# Using OPNET to Enhance Student Learning in a Data Communications Course

*Michael W Dixon and Terry W Koziniec*
*Murdoch University, Perth, Australia*

m.dixon@murdoch.edu.au t.koziniec@murdoch.edu.au

## Abstract

It is difficult to teach data communications because it involves complex, dynamic processes which are not visible to students and are hard for them to conceptualise. This paper describes a project to improve the learning of students enrolled in the data communications courses. In this project the traditional mode of teaching data communications was supplemented by a problem solving approach using OPNET® a software environment for modelling, simulating, and analyzing the performance of communications networks. This mode of teaching enabled students to gain experience solving real world data communications problems without requiring the University to invest heavily in communications hardware and software which would quickly become obsolete.

**Keywords:** Data communications, discrete network simulation

## Introduction

It is difficult to teach data communications because it involves complex, dynamic processes that are not visible to students and are hard for them to conceptualise. Yet students are required to understand and utilise complex communication protocols that depend upon this underlying knowledge. The traditional approach to teaching data communications consists of lectures (supplemented by readings), where the discipline’s theoretical, mathematical basis is presented, and laboratory sessions where students are taught how to program a communications protocol in the hope that this will enable them to better understand the communication process. This is not ideal because a verbal or text-based description does not convey the process of data communications adequately. Having students do programming assignments is very time consuming and hence many students are unable to get beyond the mechanics of programming to a full understanding of the communication process.

The ideal approach to teaching data communications would be to allow students to experiment with the behaviour of a number of real networks. This is because research in student learning indicates that skills are more easily developed in context (Ramsden, 1988). However, the cost of providing dedicated networks makes this approach financially unworkable. Similarly, allowing students to experiment on non-dedicated networks would cause the network to be too unstable for use by others.

This paper describes a project to improve the learning of students enrolled in the data communications courses at Murdoch University. In this project the traditional mode of teaching data communications was supplemented by a problem solving approach using simulation. By making use of simulation software it was possible to give students experience with a wider range of network systems than would be possible with communications hardware, and the user community is not adversely affected (Barnett, 1993).

This mode of teaching enabled students to gain experience solving real world data communications problems without requiring the University to invest heavily in communications hardware and software which would quickly become obsolete. In addition, industry is beginning to use network simulation software to design networks and analyse network performance. Students who have been taught using this approach should be able to immediately apply their knowledge of network simulations when first employed.

The use of graphical communication network simulation software was incorporated into the teaching of the data communications courses at Murdoch University in two ways. Firstly, it was used to illustrate data communication concepts dynamically in lectures and also to introduce problems in lectures. Secondly, and more importantly, it was used by the students in laboratories. This paper focuses on the use of simulation in laboratories and assignments.

The use of simulation in laboratories will not only enable the dynamic illustration of data communication concepts to facilitate the understanding of them, but enable students to gain experience modelling, simulating and analysing the performance of the communication network protocols. Also, the students will be able to explore the complex issues of internetworking of computer networks and the design issues of communication networks. Students gain experience designing communication networks and testing their designs for robustness, efficiency and expandability. This would be impossible to do without the use of simulation software as a teaching tool, and was particularly effective because the simulation software used is user-friendly enabling students to focus on the concepts rather than the mechanics of using the software.

As a result of this change in teaching approach students gained an improved understanding of communication concepts. Students were able to visualise and actively analyse the behaviour and performance of communication networks.

## What is OPNET and Why use It?

OPNET is a discrete network simulator which contains a comprehensive development environment supporting the modeling and performance evaluation of communication networks and distributed systems. OPNET was chosen over other major network simulation software packages after evaluating the leading vendors for ability to stay in touch with new communication network technologies, ease of use of software, quality of documentation, simulation platforms and whether the software meets all the needs for use in a data communications course. Another secondary criterion was the suitability of the software for use in research.

OPNET uses models that are specified in terms of objects, each with configurable sets of attributes. This allows for flexible definition of new objects with programmable characteristics and behavior in order to address the wide scope of systems that are presented in the course. Models are hierarchical to naturally parallel the structure of actual communication networks. The hierarchical approach allows for deep nesting of subnetworks and nodes and large networks can be efficiently modeled. Stochastic and/or deterministic models can be used to generate network traffic. Performance evaluation and trade-off analysis require large volumes of simulation results to be interpreted and OPNET includes a tool for graphical representation and processing of simulation output. Simulation runs can be configured to automatically generate animations of the modelled system at various levels. These are used to help the student visualize how the network operates while working. Simulation results can be plotted as time series graphics, scatter plots, histograms, and probability functions.

