Cast

- Instructor: Rodrigo Fonseca (rfonseca)
- GTA: Andrew Ferguson (adf)
- HTA: Son Nguyen (sbnguyen)
- UTA: Osmar Olivo (oolivo)
- Email: cs168tas@cs.brown.edu (all of us)
Overview

• **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
Prerequisites

• CSCI-0320/CSCI-0360 (or equivalent).
  – We assume basic OS concepts (kernel/user, threads/processes, I/O, scheduling)

• Low-level programming or be willing to learn quickly
  – threads, locking, explicit memory management, …

• We allow any* language, but really support only C
  – You will be bit twiddling and byte packing…
Administrivia

• All assignments will be on the course page
  http://www.cs.brown.edu/courses/cs168/s11

• Text: Peterson and Davie, Computer Networks - A Systems Approach, 4th Edition

• You are responsible to check the web page!
  – All announcements will be there
  – Textbook chapters corresponding to lectures: read them before class
  – Handouts, due dates, programming resources, etc…
  – Subject to change (reload before checking assignments)
Grading

• **Exams: Midterm (15%) and Final (25%)**
• **Homework: Four written assignments (20%)**
  – Short answer and design questions
• **4 Programming Projects (40%)**
  – User level networking: streaming music server
  – IP, as an overlay, on top of UDP
  – TCP, on top of your IP
  – Final (TBD, we will solicit your input)
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 (most nodes)
  – Links: wires, fiber optics, EM spectrum, composite…
Why Study Computer Networks?

• Many 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 phones
  – Intelligence in the network, control by parties vested in the status quo, …
Growth of the Internet

Source: Miguel Angel Todaro
Traceroute map of the Internet, ~5 million edges, circa 2003. opte.org
Why should you take this course?

• Networks are cool!
  – Incredible impact: social, economic, political, educational, …

• Incredible complexity

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  – Any fact you learn will be inevitably out of date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

• Learn to program the network
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…
Two-minutes for stretching
Building Blocks

- **Nodes:** Computers (hosts), dedicated routers, …
- **Links:** Coax, twisted pair, fiber, radio, …

**Physical Layer:** Several questions:
- Voltage, frequency
- Wired, wireless

**Link Layer:** how to send data?
- When to talk
- What to say (format, “language”)

Stay tuned for lectures 3 and 4…
How to connect more nodes?

Multiple wires

Shared medium
From Links to Networks

- To scale to more nodes, use *switching*
  - Nodes can connect to multiple other nodes
  - Recursively, one node can connect to multiple networks
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 class

Analogy: circuit switching reserves the highway for a cross-country trip. Packet switching interleaves everyone’s cars.
Multiplexing

• What to do when multiple flows must share a link?
- **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: allocates a frequency band for each flow
  – Same TV channels and radio stations

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

- Idea: 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
Protocol Layering

- A network packet from A to D must be put in link packets A to B, B to C, C to D
- Can view this encapsulation as a stack of layers
  - Each layer produces packets that become the payload of the lower layer’s packets
OSI Reference Model

End host
- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

End host
- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

One or more nodes within the network
Layers

• Physical – sends individual bits
• Data Link – sends frames, handles media access
• Network – sends packets, using routing
• Transport – demultiplexes, provides reliability, flow and congestion control
• Session – can tie together multiple streams (e.g., audio & video)
• Presentation – crypto, conversion between representations
• Application – what the users sees, e.g., HTTP
Addressing

• Each node typically has a unique* name
  – When that name also tells you how to get to the node, it is called an address

• Each layer can have its own naming/addressing

• **Routing** is the process of finding a path to the destination
  – In packet switched networks, each packet must have a destination address
  – For circuit switched, use address to set up circuit

• **Special addresses** can exist for broadcast/multicast/anycast

* or thinks it does, in case there is a shortage
Internet Protocol (IP)

• **Used by most computer networks today**
  – Runs over a variety of physical networks, can connect Ethernet, wireless, modem lines, etc.

• **Every host has a unique 4-byte IP address (IPv4)**
  – *E.g.*, www.cs.brown.edu \(\rightarrow\) 128.148.32.110
  – The network knows how to route a packet to any address

• **Need more to build something like the Web**
  – Need naming (DNS)
  – Interface for browser and server software (next lecture)
  – Need demultiplexing within a host: which packets are for web browser, Skype, or the mail program?
Inter-process Communication

• Talking from host to host is great, but we want abstraction of inter-process communication
• Solution: *encapsulate* another protocol within IP
Transport: UDP and TCP

- **UDP and TCP most popular protocols on IP**
  - Both use 16-bit *port* number & 32-bit IP address
  - Applications *bind* a port & receive traffic on that port

- **UDP – User (unreliable) Datagram Protocol**
  - Exposes packet-switched nature of Internet
  - Sent packets may be dropped, reordered, even duplicated (but there is corruption protection)

- **TCP – Transmission Control Protocol**
  - Provides illusion of reliable ‘pipe’ or ‘stream’ between two processes anywhere on the network
  - Handles congestion and flow control
Uses of TCP

• Most applications use TCP
  – Easier to program (reliability is convenient)
  – Automatically avoids congestion (don’t need to worry about taking down the network)

• Servers typically listen on well-know ports:
  – SSH: 22
  – SMTP (email): 25
  – Finger: 79
  – HTTP (web): 80
### Internet Layering

<table>
<thead>
<tr>
<th>Network</th>
<th>IP</th>
<th>TCP</th>
<th>UDP</th>
<th>Application</th>
</tr>
</thead>
</table>

- **Strict layering not required**
  - TCP/UDP “cheat” to detect certain errors in IP-level information like address
  - Overall, allows evolution, experimentation
Many applications protocols on top of UDP & TCP
IP works over many types of networks
This is the “Hourglass” architecture of the Internet.
  - If every network supports IP, applications run over many different networks (e.g., cellular network)
Coming Up

• **Next class: how do applications use the network?**
  – Introduction to programming with Sockets
  – Peterson & Davie 1.4
  – Beej’s Guide to Network Programming (link on the course website)

• **Then…**
  – We start our journey up the network stack, starting from how two computers can talk to each other.

• **Remember: start your projects now!**