Analysis of Computer Networks Courses in Undergraduate Computer Science, Electrical Engineering and Information Science Programs

Maurice F. Aburdene, Xiannong Meng and Gregory L. Mokodean
Bucknell University

Abstract

This paper presents an analysis of computer networks courses offered by universities and colleges in the departments of computer science, electrical engineering, or information science. The results are based on the information collected from course web sites from twenty-seven universities and colleges in computer science, electrical engineering and information science departments, primarily within the United States. The data analyzed include the course titles, course structure, textbooks used, major topics and how they are covered, projects, and laboratory exercises, if any. We found that the courses can be divided into three categories: those that cover the general topics of computer networks using some practical examples, those that specifically discuss Internet protocols, and those that work through a set of programming projects after students have had a previous network course.

1. Introduction

Pervasive use of the Internet, especially the World Wide Web (the web) has made teaching computer network courses a necessity for many universities and colleges. Students take network related courses hoping to gain first-hand knowledge of how the Internet works, how the web works, how to set up an operational network and how to program the network to deliver various applications. In this paper, we summarize characteristics of computer network courses from the web sites of 27 universities and colleges, mainly from the U.S. and from two other countries. We wanted to cover a variety of schools with different structures and objectives, including both public and private, predominately undergraduate and graduate, and different regions of the US, from what was available on the web. We examined how the course is taught, what textbooks are used, which subjects and practices are covered in laboratory exercises, if any, and the topics of course projects.

In general, three approaches are used to cover the main topics: bottom-up, which starts at the lowest (physical) layer of the protocol and works its way up; top-down, which starts from higher layers such as applications; and a mix of the two. Five textbooks dominate the required reading material. While most courses involve some projects, a few actually have dedicated weekly laboratory exercises. The contents of projects and laboratory exercises vary, and include: looking at the design and implementation protocols, programming actual network processors, writing client-server software in various programming languages, setting up operational network hardware and software, and experimenting with protocols using network simulation software.

"Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2004, American Society for Engineering Education"

We found a wealth of information about computer networks courses in this study. The analysis will help us revise our own network courses in both the electrical engineering and the computer science departments. We believe this information will be useful to our colleagues at other universities and colleges as well. There have been various surveys and studies about laboratory components of computer network courses such as [1,2,3,5,6,7,9,10,12,16,17,18]. There are also papers describing how a network course should be conducted [4, 11, 15, 20]. We have yet to find a general course survey across multiple disciplines and multiple schools such as the one we are presenting now.

The paper is organized as follows: Section 2 describes how this study was conducted. Section 3 discusses how the subject of computer networks is covered, bottom-up or top-down. Section 4 reviews the textbooks used in the network courses surveyed. Section 5 examines the subject and contents of the laboratory exercises. Section 6 describes the course projects. Section 7 addresses the issue of course objectives, outcomes, and assessments. Section 8 presents some observations and concluding remarks. The Appendix lists a summary of the data used in this study.

2. Method of Study

The information used in this study was collected from course web sites. The materials for each course contain at least a course description. Of the 61 web sites we visited, 27 have actual course materials such as syllabus, assignments, lecture schedules and assessment methods available on line; 42 listed their choice of textbooks, including the ones that indicate explicitly that no textbooks are used. The 61 courses come from a total of 27 different schools, all of which are from the U.S. except for one from Hong Kong and one from Australia. Most schools in our study offer network course(s) in more than one department and some offer courses at both the undergraduate and graduate levels.

There are 16 different department names in the 61 courses studied, including "Electrical Engineering", "Computer Science", "Computer and Information Science", and "Telecommunications and Networking". The course names have 36 different varieties, such as "Computer Networks", "Data Communications", "Network Essentials", and "Information Networks". One caveat of this type of study is that the information is collected from the web and reflects only the facts available at the time the information was collected. Also some of the information comes from course offerings in the previous semesters or years. A summary of the data can be found in the Appendix.

3. Top-down, Bottom-up or something in between

There are three different approaches to cover the network subjects among the courses we surveyed: top-down, bottom-up, and a mix of the two. These different approaches have a high correlation with the textbook chosen by the instructor. We will discuss the textbooks in detail in next section.

