A Redesigned Undergraduate Data Communication and Networking Course Incorporating Theory and Practice

Ziping Liu
Computer Science Department
Southeast Missouri State University
Cape Girardeau, MO, U. S. A.

Abstract- *Data Communication and Computer Networking is a traditional undergraduate CS course. Classic teaching focuses on communication protocol and algorithm analysis, plus socket programming, and often deploys simulator to reinforce theoretical concepts. However, in our CS department with only undergraduate students, this approach appears to be hard to meet students' expectation to handle real world networking problems. With the goal to stimulate the enthusiasm of our students and tie closely theory and practice, the course is redesigned. In this paper, we present our experience in redesign the course, describe the course structure, assignments and outcomes.*

Key Words – data communication course redesign, active learning

1. INTRODUCTION

Data Communication and Computer Networking is a traditional introductory course on communication and networking systems for junior/senior undergraduate CS students. It is also a regular offered course in other majors such as MIS, technology and computer engineering. While this course is taught, different majors emphasize different aspects of the course, for example, MIS major focuses on management of networking systems, technology major focuses on networking administration while computer engineering major focuses on design of hardware components. As for CS major, emphasis is usually put on communication algorithms/protocols and programming.

Today, we have plenty of texts on this and similar courses. These textbooks are normally organized as bottom-up approach[1], [2], or top-down approach[3], or integrated approach [4] that combines the best of top-down and bottom-up to explain principles, design approaches and standards. Most of authors consider that experiments and projects are the most effective means for reinforcing the concepts taught in class, and they often provide accompany lab manuals/instructor manuals that describe experiments/projects to be performed. The relevant lab assignments normally cover a range of topics with network measurement, packet analysis and network programming. For example, [5] offers simulation projects and performance modeling using OPNET simulator, [2] and [3] packet analysis projects using Ethereal open-source packet analysis tool, [4] mentions that their students do projects of packet capture from live network using their lab facilities. There is no doubt that using simulators or packet analyzer at certain level helps students to understand abstract concepts, especially when only limited networking resources are available. However, this approach appears to be hard to stimulate and hold enthusiasm of learning from students, especially for a CS department with only undergraduate students.

I have taught this course for a few years and I often wonder what is the best way to teach it, especially when we get students with various career goals and mixed abilities/skills. When I first taught this course, I focused on theory and programming. Based on my own research experience in writing simulation software, I had thought of the development of a simulation program for a protocol as a useful phase to help students deeply understand a protocol as well as enhance programming skills. As for socket programming, I spent a couple of lectures on the structure/syntax of windows socket. I also provided students a detailed handout about windows socket. Then students were required to write a file transfer application for both client side and server side from scratch. Even though students spent a lot of time on programming a simple protocol, the outcome did not show the effectiveness. In the next round when I offered this course, I changed the project assignments to do packet analysis using Ethereal [3], as well as program a set of client-server applications using provided API. Initially, students did show some interests in Ethereal, because they can view the real internet packets. After a couple of such projects, they lost their further interests and they simply rushed to finish their assignment whenever they had their assignment due. Even though the assignment of set of client-server applications exposed various aspects of socket programming, students felt frustrated with the complication of the provided API.

Since fewer students in our department intend to go to graduate school and do research after graduation, most of the students report that they are more interested in handling real world networking problems, such as network setup and configuration, plus the hands-on experience can help them build up their confidence when they encounter the real world challenges.

In Fall 2005, I redesigned the course. I replaced packet capture and analysis component with hands-on network setup and configuration component, and replaced socket programming using provided API with a set of templates/frameworks of socket programming. There are four components in the redesigned course: communication theory and standards, network setup and configuration, socket programming and an open topic term project for recital. Students not only show but also maintain great enthusiasm throughout the whole semester. Among other things, since they encounter quite a few network terms and concepts in their hands-on labs, they are more eager to know the underlying working mechanism, such as TCP/IP, collision domain, routing table. Posttest outcome also shows improvements compared with previous semesters.

