CS 332: Computer Networks

Introduction

Professor Doug Szajda
Thanks!

- I’ve taught this course many times, the most recent being Spring 2014. Each time calls for a rethinking of the topics and new material.

- Much of the material I’ll use in the course has been supplied by Professor Patrick Traynor of the University of Florida
  - In fact, many assignments, almost all of the slides, etc, are taken word for word.

- I don’t feel bad about this: Patrick took this course with me at UR in the Spring of 2001.
  - So in a sense, it’s payback! (And some of the material was originally mine!)
What’s this all about?
A Modern Day Silk Road

- We live with nearly constant access to the most extensive system ever built by human beings.
  - We may never build anything bigger.

- The Internet quickens the exchange of ideas, goods, news, and improves the quality of life for a large portion of the world’s population.

- It even impacts you where you least expect it.
  - E.g. Shipping/Supply Chain Management
Why Do I Need This Course?

• As engineers and scientists, you need to understand the underpinnings of our global communications networks.

• With this information, you will be able to help design and implement the next generation of networked systems.

• As everything “comes online”, you need to understand the implications and architecture of these systems.
Goals

• My Goal: To provide students with the tools to evaluate current, and develop new, networked systems.
  ‣ Networking Fundamentals
  ‣ Recognize trade-offs between different technologies.
  ‣ Design and implement software with a communications interface.
  ‣ Prepare you for advanced work in this area.

• I love this material and want to help you all love it too.
What Topics Will We Cover?

• This is an introductory course, so you will get a bit of everything:
  ‣ TCP, IP, UDP, DNS, BGP, Email, P2P, Routing Algorithms, Congestion Control, Queuing Fundamentals, Network Management, Wireless, Cellular, Security, Ethics and lots more...

• I will be maintaining the course at: http://www.richmond.edu/~dszajda/classes/cs332/Spring_2016/index.html

• Assignments, slides and other information will be made available here.
Textbooks

- There are two *required* books for this class:
  - Computer Networking: A Top-Down Approach
  - TCP/IP Sockets in C: Practical Guide for Programmers
- Readings will come from the first; projects will be made easier using the second.
Assignments/Workload

• There will be a reading assignment for nearly every class.
  ‣ Reading must be done before the class period.

• We will have a total of three homework assignments:
  ‣ Problems will come from the book and the professor.
  ‣ Check the website.

• We will also do four programming projects:
  ‣ All programming must be done in the specified language - no exceptions.
    • Generally C - check each assignment.
  ‣ Check the website.
Course Calendar

- The course calendar contains a listing of readings, assignments and deadlines.

- The page also contains links to additional readings.

- Check frequently! There will be changes! Students are responsible *(I will do my best to make announcements).*
Expectations

• This is going to be a hard course. The key to success is sustained effort. Failure to keep up with readings and assignments will result in poor grades and more critically, little understanding of the material.

• So what do we get for all our hard work?
  ‣ Perhaps a step toward helping to change the world (hopefully for the better)...

![Hard Work Never Killed Anyone]
Grading

• Grading in this class will be distributed as follows:
  10% Homework
  30% Projects
  25% Midterm
  30% Final
  5% Class Participation

• I reserve the right to give “Unannounced Learning Experiences”.

• You get the grade that you earn, so be sure that you earn a grade you like.
Lateness

• All homework is due at the beginning of class.

• Projects must be submitted as an attachment to an email sent to a special Box email.

• Late assignments are assessed a 15% per-day late penalty, with a maximum of four days.

• Students with legitimate reasons should contact the professor before the deadline to apply for an extension.
  ‣ Unless the problem is apocalyptic, don’t give me excuses.
Academic Integrity

• As scientists and engineers, we must trust each other to make progress.

• Numerous examples exist to show the consequences of this breakdown.
  ‣ Jan Hendrik Schon…

• Academic dishonesty, whether from cheating, copying, fabricating results or through any other dishonest practice will not be tolerated.
  ‣ I take this personally - you should too.
Course Outline

• Introduction to Networking (Chapter 1)
• Application layer (Chapter 2)
• Transport layer (Chapter 3)
• Network layer (Chapter 4)
• Link layer (some physical layer topics) (Chapter 5)
• Wireless, Mobility and Android (Chapter 6)
• Network security (Chapter 8)
• Multimedia networking (Chapter 7)
Chapter 1: Roadmap

1.1 What is the Internet?

1.2 Network edge
1.3 Network core
1.4 Delay & loss in packet-switched networks
1.5 Protocol layers and their service models
1.6 Networks Under Attack
1.7 History of Computer Networking and the Internet
1.8 Summary
What’s the Internet: “Nuts and Bolts” View

millions of connected computing devices:
hosts = end systems

running network apps

communication links

fiber, copper, radio, satellite

transmission rate = bandwidth

routers: forward packets (chunks of data)
Cool Internet Appliances

- IP picture frame
  http://www.ceiva.com/
- World’s smallest web server
  http://www-ccs.cs.umass.edu/~shri/iPic.html
- Web-enabled toaster + weather forecaster
- Internet phones
What’s the Internet: “Nuts and Bolts” View

**Protocols** control sending, receiving of msgs
- e.g., TCP, IP, HTTP, FTP, PPP

**Internet:**
“network of networks”
- loosely hierarchical
- public Internet versus private intranet

**Internet standards**
- RFC: Request for comments
- IETF: Internet Engineering Task Force
What’s the Internet: A Service View

communications infrastructure enables distributed applications:

- Web, email, games, e-commerce, file sharing

communication services provided to apps:

- Connectionless unreliable
- connection-oriented reliable
What’s a Protocol?

