CS 471G Computer Networks Spring 2013

Syllabus

Read this and keep it! *10 January 2013*

Instructor: Prof. Ken Calvert, `calvert@netlab.uky.edu`
Office: 102F Davis Marksbury Building
Phone: 257-3961
Office Hours: MWF 3–4pm (after class); also TuTh 2–3pm
Class Meeting Time & Place: MWF 2–2:50, FPAT 263.
Web page: `http://protocols.netlab.uky.edu/~calvert/classes/471`
Textbook: *Computer Networking: A Top-Down Approach*, Sixth Edition, by James F. Kurose and Keith W. Ross, Pearson, 2013, ISBN 0-13-285620-4. This is the main text for the course.

In addition to the main text, you will need **one** of the following **required supplementary texts.** Which one you choose will depend on which language you want to use for programming assignments. (Note that University/Department computer systems do not in general support C#; however, you are welcome to use it if you like.)

- Michael J. Donahoo and Kenneth L. Calvert, *TCP/IP Sockets in C: Practical Guide for Programmers*, second edition, Morgan Kaufman, 2009, ISBN 978-0-12-374540-8
- Kenneth L. Calvert and Michael J. Donahoo, *TCP/IP Sockets in Java: Practical Guide for Programmers*, Morgan Kaufman, 2008, ISBN 978-0-12-374255-1
- David B. Makofske, Michael J. Donahoo, and Kenneth L. Calvert, *TCP/IP Sockets in C#: Practical Guide for Programmers*, Morgan Kaufman, 2004, ISBN 978-0-12-466051-7

These books were written to provide a gentle introduction to programming applications that use the network, without the need to spend class time on it. When you get the book, you should *read it on your own*. In other words, I will not assign specific readings from the socket books, but rather will expect you to use them as a reference in doing the programming assignments. *Of particular importance is the chapter in each book on how messages are constructed. It deals with low-level details of how information is laid out in memory that you may not have encountered before.*

1 Course Statement

This course is an introduction to the principles, applications, and technologies of computer networks, with a focus on the Internet. This is a *very large* subject, and no single course can make you an expert. Rather, the goal is to help you learn how to *think about networks* (and other systems involving communication between computing devices)—a skill that is likely to come in handy in a variety of contexts in the real world.

Please note: The official title of this course is still listed as "Computer Networks and Distributed Operating Systems", but it is in the process of being changed to just "Computer Networks". There's not enough time to come close to covering networks, let alone any material on distributed operating systems.

2 Expectations: Student Background

The prerequisites for this course are:

- Engineering standing;
- CS 315, Design of Algorithms.

- Ability to program in C, C++, Java, or C#.

If you are missing any of the prerequisites, you should see me as soon as possible after the first class to discuss the matter. In addition, familiarity with basic operating system concepts and mechanisms, as might be found in CS 470, is helpful but not required.

3 Learning Outcomes (Postconditions)

- Describe the operation of common systems and protocols used in the Internet.
- Describe the organization of the Internet infrastructure, and the functions performed by its various components.
- Explain the principles and algorithms related to: reliable data transfer over unreliable media; routing in large-scale networks; and sharing a transmission channel among multiple stations.
- Analyze and explain the factors affecting performance (throughput and delay) in systems that communicate over the Internet.
- Implement an application-level communication protocol from a given specification.

4 Topical Outline

The material will be presented in the following approximate order. It is (alas) very unlikely we will have time to cover all of the topics listed below.

Chapter 1: Basic Concepts. Course Overview; Ways to look at the Internet. Packets, Circuits, and multiplexing. Protocols, clients, servers, and connections. Delay, throughput, loss. Modularity, abstraction, and layering.

Chapter 2: Applications. Architectures and Requirements—client-server, P2P; bandwidth, reliability, timing, naming; TCP vs. UDP services. The Web and HTTP: message formats, cookies, caching, subtleties. Domain Name System: architecture, namespace hierarchy, database organization, security. Email system architecture: message formats, MIME, SMTP. Peer-to-Peer file sharing architectures: Napster, Gnutella, KaZaA, bitTorrent.

