

## **CSE/EE 461: Introduction to Computer Communication Networks Autumn 2003**

**Instructor:** Tom Anderson ([tom@cs.washington.edu](mailto:tom@cs.washington.edu))  
**TAs:** Harsha Madhyastha ([harsha@cs.washington.edu](mailto:harsha@cs.washington.edu))  
David Sunderland ([dsund@cs.washington.edu](mailto:dsund@cs.washington.edu))  
**Time/Place:** MWF 3:30 - 4:20 in MGH 241

**Course Web:** <http://www.cs.washington.edu/education/courses/461/03au>  
All of the materials you need are accessible from the course website.

**Course Description:** The goal of this course is to help students learn how to design and implement network protocols that work – ones that are robust, efficient, scalable, interoperable, extensible, and secure. We cover the breadth of existing network protocols, including coding, routing, transport, congestion control, wireless and LAN media access, quality of service, content distribution, and encryption. Our focus, however, is forward-looking – what are the concepts needed to understand not only the protocols we currently use, but those that will be developed in the future. A large part of the course is the programming project -- to build a wireless ad hoc network capable of carrying real-time voice traffic.

**Prerequisites:** The course is open to any CSE or EE undergraduate major, with two requirements. First, the class involves a significant, graded programming project, and so you must have completed at least CSE 143 and preferably an additional course on data structures. The lectures and homework assignments also require a basic understanding of probability, such as provided by MATH 390/STAT 390 or STAT 391.

**Enrollment:** A signup sheet will be circulated during the first class for those who have not been able to enroll. We will admit a small number of students from this list, with names posted at my office (CSE 644) by 1:30pm on Wednesday. We will automatically enroll these students. No one will be added after Wednesday, regardless of future drops. Note that we need to strictly control the enrollment to allow every student to be assigned an IPAQ for the programming project.

**Mailing List:** Join the class mailing list right away by sending email to [majordomo@cs.washington.edu](mailto:majordomo@cs.washington.edu) with “subscribe cse461” as the contents. All messages sent to this list are archived and available on the course web page. We will use the email list for all course-related communication of general interest, in particular, to answer questions about assignments. You should feel free to use it for any class related discussion, and you are specifically encouraged to answer other student’s questions, and for posting hints for navigating the inevitable confusions that will occur in doing the programming projects.

**Textbooks:** Computer Networks: A Systems Approach, by Larry Peterson and Bruce Davie. Morgan Kaufmann, 3<sup>rd</sup> edition, 2003. This edition is nearly the same as the second edition, and so you may use that instead, at your own risk. However, we will only provide reading and homework assignments relative to the 3<sup>rd</sup> edition, so please make sure you at least know someone who has the 3<sup>rd</sup> edition so that you can make sure to stay in sync.

The programming project for the course will be implemented in Ruby, so you may also wish to purchase a Ruby book. We recommend, [Programming Ruby: The Pragmatic Programmer's Guide](#), by David Thomas and Andrew Hunt. The book's text is posted on the web at [www.rubycentral.com](http://www.rubycentral.com), and hard copies can be purchased via Amazon. Because of timing constraints, it is *not* stocked at the University Bookstore.

**Programming Project:** Over the quarter, we will design and implement a suite of protocols to turn a collection of IPAQ computers from an expensive set of paperweights into a wireless ad hoc network. You will do all your project work in pairs – no singletons will be allowed. There are four assignments, each worth 10% of your grade.

We will sign out IPAQ computers in section. You are responsible for them during the quarter, and must return them to receive your grade. IPAQ's will only be handed out to registered students. Because Ruby is highly portable, much of your development can be done on any convenient machine, including Windows, but the final IPAQ testing will require access to the UNIX machines in the CSE Labs. CSE majors should already have access and an account. Non-majors must fill out a request form and should submit that without delay.

**Grading:** Final: 40%; four Fishnet programming assignments: 10% each; two written homework assignments: 10% each. The written assignments must be done individually. The final will be held on Dec. 17, 2:30 – 4:30. There will be no midterm.

