Course Overview

Spring 2014
ECE/CS 372 Introduction to Computer Networks

Lecture 1

School of Electrical Engineering and Computer Science
Oregon State University

Credit for lecture slides to Professor Bechir Hamdaoui
Adapted from Jim Kurose & Keith Ross (original copyright)
Course website
- http://classes.engr.oregonstate.edu/eecs/spring2014/cs372-001/
- Please write down this URL—all course material and information will be provided thru this site

Lectures
- Tuesday, Thursday 12:00-1:20pm, 212 Kearney Hall

Instructor
- Stephen Redfield (redfiels@eecs.oregonstate.edu)
- Office hours: TR 1:30-2:30pm @ Kelley Engineering Center Rm 2077
Lecture/Office/Lab Hours

- Tamara AlShammari
  - Location: KEC 2063
  - Lab hours: to help you with your labs
  - Hours TBD
    - Information can be found in course’s website

- MohammadJavad NoroozOliaee (MJ)
  - Location: KEC 2063
  - Office hours: to help you with your assignments
  - Hours TBD
    - Information can be found in course’s website
Prerequisite/Textbook

- CS or ECE 271 or an equivalent course
- Basic mathematical/probability skills

- Textbook is Required
  
Grading Policy

- **Assignments:** 15%
  - Each student must hand in one copy
  - 5 assignments: approx. 1 every two weeks
  - Check, Check Minus, X Grading

- **Labs:** 20%
  - Each student must hand in one copy
  - 4 labs: approx. 1 every two weeks

- **Bonus pop quizzes:** extra 2 to 10%
  - You need to get it completely right to receive an extra 1%

- **One midterm exam:** 30%
- **Final exam:** 35%
Lectures & assignments

Objective

- Deep understanding of basic and fundamental networking concepts, architectures, and philosophies

- IMPORTANT: this course is not about setting up your router at home, or writing a twitter program!!

Approach: how to do well in this course

- Easy: attend ALL lectures and do ALL assignments
- Do your assignments individually (Don’t use Solutions)
- Do NOT miss any Bonus Quiz (i.e., do not miss class)
- Some HW problems will be solved in class: this gives you the opportunity to clarify things further
Labs

Objective

- Understand how Internet protocols work
- Force network protocols to perform certain actions
- Observe and analyze protocols’ behavior

Approach

- Software tool: Wireshark
  - Install on your Laptop
  - Do this EARLY so you can avoid problems with Lab 1
- Allows you to sniff and analyze traffic sent/received from/by your end system: real measurement of Internet traffic
- Lab 1 is posted and is due next next Tuesday
Break

- Online Course Available
- We’ll start after a 10-min break
Our goal:

- learn basic network terminologies
- more depth, detail later in course
- approach:
  - use Internet as example
Chapter 1: roadmap

1 What is the Internet?
2 Network edge
3 Network core
4 Internet structure and ISPs
5 Protocol layers, service models
6 Delay & loss in packet-switched networks
What’s the Internet: a “service” view

- communication infrastructure enables distributed apps:
  - Enables apps to communicate
  - Web, email, games, e-commerce, file sharing

- communication services provided to apps:
  - Offers services
What’s the Internet: “nuts and bolts” view

- **hosts or end systems:** millions of connected computing devices
  - e.g., Laptops, workstations
  - running network apps

- **routers & switches:**
  - forward packets (chunks of data)

- **communication links**
  - e.g., fiber, copper, radio, satellite
What's the Internet: “nuts and bolts” view

- **Internet standards**
  - IETF (Internet Eng. Task Force)
    - RFC: Request for comments
  - IEEE: for links/hardware
    - E.g., Ethernet

- **network protocols**
  - control sending/receiving of messages
  - e.g., TCP, IP, HTTP, FTP, PPP
What's a protocol?

A human protocol and a computer network protocol:

- **Human Protocol:**
  - Hi
  - Got the time?
  - 2:00

- **Computer Network Protocol:**
  - TCP connection request
  - TCP connection response
  - <file>
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 (1) format, order of msgs sent and received among network entities, and (2) actions taken on msg transmission, receipt
Chapter 1: roadmap

1 What is the Internet?
2 Network edge
3 Network core
4 Internet structure and ISPs
5 Protocol layers, service models
6 Delay & loss in packet-switched networks
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: service models

- **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-to-peer model:**
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent, KaZaA
Chapter 1: roadmap

1 What is the Internet?
2 Network edge
3 Network core
4 Internet structure and ISPs
5 Protocol layers, service models
6 Delay & loss in packet-switched networks
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”

- dedicated resources: no sharing
- call setup required
- circuit-like (guaranteed) performance
- same path for all chunks
Network Core: Circuit Switching

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

- allocated pieces per call
- no sharing resource piece *idle* if not used by owning call
Network Core: Circuit Switching

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

Freq. Division Multiplx. (FDM)

Example:
4 users

Time Division Multiplx. (TDM)
Network Core: Packet Switching

each end-to-end data stream is divided into packets
- no dedication/reservation: all streams *share* resources
- no setup is required
- resources used as needed
- each packet uses full link bandwidth
- aggregate resource demand can exceed capacity
- no guarantee
Sequence of A & B packets does not have fixed pattern, shared on demand \( \Rightarrow \) **statistical multiplexing**.

Whereas in TDM, each host gets same slot (periodically)
Packet switching versus circuit switching

Circuit switching

- 2 circuits (use TDM)
- A reserves 1 circuit
- B reserves 1 circuit

Utilization = 50% only = 1 Mb/s

B: has no packets to send

Packet switching

- statistical multiplex.
- B uses full link since A is not using it

Utilization = 100% = 2 Mb/s

2 Mb/s

A

B

A

B
# Packet switching versus circuit switching

<table>
<thead>
<tr>
<th></th>
<th>Packet-switching</th>
<th>Circuit-switching</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resources</strong></td>
<td>sharing</td>
<td>dedicated</td>
</tr>
<tr>
<td><strong>Congestion</strong></td>
<td>may lead to it</td>
<td>admission control</td>
</tr>
<tr>
<td><strong>Overhead</strong></td>
<td>less overhead;</td>
<td>more overhead;</td>
</tr>
<tr>
<td></td>
<td>no connection</td>
<td>reserve resources</td>
</tr>
<tr>
<td></td>
<td>setup</td>
<td>1st</td>
</tr>
<tr>
<td><strong>Guarantee</strong></td>
<td>Best-effort</td>
<td>provide guarantee</td>
</tr>
<tr>
<td></td>
<td>no guarantee</td>
<td>good for multimedia</td>
</tr>
</tbody>
</table>
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?
  - The link’s transmission rate = 0.64 Mbps
  - Each link uses TDM with 10 slots/sec
  - 0.5 sec to establish end-to-end circuit

Let’s work it out! You have few minutes!

- Solution:
  - Bandwidth of circuit (in kbps) = \( 0.64 \times 1000 / 10 = 64 \) kbps
  - Time to send: \( 640 \text{ kbits} / 64 \text{ kbps} + 0.5 \text{ s} = 10.5 \text{ s} \)