ECE 4380/6380
Computer Communications

Class Location/Time:	106 ZFGEC, CURI. Mondays and Wednesdays, 5 – 7 p.m. 226 Riggs Hall, Clemson campus Online
Instructor:	Associate Professor Harlan B. Russell (harlanr@clemson.edu) 209 Zucker Family Graduate Education Center 316 Fluor Daniel Building, Tel: 656-7214
Office Hours:	Mondays and Wednesdays, 4-5 p.m. at the ZFGEC Other times by appointment
Grader:	TBA
Prerequisites:	Background in C programming and familiarity with basic probability

Course Description

We will explore the issues for networked communication from local area networks up to the global Internet and we will study a range of solutions to the associated problems. The course goals include the ability to list the concepts in wide-area network architecture, describe protocol layering, and discuss the functional requirements of each layer. By the end of the semester, you will be able to analyze different methods for the design and implementation of computer networks, and evaluate the ability of various protocols and equipment to meet specific performance requirements. Preq: Senior standing in Electrical or Computer Engineering or Computer Science

Course Objectives

Competences: Upon completion of this course, students should be able to:

a) Understand issues and solutions to networked communication from local area networks to the Internet.
b) List the concepts in wide-area network architecture.
c) Describe protocol layering and discuss the functional requirements of each layer.
d) Analyze different methods for the design and implementation of computer networks and evaluate the ability of various protocols and equipment to meet specific performance requirements.

Knowledge: Upon completion of this course, students will have been exposed to the following:

a) Foundation of networks including the classes of applications and the support for common services
b) Requirements for networked communications including user, administrator, and designer perspectives
c) Approaches to achieve scalable connectivity and cost-effective resource sharing
d) Network architecture including layering and protocols with the Internet architecture as the primary example
e) Physical layer models including encoding, framing, and error detection

f) Protocols for point-to-point and multiple-access links, including Ethernet and 802.11
g) Switching at both layer 2 (link layer) and layer 3 (network layer)
h) Layer 2 extended local-area network bridging and spanning tree protocols
i) Layer 3 datagram forwarding (IP), scaling with subnets and supernets, and supporting protocols
j) Routing and tradeoffs between distance-vector and link-state algorithms
k) Hierarchical protocols to scale network connectivity to billions of nodes with BGP and IPv6
l) Support for mobile devices and Mobile IP
m) End-to-end reliability, adaptive timeout, and flow control
n) Congestion control and resource allocation including WFQ, TCP congestion control, RED queue management

Skills: Upon completion of this course, students will be able to:

a) Implement client-server network software with the socket programming interface
b) Analyze network protocols to evaluate throughput and delay
c) Calculate bandwidth-delay products and select window sizes to maximize throughput for sliding window protocols
d) Evaluate local area network performance in the presence of collisions and back off delays
e) Determine latency in the presence of packet errors for multiple ARQ protocols
f) Design extended local area networks with redundant connectivity and select configuration details to provide min-max fairness and traffic balancing
g) Select subnets, masks, and CIDR assignments to maximize utilization of address space and minimize number of entries in forwarding tables
h) Evaluate convergence time and looping problems with distance-vector and link-state protocols
i) Calculate delay and fairness for active queue management strategies including weighted-fair queueing and random-early detection
j) Deploy extended local area network bridges and select spanning tree parameters
k) Configure OSFP routing daemons and observe control traffic in response to changes in connectivity
l) Design alternative state machines to adjust TCP establishment and teardown synchronization

Judgment Skills and Critical Abilities: Upon completion of this course, students will be able to

a) Contrast alternative design approaches to forwarding and routing protocols to address convergence time and scalability
b) Critique approaches to congestion control algorithms that are router-centric and host-centric with respect to TPC friendliness.
c) Determine appropriate performance requirements for destination-based versus virtual-circuit forwarding strategies
d) Evaluate trade-off designs at multiple protocol layers to meet various application performance requirements, including bulk data transfer, streaming content, and near real-time communications

Required Materials

Required textbook:	L. Peterson and B. Davie, *Computer Networks: A systems approach*, fifth edition, Morgan Kaufmann Publishers, Inc., 2011
Available at library:	M. J. Donahoo and K. L. Calvert, *The Pocket Guide to TCP/IP Sockets*, *C Version*, Morgan Kaufmann Publishers, Inc., 2001

