APPENDIX B

PROJECTS AND OTHER STUDENT EXERCISES FOR TEACHING DATA AND COMPUTER COMMUNICATIONS

Analysis and observation, theory and experience must never disdain or exclude each other; on the contrary, they support each other.

—*On War,* Carl Von Clausewitz

Many instructors believe that research or implementation projects are crucial to the clear understanding of the concepts of data and computer communications. Without projects, it may be difficult for students to grasp some of the basic concepts and interactions among components. Projects reinforce the concepts introduced in the book, give the student a greater appreciation of how protocols and transmission schemes work, and can motivate students and give them confidence that they have mastered the material.

In this text, I have tried to present the concepts as clearly as possible and have provided nearly 400 homework problems to reinforce those concepts. Many instructors will wish to supplement this material with projects. This appendix provides some guidance in that regard and describes support material available in the **Instructor's Resource Center (IRC)** for this book accessible from Prentice Hall for instructors. The support material covers nine types of projects and other student exercises:

- Animations
- Practical exercises
- Sockets programming projects
- Wireshark projects
- Simulation projects
- Performance modeling projects
- Research projects
- Reading/report assignments
- Writing assignments
- Discussion topics

B.1 ANIMATIONS AND ANIMATION PROJECTS

Animations provide a powerful tool for understanding the complex mechanisms of network protocols. A number of Web-based animations are used to illustrate protocol behavior. Each animation allows the users to step through the operation of the protocol by selecting the next step at each point in the protocol exchange. The animations will be made available to professors at the IRC for this book in such a way as to enable online access by students.

The animations can be used in two ways. In a **passive mode**, the student can click more or less randomly on the next step at each point in the animation and watch as the given concept or principle is illustrated. The **active mode** can be used for two types of assignments. First, the student can be given a specific set of steps to invoke and watch the animation, and then be asked to analyze and comment on the

results. Second, the student can be given a specific end point and required to devise a sequence of steps that achieve the desired result. The IRC includes a set of assignments for each of the animations, plus suggested solutions so that instructors can assess the student's work.

These animations were developed at the University of Stirling in Scotland by Iain Robin and Ken Turner, with contributions from Paul Johnson and Kenneth Whyte. Larry Tan of the University of Stirling developed the animation assignments.

B.2 PRACTICAL EXERCISES

The IRC includes Web pages that provide a set of practical exercises for an introduction to the use of IP over a LAN. The exercises naturally follow one another and build on the experience of the previous exercises. They do not, however, need to be attempted one after another. The four exercises may more easily be done on four separate occasions. The practical exercises are designed to help the student understand the operation of an Ethernet LAN and an IP network. The exercises involve using simple network commands available on most computers. About an hour is needed to perform all four exercises. The exercises cover the following topics: your own network connection, computers on your LAN, computers on remote networks, and the Internet.

B.3 SOCKETS PROJECTS

The concept of sockets and sockets programming was developed in the 1980s in the UNIX environment as the Berkeley Sockets Interface. In essence, a socket enables communications between a client and server process and may be either connection oriented or connectionless. A socket can be considered an endpoint in a communication. A client socket in one computer uses an address to call a server socket on another computer. Once the appropriate sockets are engaged, the two computers can exchange data.

Typically, computers with server sockets keep a TCP or UDP port open, ready for unscheduled incoming calls. The client typically determines the socket identification of the desired server by finding it in a Domain Name System (DNS) database. Once a connection is made, the server switches the dialog to a different port number to free up the main port number for additional incoming calls.

Internet applications, such as TELNET and remote login (rlogin), make use of sockets, with the details hidden from the user. However, sockets can be constructed from within a program (in a language such as C or Java), enabling the programmer to easily support networking functions and applications. The sockets programming mechanism includes sufficient semantics to permit unrelated processes on different hosts to communicate.

The Berkeley Sockets Interface is the de facto standard application programming interface (API) for developing networking applications, spanning a wide range of operating systems. The sockets API provides generic access to interprocess communications services. Thus, the sockets capability is ideally suited for students to

learn the principles of protocols and distributed applications by hands-on program development.

Appendix T provides an overview of sockets programming prepared especially for this book plus links to sites with more information on the subject. In addition, the IRC includes a set of programming projects.

B.4 WIRESHARK PROJECTS

Wireshark, formerly known as Ethereal, is used by network professionals around the world for troubleshooting, analysis, software and protocol development, and education. It has all of the standard features you would expect in a protocol analyzer and several features not seen in any other product. Its open source license allows talented experts in the networking community to add enhancements. It runs on all popular computing platforms, including UNIX, Linux, Windows, and Mac OS X.

Wireshark is ideal for allowing students to study the behavior of protocols not only because of its many features and multiplatform capability but also because students may subsequently use Wireshark in their professional life.

The IRC includes a Student User's Manual and a set of project assignments for Wireshark created specifically for use with the book. In addition, there is a very useful video tutorial that introduces the student to the use of Wireshark.