The bottom-up approach starts from the building blocks of computer networks. Most of computer science or information science programs use the physical or MAC layer access protocols as a starting point. Electrical engineering programs may discuss a bit more at the signal level. The topics then are moved up towards data link, network, transport, and application layers. The advantage of this approach is that the students will have a good understanding of the lower layers when they study the higher layers. For example, students will know how packets and frames are formed and used when studying various routing protocols at the network layer.

The top-down approach starts from the application and the transport layers then works its way down to the physical layer. This has been a relatively new approach in the recent years. We believe this trend has been strongly influenced by an excellent recent textbook by Kurose and Ross [14]. The advantage of top-down approach is that the course can start with interesting and attractive applications such as World Wide Web and Instant Messenger. Students are more likely to be motivated with real world applications. The courses then proceed with how a system works by "peeling the onion", one layer at a time.

The third is a mix of bottom-up and top-down approaches. Typically, one would start with some interesting applications and then discuss various topics as needed, not necessarily in any fixed order. Of the 27 sets of actual on-line course materials we were able to get, ten used the bottom-up (or very similar) approach, seven used the top-down (or very similar) approach, and ten used some mixed approach. It is interesting to note that the seven courses using the top-down approach all use [14] as the textbook, while the majority of the ten courses using bottom-up approach use [19].

4. Textbooks

Among the course web sites we visited, 42 provided some information about the textbook selection, indicating textbooks used or stating no textbook is required. Thirteen different books are mentioned as the primary texts, with [14] and [19] each listed as the main text for ten courses. Comer's book [8] is used by three courses, while [21] and [13] are each used by two. The remaining books are used by a single course. We now examine briefly the five textbooks used by more than one course to see the main topics covered in each of the books and how they are covered.

Kurose and Ross [14] is the only main textbook that explicitly indicates that the subjects are covered using a top-down approach. After an introductory chapter, it starts with a discussion of the layers in the following order: application, transport, network, and data link. Other subjects of interest, such as multi-media network, network security and network management, follow the presentation of the basic protocol stack. According to the authors, the top-down approach has three main advantages. It places emphasis on the applications, which is a fast-growing area; the applications can be used as a motivation tool to inspire interest in the students; and it enables students to build real applications early on so that they can see the process of applications development. The book by Kurose and Ross is currently in its second edition.

Peterson and Davie's book [19] has a history dating back to 1996. The subtitle of their book, *A Systems Approach* reveals well the intention of the authors. It starts with lower level protocols, such as encoding, framing, and MAC protocols i.e. Ethernet and token ring. It then moves its way up to packet switching and internetworking, where routing and Internet are discussed. End-to-end protocols such as UDP and TCP are then presented. The subjects of network security and

Copyright © 2004, American Society for Engineering Education"

applications are presented last. As a motivation for students to pursue further study, the authors present application examples early in the first chapter.

Tanenbuam's book [21] is similar to that of Peterson and Davie in that it takes a bottom-up approach. Its new edition came out this year (2003). The previous (third) edition of the book is dated 1996, which is the likely reason few schools are using his book.

Comer's book [8] stands out from other textbooks in that it concentrates Internet protocols, describing the details much more clearly than any other books. Using the bottom-up approach, the book first presents data transmission followed by packet transmission, before moving on to Internetworking and network applications. Since the chapters in the book are relatively independent of each other, instructors have the flexibility to cover the subject any the way they wish. Like all books aforementioned, Comer's book starts with network applications such as the web and the chat program to motivate the students.

Keshav's book [13], *An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network,* was published in 1997, at a time when ATM was still an important subject, thus the subtitle. It follows a bottom-up approach, discussing details of bits and bytes first, then moving to higher layers.

Though all texts feature one of the three approaches, each with its own strengths and weaknesses, the instructors also have their own unique way of teaching the contents. While the survey we conducted has a relatively small sample, it still showed which textbooks are popular and how the topics are presented.

