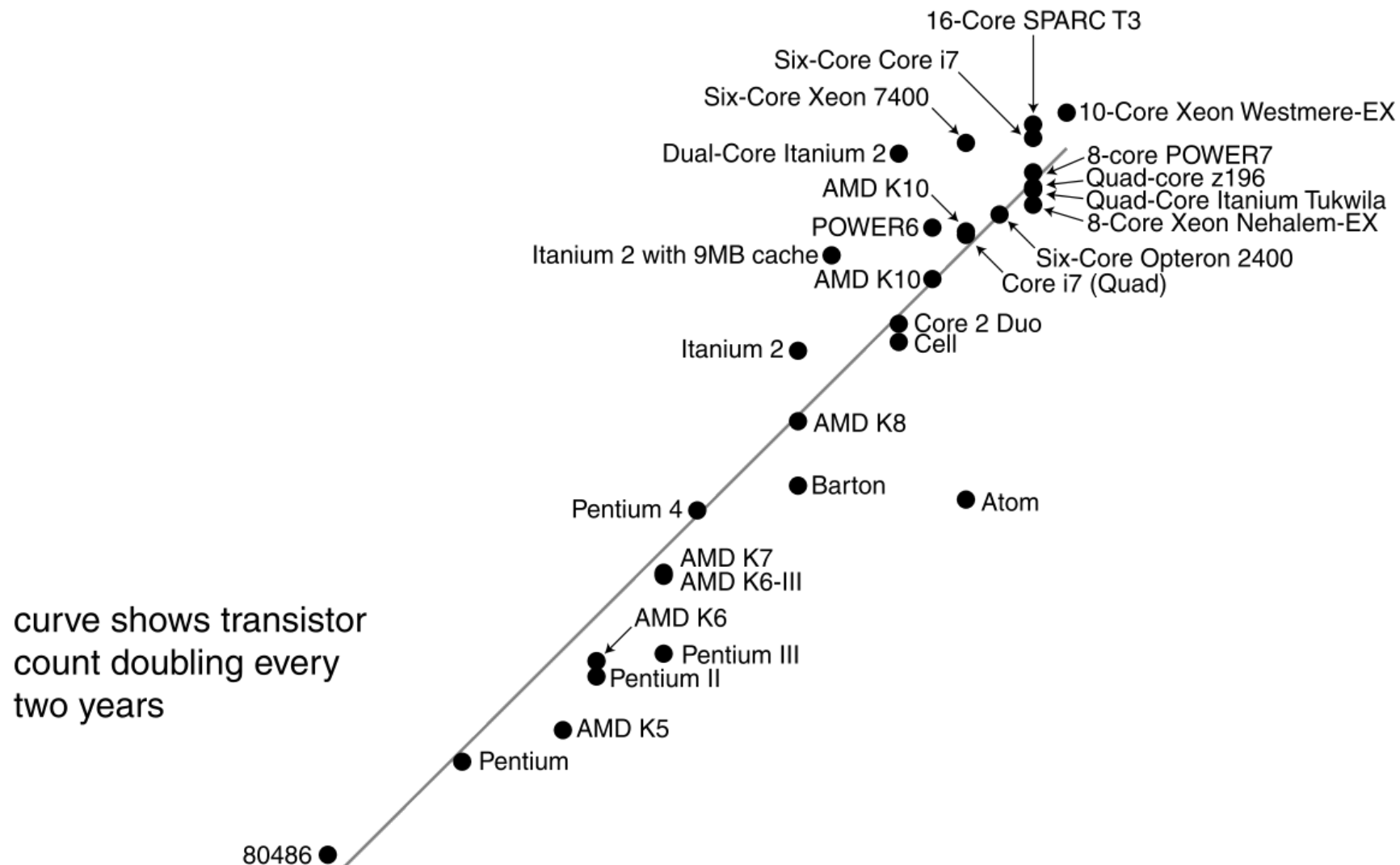
What's big shift?

- From single to multiple cores:
  - Increasing speed of single processor reached point of diminishing returns
  - Solution: put more cores on a processor!