2. COURSE OBJECTIVES AND STRUCTURE

As indicated above, there are two aims for the redesign of the course: to introduce students to the field of networking and to provide plenty of opportunities for real world experience. Toward this end, the course is redesigned to have a lecture-based component and a hands-on experiential component. The hands-on component is implemented as a set of on-going, involving system installation and configuration, software programming labs, as well as a "recital" term project.

2.1 Objectives

After completing the course, students should be able to:

1. to obtain fundamental knowledge about underlying mechanisms in data communication and computer networks
2. to learn network programming skills such as socket interface in C and protocols such as FTP/HTTP through programming projects
3. to learn network setup and configuration through hands-on LAN set-up, literature surveys and reports

Student learning outcomes are measured through a midterm and final exam, hands-on labs(including both network setup/configuration and socket programming) and term project demonstration and presentation.

2.2 Course Structure

The course structure consists of four components: 1. lectures on theoretical material; 2.lectures on socket architecture, detailed socket programming syntax, a sample echo server application, and other in-class activities; 3. network setup and configuration labs; and 4. open topic term project work. We use the first two weeks for introduction. Starting from the third week, we went through the sequence of socket programming periodically. At the $6^{th}/7^{th}$ week, we devoted whole 3 weeks for hands-on network labs. And the final two weeks are reserved for our term project recital. The resulting course plan is shown in Table 1.

Week 1	Introduction to Communication and Networking System, OSI 7 Layer and TCP/IP Five Layer Protocol Architecture
Week 2	Digital Data Communication Techniques: Asynchronous and Synchronous Transmission, Error Detection and Correction
Week 3	Socket Programming Structure, Syntax and A Simple Echo Server Application Example, Assign first socket programming lab
Week 4 & 5	Data Link Control Protocols: Stop-and-Wait, Sliding Window, Go-Back-N and Selective Reject, Assign second socket programming lab
Week 6	FDM, TDM and CDMA; Circuit Switching and Packet Switching
Week 7, 8, 9	Hands-on Network Setup and Configuration
Week 10	Routing Algorithm, IP Routing, IP Addresses and Subnet, Congestion Control, Assign term project
Week 11, 12	LAN Design, Hub, Switch, Bridge, Router, Media Access Protocols, Ethernet, Token Ring, and Wireless LAN, Assign third socket programming lab
Week 13	ARP, Transport Layer Protocols: TCP and UDP
Week 14	Work on term project
Week 15	Presentation and Demonstration of Term Project

Table 1 Course Plan

3. EXPERIENTIAL COMPONENT

There are three components in our project arrangement: network setup and configuration hands-on labs, socket programming labs and open topic term project. The textbook we adopted is [5]. Accompanying the textbook, the publisher also provides a free OPNET Lab manual, which uses OPNET simulator to experience concepts and protocols. As indicated in the introductions,

the majority of the students in our department will not go to graduate school, however they do show interests in building their confidence to handle real world networking problems. Usually, network setup and configuration is considered as a topic in the certification exam for network technician/administrator and is excluded from CS curriculum arrangement. Based on our past experiences and the expectations of our students, we think that it will be a benefit for our students to gain some experience in this aspect and be able to handle real world problems. Just like "fly simulator" is a useful tool for pilot to learn flying plane, a pilot still needs to get on a real plane to learn driving.

3.1 Platform for hands-on labs

Obviously, the available resources will influence that is possible – not all institutions will have the same equipment and there are a lot of institutions that just have limited resources. Our current model is that we have a flexible lab with 25 standalone blank desktop PCs(and the lab is not a standard pre-configured computer lab), 6 laser printers, a number of 10/100 Base Tec 4-port switches, NIC cards, CAT 5 cables and power supplies. As for the OS, we choose Microsoft XP and 2003 server. We didn't choose Linux. One reason is that our department joins the Microsoft Academy Alliance program and hence we have discounted Windows XP and 2003 server software packages. Also, Microsoft provide 180-day free trial 2003 server and this can be a benefit for students to do deeper dig at home. The other reason is that these systems are used a lot in every day life and only a few students are Linux fans. Since the lab is not a dedicated networking lab, we don't have other network equipments, such as Cisco router. The lab is a flexible lab that can be adjusted to serve individual course for its purpose. It is thus the responsibility of the instructor and students of a specific course to setup the lab, however, this does provide an opportunity and flexibility to gain experience. To plan hands-on lab assignment for this course, I spent a few weeks to go over the network setup and configuration from scratch in summer break. This helps me to identify problems that may be encountered by students. Another decision to make is that we should allow students install OS from scratch or just let students do the network configuration with OS pre-installed. From my own experience, I found installing OS from scratch is quite helpful, since students get exposed to all aspects of a real world scenario.