**human protocols:**
- “what’s the time?”
- “I have a question”
- introductions
  - specific msgs sent
  - specific actions taken when msgs received, or other events

**network protocols:**
- machines rather than humans
- all communication activity in Internet governed by protocols

Protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt.
What’s a Protocol?

- Example: A human protocol and a computer protocol:

- Question: What are some other human protocols?
1.1 What is the Internet?

1.2 Network edge

1.3 Network core

1.4 Delay & loss in packet-switched networks

1.5 Protocol layers and their service models

1.6 Networks Under Attack

1.7 History of Computer Networking and the Internet

1.8 Summary
A Closer Look at Network Structure:

• network edge: applications and hosts

• network core:
  ▶ routers
  ▶ network of networks

• access networks, physical media: communication links
The Network Edge:

- **end systems (hosts):**
  - run application programs
  - e.g. Web, email
  - at “edge of network”

- **client/server model**
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server

- **peer-peer model:**
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent, KaZaA
**Goal:** data transfer between end systems

- **handshaking:** setup (prepare for) data transfer ahead of time
  - Hello, hello back human protocol
  - set up “state” in two communicating hosts

- **TCP - Transmission Control Protocol**
  - Internet’s connection-oriented service

**TCP service** [RFC 793]

- reliable, in-order byte-stream data transfer
  - loss: acknowledgements and retransmissions
- flow control:
  - sender won’t overwhelm receiver
- congestion control:
  - senders “slow down sending rate” when network congested
Network Edge: Connectionless Service

• **Goal:** data transfer between end systems
  - same as before!

• **UDP** - User Datagram Protocol [RFC 768]:
  - connectionless
  - unreliable data transfer
  - no flow control
  - no congestion control

**App’s using TCP:**
- HTTP (Web), FTP (file transfer), Telnet (remote login), SMTP (email)

**App’s using UDP:**
- streaming media, teleconferencing, DNS, Internet telephony
Chapter 1: Roadmap

1.1 What is the Internet?
1.2 Network edge
1.3 Network core
1.4 Delay & loss in packet-switched networks
1.5 Protocol layers and their service models
1.6 Networks Under Attack
1.7 History of Computer Networking and the Internet
1.8 Summary
The Network Core

- mesh of interconnected routers

- **the** fundamental question: how is data transferred through net?

  - circuit switching: dedicated circuit per call: telephone net

  - packet-switching: data sent thru net in discrete “chunks”
Network Core: Circuit Switching

End-end resources reserved for “call”

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required
Network Core: Circuit Switching

network resources (e.g., bandwidth) divided into “pieces”

- pieces allocated to calls

- resource piece *idle* if not used by owning call *(no sharing)*

- dividing link bandwidth into “pieces”
  - frequency division
  - time division
Circuit Switching: FDM and TDM

Example:

- **FDM** (Frequency Division Multiplexing)
  - Four users sharing the same time slot by using different frequencies.

- **TDM** (Time Division Multiplexing)
  - Four users sharing the same frequency by using different time slots.
Numerical Example

• How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  
  ▶ All links are 1.536 Mbps
  
  ▶ Each link uses TDM with 24 slots/sec
  
  ▶ 500 msec to establish end-to-end circuit

Let’s work it out!
each end-end data stream divided into packets

- user A, B packets share network resources
- each packet uses full link bandwidth
- resources used as needed

resource contention:
- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - Node receives complete packet before forwarding

Bandwidth division into “pieces”

Dedicated allocation

Resource reservation
Packet Switching: Statistical Multiplexing

Sequence of A & B packets does not have fixed pattern, shared on demand: \textit{statistical multiplexing}.

TDM: each host gets same slot in revolving TDM frame.
Packet-Switching: Store and Forward

• Takes $L/R$ seconds to transmit (push out) packet of $L$ bits on to link of $R$ bps

• Entire packet must arrive at router before it can be transmitted on next link: 
  *store and forward*

• delay = $3L/R$ (assuming zero propagation delay)

Example:
• $L = 7.5$ Mbits
• $R = 1.5$ Mbps
• delay = 15 sec

more on delay shortly …
Packet Switching vs Circuit Switching

Packet switching allows more users to use network!

- 1 Mb/s link
- each user:
  - 100 kb/s when “active”
  - active 10% of time
- circuit-switching:
  - 10 users
- packet switching:
  - with 35 users, probability > 10 active less than .0004

Q: how did we get value 0.0004?
Packet Switching vs Circuit Switching

Is packet switching a “slam dunk winner?”

- Great for bursty data
  - resource sharing
  - simpler, no call setup
- Excessive congestion: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem (chapter 7)

Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?
Conclusion

- Welcome to CMSC 332 - this will be a great class.

- Go get the books and start doing the reading (if you haven’t already)!

- Go to the webpage and figure out when homeworks and projects will be due.

- Questions?