- Important issues:
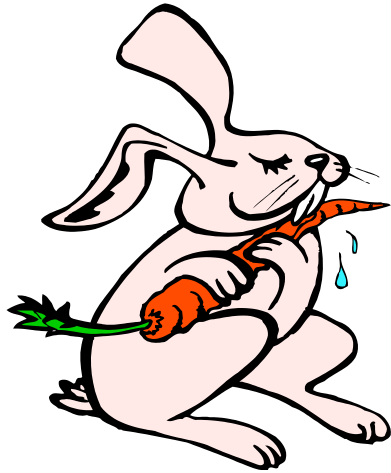 - Power
  - Cool
  - Parallelism
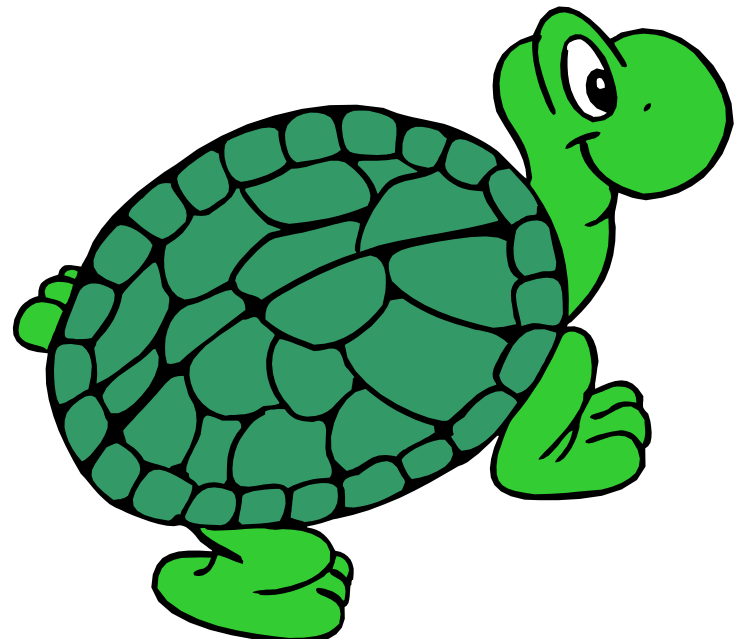
Example 2: Caching

Typical Access Time: 100 ns

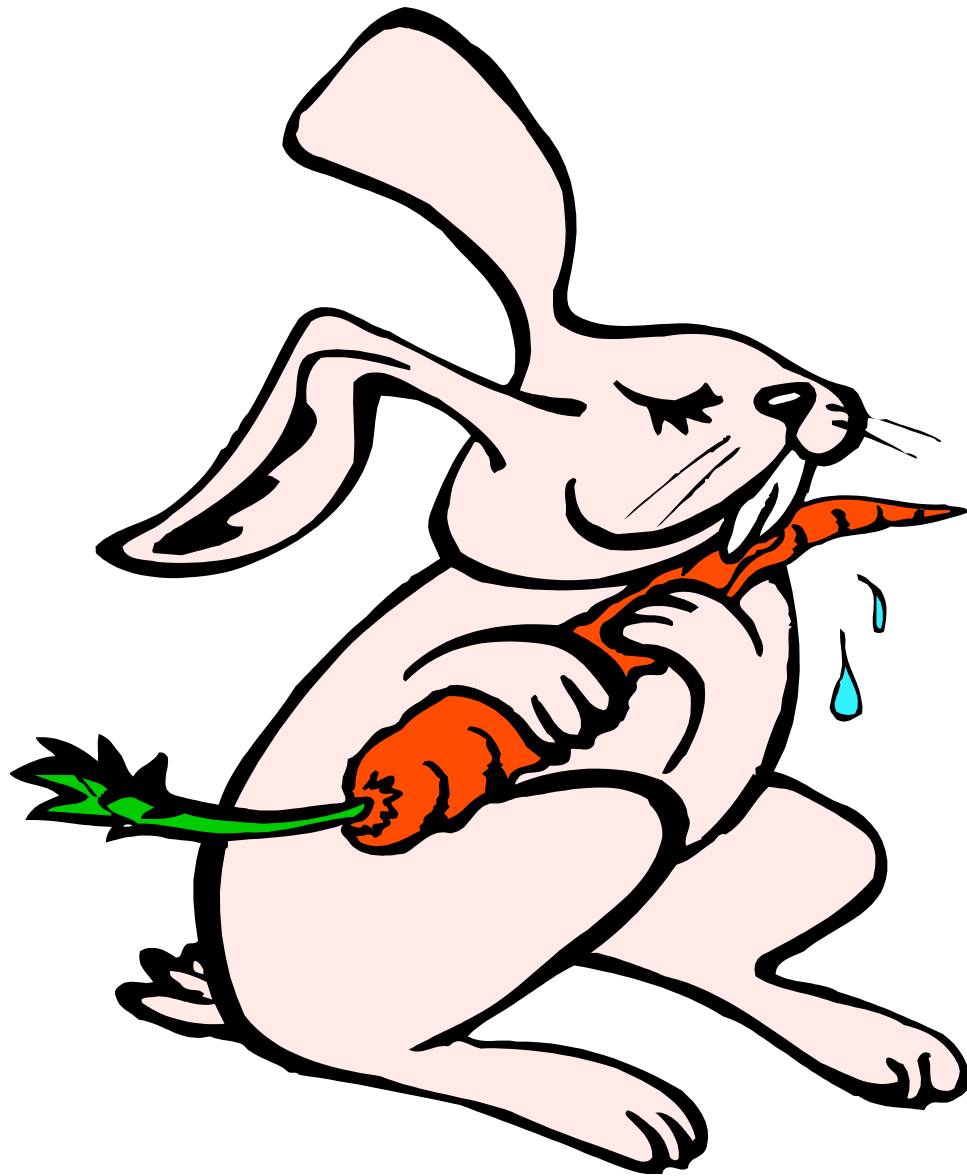Typical Access Time: 10 ms
(10,000x slower than RAM!!!)

