CSCI-1680 - Computer Networks

Nick DeMarinis

http://www.cs.brown.edu/courses/cs1680

Based partly on lecture notes by Rodrigo Fonseca, David Mazières, Phil Levis, John Jannotti, Peterson & Davie
Cast

- Instructor: Nick DeMarinis (ndemarin)
- HTA: Yongjeong Kim
- UTA: Max Heller
- GradTA: Usama Naseer
- GradTA: Amrit Singh Rana
- How to reach us: EdStem
  https://edstem.org/us/courses/18576/
Why are we here?

"The network"
"The Internet"
"The Cloud"

You (the user)  Applications
Why are we here?

Goal: learn concepts underlying networks

• How do networks work? What can one do with them?
• Gain a basic understanding of the Internet
• Gain experience writing protocols
• Tools to understand new protocols and applications

“From two communicating machines to the entire Internet”
Why should you care?

• Networks have mostly disappeared...
  – By being everywhere!

• But...
  – Almost all applications are (partly) cloud-based
  – There are important tradeoffs when using networked systems
    • ISP choice, how to build applications, etc.
  – What to do when they fail?
Networks

• What is a network?
  – System of lines/channels that interconnect
  – *E.g.*, railroad, highway, plumbing, postal, telephone, social, computer

• Computer Network
  – Moves information
  – Nodes: general-purpose computers
  – Links: wires, fiber optics, EM spectrum, composite…
Why are computer networks cooler?

• Most nodes are general-purpose computers
• Very easy to innovate and develop new uses of the network: you can program the nodes

• Contrast with the ossified Telephone network
  – Can’t program most landline phones
  – Intelligence in the network (why?)
Examples of Innovation / Disruption

• WhatsApp: as of Jan 2016, reached 1B monthly active users in 7 years (now over 2B)
  – 57 engineers by then!

• Uber: global dispatch service

• Zoom: Where would we be without it now?
  – April 2020: 300 million daily meeting participants
Mirai Botnet (Oct 2016)

- DVRs, Home Routers, Cameras disrupted DYN
  - DNS provider for Twitter, Netflix, Reddit, others
## Growth of the Internet

<table>
<thead>
<tr>
<th>Year</th>
<th>Global Internet Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>100 GB per day</td>
</tr>
<tr>
<td>1997</td>
<td>100 GB per hour</td>
</tr>
<tr>
<td>2002</td>
<td>100 GB per second</td>
</tr>
<tr>
<td>2007</td>
<td>2,000 GB per second</td>
</tr>
<tr>
<td>2016</td>
<td>26,600 GB per second</td>
</tr>
<tr>
<td>2021 (Est.)</td>
<td>105,800 GB per second</td>
</tr>
</tbody>
</table>

**By 2023:**

- 5.3B users globally (66% of population)
- 29.3 Billion devices (3.6 per capita)
- >70% will be wireless

Source: CISCO
During 2008, the number of things connected to the Internet exceeded the number of people on earth.

By 2020 there will be 50 billion.

A Dutch startup, Sparked, is using wireless sensors on cattle.
Source: Facebook
Traceroute map of the Internet, ~5 million edges, circa 2003. opte.org
Why should you take this course?

Maslow’s Hierarchy of Needs (1943)

Source: hard to track meme on the Internet
Why should you take this course?

• Impact
  – Social, economic, political, educational, ...
  – Why should you care about NetNeutrality?
  – What does it mean to run out of IP addresses?
  – How could Egypt shut down the Internet internally?
  – How could Pakistan shut down Youtube *globally*?

• Continuously changing and evolving
  – Incredible complexity
  – Any *fact* you learn will be inevitably out of date
  – Learn general underlying *principles*

Networks are cool… and you will learn to program them!
Recurring Themes

• How to find who to talk to
  – Addresses and names, discovery, routing
• Decide how to talk to them
  – Encodings, Protocols
• Make sure communication is correct, only among intended parties, works for all
  – Mediation, Error correction, encryption, …
• How to do this at scale
  – Planetary scale (or beyond)
Roadmap

• Build knowledge from the ground up
  – Link individual nodes
  – Local networks with multiple nodes
  – IP: Connect hosts across several networks
  – Transport: Connect processes on different hosts
  – Applications

• A few cross-cutting issues
  – Security, multimedia, overlay networks, P2P…
Mechanics: Resources

• Lecture slides/notes: authoritative content
  – Only what we cover in class will be tested

• Tools
  – Course website: https://cs.brown.edu/courses/csci1680/s22/
  – Discussions: EdStem
  – HW submission/grading + exam grading: Gradescope
  – Project development and submission: Github

Complete HW0 (on website) ASAP to be added to Github
Texts

• Two examples on source website
  – Peterson and Davie, Computer Networks - A Systems Approach, 4th or 5th editions or
  – Kurose and Ross, ‘Computer Networking: A Top-Down Approach (6th or 7th editions)

• May be helpful, not required
Mechanics: Projects

• Learn by implementing!
  – Build protocols and client/servers from the ground up
  – Implement fundamental network protocols

• 4 Programming projects
  – Snowcast: streaming music server
  – IP, as an overlay, on top of UDP
  – TCP, on top of your IP
  – Final (short, fun, TBD)

• First project is individual, others in groups of 2
What do you do?

