Lecture 1:
Course Introduction

CSE 222A: Computer Communication Networks
Alex C. Snoeren
Lecture 1 Overview

- Class overview
  - Expected outcomes
  - Structure of the course
  - Policies and procedures

- A brief review of undergrad networking
  - High-level concepts
  - An end-to-end example
Logistics

- Instructor: Alex C. Snoeren
  - Office hours Tuesdays 1-2pm or by appointment
  - EBU3b 3114

- TA: Danny Huang
  - Office hours 3:30-5:30pm EBU3b B260A

- Course Web page
  - http://www.cs.ucsd.edu/classes/wi14/cse222A-a/
  - Piazza is only for Q&A
CSE 222A Class Overview

- Course materials taught through class lecture, paper readings, and term project
  - Lectures are *interactive*—attendance is crucial to success

- Course grade based upon:
  - Daily paper reviews
  - In-class quiz at end of term (based on lectures/readings)
  - Term project with paper and presentation

- Piazza discussion forums
  - The place to ask questions about lectures, readings, project
Prereqs

- Undergrad networking course (e.g., CSE123)
  - You are welcome to take the course without prior background,
  - But, several parts of the course will be especially challenging
    » You are responsible for doing the extra reading on your own
    » Peterson & Davie are your friends—our undergrad textbook

- Systems programming experience
  - The term project will likely require significant implementation
  - This course will not teach you systems programming. The TA will help, but you need to learn it on your own if you don’t already know it.
Expected Outcomes

● This course *will* teach you about network architecture
  ◆ We will cover some classic literature for background
  ◆ Focus mostly on recent developments in the field

● This course *will not* teach you the fundamentals
  ◆ Layering, signaling, framing, MAC, switching, routing, naming, Internetworking, congestion control, router design, etc.
  ◆ Take the undergrad course for the basics

● Similarly, we will not cover Web/Cloud services
  ◆ CSE223B covers distributed systems design, the “cloud,” etc.
  ◆ You *will be able to* pick this up on your own with Google
CSE 222A Class Overview

- Course materials taught through class lecture, paper readings, and term project
  - Lectures are *interactive*—attendance is crucial to success

- Course grade based upon:
  - Daily paper reviews
  - In-class quiz at end of term (based on lectures/readings)
  - Term project with paper and presentation

- Piazza discussion forums
  - The place to ask questions about lectures, readings, project
Textbook

Paper reviews

- Written critique of each assigned reading
  - Submitted in advance of each class through an automated conference review system (HotCRP)
  - What are the biggest contributions of the paper?
  - What are the main shortcomings/issues with the work?
  - What are the implications of the described work?

- You should read others’ reviews
  - Help you see other points of view
  - Available after you submit your review

- Graded on a 3-point scale
  - Our expectations will go up as the term progresses
Term Project

- Group project; teams of 2—3 people
  - Your chance to explore what networking research is like
  - The very best projects can—and do—result in publication

- We will post a list of project ideas on the Website
  - You can review old lists as well while you wait

- Several milestones to keep you on track
  - Topics of interest due Jan 21\textsuperscript{st}
  - Teams formed January 23\textsuperscript{rd}
  - Project proposal due February 4\textsuperscript{th}

- Final exam period will be a mini conference
  - Each group will prepare a report and a presentation
Grading

- Paper reviews: 15%
- Quiz: 35%
- Project: 40%
- Participation: 10%
  - Attendance and engagement in class discussion is crucial
Questions

- Before we start the material, any questions about the class structure, contents, etc.?
Networking in One Slide

- Protocols & Layering
  - Manage complexity by decomposing the tasks
  - Standardizing syntax and semantics to support interoperability

- Naming
  - Agreeing on how to describe a host, application, network, etc.

- Switching & Routing
  - Deciding how to get from here to there
  - Forwarding messages across multiple physical components

- Resource Allocation
  - Figuring out how to share finite bandwidth, memory, etc.
A “Simple” Task

- Send information from one computer to another

- Endpoints are called **hosts**
  - Could be computer, iPod, cell phone, etc.