Pick two

- Speed

- Capacity

- Cost

RAM: small, expensive, fast

Hard drives: big, cheap, slow

Best of both worlds? cheap, fast, and big

Caching

- **Idea:** move data you're going to use from slow memory into fast memory

  - Slow memory is cheap so you can buy lots of it
  - Caching gives you the illusion of having lots of fast memory

- Physical analogy?

- How do we know what data to cache?

  - Spatial locality: If the system fetched x, it is likely to fetch data located near x (Why?)
  - Temporal locality: If the system fetched x, it is likely to fetch x again (Why?)

Example 3: Replication
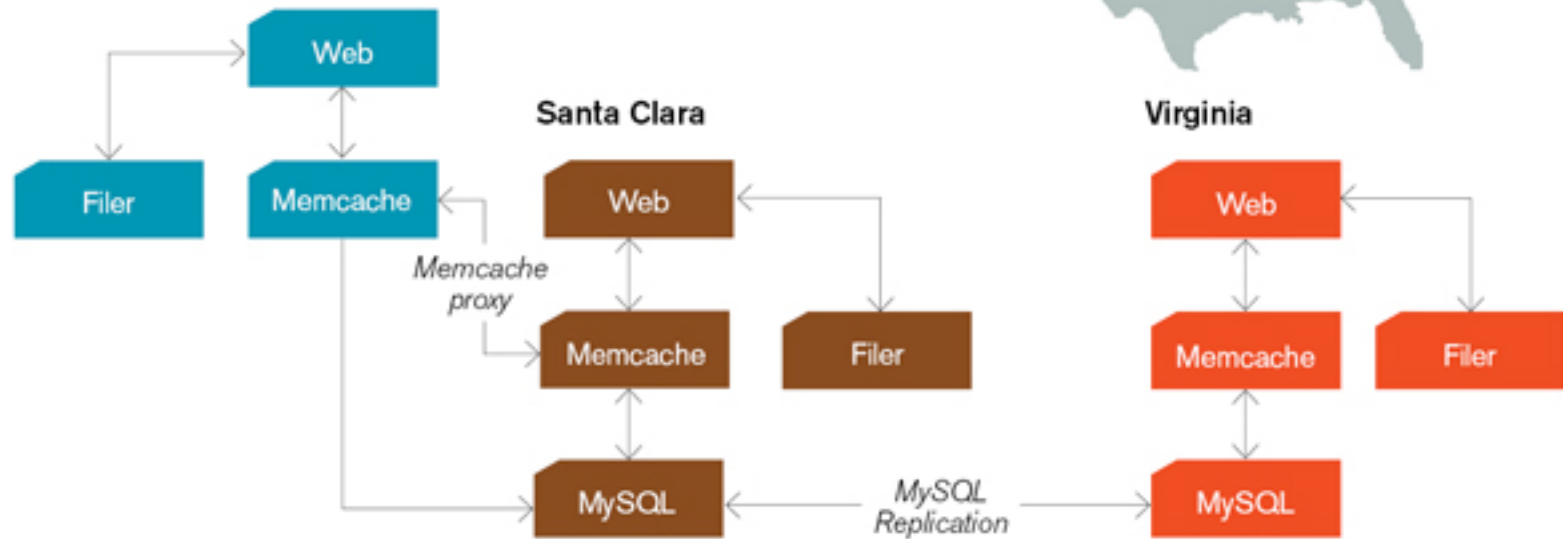
Characterizing Reliability

| "Nines"	Availability	Downtime (per year)
One nine	90%	36.5 d
Two nines	99%	3.65 d
Three nines	99.9%	8.76 h
Four nines	99.99%	52.56 m
Five nines	99.999%	5.256 m
Six nines	99.9999%	31.536 s

How do you ensure reliability?

