Course Outline for COMP3331/9331 Computer Networks and Applications
2018 Semester 1
Credit Units: 6
Moodle Portal: https://moodle.telt.unsw.edu.au/course/view.php?id=31236

Lecturer-in-Charge (LiC): Mahbub Hassan, mahbub.hassan@unsw.edu.au, Room 607, K17
LiC Consultation Hours: Wednesdays 10am-11am and Thursdays 10am-11am (Rm 607, K17)
Course Administrator: Ali Dorri, ali.dorri@unsw.edu.au

Student query: All student queries will be handled via the Moodle forum, which will be monitored regularly. **Please use the emails of LiC and Course Admin only sparingly**.

Course Summary and Syllabus

This course is an introductory course on computer networks, aimed at students with a background in computer science / electrical engineering. We will focus on common paradigms and protocols used in present data communication. Through lectures, labs, and assignments, you will learn the theory and application of medium access control, congestion control, flow control, reliable transmission, addressing and naming, routing and switching, widely used network protocols such as Ethernet, IP, TCP, UDP, HTTP, DNS, ARP, etc., security threats and common defensive techniques, and special purpose networks such as content delivery networks, peer-to-peer networks and wireless networks. This is a combined undergraduate and postgraduate course, but the written exams for the postgraduate students will contain some different and more challenging questions.

Overall Syllabus (subject to minor variations): Introduction to Computer Networks, Application Layer, Transport Layer, Network Layer, Data Link Layer, Wireless and Mobile Networks, IoT, Network Security.

Course Timetable

There will be 3 hours of lectures every week, 2-hour on Wednesday 16:00 - 18:00 followed by 1-hour on Thursday 15:00 - 16:00, both in Ritchie Theatre. There will be 2-hours of labs per week starting Week 2. Lab enrolments are done online. More details about labs will be available from Moodle portal.

Course Aims

- To gain in-depth introduction to a wide range of topics in the field of computer networks including the Internet.
- To obtain hands-on understanding of network protocols.
- To gain skills in network programming (client server, socket programming), designing and implementing network protocols, evaluating network performance and problem-solving.

- To build necessary foundational knowledge required in more advanced networking courses

Student Learning Outcomes

After completing this course, students will:

- have a working knowledge of computer networks, and will be able to demonstrate their knowledge both by describing aspects of the topics and by solving problems related to the topics,
- have a solid understanding of the current architecture of the Internet and the entities involved in its operations,
- be able to identify soundness or potential flaws in network protocols
- be equipped with the necessary skills to design networked applications and protocols
- be able to implement and write protocols and applications in C, Java or Python
- be able to analyse and evaluate the performance of computer networks
- be able to capture and analyse network traffic
- be able to understand and explain security and ethical issues in computer networking

Graduate Attributes

This course contributes to the development of the following graduate capabilities:

Graduate Capability	Acquired in
scholarship: understanding of their discipline in its interdisciplinary context	lectures, labs, assignment
scholarship: capable of independent and collaborative enquiry	labs, assignment
scholarship: rigorous in their analysis, critique, and reflection	lectures, labs, exams, sample problems
scholarship: able to apply their knowledge and skills to solving problems	labs, assignment, exams, sample problems
scholarship: capable of effective communication	labs, assignment, lectures, exams
scholarship: information literate	all aspects of the course
scholarship: digitally literate	all aspects of the course
leadership: collaborative team workers	labs, assignment
professionalism: capable of independent, self-directed practice	all aspects of the course

professionalism: capable of lifelong learning	all aspects of the course
professionalism: capable of operating within an agreed Code of Practice	labs, assignment
global citizens: culturally aware and capable of respecting diversity and acting in socially just/responsible ways	labs, course forums

Assumed Knowledge

Before commencing this course, students should:

- have a good understanding of data structures and algorithms, basic probability theory.
- be able to write working programs in C, Java or Python (the course will include programming assignments and labs).

These skills are assumed to have been acquired in the courses: COMP1921 or COMP1927 or MTRN3530 (for undergraduates) and COMP9024 (for postgraduates).

Teaching Rationale and Strategies

This course takes a top-down approach to teaching computer networks. The rationale behind this is that most students have first-hand experience using applications running over the Internet. This allows them to relate to each layer of protocol stack as we travel down the layers. Once they are committed, they participate in appropriate cognitive aspects such as learning the details with a focus to understand them. Students get mentally prepared to answer questions where very often there is no single answer or the answers can be unexpected. This results in deep learning and gives students a sense of accomplishment and confidence.

Learning will be largely facilitated through the delivery of lectures. The lecturer-in-charge will offer 2 hours of first-come-first-served consultation hours every week for students wishing to seek further clarifications on any topics.

