Course Introduction

Jasleen Kaur

Fall 2014

Today’s Overview

- Introductions
- What is this course about?
- Administrative details
  - Course requirements, policies, resources
Introduction

- Instructor – Jasleen Kaur (me)
- You
  - Attendance
  - Graduate or undergraduate?
  - Why this course?

WHAT IS THIS COURSE ABOUT?

What will we cover? What will we not?
So What Is This Course About?

- Focus: Internet’s protocols and distributed services
  - Other networks exist too: Phone networks, ATM, …

- Course can be broken into two parts:
  - Part 1: Design of Computer Networks
    - How do you design a global-scale network that can be used to transfer information efficiently between end-users and applications?
  - Part 2: Design of Internet-wide Distributed Systems
    - How do you design (massively) multi-user and global-scale systems and applications on top of such a network?

- Emphasis on common design principles
  - Service models, Hierarchy, Randomization, Virtualization, Indirection, …

Part 1: Requirements for a Global Network

- Small Networks
  - Point-to-point links
    - Issues: error detection, error recovery
  - Multiple access links
    - Issue: contention-resolution
  - Switched networks
    - Issue: datagram-switching vs virtual-circuit switching

- Global scale and Autonomous co-existence
  - Scalable Addressing
  - Routing and Forwarding

- Reliability and Congestion-control
  - Challenges: scale, estimation, …
Part 1 Topics

- Transport protocols (TCP)
  - Congestion control and Reliability
- Internet routing architecture and algorithms
  - Distance vector, Link state, BGP
- The Internet Protocol (IP)
  - Scalable addressing
- Link-layer media access protocols
  - Token rings, 802.11

Part 2: Large-scale Distributed Systems

- Name resolution:
  - Domain Name System
    - Issues: scale and autonomy
- Information sharing:
  - Peer-to-peer file-sharing systems
    - Issues: scale, churn, fault-tolerance
- Content distribution:
  - Overlay Networks
    - Issues: resilience, autonomy
  - P2P Content-distribution systems
    - Issues: scale, incentives
- Cloud Networks
  - Target: Google, EC2
    - Issues: scale, consistency
What Will Not Be Covered?

- We will not discuss fine details of most protocols
- We will not discuss implementation details of most systems we study
- We will not cover:
  - Any specific technology
  - Socket programming
  - Application-layer protocols (other than DNS)

ADMINISTRIVIA

Requirements, Policies, Resources, …
Prerequisites

- COMP 431: Internet Services and Protocols
- COMP 530: Operating Systems
- A working knowledge of the UNIX program development environment
- Comfort with socket programming (in any language)

References

- Computer Networks: A Systems Approach
  - Peterson and Davie
- Several research papers
Course Grading (Within ± 10%)

- Programming and written assignments (25%)
  - Roughly 3 – 4 in the semester

- Course Project (30%)
  - Implement (as an application-layer overlay), or evaluate (through measurements), a network protocol or a distributed system
  - Groups of 2 – 4

- Exams
  - All exams are likely to be oral
  - Considering a midterm examination (around mid-October)
  - Final examination (25%)

- Class participation (10%)

Course web page

- [http://www.cs.unc.edu/~jasleen/Courses/Fall14-631](http://www.cs.unc.edu/~jasleen/Courses/Fall14-631)

- Go-to resource for
  - Handouts and slides
  - Assignments
  - Schedule (including exam dates)

- Monitor this page regularly!
Classroom Etiquette

- Class attendance is required
- Please arrive on time
  - Occasionally late is ok
    - Make sure you do not disrupt the class (sit in last 2 rows)
    - Habitual is NOT ok
- Please do NOT browse in class

Honor Code

- Working in groups on assignments is OK but...
  - You can only collaborate with students in this course
  - You can only collaborate on understanding the assignment and possible approaches
    - Every student must craft their own final solution
    - Every student must fully write up their own solution
  - All collaborators must be acknowledged in writing
- Code may never be shared
  - Collaboration on the mechanics of programming is OK
    - Debugging or designing each other’s programs is not OK
Reading Assignment: Networking Basics

- Layered architecture
- Packets, headers
- Encoding
- Framing