- Keep multiple copies:
  - On different machines
  - On different machines far apart

- What are the challenges with this?
  - Synchronous vs. Asynchronous
  - Active-Active vs. Active-Passive
  - …
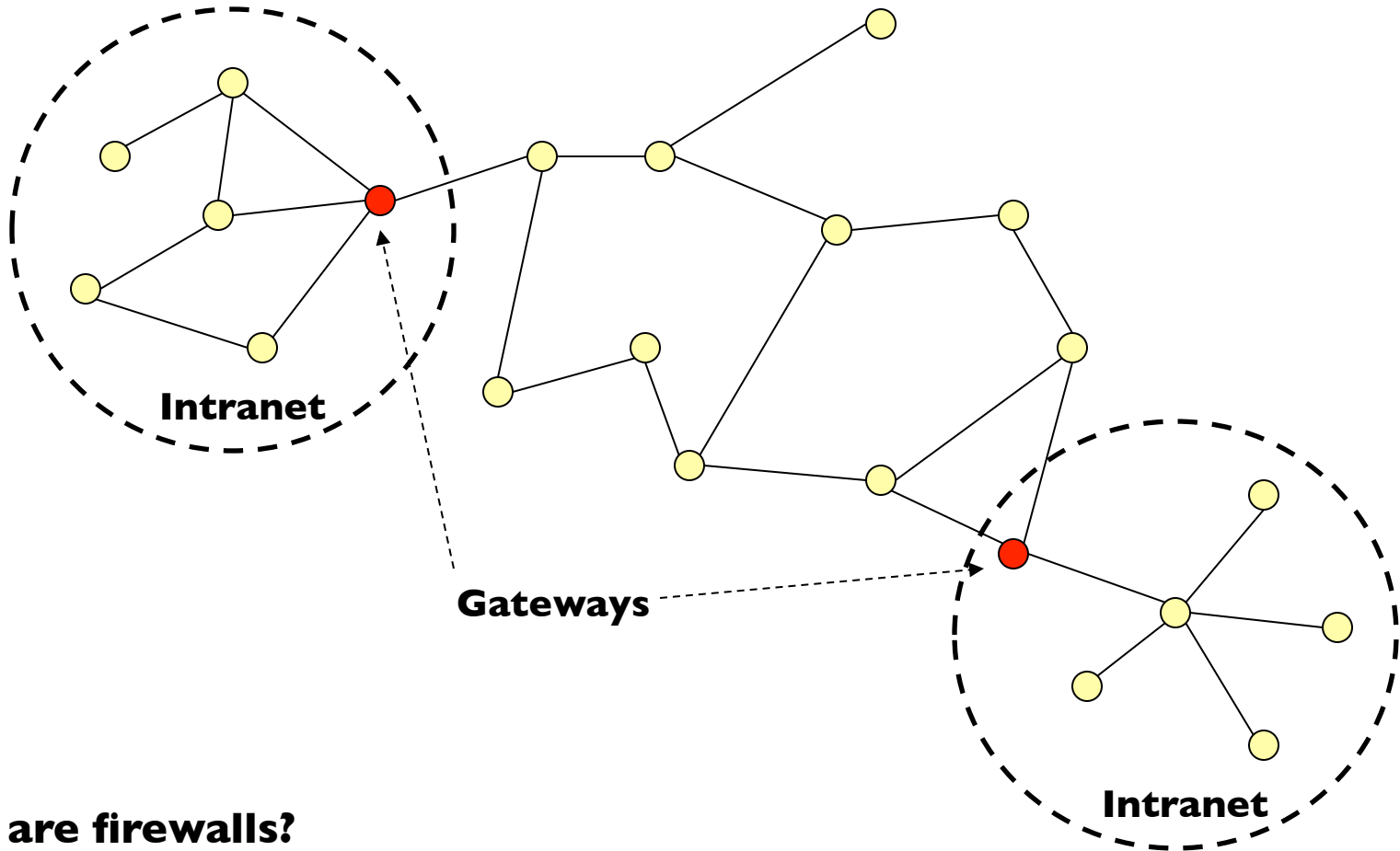
Facebook architecture
(circa 2008)
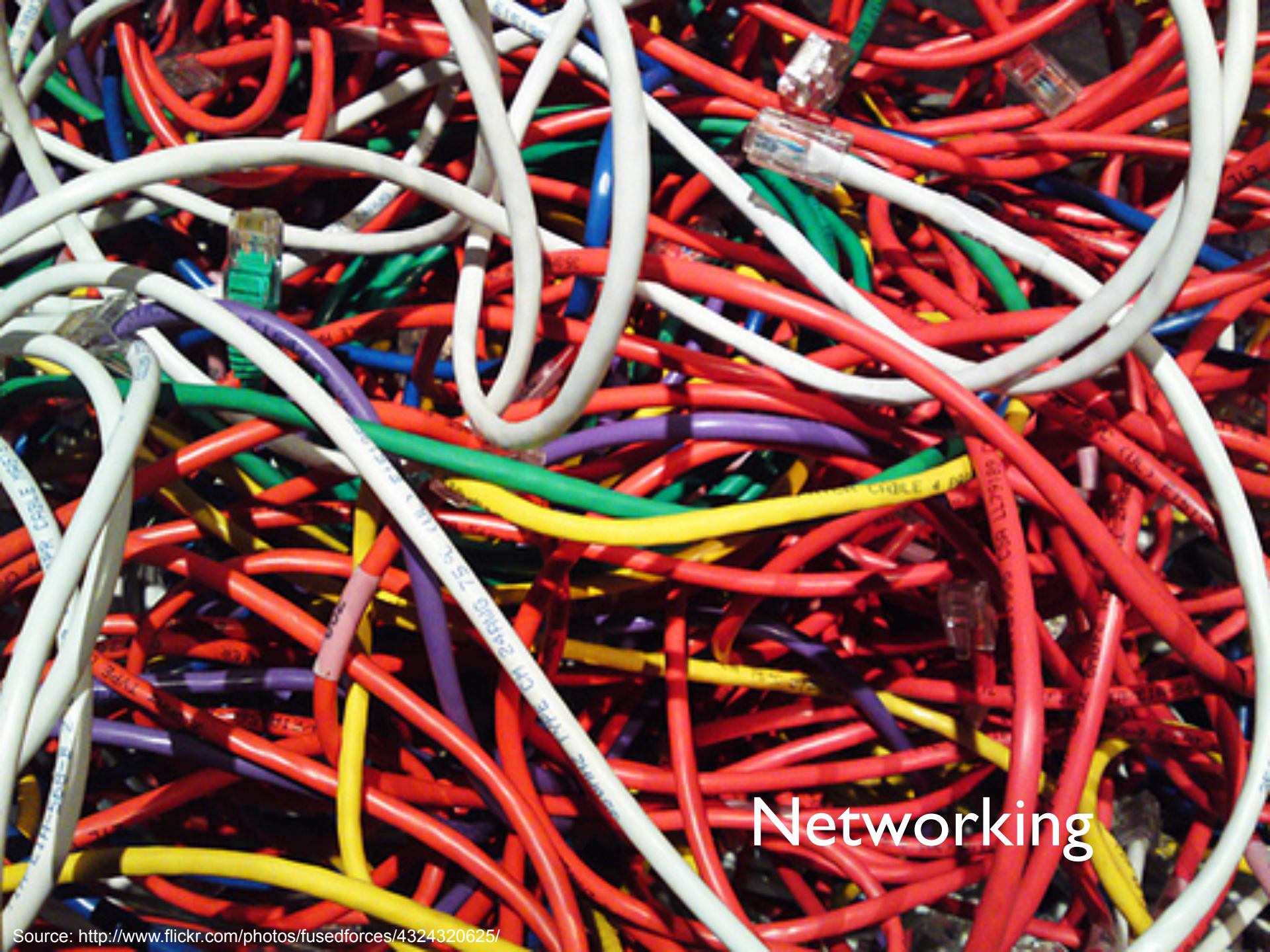
Networking

