Lab Exercises for Computer Networking Courses

Michael Erlinger
Computer Science Department, Harvey Mudd College
Claremont, CA 91711, USA
mike@cs.hmc.edu

Categories and Subject Descriptors
K.3.2 [**Computer and Information Science Education**]: Miscellaneous

General Terms
Design

Keywords
Networking, Laboratory Exercises

1. INTRODUCTION

As computing grows and matures, we are continually adding layers of abstraction and encapsulation to make our day-to-day usage and programming tasks easier. Abstracting away the growing complexity of a modern computer is a necessary part of computing today. However, it is occasionally both useful and important to be able to pull back those interfaces and see the actual workings of the systems we build on. It is important for students to understand the workings of some of our more complicated systems, e.g., operating systems and computer networks. There is a history of computer science labs and programming environments that allow just that. These systems remove any unnecessary complexity and leave exposed the features most important for students to gain the all-important hands-on understanding that is otherwise lacking. Our system, TinkerNet, provides direct, real-world access to Ethernet packets, and gives students the features necessary to implement an OSI network stack from the data link layer all the way up through the application level.

While it was educational and fun for us to build, TinkerNet or any specific laboratory environment is not what is most important to educators. Rather educators are interested in the laboratory exercises. These exercises give students a hands-on understanding of how the protocols that have been hidden away by the now-universal Berkeley Socket API behave. Through such lab exercises students gain a better grasp of the theoretical workings of an internet. While the principles of networking can be presented in lectures, real understanding occurs when students actively develop and evaluate systems based on those principles – there is no good substitute for hands-on experience with real networks.

ITiCSE'06, June 26–28, 2006, Bologna, Italy.

2. LABORATORY EXPERIMENTS

We have created a semester-long set of laboratory experiments focused on student development of a fully functional network protocol stack. In this set of experiments each new experiment builds on previous experiments. We begin with an experiment to review some issues around programming in C, and then work our way up from raw Ethernet packets to fully functional IP, UDP, a student created protocol, and Blast (a microprotocol which fragments and reassembles large messages).

3. CURRENT SET OF LABORATORY EXPERIMENTS

- Lab 1: The goals of this assignment are to implement functions that send and receive Ethernet packets.
- Lab 2: The goals of this assignment to implement functions that send and receive ARP packets.
- Lab 3: The goals of this assignment are to implement an end-host version of the Internet Protocol. The implementation must be able to recognize IP packets addressed to a specific IP address and ignore those addressed to other IP addresses.
- Lab 4: The goals of this assignment are to implement the sending and receiving of UDP datagrams.
- Lab 5: The goals of this assignment are to design, to create, and to implement a peer-to-peer protocol that will be used to locate other hosts on the network running the same protocol.
- Lab 6: The goals of this assignment are: to implement the microprotocol Blast.

4. FUTURE WORK

We believe that there are many additional experiments which could be created, or have already been created by others. Our goal is to develop a repository of such exercises, making the repository available to the community. No matter what laboratory environment (simulation, close network, etc.) they were created for, we believe that a standard format could be developed for all such exercises.