5. Contents of Laboratory Exercises

The subject of computer networks is application-oriented. Having a hands-on component is a very critical and integral part of the course, and most courses in the survey have such a component. Hands-on experience can come in different forms: weekly programming exercises in a closed-lab environment; semester-long projects; and independent programming projects each of which is designed for a few weeks of work. There is no common definition of what constitutes a lab exercise, however, we considered a lab exercise to be a hands-on experience designed to be finished in a few hours and typically in a supervised, closed lab environment. We leave the subject of a programming project, including semester projects, to the next section.

We found seven courses that describe their lab exercises in some detail (title, main contents, and how the lab can be accomplished). Although this is a relatively small sample, it shows some common features. Of the seven courses that made their lab description available on-line, only two courses require a weekly lab, Renselaer Polytechnic Institute (RPI) and Bucknell University. Some courses list a few lab exercises for a semester and others use lab sessions only for tutorials. The lab exercises can be categorized into the following groups:

- UNIX system programming – introduce UNIX system calls and file systems
- Socket interface – familiarize students with socket interface so they can write application programs

Copyright © 2004, American Society for Engineering Education"

- Client/server programming – write client/server applications such as file transfer, or web service
- Packets collection and analysis – analyze packet traces and familiarize students with packet format
- Routing – implement and test various routing algorithms
- Performance measurements – measure protocol performance

Most of the lab exercises are done in an actual programming environment; some use simulation packages, while others use laboratory environment to have students read and experiment using tutorials.

6. Course Projects and Programming Assignments

Course projects and programming assignments are defined roughly as programming work that would take more than a few days. Typical programming assignments take a week or two to finish while projects last longer. Many courses assign research paper based projects. We concentrate here only on the projects involving substantial amount of programming and we found 12 courses describing such projects. Some courses have a specified project, others let students choose from a given set of projects, and still others allow students to define their own projects within some constraints. The levels of difficulty in these projects vary greatly.

- Implement some simplified but real protocols, e.g. TCP, IP, ICMP, on top of the socket interface
- Implement reliable transport protocol, e.g. TCP, on top of an un-reliable channel, e.g. UDP
- Use simulation to study the behaviors of various protocols
- Implement an application protocol such as FTP or HTTP

7. Course Objectives, Outcome, and Assessments

Computer networks is an important course in many of the engineering schools, whether it is offered in a computer science, electrical engineering, or an information science department. Here we try to summarize stated course objectives and outcomes, and examine the common assessment tools used in these courses.

The stated objectives vary from course to course. Some common elements of course objectives can be summarized as follows:

- Students master the basic concepts of modern computer networks
- Students learn how to develop network applications using a programming language
- Students understand general architecture of computer networks and how layered protocols of computer networks work
- Students are able to identify and explain current topics in computer networks, such as security and quality of service, among others

"Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition Copyright © 2004, American Society for Engineering Education"

While some course descriptions give more detailed goals or outcomes, most stated objectives are general in nature as seen from the above summary. We present the course outcomes of one author as an example here.

As a result of taking the network course, students should be able to

- *analyze point-to-point networks*
- *determine packet delay in point-to-point networks*
- *determine the important issues in the design of point- to- point networks*
- *analyze Aloha type networks*
- *analyze Ethernet type networks*
- *understand the layering issues in network design*
- *understand routing issues in network design*
- *analyze error detection and error correction codes (CRCs, Hamming codes)*
- *understand fundamentals of TCP/IP*
- *analyze distributed algorithms*
- *understand networking design issues*
- *read the networking technical literature*
- *analyze wireless networks*
- *realize the importance of security issues*

As far as assessments are concerned, we found most courses use traditional tools such as exams, quizzes, written homework assignments, programming assignments, semester projects, and research papers and their presentations. We have yet to find a course that doesn't have some type of written exam, with the exception of two pure project courses. One course has multiple laboratory tests, in addition to other assessment schemes.

