Lecture 1
Introduction

David Andersen
School of Computer Science
Carnegie Mellon University

15-441 - Computer Networks
http://www.cs.cmu.edu/~dga/15-441/F08/

Today's Lecture

- Course outline and goals.
- Whirlwind Tour of Networking ™

Course Staff

- Instructors
  - David Andersen
    - <dga@cs.cmu.edu> , Wean Hall 8206
  - Seth Goldstein
    - <seth@cs.cmu.edu>, Wean Hall 7122
- Teaching assistants:
  - Joshua Primero - <jprimero@andrew.cmu.edu>
  - Xunnan Fu - <xfu@andrew.cmu.edu>
  - Vijay Panghal <vpanghal@andrew.cmu.edu>

Course Goals

- Become familiar with the principles and practice of data networking.
  - Routing, transport protocols, naming, ...
- Learn how to write networked applications:
  - An IRC server
  - A peer-to-peer file transfer program
- Get some understanding about network internals in a hands on way.
  - You'll implement a routing protocol for your IRC server
  - TCP-style congestion control

Course Format

- ~30 lectures
  - Cover the "principles and practice"
  - Readings are posted beforehand
- 4 homework assignments
  - "Paper": Do you understand and can you apply the material?
  - Feedback to students and instructors
  - "Lab": Illustrate networking concepts
- Mid-term and final.
- 2 programming projects.
  - How to use and build networks / networked applications
  - Application layer; include key ideas from kernel
  - Larger, open-ended group projects. *Start early!*

Recitation Sections

- Key 441 objective: system programming
- Different from what you've done before!
  - Low level (C)
  - Often designed to run indefinitely. Handle all errors!
  - Must be secure
  - Interfaces specified by documented protocols
  - Concurrency involved (inter and intra-machine)
  - Must have good test methods
- Recitations address this
  - "A system hackers' view of software engineering"
  - *Practical* techniques designed to save you time & pain!

Sounds Great! How Do I Get In?

- Currently 55 people are enrolled, and no people are on the waiting list.
  - Cool.
  - We'll update more if we end up with a waitlist due to unexpected, sudden popularity because the class is *just that cool.*
- But just to be sure:
  - If you do not plan to take the course, please drop it within a reasonable amount of time
  - And if you do, please make sure you're registered!
    - We'd like a reasonable headcount
    - Lets us use the online roster to create your logins/ etc. for assignments

Administrative Stuff

- Watch the course web page.
  - Handouts, readings, ..
- Read courses bboards.
  - "Announce" for official announcements
  - "General" for questions/answers
- Office hours posted on web page.
- Course secretary
  - Angela Miller, Wean Hall 8215
- Office hours this week by email / appointment
  - Final office hours posted Thursday
- Books – have people gone to the bookstore? How many copies? *Should* be there…

Grading

- Roughly equal weight in projects and testing on course contents.
- ~20% for Project I & II
- ~25% for Project III
- 15% for Midterm
- 25% for Final exam
- 15% for Homeworks
- You need to demonstrate competence in both projects and tests to pass the course. *Don't fail any component.*

Policy on Collaboration

- Working together is important.
  - » Discuss course material in general terms
  - » Work together on program debugging, ..
- Parts *must* be your own work
  - » Homeworks, midterm, final
- Projects: Solo (P1) + Teams of two (P2,P3)
  - » Collaboration, group project skills
  - » Both students should understand the entire project
- Web page has details.
- Things we don't want to have to say: We run projects through several cheat-checkers against *all* previously and concurrently handed in versions…

Policy on Late Work and Regrading

- No assignments with a "short fuse".
  - » Homeworks: ~1 week
  - » Projects: ~5 weeks
- Late work will receive a 10% penalty/day.
  - » No penalty for a limited number of handins - see web page
  - » No assignment can be more than 2 days late
- Only exception is documented illness and family emergencies
- Start on time!
  - » Every year some students discover that a 4 week project cannot be completed in a week
- Requests for regrading must be submitted in writing with course secretary within 2 weeks.
  - » Regrading will be done by original grader

This Week

- Intro – what's this all about?
- Protocol stacks and layering
- Next week? Applications and Network programming review.
  - » Socket programming (213 review++)
  - » Recitations start next week: Project management (SVN, etc.)
- Course outline:
  - » Low-level (physical, link, circuits, etc.)
  - » Internet core concepts (addressing, routing, DNS)
  - » Advanced topics
- On to the good stuff…

What Is a Network?

