Teaching Computer Networks Through Modeling

by
Capt. Noël Davis, Capt. Scot Ransbottom and Lt.Col. Drew Hamilton
Electrical Engineering and Computer Science
United States Military Academy
West Point, New York 10996

INTRODUCTION

Modern computer scientists need to understand both the theoretical and practical aspects of computer networking. As computer science teaching methods continue to mature, experiences in mixing theory and application have been shared in the community [Marti 96]. While the literature is rich with experiences in applied or project based courses in artificial intelligence, architecture, software engineering, data structures and algorithms, few have addressed the challenge of coordinating theory and practice in computer networking courses.

Combining theory and practice in a single course on computer networks is difficult because of the complexity and scale of modern networks. Classroom modeling of networks is a technique to illustrate the theoretical aspects of networking through practical models of computer networks. At West Point we are evolving our network course to use network modeling and simple simulation as a visualization tool to show how networks operate and the performance tradeoffs associated with network design decisions. This approach provides the opportunity to do more than teach an undergraduate survey course in computer networking.

RATIONALE FOR ACADEMIC NETWORK MODELING

Most networks are built to accommodate the needs of a single organization or group. Internetworking is a technology that accommodates multiple, diverse, underlying hardware by providing the means of interconnecting heterogeneous networks. [Comer 95]. Comparatively few individuals, academic institutions or corporations have networks *exclusively dedicated* for student use and experimentation. Prudent administrators limit student or general user access to operational networks. Modeling efforts are not limited by these operational considerations.

Network courses are often based one or more of the following areas:
- The OSI Model
- Performance Analysis
- Network Simulation

As we summarize each area, we must point out that most networks courses will cover most of these areas and but may widely vary the degree of coverage.

OSI Model - based instruction is very common. Typically, the functionality of each layer of the OSI Model is explained and then examples of this functionality in a specific network protocol is presented. However, there are few, if any, implemented network protocols in which the architecture layers are strictly aligned with the OSI model layers. Another way to approach the model technique is to teach the basics of a protocol then describe how it reacts at each level of the OSI model, if it engages a particular level at all.

Performance Analysis - an emphasis on analytical models of networking and the associated probability is also challenging and worthwhile. Not all undergraduates will have the necessary math/statistical background for serious work in analytical modeling of computer networks. While the academic value of analytical modeling is undoubted, it should be noted that an increasing number of researchers are questioning how well classical queuing theory approximates actual network performance [Paxson 95].

Simulation of computer networks provides students with simulated observations of the operation of a network. [Barnett 93] proposes the use of the NetSim simulator to support both major project assignments and more focused homework assignments. We have also found NetSim valuable in allowing students to observe and experiment the effects of bottlenecks, delays and other network phenomena. NetSim is available via anonymous FTP from lcs.mil.edu and runs over X-Windows. Whatever the simulation tool, a prudent technique is to incorporate the tool into supervised lab/project assignments, individual homework assignments, and classroom demonstrations.

USE OF OPNET

Those institutions that do have computer network test beds often find the maintenance and operation of such facilities to be manpower intensive. For example, once a test network is set up and operational, how is it stressed? How are comparisons between design decisions made? How many labs can purchase the newest ATM switches with the latest features?

Network modeling does not suffer from these shortcomings. The U.S. Army has adopted OPNET as a standard under the auspices of the Army Enterprise Strategy under the leadership of the U.S. Army Office of the Director of Information Systems for Command, Control, Communications and Computers. OPNET is widely used in universities as well as many parts of the DOD. Our preliminary work with OPNET indicates that it will support our visualization through modeling objectives.

OPNET may be described as a communications-oriented simulation language. The name OPNET is derived from Optimized Network Engineering Tools. The single most significant aspect of OPNET is that it provides direct access to the source code coupled with an easy-to-use front end.

A generic approach to network modeling can be constructed using the OSI Reference Model as its basis. This approach allows the implementation of different network protocols which are compatible at the OSI layer boundaries. Pedagogically, this approach has limitations. As illustrated in Figure 1 below, any detailed implementation of an Ethernet model will not directly align with the OSI Reference Model. Other protocols such as Fiber Distributed Data Interface also do not perfectly align with the OSI Reference Model.