Michael Harris of Indiana University initially developed the Ethereal exercises and user's guide. Dave Bremer of Otago Polytechnic in New Zealand updated the material for the most recent Wireshark release; he also developed the online video tutorial.

B.5 SIMULATION AND MODELING PROJECTS

An excellent way to obtain a grasp of the operation of communication protocols and network configurations, and to study and appreciate some of the design trade-offs and performance implications, is by simulating key elements. A tool that is useful for this purpose is *cnet*.

Compared to actual hardware/software implementation, simulation provides two advantages for both research and educational use:

- With simulation, it is easy to modify various elements of a network configuration or various features of a protocol, to vary the performance characteristics of various components and then to analyze the effects of such modifications.
- Simulation provides for detailed performance statistics collection, which can be used to understand performance trade-offs.

The *cnet* network simulator [MCDO91] enables experimentation with various data link layer, network layer, routing and transport layer protocols, and with various network configurations. It has been specifically designed for undergraduate computer networking courses and used worldwide by thousands of students since 1991.

The *cnet* simulator was developed by Professor Chris McDonald at the University of Western Australia. Professor McDonald has developed a Student User's Manual and a set of project assignments specifically for use with *Data and Computer Communications* and available to professors on request.

The *cnet* simulator runs under a variety of UNIX and Linux platforms. The software can be downloaded from the *cnet* Web site. It is available at no cost for noncommercial use.

B.6 PERFORMANCE MODELING

An alternative to simulation for assessing the performance of a communications system or networking protocol is analytic modeling. As used here, analytic modeling refers to tools for doing queuing analysis, as well as tools for doing simple statistical tests on network traffic data and tools for generating time series for analysis.

A powerful and easy-to-use set of tools has been developed by Professor Kenneth Christensen at the University of South Florida. His *tools page* contains downloadable tools primarily related to performance evaluation of computer networks and to TCP/IP sockets programming. Each tool is written in ANSI C. The format for each tool is the same, with the program header describing tool purpose, general notes, sample input, sample output, build instructions, execution instructions, and author/contact information. The code is documented with extensive inline comments and header blocks for all functions. The goal for each tool is that it can serve as a teaching tool for the concept implemented by the tool (and as a model for good programming practices). Thus, the emphasis is on simplicity and clarity. It is assumed that the student will have access to a C compiler and have at least moderate experience in C programming.

Professor Christensen has developed a Student User's Manual and a set of project assignments specifically for use with *Data and Computer Communications* and available to professors on request. The software can be downloaded from the *tools* Web site. It is available at no cost for noncommercial use.

In addition, OPNET, a professional modeling tool for networking configurations, can be used. An academic version is available and a student lab manual prepared for this book is available from Prentice Hall.

B.7 RESEARCH PROJECTS

An effective way of reinforcing basic concepts from the course and for teaching students research skills is to assign a research project. Such a project could involve a literature search as well as a Web search of vendor products, research lab activities, and standardization efforts. Projects could be assigned to teams or, for smaller projects, to individuals. In any case, it is best to require some sort of project proposal early in the term, giving the instructor time to evaluate the proposal for appropriate

topic and appropriate level of effort. Student handouts for research projects should include the following:

- A format for the proposal
- A format for the final report
- A schedule with intermediate and final deadlines
- A list of possible project topics

The students can select one of the listed topics or devise their own comparable project. The IRC includes a suggested format for the proposal and final report plus a list of possible research topics.

B.8 READING/REPORT ASSIGNMENTS

Another excellent way to reinforce concepts from the course and to give students research experience is to assign papers from the literature to be read and analyzed. The IRC includes a suggested list of papers, one or two per chapter, to be assigned. The IRC provides a PDF copy of each of the papers. The IRC also includes a suggested assignment wording.

B.9 WRITING ASSIGNMENTS

Writing assignments can have a powerful multiplier effect in the learning process in a technical discipline such as cryptography and network security. Adherents of the Writing Across the Curriculum (WAC) movement (http://wac.colostate.edu/) report substantial benefits of writing assignments in facilitating learning. Writing assignments lead to more detailed and complete thinking about a particular topic. In addition, writing assignments help to overcome the tendency of students to pursue a subject with a minimum of personal engagement, just learning facts and problem-solving techniques without obtaining a deep understanding of the subject matter.

The IRC contains a number of suggested writing assignments, organized by chapter. Instructors may ultimately find that this is an important part of their approach to teaching the material. I would greatly appreciate any feedback on this area and any suggestions for additional writing assignments.

B.10 DISCUSSION TOPICS

One way to provide a collaborative experience is discussion topics, a number of which are included in the IRC. Each topic relates to material in the book. The instructor can set it up so that students can discuss a topic either in a class setting, an online chat room, or a message board. Again, I would greatly appreciate any feedback on this area and any suggestions for additional discussion topics.