Chapter 3: Transport. Types of transport service; relationship to network layer; addressing; UDP protocol operation. Reliable data transfer principles: detecting losses, ARQ operation, go-back-N, selective repeat. TCP Details: packet formats, detecting loss, sliding window operation, connection management, 3-way handshake. Congestion control principles: where congestion comes from, solution space, TCP's congestion control mechanism

Chapter 4: Network. Network layer's job: routing, forwarding and switching; scalability and hierarchy. Anatomy of a router; packet processing; virtual circuit and datagram networks. Internet Protocol details: functionality, packet formats, addressing, subnetting. Routing algorithms: Link-state, distance-vector approaches; Interdomain routing principles. Protocol examples: RIP, OSPF, BGP

Chapter 5: Link Layer and Local Area Networks. Motivation. Broadcast channel characteristics; the framing problem. Error detection and correction. The multiple access problem; solution space; random-access protocols; ALOHA. Turn-taking protocols: polling, token-passing, probing. Ethernet and IEEE 802 details. Bridging: interconnecting Ethernets to make a big network; the limits of scalability.

Chapter 6: Wireless and Mobile. Types of wireless network; "spectrum"; FDMA, TDMA, and CDMA. IEEE 802.11 (WiFi): architecture, message formats, algorithms. Cellular networks: architecture; 2G, 3G, 4G. Mobile IP.

Chapter 7: Multimedia Networking. Applications: streaming audio and video; interactive applications. Compression principles and protocols. Voice telephony over IP best-effort service.

Chapter 8: Network Security. Security requirements; policy and mechanism; threats. Non-cryptographic solution space: firewalls, application gateways. Cryptographic building blocks; why they are not a panacea. SSL and ssh protocol details. IPSec.

5 Policies

This section describes a number of policies relevant to the course. Some of these I have derived over many years of experience; others are experimental. (I'll let you guess which is which.)

5.1 Grading

Your grade will be determined by your achievement of the course learning outcomes, as measured by your attendance, your performance on the homework, laboratory and programming assignments, and examinations. The weights assigned to each of these are as follows:

Attendance	10%
Homework assignments	25%
Laboratory/Programming assignments	30%
In-class Exam	15%
Comprehensive Final Examination	20%

All assignments of each type will have equal weight in determining your marks for that component. Final grades are "curved" by normalizing them to the highest score obtained by anyone in the class. For example, if the highest raw score obtained by anybody is 93, a student with a raw score of 85 would have a final mark of 91.40%. Letter grades are assigned on the basis of final scores, with the dividing line between A and B falling roughly around 90, between B and C around 80, between C and D around 70, and between D and E around 60. I *may* move the lines to match gaps in the final mark distribution. In all cases, however, your grade is completely determined by your numerical score on the above components and where the lines are drawn. Graduate students will do the same assignments, but their final score will be normalized to other graduate students, rather than undergraduates.

5.2 Late Assignments

Late assignments will be assessed a penalty of 15% per calendar day. That is: when an assignment comes in late, I mark down the number n of days late it is, and grade the assignment to get the score x. Then I subtract $0.15nx$ from x, and the result, if positive, becomes your score for that assignment. (A positive score requires that $n \leq 6$.) The number of calendar days an assignment is late is the number of midnights that have passed since it was due. Thus an assignment due on January 22nd is one day late whether it is handed in at 2:00am on January 23rd or at 11:30pm on January 23rd. An assignment (hard copy) is considered "handed in" when I have it in my hand.

Under no circumstances will an assignment be accepted once a solution for that assignment has been made available. Solutions for homeworks and programming assignments will typically be made available between three and seven days after the due date.

5.3 Accommodations for Disability

If you have a documented disability that requires academic accommodations, please see the instructor as soon as possible during scheduled office hours. In order to receive accommodations in this course, you must provide the instructor with a Letter of Accommodation from the Disability Resource Center (Room 2, Alumni Gym, 257-2754) for coordination of campus disability services available to students with disabilities.

5.4 Academic Conduct

I expect each student to do his or her own work. I don't mind if you help each other with understanding the material; in fact, I encourage it. However, anything that you turn in —homework, examinations, projects, anything— *must* be your own work. There will be **no tolerance** of plagiarism or other academic misconduct. Descriptions of these offenses are given in Section 6.3, Part II of the U.K. Student Rights and Responsibilities,[1] and students are expected to have read and understood that document. In case there are group assignments, the specific expectations will be spelled out in the assignment. Cases of suspected academic misconduct will be handled according to the procedures described in Section 6.4 of Part II of the Student Rights and Responsibilities. **Please Note** that these procedures have recently been extensively revised.

If you have a question or problem with any of these expectations, *inform me at the outset of the course*.

5.5 Contacting the Instructor

My schedule, including my office hours, is posted on my web page. The best way to contact me is by email. Note, however, that email is an *asynchronous* medium—if you need an immediate response, call me on the telephone. If you absolutely cannot see me during my regularly-scheduled office hours, send me email to make an appointment.

[1] `http://www.uky.edu/StudentAffairs/Code/`