EE 122: Introduction To Communication Networks

Fall 2007 (WF 4-5:30 in **Cory 277**)

Vern Paxson

TAs: Lisa Fowler and Jorge Ortiz

http://inst.eecs.berkeley.edu/~ee122/

Materials with thanks to Jennifer Rexford, Ion Stoica, and colleagues at Princeton and UC Berkeley

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

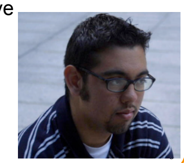
Instructor

- Vern Paxson (vern@cs.berkeley.edu)
  - Research focuses on network security & network measurement
  - Also affiliated with *International Computer Science Institute* and the *Lawrence Berkeley National Lab*
  - http://www.icir.org/vern/
  - Office hours WF 3-4PM in 615 Soda
    - And by appointment, sometimes at ICSI
      - http://www.icsi.berkeley.edu/where.html
  - Phone: 643-4209, 666-2882
    - Email works *much* better!
  - Hearing impaired: please be ready to repeat questions & comments!

7

TAs

- Lisa Fowler (fowler@eecs.berkeley.edu)
  - Office hours F 11-12 in 283E Soda
    - And by appointment
  - Section F 10-11 in 293 Cory
- Jorge Ortiz (jortiz@cs.berkeley.edu)
  - Office hours W 1-2 in 711 Soda alcove
    - And by appointment
  - Section W 12-1 in 299 Cory



- Co-teach 3rd section
  - T 10-11 in 521 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/2wynmu**

9

Course Materials

- Textbooks
  - J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach,* 4th Edition, Addison Wesley, 2007.
    - Note, we jump around in it a lot
  - Recommended & on reserve:
    - W. R. Stevens, B. Fenner, A. M. Rudoff, *Unix Network Programming: The Sockets Networking API*, Vol. 1, 3rd Ed., Addison-Wesley, 2004.
    - W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1993.*
- Web site: *http://inst.eecs.berkeley.edu/~ee122/*
  - Updated frequently, including lecture slides
  - Note: if you are following the slides during lecture, please don't use them to answer questions I ask
- Mailing list: ee122@icsi.berkeley.edu
  - Sign up: http://mailman.icsi.berkeley.edu/mailman/listinfo/ee122

10

Class Workload

- Four homeworks spread over the semester
  - Strict due dates (no slip days!)
  - Deadlines are generally 3:50PM prior to lecture
- Two projects
  - Web server (solo)
    - Includes socket programming, client/server
  - Peer-to-peer file sharing app (solo or teams of two)
    - You design and implement the transport protocol
  - C (or C++) required
  - Deadlines 11PM
  - These are **extensive** undertakings, particularly the second
- Exams
  - Midterm: Friday October 12 in class, 4-5:30PM
  - Final: Tuesday Dec 18 location TBD, 8-11AM
  - Closed book, open crib sheet

11

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

12

Grading

Homeworks	20% (5% each)
Projects	40% (20+20)
Midterm exam	15%
Final exam	25%

- 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
    - For which the TAs have significant input

13

No Cheating

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

14

Will The Class Size Increase?

- Not clear at this point. We are at the limit for what we can handle with the current TA resources.
- I'm looking into additional TA/readers.
- This probably won't resolve until next week.

15

5 Minute Break

Questions Before We Proceed?

16

Networking: Actually Not Boring

- How hard can it be??
- You just string a wire (or other signaling path) between two computers …
- … first one squirts bits down the link …
- … and the second one slurps them up. Right?
- Where does it get tricky?
  What are the challenges?