The hands-on laboratories will provide an opportunity to gain deeper understanding of the concepts discussed in the lectures. Students are required to attend laboratory sessions 2 hours per week. Labs are designed for students to learn and demonstrate the concepts learned in the lecture as well as gain network programming skills. Students are strongly encouraged to prepare well before attending each lab to ensure they can complete the main tasks on time. Each lab session will carry marks.

The homework problem sets will help in the development of problem-solving skills and in preparing for the exams.

The programming assignment will help students gain confidence with basic network programming and designing network protocols.

Teaching Strategies

Assessment

There will be four assessment components as listed below:

Lab Exercises (best of 5; 4 marks each)	20%
Programming (implementation) Assignment	20%
Mid-session Exam (closed book, MCQ)	20%
Final Exam (closed book, written)	40%

There is no supplementary for laboratory works and mid-session test. If you miss your mid-session test for some unavoidable reasons, the only possibility for you to cover the loss of marks is to apply (with proper evidence of misfortune) for a greater weighting in the final exam. For late submission of the assignment, 10% of the awarded marks will be deducted for each late day. No assignments will be accepted after one week of the original due date.

Overall and Final Assessment

Minimum requirements for obtaining a Pass grade or better are: a mark of 40%
in the Final Exam, and an overall grade of at least 50%. If you cannot clear the bar in the final exam (i.e. you get less than 40% in the final exam), the maximum final mark that will be reported is 40%. Under special circumstances, the LIC reserves the right to scale marks up or down.

Academic Honesty and Plagiarism

Plagiarism is defined as using the words or ideas of others and presenting them as your own . UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:
Learning Centre: Plagiarism and Academic Integrity
MyUNSW: Plagiarism and Academic Misconduct
CSE: Addendum to UNSW Plagiarism Guidelines
CSE: Yellow Form (whose terms you have agreed to)
Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

Resources for Students

Course Textbook: Computer Networking - A Top-Down Approach Featuring the Internet, J. Kurose and K. Ross, Pearson , Seventh Edition, 2017.

Reference Texts:
Unix Network Programming Volume 1 - Networking APIs: Sockets and XTI, W. Richard Stevens, Prentice Hall, Second Edition, 1998.
Java Network Programming, E. R. Harold, O'Reilly, Third Edition, 2004.
Learning Python, Mark Lutz, O'Reilly, Fifth Edition, 2013.
Computer Networks: A Systems Approach, Larry Peterson and Bruce Davie, Morgan Kaufmann, Fifth Edition, 2011.
Introduction to Computer Networks and Cybersecurity, John Wu and J. David Irwin, CRC Press, 2013.
Computer Networks, Andrew Tanenbaum and David Wetherall, Fifth Edition, Pearson, 2010.
Links to additional reading material will be available on Moodle.

Software:
For the labs, we will be using several Unix-based network utility programs. The purpose of these programs and information on how to use them will be provided in the lab handouts. We will also use a packet sniffing tool called Wireshark , which has been widely deployed on CSE machines. In addition, we will also use ns-2 , a widely used network simulator for few labs. Ns-2 is installed on the CSE lab machines. The simulator is written in C++. However, it uses OTclas its command and configuration interface. In lab exercises, we will use scripts written in OTcl. We will provide the OTcl scripts for the lab exercises . You will expected to run the scripts, make some minor changes in the scripts, and analyse certain performance metrics. You will not be required to write C++ code. Detailed resources for all tools used will be made available on Moodle.

Programming assignments are expected to be developed in C, Java or Python. Students are assumed to have sufficient expertise in one of these programming languages. Links to network programming in C, Java and Python will be available on Moodle. Sample code demonstrating a simple client/server application will also be supplied as a starting point for students.

Student Feedback and Continual Course Improvement

Student feedback on this course, and on the lecturing in this course, will be gathered towards the end of the session. Student feedback is taken seriously, and continual improvements are made to the course based in part on this feedback. The course questionnaire results go to the Head of the School of Computer Science and Engineering, who reads the results and follows up in cases where action is clearly needed.

Student feedback from last year called for better support for tutorials and programming, streamlining total student submission load, more detailed up-front marking guideline for assignment, more engaging mid-session exam questions, and a better platform for course delivery. We have listened to these student comments and revised the relevant processes and documents to improve the course delivery. In particular, we have (i) combined two separate assignment submissions into a single, but more involved, assignment with a detailed marking guideline included in the specification, (ii) revised the mid-session exam questions, (iii) made provision for getting help with programming tasks in the tutorials with additional programming consultations for different languages (Java/C/Python) after tutorial hours, (iv) removed student submission requirements for each lab sessions, but instead demonstrate completion of key tasks right during the session, and (v) switched to Moodle, which provides a fast and simpler interface

to course contents and activities. We hope that these changes will help improve student experience.

End of Course Outline for COMP3331/9331