Today’s Lecture

• Administrivia

• Whirlwind tour of networking

Instructors

• Instructors.
  • Srini Seshan
    • srini@cs.cmu.edu, Gates Hall 8123
  • Seth Goldstein
    • seth@cs.cmu.edu, Gates Hall 7111

• Teaching assistants.
  • Kaushik Lakshminarayanan
  • Rui Meireles
  • Dae Gun Won

Course Goals

• Become familiar with the principles and practice of data networking
  • Routing, transport protocols, naming, ...

• Learn how to write applications that use the network
  • 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”
  - Complete readings before lecture
- 4 homework assignments
  - "Paper": Do you understand and can you apply the material?
  - "Lab": Illustrate networking concepts
  - Loosely tied to lecture materials
  - Teach networking concepts/tools
- 3 programming projects
  - How to use and build networks / networked applications
  - Application-layer programming; include key ideas from kernel
  - Larger, open-ended group projects. Start early!
- Midterm and final
  - Covers each of the above 3 parts of class

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 76 people are enrolled, and 33 people are on the waiting list.
  - If you do not plan to take the course, please drop it ASAP so somebody else can take your place!
- We give preference to:
  1. Students attending class (sign in sheet)

Administrative Stuff

- Watch the course web page
  - Handouts, readings, ..
- Read bboards
  - academic.cs.15-441[announce] for official announcements
  - cyrus.academic.cs.15-441.discuss for questions/answers
- Office hours posted on web page
  - By appointment this week
- Course secretary
  - Angela Miller, Gates 9118
Grading

- Roughly equal weight in projects and testing
  - 45% for Project I, II and III
  - 15% for Project II
  - 15% for Midterm exam
  - 25% for Final exam
  - 15% for Homework
- You MUST demonstrate competence in both projects and tests to pass the course
  - Fail either and you fail the class!

Policy on Collaboration

- Working together is important
  - Discuss course material in general terms
  - Work together on program debugging, ...
  - Final submission 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...

Late Work and Regrading

- Late work will receive a 15% penalty/day
  - No assignment can be more than 2 days late
  - No penalty for a limited number of handins - see web page
  - Only exception is documented illness and family emergencies
- Requests for regrading must be submitted in writing to course secretary within 2 weeks.
  - Regrading will be done by original grader
- No assignments with a “short fuse”
  - Homeworks: ~1-2 weeks
  - Projects: ~5 weeks
  - Start on time!
  - Every year some students discover that a 5 week project cannot be completed in a week

This Week

- Intro – what’s this all about?
- Protocol stacks and layering
- Recitations start this week: Socket programming (213 review++)
- On to the good stuff…Whirlwind tour of networking
  - Course outline:
    - Low-level (physical, link, circuits, etc.)
    - Internet core concepts (addressing, routing, DNS)
    - Advanced topics
What is the Objective of Networking?

- Enable communication between applications on different computers
  - Web (Lecture 22)
  - Peer to Peer (Lecture 23)
  - Audio/Video (Lecture 20)
  - Funky research stuff (Lecture 27)
- Must understand application needs/demands (Lecture 3)
  - Traffic data rate
  - Traffic pattern (bursty or constant bit rate)
  - Traffic target (multipoint or single destination, mobile or fixed)
  - Delay sensitivity
  - Loss sensitivity

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, …
- Class focuses on Internet, but explores 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
- Scale
  - The Internet vs. your home network
- Application types
  - Email vs. video conferencing
- Trust and Administration
  - Corporate network – one network “provider”
  - Internet – 17,000 network providers
How to Draw a Network

Basic Building Block: Links

• Electrical questions
  • Voltage, frequency, …
  • Wired or wireless?

• Link-layer issues: How to send data?
  • When to talk – can either side talk at once?
  • What to say – low-level format?
  • Lecture 5

• Okay… what about more nodes?