8. Summary

This paper presented the results of our survey of computer network courses offered by 27 universities and colleges. First, we selected a set of schools that represent different types of institutional settings. We then visited the course web sites and collected information, such as course syllabi, course schedule, textbooks used, project descriptions, and laboratory exercises. We categorized the information into five groups: how the subjects were covered, which textbooks were used, the contents of the laboratory exercises, course projects, and course objectives and outcomes. The information reflected the facts at the time of the survey. The findings are empirical and informal. We hope the information is useful for colleagues who are teaching, or are preparing to teach computer network related courses.

References

1. J.P. Abraham "The Undergraduate Networking Course", white paper, ACM SIGCOMM Workshop on Computer Networking: Curriculum Designs and Educational Challenges, August 2002. Available at: http://www-net.cs.umass.edu/sigcomm_education_2002_compiled_white_papers.pdf

"Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition Copyright © 2004, American Society for Engineering Education"

2. M. Aburdene, D. Hyde, X. Meng, J. Janzi, B. Hoyt, R. Droms, "An Undergraduate Networked Systems Laboratory", in *Proceedings of the 2002 American Society for Engineering Education Annual Conference and Exposition,* Montreal, Quebec, Canada, June 2002.

3. M. Aburdene, J. Jantzi, X. Meng, "Creating and Testing VLANs using Network Switches", *Proceedings of the 2003 American Society for Engineering Education Annual Conference and Exposition,* Nashville, Tennessee, June 2003.

4. S. Akhtar, N. Al-Holou, M. Fienup, G. T. Finley, R. S. Roos, S. Tannouri, "The network course: old problems, new solutions", *ACM SIGCSE Bulletin, The proceedings of the 30th SIGCSE technical symposium on computer science education*, March 1999.

5. N. Al-Holou, K.K. Booth, and E. Yaprak, "Using Computer Network Simulation Tools as Supplements to Computer Network Curriculum," Session S2C, in *Proceedings of the 30th ASEE/IEEE Frontiers in Education Conference*, Kansas City, MO, October 18-21, 2000.

6. G. Bressan, M.V.S.O. Paula, T.C.M. Carvalho, and W.V. Ruggiero, "Infrastructure and Tools for a Computer Network and Data Communication Laboratory for a Computer Engineering Undergraduate Course," Session T4C, in *Proceedings of the 31st ASEE/IEEE Frontiers in Education Conference*, Reno, NV, October 10-13, 2001.

7. D.E. Comer, *Hands-on Networking with Internet Technologies*, Prentice Hall, Upper Saddle River, NJ 07458, 2002.

8. D. E. Comer, *Computer Networks and Internets*, Prentice Hall Publishing, 1st edition (1997), 2nd edition (1999), 3rd edition (2001), 4th edition (2003).

9. L. Fabrega, J. Massaguer, T. Jove, and D. Merida, "A Virtual Network Laboratory for Learning IP Networking," in *Proceedings of the 7th Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, pp. 161-164, Aarhus, Denmark, June 24-26, 2002.

10. G.A. Francia III and R.K. Smith, "The Design and Implementation of a Heterogeneous Computer Networking Laboratory," in *The Journal of Computing Sciences in Colleges* 16, pp. 234-241, the Consortium for Computing in Colleges, May 2001.

11. N. Greca, R. P. Cook, J. K. Harris, "Enhancing learning in a data communication and networking course with laboratory experiments", *The Journal of Computing in Small College*, vol. 19, issue 3, January 2004.

12. J.M.D. Hill, C.A. Carver, Jr., J.W. Humphries, and U.W. Pooch, "Using an Isolated Network Laboratory to Teach Advanced Networks and Security," in *Proceedings of the 32nd SIGCSE technical symposium on Computer Science*, pp. 36-40, Charlotte, NC, April 2001.

13. S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network*, Addison-Wesley Publishing, 1997.

14. J. F. Kurose and K. W. Ross, *Computer Networking – A Top-Down Approach Featuring the Internet*, Addison-Wesley Publishing, 1st edition (2001), 2nd edition (2003).

15. C. Liang, "A course on TCP/IP networking with Linux", *The Journal of Computing in Small College, Proceedings of the fifth annual CCSC northeastern conference on computing in small colleges*, vol. 15, issue 5, April 2000.

