Outline

1. Introducing networked services
2. CSE 124 overview
3. Internet naming and addressing
Outline

1. Introducing networked services
2. CSE 124 overview
3. Internet naming and addressing
Our lives are (largely) online!
Networked services driven by data

Data + amazon.com® = Product Recommendations

Data + Spotify® = Custom Stations

Data + Google = Personalized Search
Data-driven, per-user customization

Data + Amazon.com = Product Recommendations

App 1

App 2

App 3

App ...

App ...

App ...

App ...

App ...

App ...

App ...

App ...
Data centers with 100,000+ servers

Microsoft

Google

Facebook
Massive networked infrastructure

• To build:
  – Google spends about $3B per year
  – Microsoft spent $15B in total

• To operate:
  – 1-2% of global energy consumption*

*LBNL, 2013.
Immense scalability

Google
100 billion searches per month

Facebook
1.15 billion users

Amazon.com
120+ million users
How do you build a service that scales to O(1B) people?

...a key topic of CSE 124
The network has seen rapid growth
The network has seen rapid growth


Web Created
The network has seen rapid growth


Web Created

Google’s 1st cluster (15 years)
The network has seen rapid growth
Techniques for handling growth

• Network primitives are designed to scale
• Techniques we learn are directly applicable to global-scale services like Google, Facebook, ...
• Our course work and projects are tested in small scale
  – Yet could scale immensely with minimal to no modification
Course objective

• You will learn how to design, build, and reason about networked services
  – That are correct
  – That are performant
  – That are secure

• But why worry about all that given Facebook’s motto of “Move fast and break things”?
Lives depend on correctly programmed networked applications!

http://www.motortrend.com/news/google-self-driving-cars-begin-tests-on-city-roads-this-summer/
Lives depend on correctly programmed networked applications!

http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
Lives depend on correctly programmed networked applications!

Outline

1. Introducing networked services
2. CSE 124 overview
3. Internet naming and addressing
Resources

• Web page:
  – Linked from www.cs.ucsd.edu/~gmporter

• Piazza discussion forum
  – Linked off the course web page
  – **Mandatory.** Official announcements / updates will occur here. You are responsible for monitoring this site.

• Books
• TA discussions and TAs
• Class meetings and the instructor
• Each other
Books

• Practical TCP/IP Sockets in C, 2nd ed., by Donahoo and Calvert (required)
  – Should be in bookstore this week (or Amazon, ...)
  – First two chapters posted to Piazza in mean time

• The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, 2nd ed., by Barroso, Clidaras, and Hölzle
  – Available free online from course webpage

• Computer Networks: A Systems Approach, 5th ed., by Larry Peterson and Bruce Davie
  – Optional background material
Introducing the TAs

Shreeja Kumar  Vishal Parekh  Sreejith Unnikrishnan
TA discussions

• Small(er) group meetings to work through examples, ask questions, seek out help on the projects/homeworks, etc.

• Each one led by a TA

• Times:
  – Wednesday 3-3:50pm (Shreeja)
  – Wednesday 12-12:50pm (Vishal)
  – Monday 4-4:50pm (Sreejith)
Class meetings (Tu/Th 8-9:20am)

IT'S TOO EARLY
Class meetings (Tu/Th 8-9:20am)

• Overview of material, work through examples/demos, small-group activities
• To help you do what you need to do for your projects/homeworks
• **Be involved**--don’t expect 90 minute speeches!
  – Corollary: **Do not come to class** if you are going to stay unengaged and check Facebook
    • Attendance is not taken!
    • Unengaged students sap energy from your peers and from me
CSE 124 topics

• [124 syllabus]
Grading

• Six projects (60%)
  1. Three related projects on building a fully functioning web server
  4. Data center performance
  5. Remote procedure calls
  6. Overlay and peer-to-peer networks

• Five homeworks (15%)

• Final exam (25%)
GitHub policy

• You will be given a ‘classroom’ GitHub account to use for the course
• All projects and homeworks will be turned in through this (and only this) GitHub account
• You may not post any course materials anywhere else online!
  – Will be considered an academic dishonesty case
Collaboration policy

• You may discuss the homeworks and projects with your peers
  – At a high level, to understand the specification
  – To understand e.g., the tool chain, the APIs used in the course, etc.
  – But you must code up the project on your own
  – You must also explicitly list any students you spoke with in your write-up

• Any material (outside the book and lecture notes) must be cited clearly in your submission

• Do not just Google to find the answer
Why can’t I just Google the answer?

• Isn’t that what “real” programmers do?
• No. In fact, let me tell you a story about working in industry...
First homework

• Due this Thursday!
  – But very simple
• Get a free GitHub.com account first
Q & A

• Before we move on, any questions?
Outline

1. Introducing networked services
2. CSE 124 overview
3. Internet naming and addressing
Terminology

- **Host** – computer, iPad, ps4, Xbox One, …
- **Packet** – unit of transmission across a network
- **Link** – transmit bits
  - Wire or wireless, broadcast or switched (or both)
- **Router** – forwards packets through the network
- **Address** – identifies endpoints in the network
- **Network separation of concerns**
  - Hosts worry about addresses
  - Routers worry about getting packets to the destination
Hosts and routers

Hosts

Routers
Packets

• Unit of transmission in the network

• Header:
  – Metadata specific to the type of packet
    • Source and destination addresses
    • Lengths
    • Checksums, ...

• Payload
  – Contents of the packet
    • Web request, part of an image, frames of video, ...
Demo 1: Examining packets with WireShark
Internet Protocol (IP)

- Datagram (packet) protocol
- Best-effort service
  - Loss
  - Reordering
  - Duplication
  - Delay
- Host-to-host delivery
  (not application-to-application)
Transport Protocols

• Add services on top of IP
• User Datagram Protocol (UDP)
  – Data checksum
  – Best-effort
• Transmission Control Protocol (TCP)
  – Data checksum
  – Reliable byte-stream delivery
  – Flow control
    • Prevents receiver from being overloaded
  – Congestion control
    • Prevents network from being overloaded
Sockets

• Operating system abstraction of a network connection
  – Similar to a file descriptor
• A process can have many sockets
  – Very common
• A socket can be shared by many processes
  – Pretty uncommon
Terminology: putting it together
Demo: TCP in action
IP Address

- 32-bit identifier
- Dotted-quad: 192.118.56.25
- www.mkp.com -> 167.208.101.28
- Identifies a host interface (not a host)
Ports

- IP addresses identify hosts
- Host has many applications
- Ports (16-bit identifier) identify a process

<table>
<thead>
<tr>
<th>Application</th>
<th>WWW</th>
<th>E-mail</th>
<th>Telnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>80</td>
<td>25</td>
<td>23</td>
</tr>
</tbody>
</table>

192.18.22.13
Special addresses

• Localhost
  – Refers to the local server
  – IPv4: 127.0.0.1

• Why would we need localhost?

• Private IP address ranges
  – 10.xx.yy.zz, 192.168.xx.zz, ...
  – For experimenting, local testing, ...

• Link-local addresses
  – 169.254.xx.yy
  – “LAN parties” (do these still exist??)
Names

• Maps easy to remember name to hard to remember (and changing) IP address
• www.cs.ucsd.edu -> 132.239.8.67
• Managed by the Domain Name System (DNS)
DNS Hierarchical Design
Demo 3: DNS in action
Demo 4: Finding TCP/UDP ports with /etc/services