- Collection of nodes and links that connect them
- This is vague. Why? Consider different networks:
  - Internet
  - Andrew
  - Telephone
  - Your house
  - Others – sensor nets, cell phones, …
- Focus on Internet, but understand important common issues and challenges

Networks Juggle Many Goals

- Efficiency – resource use; cost
- The "ilities":
  - Evolvability
  - Managability
  - Security (securability, if you must)
  - Ease of:
    - Creation
    - Deployment
    - *Creating useful applications*
  - Scalability

Challenges for Networks

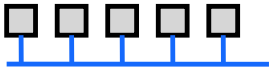
- Geographic scope
  - The Internet vs. Andrew, etc.
- Scale
  - The Internet vs. your home network
- Application types
  - Email vs. Videoconferencing
- Trust and Administration
  - Corporate network – one network "provider"
  - Internet – 17,000 network providers

How to Draw a Network

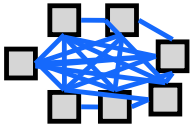
Building block: The Links

- Electrical questions
  - » Voltage, frequency, …
  - » Wired or wireless?
- Link-layer issues: How to send data?
  - » When to talk – can everyone talk at once?
  - » What to say – low-level format?
  - » Stay tuned for lecture 5
- Okay… what about more nodes?

- … But what if we want more hosts?



One wire

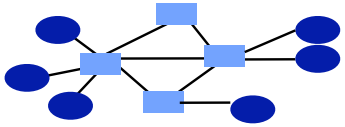


Wires for everybody!

- Scalability?!

Multiplexing!

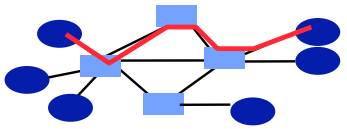
- Need to share network resources



- How? Switched network
  - » Party "A" gets resources sometimes
  - » Party "B" gets them sometimes
- Interior nodes act as "Routers" or "Switches"
- What mechanisms can share resources?

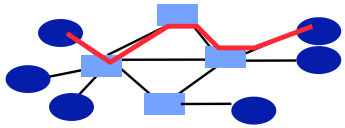
Circuit Switching

- Source first establishes a connection (circuit) to the destination.
- Source sends the data over the circuit.
  - » then the connection is torn down.
- Example: telephone network
  - » Early early versions: Human-mediated switches.
  - » Early versions: End-to-end electrical connection!
  - » Today: Virtual circuits or lambda switching



Circuit Switching 2

- What about many connections?
  - » Many wires (e.g., those big 200-pair cables you sometimes see)
- A more practical approach is to multiplex multiple circuits over a single "fast" wire.
  - » Can benefit from improvements in technology
  - » Fewer wires
  - » Multiplexing is discussed in more detail in Lecture 5

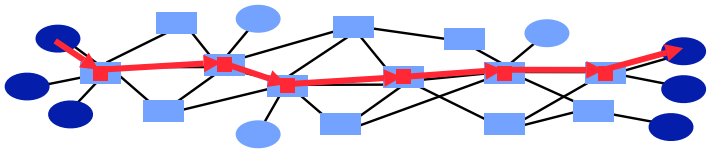


Circuit Switching Discussion

- Circuits have some very attractive properties.
  - » Fast and simple data transfer, once the circuit has been established
  - » Predictable performance since the circuit provides isolation from other users
  - » E.g. guaranteed bandwidth
- But it also has some shortcomings.
  - » How about bursty traffic
    - – circuit will be idle for significant periods of time
  - » How about users with different bandwidth needs
    - – do they have to use multiple circuits
- Alternative: packet switching.

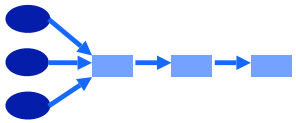
Packet Switching (our emphasis)

- Source sends information as self-contained packets that have an address.
  - » Source may have to break up single message in multiple
- Each packet travels independently to the destination host.
  - » Routers and switches use the address in the packet to determine how to forward the packets
- Analogy: a letter in surface mail.