16. S. Liu, W. Marti, and W. Zhao, "Virtual Networking Lab (VNL): Its Concepts and Implementation," Session 3532, in *Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition*, 2001.

Copyright © 2004, American Society for Engineering Education"

17. J. Mayo and P. Kearns, "A Secure Networked Laboratory for Kernel Programming," in *Proceedings of the 3rd Conference on Innovation and Technology in Computer Science Education(ITiCSE*), pp. 175-177, Dublin, Ireland, July 1998.

18. D. Nelson and Y.M. Ng, "Teaching Computer Networking Using Open Source Software," in *Proceedings of the 5th Conference on Innovation and Technology in Computer Science Education(ITiCSE*), pp. 13-16, Helsinki, Finland ,July 2000.

19. L. L. Peterson and B. S. Davie, *Computer Networks – A Systems Approach*, Morgan Kaufmann Publishing, 1st edition (1996), 2nd edition (2000), 3rd edition (2003).

20. P. Steenkiste, "A network project course based on network processors", *ACM SIGCSE Bulletin, Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, vol. 35, issue 1, January 2003.

21. A.S. Tanenbaum, *Computer Networks*, 4th edition, Prentice Hall Publishing, 2003.

Biographical

MAURICE F. ABURDENE is the T. Jefferson Miers Professor of Electrical Engineering and Professor of Computer Science at Bucknell University. He has taught at Swarthmore College, the State University of New York at Oswego, and the University of Connecticut. His research areas include, parallel algorithms, simulation of dynamic systems, distributed algorithms, computer communication networks, control systems, computer-assisted laboratories, and signal processing.

XIANNONG MENG is an Associate Professor in the Department of Computer Science at Bucknell University in Lewisburg, Pennsylvania, U.S.A. His research interests include distributed computing, data mining, intelligent Web search, operating systems and computer networks. He received his Ph.D. in computer science from Worcester Polytechnic Institute in Worcester, Massachusetts, U.S.A.

GREGORY L. MOKODEAN is a first-year computer science student at Bucknell University. He is a Bucknell University Presidential Fellow.

Appendix A: List of Universities

Bucknell University
Carnegie Mellon University
Cooper Union
Harvard University
Hong Kong Polytechnic University
Johns Hopkins University
Lafayette College
Lehigh University
University of Massachusetts at Amherst
Massachusetts Institute of Technology
New South Wales University
University of Pennsylvania
Pennsylvania State University at State College

Copyright © 2004, American Society for Engineering Education"

Princeton University
Purdue University
Rensselaer Polytechnic Institute
Rose-Hulman Institute of Technology
University of South Carolina
University of South Florida
Stanford University
Stevens Institute of Technology
Swarthmore College
University of Texas -- San Antonio
University of California -- Berkeley
University of Illinois at Urbana-Champaign
University of Texas -- Austin
Worcester Polytechnic Institute

Appendix B: List of Course Titles

Advanced Computer Networks
Advanced Distributed Systems and Networks
Advanced Topics in Data Networking Protocols and Network Architecture
Advanced Topics in Wireless Networks
Communication Networks
Communications Networks for Computers
Computer Communications Networks
Computer Network Fundamentals
Computer Networking
Computer Networks
Computer Networks and Applications
Computer Networks and Computer Programming
Computer Networks and Distributed Systems
Data Communications
Data Communications Systems and Networks
Digital Communications and Networks
Elements of Networking
Experimental Networking
High-Speed Networks
Information Networks II
Internet Protocols
Internetworking Protocols and Software I
Internetworking Protocols and Software II
Introduction to Computer Networks
Introduction to Local and Wide Area Networks
Introduction to Networks and Protocols
Network Architecture and Performance Engineering
Network Design and Evaluation (capstone)
Network Essentials

Copyright © 2004, American Society for Engineering Education"

Network Synthesis
Networking - Theory and Fundamentals
Optical Networking
Packet Switch Architectures I, II
Projects in Computer Networks
Wireless Local Area Networks
Wireless Systems