17

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

18

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

19

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 cyles/sec * 0.08 sec = 240,000,000 cycles

= **An Eon**
  - Communication **feedback** is always *dated*
  - Communication fundamentally asynchronous

20

Fundamental Challenge: Speed of Light

- Question: what about between machines directly connected (via a *local area network* or LAN)?
- Answer:

```
% ping www.icir.org
PING www.icir.org (192.150.187.11): 56 data bytes
64 bytes from 192.150.187.11: icmp_seq=0 ttl=64 time=0.214 ms
64 bytes from 192.150.187.11: icmp_seq=1 ttl=64 time=0.226 ms
64 bytes from 192.150.187.11: icmp_seq=2 ttl=64 time=0.209 ms
64 bytes from 192.150.187.11: icmp_seq=3 ttl=64 time=0.212 ms
64 bytes from 192.150.187.11: icmp_seq=4 ttl=64 time=0.214 ms
```

- 200 μsec = 600,000 cycles
  - Still a loooong time …
  - … and asynchronous

21

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

22

Why Networking Is Challenging, con't

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

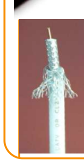
23

Examples of Network Components

Links

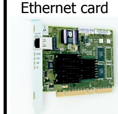
Fibers

Coaxial Cable



Interfaces

Ethernet card



Wireless card

Switches/routers

Large router

Telephone switch

24

Why Networking Is Challenging, con't

- Fundamental challenge: **components fail**
  - Network communication involves a chain of interfaces, links, routers and switches …
  - … **all** of which must function correctly.
- 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)

25

Why Networking Is Challenging, con't

- Challenge: **enormous dynamic range** (because **we want it all**)
  - Round-trip times (latency) vary 10 µsec'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

26

Why Networking Is Challenging, con't

- Challenge: **different parties must work together**
  - Multiple parties with *different agendas* must agree how to divide the task between them
- 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

27

Why Networking Is Challenging, con't

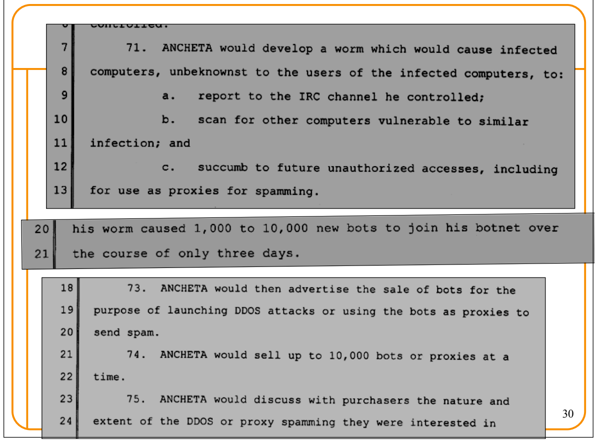
- Challenge: **incessant rapid growth**
  - Utility of the network scales with its size
    - ⇒ Fuels exponential growth (for more than 2 decades!)
  - Currently about half a billion hosts
- Adds another dimension of dynamic range …
  - … and quite a number of ***ad hoc*** artifacts
    - "Success disaster"

28

Why Networking Is Challenging, con't

- Challenge: **there are Bad Guys out there**
- As the network population grows in size, so does the number of
  - Vandals
  - Crazies
- What really matters, though: as network population grows, it becomes more and more attractive to
  - **Crooks**
  - (and also spies and militaries)

29



71. ANCHETA would develop a worm which would cause infected computers, unbeknownst to the users of the infected computers, to:

a. report to the IRC channel he controlled;

b. scan for other computers vulnerable to similar infection; and

c. succumb to future unauthorized accesses, including for use as proxies for spamming.

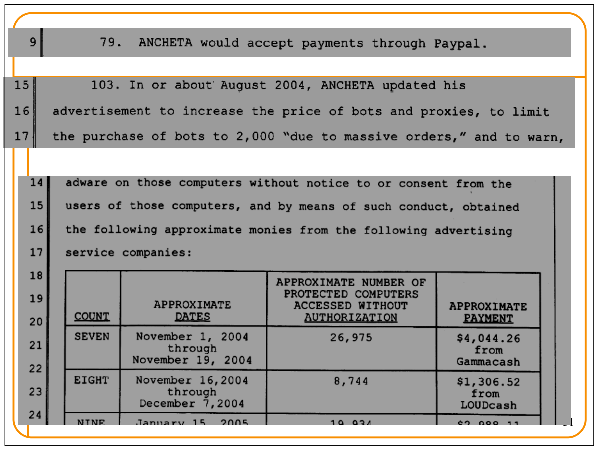
his worm caused 1,000 to 10,000 new bots to join his botnet over the course of only three days.