- The plumbing is called a **link**
  - We don’t care what the physical technology is: Ethernet, wireless, cellular, etc.
Measures of success

- How fast?
  - **Bandwidth** measured in bits per second
  - Often talk about KBps or Mbps – Bytes vs bits
- How long was the wait?
  - **Delay** (one-way or round trip) measured in seconds
- How efficiently?
  - **Overhead** measured in bits or seconds or cycles or…
- Any mistakes?
  - **Error rate** measured in terms of probability of flipped bit
How long to send a message?

- Transmit time \( T = \frac{M}{R} + D \)
  - 10 Mbps Ethernet LAN (\( M=1\)KB)
    - \( \frac{M}{R}=1ms, \ D \approx 5us \)
  - 155 Mbps cross country ATM link (\( M=1\)KB)
    - \( \frac{M}{R} = 50us, \ D \approx 40-100ms \)

- Where are the bits in the mean time?
  - In transit inside the network

- \( R \times D \) is called the **bandwidth delay product**
  - How many bits can be “stored” be stored in transit
  - Colloquially, we say “fill the pipe”
Is Not Really So Simple
Layering: A Modular Approach

- Sub-divide the problem
  - Each layer relies on services from layer below
  - Each layer exports services to layer above

- Interface between layers defines interaction
  - Hides implementation details
  - Layers can change without disturbing other layers

- Interface among peers in a layer is a protocol
  - If peers speak same protocol, they can interoperate
Protocol Standardization

- Communicating hosts speaking the same protocol
  - Standardization to enable multiple implementations
  - Or, the same folks have to write all the software

- Internet Engineering Task Force
  - Based on working groups that focus on specific issues
  - Produces “Request For Comments” (RFCs)
    » Rough consensus and running code
    » After enough time passes, promoted to Internet Standards

- Other standards bodies exist
  - ISO, ITU, IEEE, etc.
Encapsulation

HTTP
TCP
IP
Ethernet interface

HTTP
Payload
Headers
TCP
IP
Ethernet interface
Internet Protocol Suite

The Hourglass Model

Applications
Transport
Data Link
Physical

“Thin Waist”

FTP
HTTP
NV
TFTP
TCP
UDP
IP

NET_1
NET_2
...
NET_n
Physical layer

2.4Ghz Radio
DS/FH Radio
(1-11Mbps)

802.11b Wireless
Access Point

Cat5 Cable (4 wires)
100Base TX Ethernet
100Mbps

Ethernet switch/router

To campus backbone

62.5/125um 850nm MMF
1000BaseSX Ethernet
1000Mbps
Link Layer (e.g. Ethernet)

- Break message into frames
- Media Access Control (MAC)
  - Can I send now? Can I send now?
- Send frame
Connecting links

- **Routers/Switches**: moves bits between links
  - *Circuit switching*: guaranteed channel for a session (Telephone system)
  - *Packet switching*: statistical multiplexing of independent pieces of data (Internet)
Putting this all together

- **ROUGHLY**, what happens when I click on a Web page from UCSD?

My computer  www.google.com

Internet
Web request (HTTP)

- Turn click into HTTP request

GET http://www.google.com/ HTTP/1.1
Host: www.google.com
Connection:keep-alive
...
Name resolution (DNS)

- Where is www.google.com?

My computer
(132.239.9.64)

What’s the address for www.google.com

Oh, you can find it at 66.102.7.104

Local DNS server
(132.239.51.18)
Data transport (TCP)

- Break message into packets (TCP segments)
- Should be delivered reliably & in-order

GET http://www.google.com HTTP/1.1
Host: www.google.com
Connection: keep-alive
...
Global Network Addressing

- Address each packet so it can traverse network and arrive at host

My computer
(132.239.9.64)

Destination
Source
Data

66.102.7.104
132.239.9.64
1
GET htt

www.google.com
(66.102.7.104)
Resource Allocation: Queues

- Sharing access to limited resources
  - E.g., a link with fixed service rate
- Simplest case: first-in-first out queue
  - Queue/serve packets in the order they arrive
  - Drop packets when the queue is full
- Anybody hear of “Network Neutrality”?
For Next Class…

- Browse the course Web site
  - http://www.cs.ucsd.edu/classes/wi14/cse222A-a

- Read P&D Chapters 1 & 2

- Read and review Saltzer, Reed, and Clark ‘84
  - Submit review in HotCRP

- Start thinking about term project ideas/groups
  - Suggestions will be available on the web soon