Appendix C: List of Department Names

Computer and Information Sciences
Computer Engineering / Electrical Engineering
Computer Science
Computer Science / Computer Engineering
Computer Science and Electrical Engineering
Computer Science and Engineering
Computer Science and Software Engineering
Computer Science/Electrical Engineering/Computational Science
Computing
Electrical and Computer Engineering
Electrical Engineering
Electrical Engineering and Computer Science
Electrical, Computer, and Systems Engineering
Information Sciences and Technology
Information Systems
Telecommunications and Networking

Appendix D: List of Textbooks

An Engineering Approach to Computer Networking by S. Keshav, Addison-Wesley, 1997
Communications Networks by A. Leon-Garcia and I. Widjaja, McGraw Hill, 2000
Computer Networking: A Top-Down Approach Featuring the Internet by J.F. Kurose and K.W. Ross, Addison-Wesley, 2003
Computer Networks (4th edition) by A.S. Tanenbaum, Prentice Hall, 2003
Computer Networks and Internets by D. Comer, Prentice Hall, 2003
Computer Networks: A Systems Approach by L.L. Peterson and B.S. Davie, Morgan Kaufmann, 2003
Data and Computer Communications by W. Stallings, Prentice Hall, 2004
Distributed Systems: Concepts and Design, by G. Coulouris, J. Dollimore, and T. Kindberg, Addison-Wesley, 2001
Elements of Network Protocol Design by M.G. Gouda, John Wiley & Sons, 1998
Engineering Internet QoS by S. Jha and M. Hassan, Artech House, 2002
IXP 1200 Programming by E.J. Johnson and A Kunze, Intel Press, 2002
UNIX Network Programming, Volume I by W.R. Stevens, Prentice Hall, 1990

"Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2004, American Society for Engineering Education"

The Computer and Information Science master's degree program is designed to prepare students for professional practice, as well as further studies and research in the computing field. The program offers a 30-credit hour curriculum consisting of required core courses and technical electives. The department schedules all CIS courses during late afternoons or evenings to enable students to earn their master's degree through part-time study.Â All courses have access to a wide variety of computing resources: local area networks, as well as the Game and Multimedia Environment Laboratory.Â One course in data structures with algorithm analysis. One course in computer architecture. One course in operating systems. Computer Engineering or Computer Science. Computer science is offered by the Faculty of Arts & Science at the University of Toronto. It is primarily concerned with the programming side of computers as well as computational theory, numerical analysis, etc. Computer engineering is offered by the Faculty of Applied Science & Engineering at U of T, which means you will be an engineer when you graduate.Â Computer engineering students learn the hardware and software of computers.Â The computer engineering program at U of T allows students to focus their studies in the area of computers that most interests them. The ECE Department is within the Faculty of Applied Science & Engineering at the University of Toronto. What is an undergraduate pathway in computer science? This pathway may allow students to delve deep into the field of computer programming, coding, and technology. Participants may study the ways computers process information and how that can be used to better our world. Additionally, students typically learn a number of computing skills that can be used in other fields. Courses may include information security, computer engineering, database administration, information technology, and computer networking. Computer science education offers students a number of skills that can be used in many d The Department of Computer Science offers the undergraduate degree Bachelor of Science in Computer Science (BSCS). As per NU policy and Bologna guidelines, the undergraduate degree program requires a minimum of 240 total ECTS credits. Technical Electives for the BSCS degree can be satisfied by any non-required course at 200-level or above offered by the CS department, as well as the following courses offered by other departmentsÂ The two specifically required courses are: CSCI 151 Programming for Scientists and Engineers (8 ECTS credits). 1010 - discovery computer science (non-majors). A breadth-first introduction to computer science based upon 7 â€œBig Ideas,â€ namely: 1) computing is a creative activity, 2) abstraction reduces information and detail to facilititate focus on relevant concepts, 2) data and information facilitate the creation of knowledge, 4) algorithms are used to develop and express solutions to computational problems, 5) programming enables problem solving, human expression and creation of knowledgeÂ Design and analysis of parallel algorithms for engineering and scientific applications. Parallel programming using message passing and shared memory paradigms. Credit Hours: 3.