Internet ≠ Web

- Internet = collection of global networks

- Web = particular way of accessing information on the Internet
  - Uses the HTTP protocol

- Other ways of using the Internet
  - Usenet
  - FTP
  - email (SMTP, POP, IMAP, etc.)
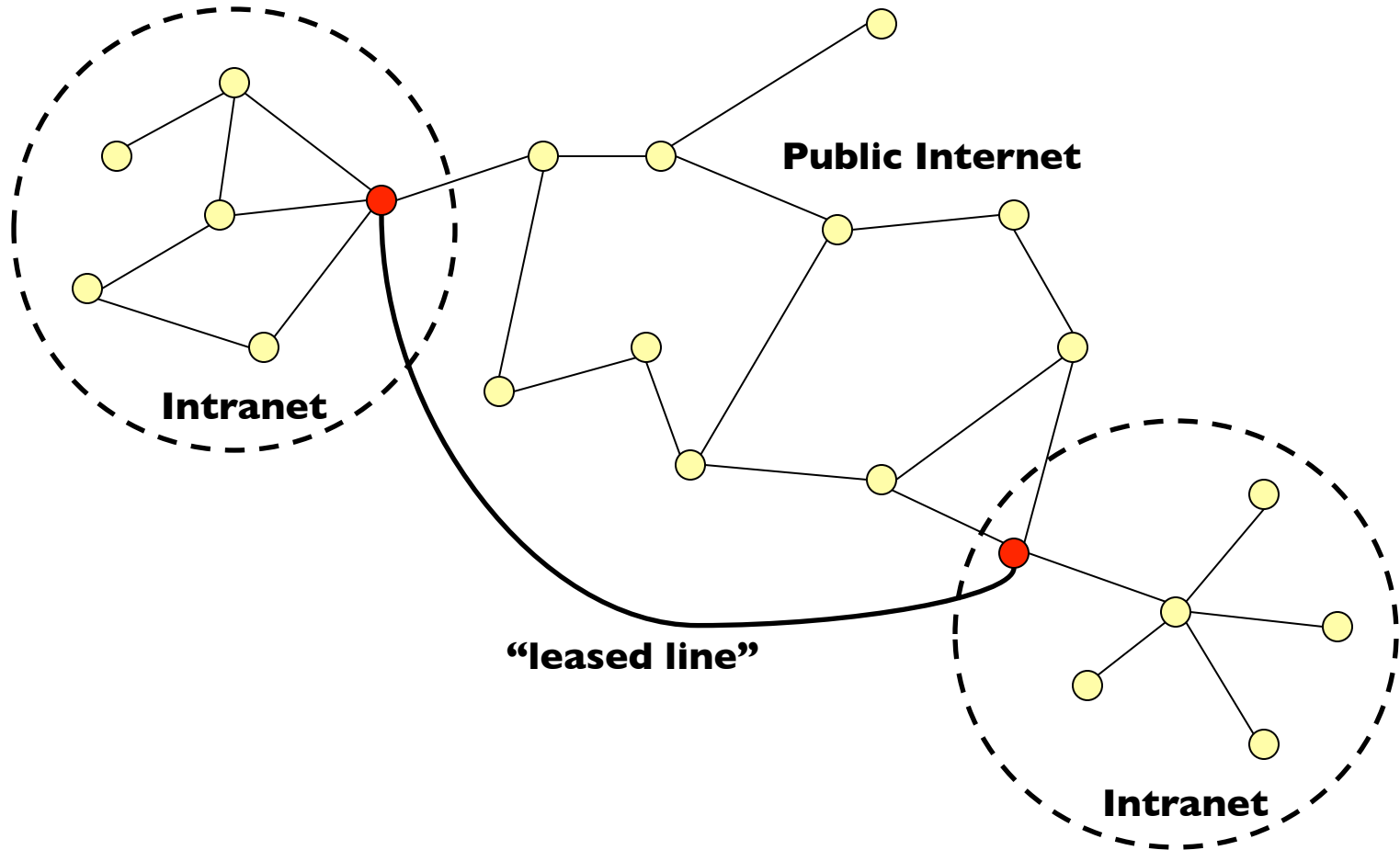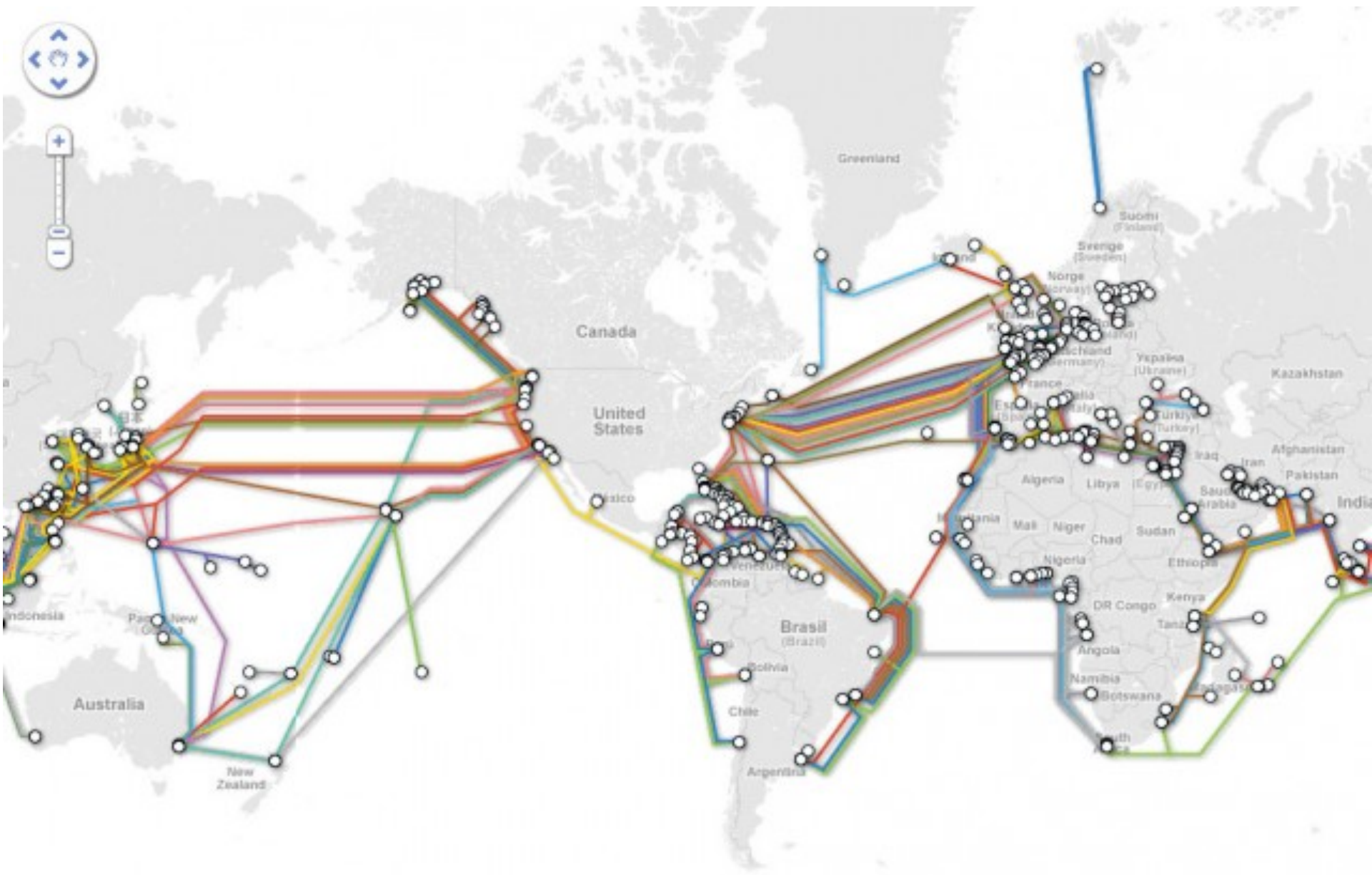 - Internet Relay Chat