Statistical Multiplexing

- Switches arbitrate between inputs



- Can send from *any* input that's ready
  - » Links never idle when traffic to send
  - » (Efficiency!)
- What networks can we build with these tools?

Local Area Networks (LANs)

- Benefits of being "local":
  - Lower cost
  - Short distance = faster links, low latency
    - Efficiency less pressing
  - One management domain
  - More homogenous
- Examples:
  - Ethernet (Lecture 7)
  - Token ring, FDDI
  - 802.11 wireless (Lecture 21)

Wide Area Networks

Distance makes things harder:

- High(er) delays and cost → Need efficiency
- Larger size → Need scalability
- Heterogeneity:
  - Traffic types
  - Host needs
- Administrative diversity → Management harder

Let's look at one prominent example:

"The Internet"

- An inter-net: a network of networks.
  - A set of networks that are connected with each other
  - Networks are connected using routers that support communication in a hierarchical fashion
  - Often need other special devices at the boundaries for security, accounting, ..
- The Internet: the interconnected set of networks of the Internet Service Providers (ISPs) providing data communications services.
  - About 17,000 different networks make up the Internet
- In order to inter-operate, all participating networks have to follow a common set of rules.

Challenges of the Internet

- Scale: 100,000,000s of hosts
- Heterogeneity:
  - 18,000+ administrative domains
  - Thousands of applications
  - Lots of users
  - Fast links, slow links, satellite links, cellular links, carrier pigeons
- Diversity of network technologies
- Adversarial environment
- Oh, and let's make it easy to use…

Implementing Packet-Switched Networks

- Requirements for packets:
  - Header information: Addresses, etc. (Lecture 9)
  - Data. What is packet size limit? (Lectures 5—9)
  - Everybody has to agree on these for interoperability
- How do packets reach destination? Routing
  - Nodes in network forward packets towards destination
  - Routing tells nodes where to send the packets they receive
    - Design questions: What criteria to decide?
      - Destination is a must
      - Source?
      - "Type"?

Routing

- Who chooses the routes?
  - A human: Static routing
  - Centralized routing (telenet, c.a. 1980s)
  - Distributed routing (Internet, …)
- Distributed routing uses a *Routing Protocol*
  - Many different protocols are in use.
  - Inside an organization: RIP, OSPF, etc (Lecture 11)
  - Between organizations: BGP (Lecture 12)

Network Service Model

- What is the *service model?*
  - Ethernet/Internet: *best-effort* – packets can get lost, etc.
- What if you want more?
  - Network can do it – Quality of Service
    - Benefits of circuit switching in packet-switched net
    - Hard in the Internet, easy in restricted contexts
    - Lecture 20
  - Hosts can do it – end-to-end *Transport protocols*
    - TCP performs end-to-end retransmission of lost packets to give the illusion of a reliable underlying network.

    (Lectures 16—19)

Using Networks

- Layering and abstraction
  - Protocol stacks failicate re-use
  - Hide underlying complexity from the programmer
  - (Lecture 3)
  - Protocol reuse *and* code/library reuse
- Tuesday's lecture: Programmer API
- Many "human-friendly" abstractions:
  - Higher-level protocols (e.g., reuse the Web's HTTP instead of writing your own!).
  - Naming (www.google.com vs. 64.233.161.99)
    - The Domain Name System, or DNS (Lecture 13)

Using Networks *Securely*

- The Internet is an unfriendly place
  - » Hacking, viruses, denial-of-service, etc.
- Cryptography to the rescue:
  - » Secure Sockets Layer (SSL) – https://www.foo.com/
  - » Key management, etc.
  - » Lecture 25
- Policy control to the rescue:
  - » Firewalls / Denial of Service (Lecture 26)
  - » Network address translation / virtual private networks (NAT, VPN) – Lecture 14

Applications

- All well and good to have networks that deliver packets, but what do we actually *do* with them?
- The Web (Lecture 23)
- Peer to Peer (Lecture 24)
- Funky research stuff (Lecture 27)

- Class Projects (…)
  - » Remember, get started early. ☺