73. ANCHETA would then advertise the sale of bots for the purpose of launching DDOS attacks or using the bots as proxies to send spam.

74. ANCHETA would sell up to 10,000 bots or proxies at a time.

75. ANCHETA would discuss with purchasers the nature and extent of the DDOS or proxy spamming they were interested in

30

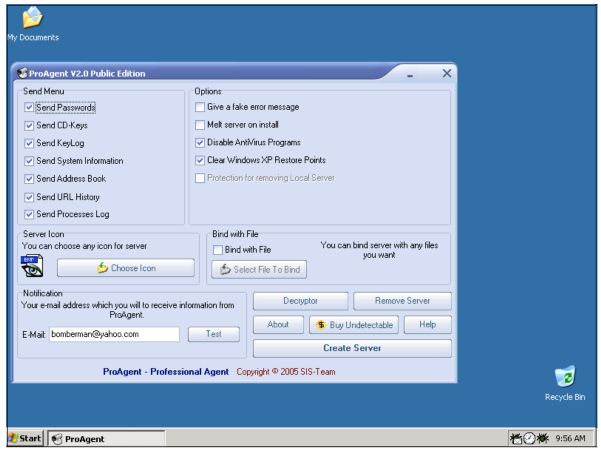


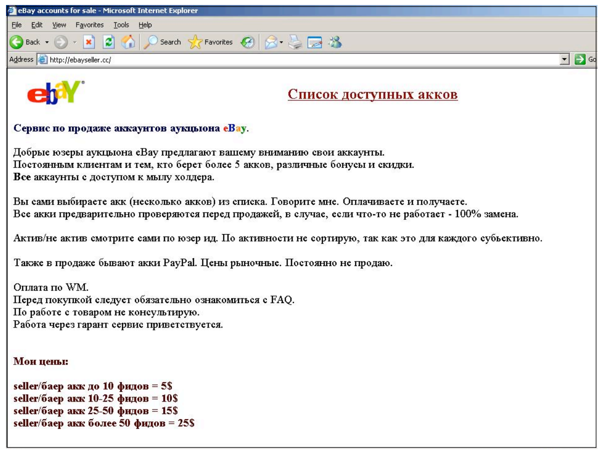
79. ANCHETA would accept payments through Paypal.

103. In or about August 2004, ANCHETA updated his advertisement to increase the price of bots and proxies, to limit the purchase of bots to 2,000 "due to massive orders," and to warn,

adware on those computers without notice to or consent from the users of those computers, and by means of such conduct, obtained the following approximate monies from the following advertising service companies:

COUNT	APPROXIMATE DATES	APPROXIMATE NUMBER OF PROTECTED COMPUTERS ACCESSED WITHOUT AUTHORIZATION	APPROXIMATE PAYMENT
SEVEN	November 1, 2004 through November 19, 2004	26,975	$4,044.26 from Gammacash
EIGHT	November 16,2004 through December 7,2004	8,744	$1,306.52 from LOUDcash





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
  - Torrents of traffic to overwhelm a service (denial-of-service) for purposes of extortion / competition
  - Passively recording network traffic in transit (sniffing)
  - 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

35

Summary

- Networking is about design in the presence of challenges/constraints:
  - Not akin to e.g. programming languages / compilers
    - Which have well-developed theories to draw upon
  - Much more akin to operating systems
    - Abstractions
    - Tradeoffs
    - Design principles / "taste"
- Next lecture: types of networks, protocols
  - Read through 1.3 of the Kurose/Ross book
  - Take the survey (**http://tinyurl.com/2wynmu**)
  - Dust off your C/C++ programming skills if need be

36