Intranets

Intranet

Intranet

Gateways

What are firewalls?
Why can't you do certain things behind firewalls?

Intranets

Problem: How do you securely connect separate networks?

Public Internet

Intranet

Intranet

"leased line"

VPN = Virtual Private Network
a secure private network over the public Internet

Foundations

- Basic protocols for the Internet:
  - TCP/IP (Transmission Control Protocol/Internet Protocol):
    basis for communication
  - DNS (Domain Name Service):
    basis for naming computers on the network

- Protocol for the Web:
  - HTTP (HyperText Transfer Protocol):
   protocol for transferring Web pages

IP Address

- Every computer on the Internet is identified by a address

- IP address = 32 bit number, divided into four "octets"
  - Example: go in your browser and type "http://74.125.131.147/"

Are there enough IP addresses to go around?
What is the difference between static and dynamic IP?

Packet Routing (TCP/IP)

128.0.1.5

63.6.9.12

4.8.15.2

52.55.64.2

192.28.2.5

18.1.1.4
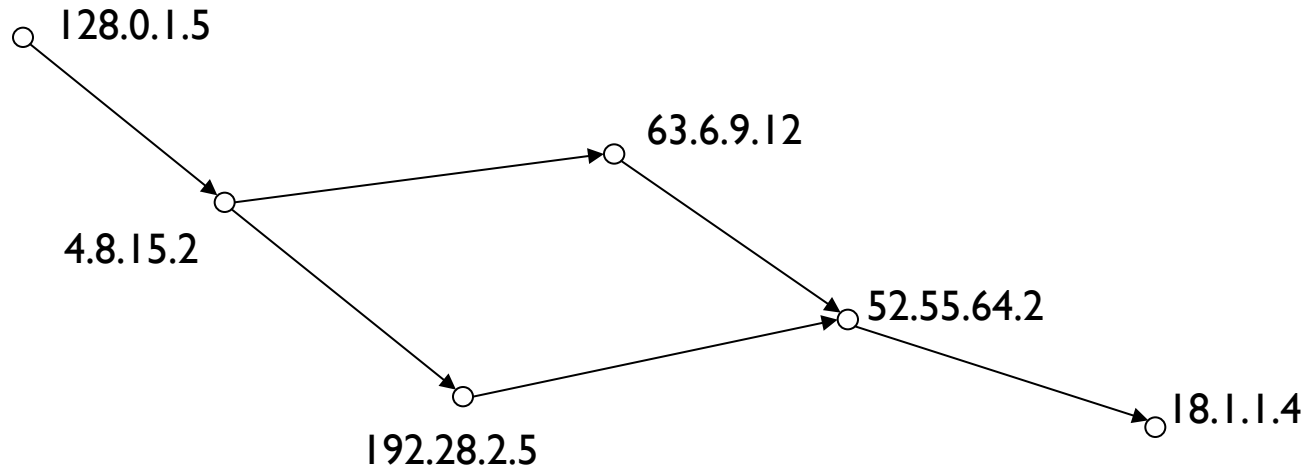
(Much simplified) Routing table for 4.8.15.2

Destination	Next Hop
52.55.*.*	63.6.9.12
18.1.*.*	192.28.2.5/63.6.9.12
4.*.*.*	225.2.55.1
…	