Basic Building Block: Links

• … But what if we want more hosts? (Lectures 6 & 7)

  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 “Switches”

• What mechanisms to share resources?
Back in the Old Days…

• Source first establishes a connection (circuit) to the destination
  • Each switch along the way stores info about connection (and possibly allocates resources)
• Source sends the data over the circuit
  • No need to include the destination address with the data since the switches know the path
• The connection is explicitly torn down
• Example: telephone network (analog)

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.
  • Switches use the address in the packet to determine how to forward the packets
  • Store and forward
• Analogy: a letter in surface mail.
Packet Switching – Statistical Multiplexing

- Switches arbitrate between inputs
- Can send from any input that’s ready
  - Links never idle when traffic to send
  - (Efficiency!)

Packet Switching Discussion

- Efficient
  - Can send from any input that is ready
- General
  - Multiple types of applications
  - Accommodates bursty traffic
    - Addition of queues
- Store and forward
  - Packets are self contained units
  - Can use alternate paths – reordering
- Contention (i.e. no isolation)
  - Congestion
  - Delay

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 6)
  - Token ring, FDDI
  - 802.11 wireless (Lecture 25)

Internet

  - 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)
  - About 17,000 different networks make up the Internet
Challenges of the Internet

- Heterogeneity
  - Address formats
  - Performance – bandwidth/latency
  - Packet size
  - Loss rate/pattern/handling
  - Routing
  - Diverse network technologies → satellite links, cellular links, carrier pigeons

- Scale
  - 100,000,000s of hosts
  - 18,000+ administrative domains,
  - Thousands of applications
  - Adversarial environment
  - Oh, and let’s make it easy to use…

  - How to translate between various network technologies?

Internet Design

- In order to inter-operate, all participating networks have to follow a common set of rules

- E.g., requirements for packets:
  - Header information: Addresses, etc. (Lecture 9)
  - Data. What is packet size limit? (Lectures 5—9)

How To Find Nodes?

Need naming and routing
Lectures 8-13
**Naming**

What’s the IP address for www.cmu.edu?

**It is 128.2.11.43**

Computer 1  
Local DNS Server

Translates human readable names to logical endpoints

**Routing**

Routers send packet towards destination

H: Hosts  
R: Routers

**Network Service Model**

- What is the service model?
  - Ethernet/Internet: best-effort – packets can get lost, etc.
- What if you want more?
  - Performance guarantees (QoS)
  - Reliability
    - Corrupt
    - Lost packets
  - Flow and congestion control
  - Fragmentation
  - In-order delivery
  - Etc…

**What if the Data gets Corrupted?**

Problem: Data Corruption

Solution: Add a checksum

0,9 → 6,7,8,21 → 4,5,7 → 1,2,3
What if Network is Overloaded?

Problem: Network Overload
- Short bursts: buffer
- What if buffer overflows?
  - Packets dropped
  - Sender adjusts rate until load = resources → "congestion control"

Solution: Buffering and Congestion Control

What if the Data gets Lost?

Problem: Lost Data

Solution: Timeout and Retransmit

What if the Data Doesn't Fit?

Problem: Packet size
- On Ethernet, max IP packet is 1.5kbytes
- Typical web page is 10kbytes

Solution: Fragment data across packets

What if the Data is Out of Order?

Problem: Out of Order

Solution: Add Sequence Numbers
Networks Implement Many Functions

- Link
- Multiplexing
- Routing
- Addressing/naming (locating peers)
- Reliability
- Flow control
- Fragmentation
- Etc….

Meeting Application Demands

- Sometimes interior of the network can do it
  - E.g., Quality of Service
    - Benefits of circuit switching in packet-switched net
    - Hard in the Internet, easy in restricted contexts
    - Lecture 21
  - OR hosts can do it
    - E.g., 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

Next Lecture

- How to determine split of functionality
  - Across protocol layers
  - Across network nodes

- Read two papers on the motivations for the Internet architecture:
  - "The design philosophy of the DARPA Internet Protocols", Dave Clark, SIGCOMM 88
  - "End-to-end arguments in system design", Saltzer, Reed, and Clark, ACM Transactions on Computer Systems, November 1984