3.2 Socket Programming Platform

We choose C language socket programming in windows and didn't choose socket programming in Linux. The reasons are: first, most of our students have Windows OS for their personal computers; secondly, our department joins the Microsoft academy alliance program and Visual studio 6.0 is free for our students and Visual studio .Net can be purchased at a discount price at around $16.9; third, it is a good opportunity to expose our students to basic skills regarding project settings in visual studio and the basic use of MSDN library, which are considered as fundamental skills in industry. There is always debate regarding whether we should use simple compiler tool or professional compiler tool in our CS undergraduate education. Even though it is effective to use simple tool in introductory programming language course, like CS1, it would be more beneficial for students to get a chance to expose professional compiler tool and development library. Though a few students usually will find resource to learn the skills, but most of our students still depend on their teachers to lead them into a door. Socket programming using C language is usually considered as a hard topic for undergraduate students. To facilitate the learning of students, I spent 3 lectures in department closed lab to explain the architecture and each individual system procedure, using a simple echo server and echo client example as the framework. Meanwhile, I also demonstrate them how to search information such as description of a procedure or sample codes in MSDN library online or offline in visual studio documentation. I also led them to go through building the client-server programs step-by-step, first on one machine with two opened windows serving as client and server respectively using "localhost" in their program, next across multiple machines with one as server and others as clients using "ipconfigure" to find the server name/IP address.

3.3 Network Setup and Configuration Hands-on Lab Assignments

The overall philosophy of the lab assignments is to provide a hands-on, network setup and configuration experience that complements the lecture material. This approach to teaching creates an active learning environment in which students can understand how networking protocols work in real world, analyze problems and solve problems, and explore more related topics in depth on the basis of their knowledge and experience. Studio-type teaching and learning model was used for our hands-on labs. Below, we list our hands-on lab assignments which we adopted from [6]. We also made some modifications on them to favor our own purposes.

Lab1 Experiencing A Simple Peer-to-Peer Network

Peer-to-Peer networks are simple and inexpensive networks and are commonly found in homes/small business offices. The goal of this lab is to experience a simple peer-to-peer network, be familiar with Windows System Properties screen, computer name, workgroup, My Network Places screen, Network connections screen, Local Area Connection Properties screen, Internet Protocol (TCP/IP) Properties screen, and manually configure IP address and subnet mask. During this lab, students are required to install windows XP OS, connect two computers with a provided TEC 4-port switch, attach the printer to one computer, and share each other's resources, such as files, folders, drives and printer.

Lab2 Building A Small Client/Server Network

Clent/Server networks are common in business/organizations. They are more scalable, manageable and secure than peer-to-peer networks. The goal of this lab is to establish a simple client-server network, be familiar with installing windows 2003 server as a domain controller, creating domain, connecting a Windows XP workstation to server and the basic concepts and operations regarding Active Directory, DHCP and DNS, creating a user account at the server and logging onto domain using the created account from the client. During this lab, students are required to build a simple client/server network, add a Windows XP client computer to a Windows Server 2003 domain, share files/folders on the server and share a networked printer on the network. Students are suggested to use the step-by-step guide to a common infrastructure for window server 2003 deployment provided from Microsoft TechNet[7].

Lab 3 Creating a Software Router and activating routing and routing protocol using a Windows 2003 Server

There are 3 parts in this lab. In the first part, students are required to physically install an additional NIC on the Windows 2003 server to make the computer using two NICs, and assign a second set of IP address(belongs to a different subnet from the first set of IP address) for the newly installed NIC. In the second part, the students are required to connect two workstations(works as two LANs) with two NICs on the server, then start Windows 2003 server Routing and Remote Access service to activate the function of routing on the configured software router, and manually add a static route to the router. Students are also directed to use network commands, such as "ping", "route print" for printing routing table, "route add" for adding a static route, and "tracert". In the third part, the students are directed to explore a routing protocol configuration using Routing and Remote Access service.