Topical Outline

Week 1	Introduction to computer network architecture (Reading: 1.1 – 1.3) Implementing network software (Reading: 1.4) Performance and bandwidth-delay product (Reading 1.5)
Week 2	Direct link networks, encoding, framing, and error detection (Reading: 2.1 – 2.4) Reliable transmission, stop-and-wait, sliding windows, (Reading 2.5)
Week 3	Concurrent logical channels and separation of concerns (Reading 2.5) Shared access networks and carrier-sense multiple-access with collision detection Ethernet and multiple-access networks (Reading 2.6)
Week 4	Wireless networks and 802.11 Carrier-sense multiple-access with collision avoidance (Reading: 2.7) Exam 1
Week 5	Switching and Forwarding Bridges, learning bridges Spanning tree algorithm (Reading: 3.1)
Week 6	Basic Internetworking, datagram forwarding (IP) Subnetting, classless addressing, ARP, DHCP, ICMP, VPN (Reading: 3.2)
Week 7	Routing, distance vector and link state algorithms, RIP and OSPF (Reading 3.3) Advanced Internetworking, scalability, and routing areas (Reading: 4.1) BGP, IPv6 (Reading: 4.1)
Week 8	Multicast, multiprotocol label switching (Reading: 4.2 - 4.3) Mobile IP (Reading: 4.4) Exam 2
Week 9	End-to-end protocols, UDP (Reading: 5.1) Reliable byte stream (TPC) Connection establishment and termination, state diagram Sliding window and flow control with advertised window (Reading 5.2)
Week 10	Congestion control and resource allocation Queueing disciplines and weighted fair queueing (Reading: 6.1 – 6.2) TCP congestion control, slow start, fast retransmit/recovery (Reading: 6.3) Congestion-avoidance, random early detection, source-based avoidance (Reading: 6.4)

Grading

Homework assignments:	15%
Projects:	15%
2 Midterm exams:	15% each
Final Exam:	40%

Three projects will be assigned. The first will explore network programming in C. The second and third will use a virtual environment to experiment with distributed spanning trees and routing protocols.

Students receiving credit for 6380 will be required to cover some topics in more in-depth (e.g., multicasting and MPLS), and will have additional homework problems and exam questions.

A – 90% - 100%; B – 80 to < 90%; C – 70 to < 80%; D – 60 to < 70 & F – < 60% (4380 Students)
A – 90% - 100%; B – 80-89%; C – 70-79%; & F – < 70% (6380 Students)

Additional Policies

Attendance: You are expected to participate in all lectures, either at the ZFGEC, Riggs 226, or online. If you miss class, you are responsible for the material covered in the lecture and for any assignment made. All lectures are recorded and will be available through Canvas. Classes begin on Wednesday May 13 and the last day of class is Wednesday July 29. Our class does not meet on Monday May 25, or the week of June 15.

Online students: Students enrolled in an online section are required to connect to the live lecture and participate in course discussions. This ensures instructor-student interaction during each lecture. Additional interactions include communications using Canvas, email, video conferencing, and conference calls. One-to-one conference calls can also be scheduled as needed.

Examinations: We shall have two in-class exams, and a final exam. The final exam is scheduled for 3:00 p.m. to 5:30 p.m. on Friday, July 31. No make-up exams will be given unless an acceptable reason is presented to the instructor at least one week before the examination date. In the event of an emergency, the student must make direct contact with the instructor before an exam takes place or an assignment is due, preferably via email. If it is not possible to make arrangements before the scheduled event, then the student must contact the instructor as soon as it is safe and reasonable to do so. It is the student's responsibility to secure documentation of emergencies.

Exams must either be taken on site (at Clemson University or the ZFGEC) or, for online students only, remotely with a proctor. Except in an exceptional case, using a certified academic testing center will be required. Online or video proctoring services are not acceptable. Due to testing requirements, for the duration of the course, students must reside within 3 time zones of the Eastern US time zone. Detailed information regarding proctors will be provided to online students. Proctors must complete a proctor form by the date indicated in the course schedule and be approved by the instructor. Without proctor approval, a student will be required to take exams at a Clemson site.

Re-grades: Re-grade requests must be submitted in writing on a separate piece of paper within one week of the return of the graded item.

Homework: All problem sets are due at the time and date specified on the assignment. No late assignments will be accepted. Assignments must be submitted through Canvas.

Projects: Submission of machine problems will be electronically, and policies for late submission will be defined for each machine problem. A passing grade in the course will not be awarded without completion of all programming projects.

We encourage you to discuss interpretations of problems and assignments with each other but we expect that you will construct and write up your own solutions to any assignments that you turn in for credit. If students are found to have collaborated excessively or to have blatantly cheated (e.g., by copying or sharing answers or computer code), all involved will at a minimum receive grades of 0 for the first infraction. Further infractions will result in failure in the course and possibly recommendation for dismissal from the university.

For the programming assignments it is okay to talk with your classmates about the ideas. But when it comes time to write up your answers we expect your words and computer code to be yours alone. Do not share your work with your classmates, as they may not have the same work ethic as you do. Do not ask your classmates to share their files with you, either. In the end, your work should be a reflection of what you understand about the topic, presented in your own words and computer code.