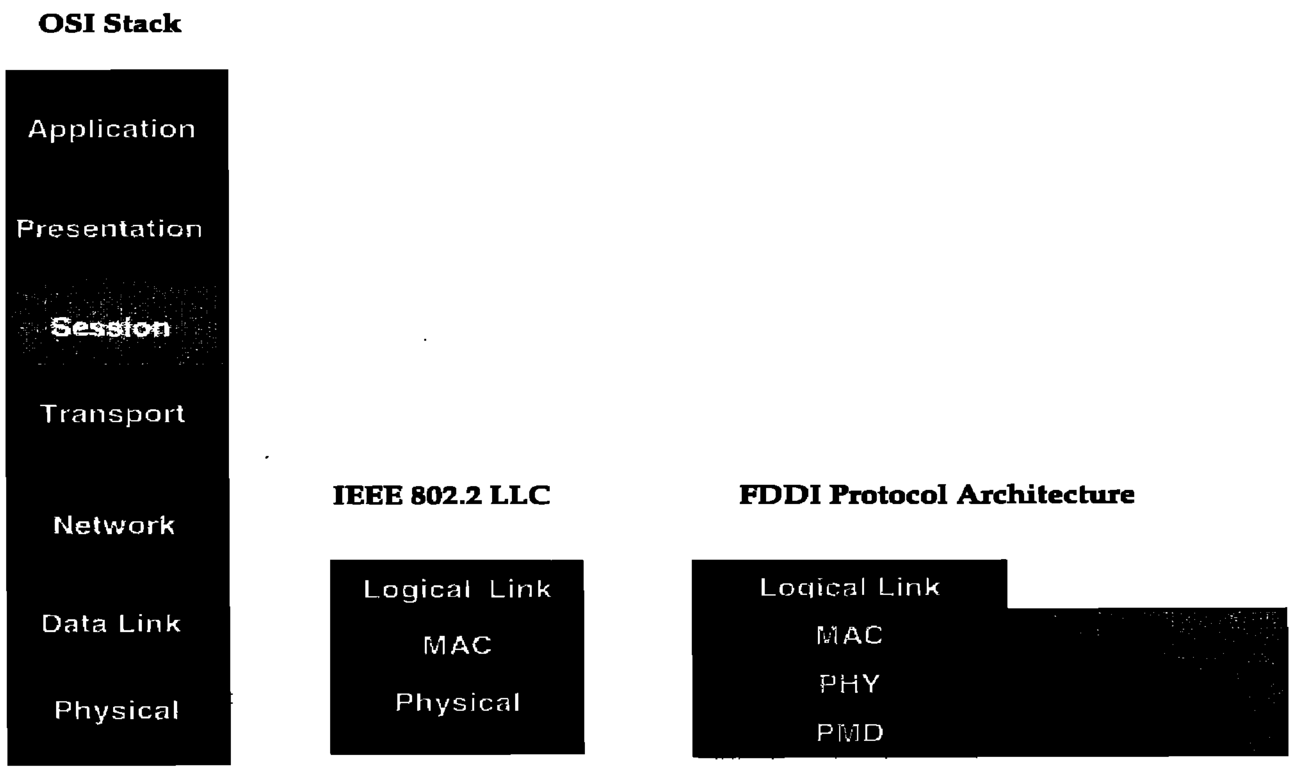

Figure 1. Reference Models and Architectures

OPNET models are composed of three primary model layers: the process layer, the node layer and the network layer. The lowest modeling layer is the process layer.

Network Models	networks and subnetworks
Node Models	individual nodes and stations
Process Models	STD that defines a node

Figure 2 OPNET model hierarchy.

This modeling hierarchy is illustrated in Figure 2. The process model in Figure 3 shows a state transition diagram (STD) for the generation of packets. Process models are built using finite state machines (FSMs) described by STDs. Finite state machines are an effective means of defining discrete-event systems that maintain state information. FSM-based design provides a means to manage complexity. Complex networks can be broken down into individual states and then each state is defined and implemented.