Domain Name Service (DNS)

- Domain names improve usability

  - Easier to remember than IP addresses
  - DNS provides a lookup service

- Each name server knows one level of names

  - "Top level" name server knows .edu, .com, .mil, …
  - .edu name server knows umd, mit, stanford, …
  - .umd.edu name server knows ischool, wam, …

Demo

- Play with various utilities at
  - http://network-tools.com/
  - http://www.yougetsignal.com/tools/visual-tracert/
  - http://en.dnstools.ch/visual-traceroute.html

HyperText Transfer Protocol

- Send request

  GET /path/file.html HTTP/1.0
  From: someuser@somedomain.com
  User-Agent: HTTPTool/1.0

- Server response

  HTTP/1.0 200 OK
  Date: Fri, 31 Dec 1999 23:59:59 GMT
  Content-Type: text/html
  Content-Length: 1354
  <html><body> <h1>Happy New Millennium!</h1> … </body> </html>

Tell me what happens…

- From the moment you click on "check messages" to the moment you start reading your email

- From the moment you click "send" to the moment the other party receives the email

- From the moment you type a URL and hit "enter" to the moment you see the Web page

Tables

Tables

<table>

\<tr>\<td> eenie \</td>	\<td> mennie \</td>	\<td> miney \</td>\</tr>
\<tr>\<td> mo \</td>	\<td> catch \</td>	\<td> a tiger \</td>\</tr>
\<tr>\<td> by \</td>	\<td> the \</td>	\<td> toe \</td>\</tr>

</table>
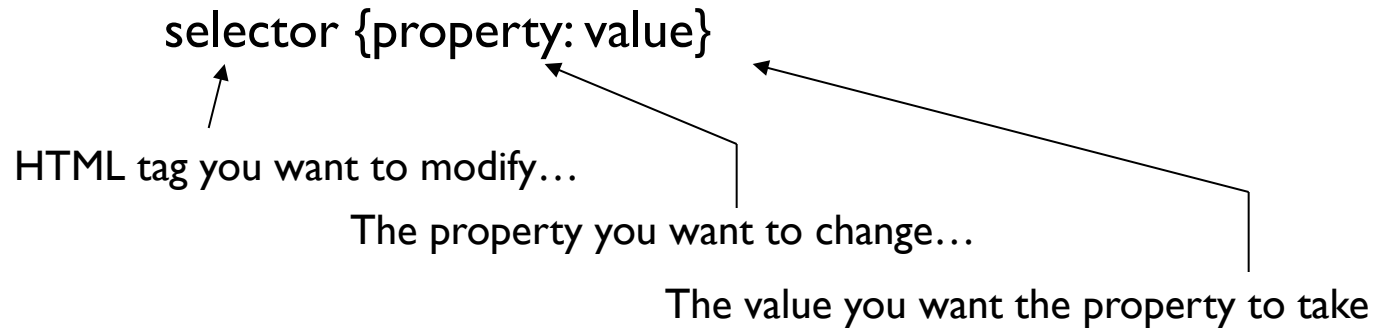
CSS

What's a Document?

- Content

- Structure

- Appearance

- Behavior

CSS: Cascading Style Sheets

○ Separating content and structure from appearance

○ Rules for defining styles "cascade" from broad to narrow:

- Browser default
- External style sheet
- Internal style sheet
- Inline style

Basics of CSS

- Basic syntax:

selector {property: value}

HTML tag you want to modify…

The property you want to change…

The value you want the property to take

- Example:

p { text-align: center;
    color: black;
    font-family: arial }

Causes

- Font to be center-aligned
- Font to be Arial and black

Different Ways for Using CSS

○ Inline style:

    ● Causes only the tag to have the desired properties

        `<p style="font-family:arial; color:blue">…</p>`

○ Internal stylesheet:

    ● Causes all tags to have the desired properties

```
…
<head>…
<style type="text/css">
p { font-family:arial; color:blue}
</style>
</head>
<body>
<p>…</p>
…
```

Customizing Classes

- Ability to define customized styles for standard HTML tags:

```
…
<head>…
<style type="text/css">
p.style1 { font-family:arial; color:blue}
p.style2 { font-family:serif; color:red}
</style>
</head>
<body>
<p class="style1">…</p>
<p class="style2">…</p>
…
```

External Style Sheets

○ Store formatting metadata in a separate file

```
p.style1 { font-family:arial; color:blue}
p.style2 { font-family:serif; color:red}
```

```
…
<head>…
<link rel="stylesheet" href="mystyle.css" type="text/css" />
</head>
<body>
<p class="style1">…</p>
<p class="style2">…</p>
…
```

Why Use CSS?

- What are the advantages of CSS?

- Why have three separate ways of using styles?