Goals for Today’s Class

• EE 122 overview
  – Goals of the course
  – Structure of the course
  – Instructor & TAs
  – Prereqs & assignments
  – Course grading
  – Academic policies

• What makes networking challenging
  – The fundamental issues we must grapple with to build a
global Internet

What You Learn in This Course

• Insight: key concepts in networking
  – Protocols
  – Layering
  – Resource allocation
  – Security
  – Naming

• Knowledge: how the Internet works
  – Internet architecture
  – IP protocol suite
  – Applications (Web, e-mail, P2P, …)

• Skill: network programming
  – Socket programming
  – Designing and implementing protocols

What This Course Is and Isn’t

• EE122 comes in two flavors:
  – Spring offering: taught by EE faculty
    – More emphasis on diverse link technologies, wireless & mobility,
      communication theory & simulation
  – Fall offering: taught by CS faculty
    – More emphasis on Internet technology, applications, practice &
      empiricism / hands-on
    – Differences aren’t huge, though

• My particular emphasis:
  – Today’s actual (messy) Internet
  – Not yesterday’s, and not much about tomorrow’s
  – Security perspectives

Structure of the Course (1st Half)

• Start at the top
  – Protocols: how to structure communication
  – Sockets: how applications view the Internet

• Then study the “narrow waist” of IP
  – IP best-effort packet-delivery service
  – IP addressing and packet forwarding

• And how to build on top of the narrow waist
  – Transport protocols (TCP, UDP)
  – Domain Name System (DNS)
  – Applications (Web, email, file transfer)

• Looking underneath IP
  – Link technologies (Ethernet, bridges, switches)

Structure of the Course (2nd Half)

• How to get the traffic from here to there …
  – Glue (ARP, DHCP, ICMP)
  – Routing (intradomain, interdomain)

• … in a way that’s both efficient and stable
  – How much data to keep in flight (the window)
  – Without clogging the network (congestion)
  – With some assurance (quality of service) … or not

• How to control network traffic …
  – Enforcing policy
  – Defending against attacks

• … and scale it to potentially huge structures
  – Peer-to-peer & overlays
Instructor

• Vern Paxson (vern@icsi.berkeley.edu)
  – Senior scientist at the International Computer Science Institute and also the Lawrence Berkeley National Lab
  – Research focuses on network security & network measurement
  – http://www.icir.org/vern/
  – Office hours W 2:30-3:30PM in 329 Soda
    • And by appointment at ICSI
    • http://www.icsi.berkeley.edu/where.html
    • This week only by appointment
  – Phone: 666-2882
  • Email works much better!
  • Hearing impaired: please be ready to repeat questions & comments!

TAs

• Dilip Anthony Joseph (dilip@eecs.berkeley.edu)
  – Office hours F 11-12 in 311 Soda
    • And by appointment
  – Section F 10-11 in 293 Cory

• Sukun Kim (binetude@eecs.berkeley.edu)
  – Office hours T 11-12 in 410 Soda
    • And by appointment
  – Section T 10-11 in 400 Cory

  • Co-teach 3rd section
  – W 12-1 in 293 Cory

Interact!

• Inevitably, you won’t understand something(s) … that’s my fault, but you need to help.
• Come to office hours, request an appointment, communicate by e-mail
  – We are here to help, including general advice!
  – TAs first line for help with programming problems
• Give us suggestions/complaints/feedback as early as you can
• What’s your background? Tell us at
  – http://tinyurl.com/fbc7u

Course Materials

• Textbooks
    • Note, we jump around in it a lot
  – Recommended & on reserve:
• Web site: http://inst.eecs.berkeley.edu/~ee122/
  – Updated frequently, including lecture slides (generally in advance)
• Mailing list: ee122@icsi.berkeley.edu
  – Sign up: http://mailman.icsi.berkeley.edu/mailman/listinfo/ee122

Class Workload

• Four homeworks spread over the semester
  – Strict due dates (no slip days!)
  – Deadlines are generally 3:50PM prior to lecture
• Three (mini-)projects
  – Simple “echo” server (socket programming)
  – Simple Web crawler
  – “Chat” tool
  • 1st phase: design protocol
  • 2nd phase: implement to reference protocol design
  • C (or C++) required
  – Deadlines 11PM
• Exams
  – Midterm: Monday October 16
  – Final: Saturday Dec 16
  – Closed book, open crib sheet

Prerequisites

• CS 61A, 61B
  – Programming, data structures, software engineering
  – Knowledge of C or C++
• Math 53 or 54
  – In fact, we’ll be relatively light on math, though your algebra should be very solid, you should know basic probability, and you’ll need to be comfortable with thinking abstractly
• Background material will not be covered in lecture. TAs will spend very little time reviewing material not specific to networking
Grading

- Course graded to mean of B
  - Relatively easy to get a B, harder to get an A or a C
  - 10% A, 15% A-, 15% B+, 20% B, 15% B-, 15% C+, 10% C
  - A+ reserved for superstars (1 or 2 per class)
  - Mean can shift up for an excellent class

Projects 40% (10+10+20)
Midterm exam 20%
Final exam 20%

No Cheating

- Cheating means not doing the assignment by yourself.
- Fine to talk with other students about assignments outside of class.
- No copying, no Google, etc.
- If you’re unsure, then ask.
- We will do automated similarity detection on assignments.

Networking: Actually Not Boring

5 Minute Break

Questions Before We Proceed?