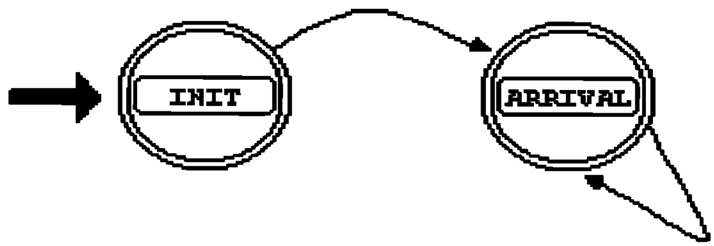

Figure 3. State transition diagram in a process model.

The source code for each state is readily accessible and modifiable by the student. Each state has entry execs and exit execs. Execs is the term used to describe the code executed when a state is entered and when a state is exited. The code defining the state transition between states is also accessible.

The next level of abstraction up from the process model is the node model. Each element in the node model is either a predefined OPNET artifact or defined by its own STD. Double-clicking on a node model element brings up its underlying process model. Figure 3 is an example of a node model that defines a station on a FDDI network. Packets are generated from the source *llc_src*, processed in the *mac* module and are put on the ring by the *phy_tx* module. Traffic from the ring is received via the *phy_rx* module processed in the *mac* module and finally received and discarded by the *llc_sink* module.

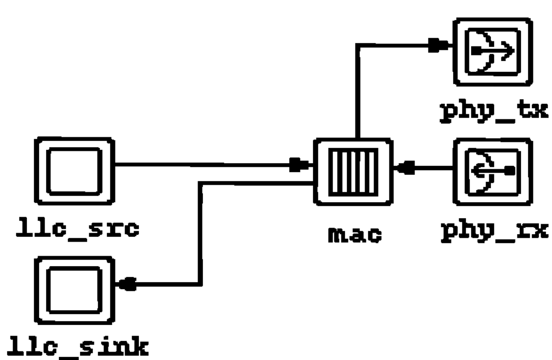

Figure 4. Node model (FDDI node).

The heart of a node model is either a processor module or a queue module. Processor modules are used to perform general processing of data packets as specified in the applicable protocol. Queue modules are supersets of processor modules with additional data collection capabilities built-in. The *mac* module in Figure 3 is an instantiation of a queue module.

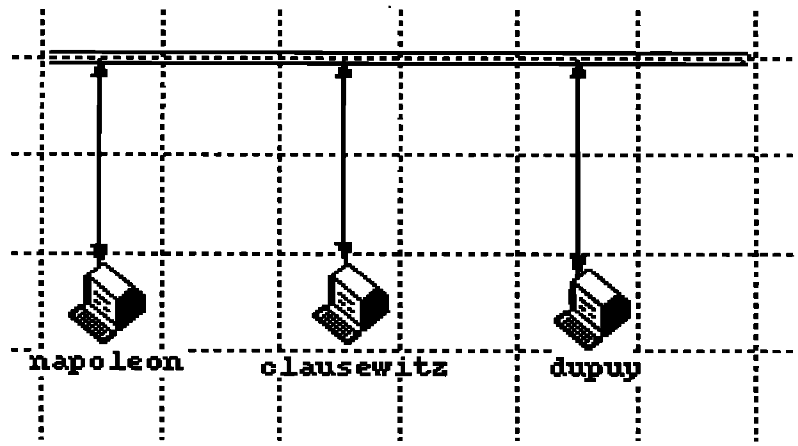

Figure 5. Three node network model.

The network model is the highest modeling layer in the OPNET model hierarchy. The network model may represent a hierarchy of subnetworks. A network model is shown in Figure 5. Each of the stations (nodes) shown in Figure 5 is defined by a node model such as the one shown in Figure 4. Again, each module in a node model is defined by state transition diagram as shown in Figure 3 thus conforming to the modeling hierarchy shown in Figure 2.

Figure 6. Subnetwork of aggregated segments.

The network model may be used to model a single network, subnet or segment or a hierarchy of networks, subnetworks or segments. The segment in Figure 5 may be joined with other segments and aggregated into a single subnet icon as shown in Figure 6.