A couple of students also extended their lab assignments to configure other services, such as file server, web server.

3.4 Socket Programming Lab Assignments

The aim of the lab assignments here is to develop a set of client-server applications using windows socket. In previous semesters, students were assigned to work on a single client-server application like FTP from scratch. However, they tended to struggle with low-level system procedures, as well as are limited to exposure to various communication applications. In fall 2005, I changed that model. While this is no doubt a good experience, it does not provide as broad an experiential activity as does programming a sequence of various applications. Meanwhile, students also learn the working mechanisms for some of the communication applications.

Lab 1 Modifying Echo Sample to A Chat Application

Modify the echo client and the echo server so they both can chat with each other, in this case, it is a chat application. During the chat session, both of the client and server require the user to enter a "user name" (an arbitrary string). Then prepend the user name followed by a greater-than sign to each line of text before sending the text across the network. If both sides like to keep a record for their chatting, a log file would certainly serve this purpose.

Lab 2 Extending Sample FTP to Support PUT and LIST

A file transfer service that consists of a client and server, and the server export a set of files from the computer on which it runs (i.e., make them available for download), and have the client obtain a file on demand (this is what we call "GET" command). That is, arrange to have the client send a file name and for the server to respond by sending the file. Here both plain text file and binary file (such as an image .gif file) are considered. Modify the server and the client so that client can send a plain text file to the server, which is the *PUT command* in FTP. Modify the server and the client so a *.gif* file can be sent from the client to the server, which is to PUT a binary file to the server. Extend the protocol to permit the client to send a *LIST* command that requests a list of all the files plus subdirectory names of the server directory and display them on the client machine. To help students with directory handling, I include the detailed instruction to go to MSDN library[8] on line to find *directory* related run-time functions and sample codes.

Lab 3 Extending Sample Web Application

The sample is a simple hypertext transfer protocol service that consists of a client and server, where the server holds a hard-wired web page in its code and the client retrieves the web page. The web server should also be compatible with other common Internet browsers, such as IE. Also, the web client should be compatible with other Internet web servers, such as IIS. Here both plain text file and binary file (such as an image .gif file) are considered. Modify the web server so 1. it extracts the contents of each page from a file instead of having them hard-wired into the code, 2. it recognizes file names that end in *.gif* and send them using a *Content-type* header with a value *image/gif* instead of the string *text/html*, and send the *.gif* file to the client, 3. to allow it to browse a directory: when the *path* given in the GET request corresponds to a directory, generate a list of files in the directory, add HTML tags to format the list, and return the result to the browser.

3.5 Open Topic Term Project

The term project topics and contents can be open, but they must be in the field of data communication and computer networking. The project can be done by a team or an individual. Like recitals for music students, the aim of the open topic term project is to foster the continuous enthusiasm and creativity of our students for the knowledge they learned, as well to provide our students an opportunity

to do their show biz among peers. The final demonstrations and presentations are very encouraging, and the topics are diversified. Some did socket programming with their own applications, some did network setup and configuration other than the ones learned in class, some did new technique exploration. Below, we list some of the projects:
- Two player networked Tic-Tac-Toe game, in which two players on two connected machines can play Tic-Tac-Toe game with each other and is written using WinSock in C language
- A Python POP proxy for Southeast Student Email, which is a POP proxy service for student campus email using socket programming in Python language
- The Higgins Home Network: Adding a print server to the LAN
- A small wireless network setup and configuration
- Exploration of more advanced services in Windows 2003 server

Over all, students in their term project produced work of very high quality and originality.