Why Networking Is Challenging

- Fundamental challenge: the speed of light
- Question: how long does it take light to travel from Berkeley to New York?
- Answer:
  - Distance Berkeley → New York: 4,125 km (great circle)
  - Traveling 300,000 km/s: 13.75 msec

Fundamental Challenge: Speed of Light

- Question: how long does it take an Internet "packet" to travel from Berkeley to New York?
- Answer:
  - For sure >= 13.75 msec
  - Depends on:
    - The route the packet takes (could be circuitous!)
    - The propagation speed of the links the packet traverses
      - E.g., in optical fiber light propagates at about 2/3 C
      - The transmission rate (bandwidth) of the links (bits/sec)
      - and thus the size of the packet
    - Number of hops traversed (store-and-forward delay)
    - The "competition" for bandwidth the packet encounters (congestion). It may have to sit & wait in router queues.
    - In practice this boils down to:
      - >= 40 msec
Fundamental Challenge: Speed of Light

• Question: how many cycles does your PC execute before it can possibly get a reply to a message it sent to a New York web server?
  
• Answer:
  – Round trip takes >= 80 msec
  – PC runs at (say) 3 GHz
  – 3,000,000,000 cycles/sec * 0.08 sec = 240,000,000 cycles

  = An Eon
  – Communication feedback is always dated
  – Communication fundamentally asynchronous

Why Networking Is Challenging, con’t

• Fundamental challenge: we are cheapskates who want it all

• Cheapskates: computer science is all about cost
  – Or, put another way: efficiency
  – If cost didn’t matter, networking would be oh-so-easy!
    • E.g., string wires between each pair of computers in the world
    • Though, um, pesky speed-of-light issues remain …

• Want it all: goal of the Internet is to interconnect
  – A huge number of devices
  – Using all sorts of link technologies
  – Across a very wide range of conditions

• So need to be vast in scope yet affordable

Examples of Network Components

<table>
<thead>
<tr>
<th>Links</th>
<th>Interfaces</th>
<th>Switches/routers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fibers</td>
<td>Ethernet card</td>
<td>Large router</td>
</tr>
<tr>
<td>Coaxial Cable</td>
<td>Wireless card</td>
<td>Telephone switch</td>
</tr>
</tbody>
</table>

Why Networking Is Challenging, con’t

• Fundamental challenge: components fail
  – Network communication involves a chain of interfaces, links, routers and switches …

• Question: suppose a communication involves 50 components which work correctly (independently) 99% of the time. What’s the likelihood the communication fails at a given point of time?
  – Answer: success requires that they all function, so failure probability = 1 - 0.99^{50} = 39.5%.

• So we have a lot of components (we want it all), which tend to fail (cheapskates) …
  – … and we don’t find out for an eon (speed-of-light)
Why Networking Is Challenging, con’t

• Challenge: enormous dynamic range  
  (because we want it all)
  – Round-trip times (latency) vary from 10 usec’s to sec’s (10^5)
  – Data rates (bandwidth) vary from kbps to 10 Gbps (10^7)
  – Queuing delays inside the network vary from 0 to sec’s
  – Packet loss varies from 0 to 90+%  
  – End system (host) capabilities vary from cell phones to supercomputer clusters
  – Application needs vary enormously: size of transfers, bidirectionality, need for reliability, tolerance of jitter
• Related challenge: very often, there is no such thing as “typical”. Beware of your “mental models”!
  – Must think in terms of design ranges, not points
  – Mechanisms need to be adaptive

Why Networking Is Challenging, con’t

• Challenge: different parties must work together
• Comes about due to network’s scope
  – Once larger than a single institution, you have multiple parties with different agendas who still must agree how to divide the task between them (“coopetition”)
• Working together requires:
  – Protocols (defining who does what)
    • These generally need to be standardized
    – Agreements regarding how different types of activity are treated (policy)
• Different parties very well might try to “game” the network’s mechanisms to their advantage

Why Networking Is Challenging, con’t

• Challenge: incessant rapid growth
• Internet has sustained energetic, compound growth for more than two decades
  – Utility of the network scales with its size
    → Fuels exponential growth
  – Currently about half a billion hosts
• With growth comes
  – Rapid evolution & innovation ...
    – ... among both the networking technology and (especially) the applications it supports.
• Adds another dimension of dynamic range ...
  – ... and quite a number of ad hoc artifacts
  – “Success disaster”
Why Crooks Matter for Networking

- They (and other attackers) seek ways to misuse the network towards their gain
  - Carefully crafted "bogus" traffic to manipulate the network’s operation
    - E.g., altering Internet routing or name lookups
  - Torrents of bogus (or even legitimate) traffic to overwhelm a service (denial-of-service)
    - E.g., as an extortion threat against an ecommerce site
  - Passively recording network traffic in transit (sniffing)
    - E.g., to steal information or aid in crafting manipulative traffic
  - Exploit flaws in clients and servers using the network to trick into executing the attacker’s code (compromise)
- They do all this energetically because there is significant $$$ to be made

Summary

- A number of deep challenges
  - Speed-of-light
  - Desiring a pervasive global network
  - Need for it to work efficiently/cheaply
  - Failure of components
  - Enormous dynamic range ("no such thing as typical")
  - Disparate parties must work together
  - Rapid growth/evolution
  - Crooks & other bad guys
- Next lecture: types of networks, protocols
  - Read through 1.2 of the Peterson/Davie book
  - Take the survey (http://tinyurl.com/fbc7u)
  - Dust off your C/C++ programming skills if need be