The operation of a single network segment may now be studied. The implemented functionality of the physical and link layers of the OSI Reference Model are sufficient to model the operation of a single segment. At this point the individual stations on the segment may be customized if a more detailed representation is desired. Individual workstations or types of workstations may be specially modeled. Special characteristics could be implemented by modifying the individual modules of the station of interest or the physical network line connecting the stations. Many modifications can be made via the built-in menus. However, modifications may be made at the source code level should the menu choices not be fully satisfactory.

Adding network services in a TCP/IP network requires the implementation of the Internet Protocol (IP). To simulate the operation of more than one segment, the functionality of the network layer services must be added to the model. The node model of Figure 4 may be extended by adding modules to implement the Internet Protocol (IP) and Address Resolution Protocol (ARP). ARP tables are implemented statically with entries matching IP addresses to MAC addresses.

CONDUCT OF THE COURSE

Three pedagogical uses for network modeling are: animations used in class to illustrate operating principles, supervised laboratory projects and individual homework assignments. Animation is a powerful teaching tool and textbook publishers are already encouraging authors to provide ready-to-run animations with their text. While OPNET has powerful built-in, animation capabilities, there are many other types of desktop animation packages available.

It is reasonable to provide complete network models to students and require the students to make modifications and measure and report back the results.

OPNET Modeler allows source code access to the network models. Students can create and modify network components and see the results of their design decisions. Projects offer an extensive design experience by allowing the construction of network models and evaluation of network principles applied to those designs. OPNET Modeler supports the design, implementation and evaluation of a network system/subsystem given requirements and basic components

Through in-class animations, homework exercises and laboratory projects, students learn to visualize and therefore understand the effects of parametric changes to a network design without the risks and costs associated with experimentation on an operational network. This active learning model will benefit practitioners, students and network enthusiasts.

CONCLUSIONS

Network modeling is not limited by the availability of hardware. Modeling a network in software allows the student to visualize the impact of design decisions. Once a realistic network model is built then simulation-based experiments may be conducted. Increasingly, network simulation studies are being conducted in industry and government to determine the feasibility of network designs before they are implemented. Effective use of these studies requires a good theoretical understanding of how network protocols are layered and detailed knowledge of how specific systems are implemented. Better *understanding* of the network through modeling and experimentation provides the means to make better design decisions. Emphasis on network design decisions makes network modeling suitable for use in educational programs that stress engineering design principles.

REFERENCES

[Comer 95] Comer, D. E., *Internetworking with TCP/IP, 3d ed.*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[Barnett 93] Barnett, III, B. L., "An Ethernet Performance Simulator for Undergraduate Networking," *Twenty-Fourth SIGCSE Technical Symposium on Computer Science Education*, vol. 25, no. 1, Miller, E., ed., March 1993, pp 145-150.

[Marti 96] Marti, W.F., Hamilton, J.A., Jr., and Pooch, U.W., "PACKET tracing: a new paradigm for teaching computer network courses," *Integrating Technology into Computer Science Education, SIGCSE Bulletin Special Issue*, vol. 28, pp 162-164.

[Paxson 95] Paxson, V., Floyd, S., "Wide Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3. June 1995, pp 226 - 244.

AUTHORS

Captain Noël J. Davis, USAF, (dn5977@exmail.usma.edu) is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the United States Military Academy, West Point, NY. Captain Davis is the course director for CS 484, Computer Networks. http://www.eecs.usma.edu/usma/academic/eecs/instruct/davis.

Captain Jeffrey R. (Scot) Ransbottom), US Army, (dj3527@exmail.usma.edu) is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the United States Military Academy, West Point, NY. Captain Ransbottom teaches CS 360, Foundations of Computer Science and is the department Webmaster. http://www.eecs.usma.edu/usma/academic/eecs/instruct/ransbottom.

Lieutenant Colonel John A. (Drew) Hamilton, Jr., Ph.D., US Army, (dj7560@exmail.usma.edu) is Research Director and Assistant Professor in the Department of Electrical Engineering and Computer Science at the United States Military Academy, West Point, NY. Colonel Hamilton previously served as Chief of the Ada Joint Program Office. http://www.eecs.usma.edu/usma/academic/eecs/instruct/hamilton.