• 4 Programming Projects (45%)

• “Written” component
  – Exams: Midterm (15%, take-home) and Final (25%, probably also take-home)
  – Homework: 3 written assignments (15%)
    • Short answer and design questions

• Must pass two components individually
Prerequisites

• CSCI 0330, CSCI 0300 (or equivalent)
  – We assume basic OS concepts (kernel/user, threads/processes, I/O, scheduling)

• Should be comfortable with systems programming or be willing to learn quickly
  – threads, locking, explicit memory management, …
Prerequisites: languages

• You can use any systems language
  – C
  – C++
  – Go
  – Rust*

• If you have other ideas, talk to us
  – No high-level networking APIs, though (unless you write them yourself)
  – You will be bit twiddling, multi-threading, and byte packing…
Regular Administrivia

• Most material on course website: https://cs.brown.edu/courses/csci1680/s22/

• You are responsible to check the web page and EdStem!
  – All announcements will be there
  – Textbook chapters corresponding to lectures
  – Handouts, due dates, programming resources, etc…
  – New documents are subject to change (reload before checking assignments)
Asking for help

• Online help: EdStem
• Office hours: calendar on course website

• Can help with..
  - Debugging
  - Project planning/design
  - Concepts
  - We’re here for you—don’t be afraid to ask!

I don’t bite—Instructor’s office hours are just like regular hours!
Asking for help

• Collaboration: work with your peers!
  – Collaboration policy on course website
  – I encourage you to collaborate, **so long as the code you write down is your own**

• Mental health is important!
  – If you have concerns, feel free to talk to us
  – We encourage you to contact University resources like CAPS
Remote learning

This is an in-person class, but we will do our best to accommodate remote needs

• All lectures live-streamed on Zoom, posted online
• Based on need, we will add remote office hours
• Please follow University guidelines—don’t come in if you’re sick (I won’t either)
• We can be flexible!

If you need to be regularly remote, please note this on HW0.
Feedback

• Anonymous feedback form on course website

• Please tell us how we can improve the course!
  – Clarity of assignments
  – Improving accessibility
  – Concerns about presentation of content, interactions with staff
Why should you listen to me?

• Received my PhD from Brown in 2021
  – My areas: software security, networking, network security

• This is my first class at Brown, but not my first time teaching, or working with this class

• No one knows everything about networks, and I am no exception!
Some more honesty

- Huge demand for the course this year!
- We are working hard to scale the class—we appreciate your patience and understanding!
The Waitlist

• If you are not enrolled, sign up for the waitlist
  – As enrollment changes, we will admit students from the waitlist, prioritizing those who cannot take the course again

• During shopping period, you are expected to participate in the course and submit assignments
Stretch

(and I won’t look if you are shopping and want to flee)
Building Blocks

• Nodes: Computers (hosts), dedicated routers, …
• Links: Coax, twisted pair, fiber, radio, …
How to connect more nodes?

Multiple wires

Shared medium
To scale to more nodes, use *switching*

- Nodes can connect to multiple other nodes
- Recursively, one node can connect to multiple networks
• What to do when multiple flows must share a link?
Multiplexing strategies

- **Physical/Link layer** (Copper wires, wireless): Signalling strategies

- **Network layer**: IP addresses to identify one host

- **Transport layer**: One host has many ports, used for different applications

- **Application layer**: One program can have multiple “sockets”, which are an interface to one network “connection”
Extra content we will cover later
Switching Strategies

- **Circuit Switching** – virtual link between two nodes
  - Set up circuit (e.g. dialing, signaling) – may fail: busy
  - Transfer data at known rate
  - Tear down circuit

- **Packet Switching**
  - Forward bounded-size messages.
  - Each message can have different senders/receivers
  - Focus of this course

Analogy: circuit switching reserves the highway for a cross-country trip. Packet switching interleaves everyone’s cars.
Synchronous time-division multiplexing
- Divide time into equal-sized quanta, round robin
- Illusion of direct link for switched circuit net
- But wastes capacity if not enough flows
- Also doesn’t degrade gracefully when more flows than slots
FDM

Frequency-division multiplexing: allocate a frequency band for each flow

- Same as TV channels and radio stations

• Similar drawbacks to STDM
  - Wastes bandwidth if someone not sending
  - Can run out of spectrum
Statistical Multiplexing

- Like STDM but with no pre-determined time slots (or order!)
- Maximizes link utilization
  - Link is never idle if there are packets to send
Statistical Multiplexing

- **Cons:**
  - Hard to guarantee fairness
  - Unpredictable queuing delays
  - Packets may take different paths

- **Yet...**
  - This is the main model used on the Internet

- **Think of running a restaurant**
  - For a fixed set of people that go there every day
  - Or on a busy corner of Manhattan
    - When would you take reservations?
Roadmap

• **Assignments: learn by implementing**
  – Warm up: Snowcast, a networked music server
    • Get a feel for how applications use the network

• **Build knowledge from the ground up**
  – Link individual nodes
  – Local networks with multiple nodes
  – IP: Connect hosts across several networks
  – Transport: Connect processes on different hosts
  – Applications

• **A few cross-cutting issues**
  – Security, multimedia, overlay networks, P2P…
• Snowcast: start TODAY!
• Next class:
  – More on layering
  – How to use the network from the application: sockets
• Then…
  – We start moving up the network stack, starting from how two computers can talk to each other.