4. OUTCOMES

Since the structure of the course has been redesigned to incorporate more practices, the students were surveyed at week 11 regarding the hands-on labs and socket programming. The students spoke strongly, especially for network setup and configuration hands-on labs. While 16.7% of the student felt not advanced enough or preferring Linux box, 83.3% of the students felt they are helpful or very helpful. Such results reinforce the positive impact that the significant network setup/configuration and diversified socket programming labs provide. A sample of student comments includes:

"*I enjoyed the labs very much. I had never set up a switch or used windows server before and I found it very interesting.*"

"*I like hands-on because I learn it better. I thought it would be very helpful for the future.*"

"*I think the labs were very helpful…but too little time spent to fully explore some of the complexities involved.*"

"*I thought the hands-on labs were very useful. I think this class should actually consist of more of them.*"

"*It taught me a lot about networks and windows server. It was fairly easy to comprehend*."

"*I thought the socket programming assignments were very useful."*

"*Socket programming was a little difficult, but it was interesting to learn and needed. And the assignments were interesting.*"

"*Socket programming assignments were fun topics.*"

"*I think socket programming assignments were helpful in learning communication, but was hard to understand."*
"*I enjoyed the labs all thought they were very helpful. Working with the code and use the servers were better than just talking about it.*"

At the end of the semester, students also conducted the course evaluation. In order to assess the course effectiveness, we analyzed the Progress on Relevant Objectives and Attitudes of the students as below:

1. Gaining factual knowledge (terminology, classifications, methods, trends)
2. Learning fundamental principles, generalizations, or theories
3. As a result of taking this course, I have more positive feelings toward this field of study

Statistic scales are given in the table 2

	Below average	average	Above average
Objective 1	14.2%	28.6%	57.2%
Objective 2	0	42.8%	57.2%
Objective 3	7.0%	28.6%	64.3%

Table 2 Statistic Scale of Objectives and Attitudes

The above survey and statistic analysis indicates that the general perspective was that the redesign of the course held more students' enthusiasm on the course and had fewer complaints than previous course offerings.

Student mid-term and final exam have average 83.1 and 77.1 out of 100 respectively, which shows great improvement compared with past semesters' exams. Even though network related math calculation did not show encouraging results in the final exam, the overall performance on main concepts and theories in both exams is very encouraging, especially for the topics listed in Table 3.

Go-Back-N and Selective Reject
Circuit Switching, virtual circuit packet switching and datagram packet switching
Routing Algorithm, IP Routing, IP Addresses and Subnet
Hub, Switch, Bridge, Router, Media Access Protocols, Ethernet
Transport Layer Protocols: TCP and UDP concepts

Table 3 Main Topics in Mid and Final Exams

5. CONCLUSION

We have described a redesigned undergraduate data communication and computer networking course that incorporates theory and practice. From our past experience, we have shown the necessity to integrate the component of the network setup and configuration and the component of diversified socket programming applications in this course. After our careful planning for the course, it is shown that students are able to learn the theoretical part required by curriculum from traditional lectures, as well as real world and market demanded skills from hands-on labs, which also shows efficacy in holding students' enthusiasm throughout the course. The survey conducted at mid-term and at the end of the semester, programming and network setup and configuration hands-on labs, term projects, plus post-test results altogether indicate desirable and encouraging outcomes.

All of the course materials including lecture slides, hands-on lab handout, socket programming samples and handout, and all of the labs and project assignment handouts are available on the web at:
http://cstl-csm.semo.edu/liu/cs480_fall05/

REFERENCES

[1].Andrew S. Tanenbaum, *Computer Networks, 4/E*. Prentice Hall, 2003
[2].Alberto Leon-Carcia and Indra Widjaja. *Communication Networks, Fundamental Concepts and Key Architectures*. McGraw Hill, 2004
[3].James F. Kurose, Keith W. Ross, *Computer Networking: A Top-Down Approach Featuring The Internet*. Addison Wesley, 2005
[4]. Douglas E. Comer. *Computer and Networks Internets with Internet Applications*. Prentice Hall, 2004
[5]. William Stallings, *Data & Computer Communications*. Prentice Hall, 2000
[6]. Tamara Dean. *Network+ Guide to Networks*. Course Technology, 2006
[7]. http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/directory/activedirectory/stepbystep/default.mspx
[8]. http://msdn.microsoft.com/