**Late Policy:** All assignments are due at the beginning of class and will be considered late once we leave the classroom; this includes any electronic and written turnins. We will use flexible slip days for both the project and homeworks; you may turn in the project assignments up to an aggregate of four calendar days late, and the homework assignments up to an aggregate of two days late. Solution sets will be therefore handed out shortly after assignments are due, and once solutions are handed out, no further late homework will be accepted

**Collaboration Policy:** Unless we specifically state otherwise, we encourage you to collaborate on homework provided (1) You spend at least 15 minutes on each problem alone, before discussing it with others, and (2) You write up every solution on your own (or with your partner for programming assignments), using your own words, and understand the solution fully. Copying someone else's written homework or programs is cheating (see below), as is copying from another source (prior year's material, etc.).

**Cheating Policy:** Cheating is a serious offense. If you are caught cheating, you can expect a failing grade and initiation of a cheating case in the University system. Basically, cheating is an insult to the instructor, to the department, and most importantly, to you. If you feel that you are having a problem with the material, or don't have time to finish an assignment, or have any number of other reasons to cheat, talk with the instructor. Just don't cheat. To avoid creating situations where copying can arise, never publicly post your solutions. If in doubt about what might constitute cheating, send the instructor email describing the situation.

## CSE/EE 461 Syllabus

### Introduction (1: Sept 29)

Peterson and Davie, Chap. 1.1 – 1.3

Thomas and Hunt, Programming Ruby: The Pragmatic Programmer's Guide, Chap 2 – 10

### Phase 1: Reliability: How do we build a network out of unreliable, distributed components?

a. Coding: encoding, framing, error detection (1: Oct 1)

PD Chap. 2.1 – 2.4

a. One hop: stop and wait, sliding window, RTT est., flow control (3: Oct 3-8)

PD Chap. 1.5, 2.5, 5.2

b. Multi-hop: flooding, forwarding, bridging, routing, mobility, circuits (3: Oct 10-15)

PD Chap. 3.1 – 3.2, 4.2

d. End to end principle (1: Oct 17)

Saltzer et al., End to End Arguments in System Design

### Phase 2: Resource Sharing: How do we share network resources without centralized control?

a. Media access: wired and wireless (2: Oct 20 - 22)

PD Chap. 2.6, 2.8

Bharghavan et al., MACAW: A Media Access Protocol for Wireless LANs.

b. Congestion control: TCP, AQM, credits, traffic engineering (4: Oct 24 - 31)

PD Chap. 3.4, 6.1 - 6.4

c. Real-time delivery: weighted fair queuing, reservations (2: Nov 3 - 5)

PD Chap. 6.5

### Phase 3: Growth and evolution: How do we cope as networks grow and evolve over time?

a. Layering: information hiding, TCP/IP, sockets, applications (2: Nov 7 - 10)

PD Chap. 1.4, 4.1, 9.2

D. Clark, Design Philosophy of the Internet Protocols

b. BGP: address aggregation and inter-domain routing (1: Nov 12)

PD Chap. 4.3

c. DNS, distributed hash tables, peer-to-peer discovery (2: Nov 14 - 17)

PD Chap. 9.1, 9.4.2

- d. NATs, load balancers, proxies, CDN's, overlays (2: Nov 19 - 21)  
PD Chap. 9.4
- e. Internet astronomy (1: Nov 24)
- f. Robustness (1: Nov 26)  
Anderson et al., Design Guidelines for Robust Internet Protocols

Phase 4: Cooperation and competition: How do we manage conflicting interests?

- a. Internet pricing, policy routing, and peering (1: Dec 1)  
W. Norton, Internet Service Providers and Peering
- b. Authentication: public and secret key encryption (1: Dec 3)  
PD Chap. 8.1-8.3
- c. Boundary devices: firewalls, VPNs (1: Dec 5)  
PD Chap. 4.5, 8.4
- d. Why you should never trust a computer (1: Dec 8)  
N. Borisov et al., Intercepting Mobile Communications: The Insecurity of 802.11.

Course Summary and Review (1: Dec 10)

Key dates:

Oct 10: Fishnet 1 due (neighbor discovery)

Oct 24: Fishnet 2 due (routing)

Nov 4: Problem set 1 due

Nov. 14: Fishnet 3 due (congestion control)

Nov. 25: Problem set 2 due

Dec. 5: Fishnet 4 due (sockets, real-time)

Dec. 17: Final exam