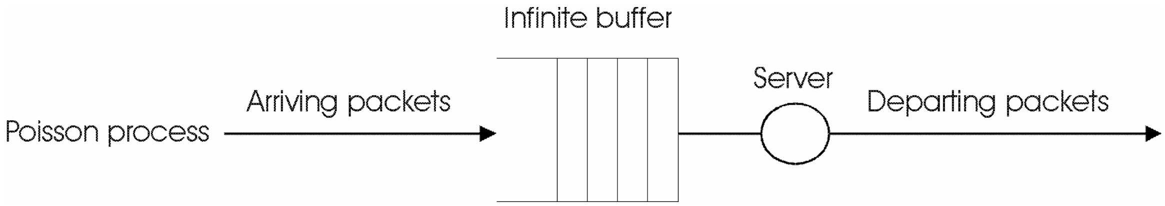

**Figure 1: M/M/1 Queue System**

## How OPNET will be integrated into the course

There are some basic communication building blocks that the students must understand before they are able to design and simulate a communication network (Baxter et al, 1994). A basic understanding of queuing theory is required of students because it plays an important role in the design of computer communication networks and systems. Many of the students in our data communications course do not have a strong mathematics background and this prohibits us from going into a detailed mathematical explanation of queuing theory. Thus, we use OPNET to help the students learn some of the basic fundamentals of the M/M/1 queuing model. For the most part, simple models can be used to obtain useful results. Also, students need to understand how finite state transition diagrams are used to represent process states and the logical flow between them. OPNET uses a graphical process editor that allows the students to design models of protocols, resources, applications, algorithms, queuing policies, and other processes. We use the animation capabilities of OPNET for visualising simulation events to help students understand how the communication process operates. Animations can depict messages flowing between objects, control flow in a process, and dynamics values such as queue size or resource status.

Apart from the use in lectures, OPNET was used in our course in two ways: firstly in laboratory sessions to help the student learn the basics of OPNET and then in two assignments. There is one individual assignment and a group assignment that involves pairs of students. A brief description of the laboratory exercises and the two assignments is provided below in the sequence students will encounter them.

### *M/M/1 Laboratory Exercise*

The students are introduced to the M/M/1 queuing model to gain a basic understanding of how queuing models play an important role in communication networks. An M/M/1 queue consists of a first-in-first-out (FIFO) buffer with packets arriving randomly in accordance with a Poisson process, and a processor, called a server, which retrieves packets from the buffer at a specified service rate (see figure 1). The three parameters that are examined are: packet arrival rate, packet sizes, and service capacity. If the combined effect of the average packet arrival rate and the average packet size exceeds the service capacity, the queue size will grow indefinitely (Daigle, 1992). The task of the students is to construct an M/M/1 queue model to allow an analyst to observe the performance of the queuing system with varying packet arrival rates, packet sizes, and service capacities. The students run simulations that will measure average delay (waiting time) experienced by packets in the queue, number of packets in the queue at any one time and average number of packets in the queue over time, and then analyze the results.

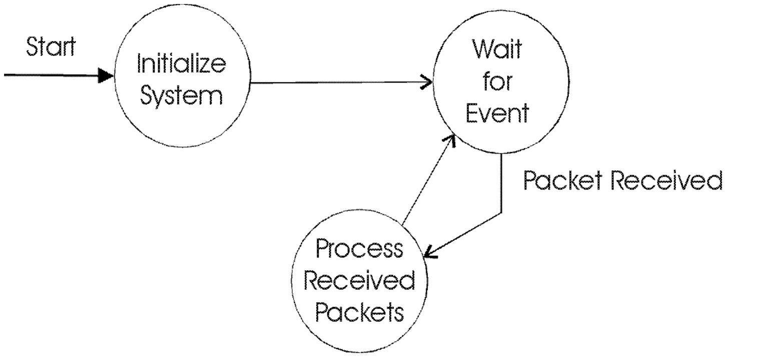

**Figure 2: Example Finite State Machine**

### Process Model Laboratory Exercise

This laboratory illustrates the use of finite state machine diagrams and Simulation Kernel library procedures. The exercise consists of the construction, execution, and analysis of a simple process which generates and counts packets. The students gain experience in defining process variables, macros, and transitions; creating process and node models; compiling and running a simulation executable; and analysing the simulation results. The objective of the exercise is to have the student design a process that will count and present the number of packets received over time from any number of packet sources.

The components of a process model include a finite state machine diagram with embedded C statements, and various blocks containing code for variable declarations, macros, constant, and function definitions. The finite state machine diagram represents the functional flow of the process with an easily interpreted diagram of states and transitions. Events and conditions determine the migration from state to state through transitions (Tanenbaum, 1999). Figure 2 shows a generic representation of a simple finite state machine. Each circle represents a state; the arrows between states represent transitions.

Many data communications problems can be thought of in terms of events, actions, and states. The students are required to express the packet counting problem as a finite state machine diagram. For the above diagram the expected events would be something like: (1) the simulation begins; (2) a packet arrives. The actions that could take place given these two events would be to initialise the packet count to zero, and as each packet arrives, update the packet count. The final element is to design a logical structure which will best accommodate the incoming events and provide the appropriate responses. Thus, the logical structure of a finite state machine is defined by its states and transitions. The above model employs three states: initialise, the starting state where the packet count is initialised; idle, where the finite state machine waits for a packet arrival; and arrival, where it responds to packet arrivals and updates the packet count. Once the initialisation state is executed, the finite state machine will transition between the idle and arrival states.

The finite state machine was used through out the course to describe various communication processes. The students are required to understand how it is used to describe a communication process and be able to apply this technique to new processes.

### Packet Switching Laboratory Exercise

The students simulate the behavior of a packet switched network consisting of nodes connected by point-to-point links in a star topology. Packets will travel from source to destination via a packet switching or hub node which has point-to-point connections with each satellite node. The initial configuration will contain four satellite nodes and one hub node. The hub node will only relay packets from the source to the destination. The satellite will generate traffic randomly according to an exponential distribution for inter-arrival times, and each packet will contain a random destination address.

Performance of the network is measured by the end-to-end delay experienced by packets. End-to-end delay is calculated as the number of seconds from the time a packet is created to the time it is received by the destination node. Packet sizes and link data rates are constant, so the end-to-end delay will be constant except when packets experience queuing delays in transmitters. Queuing delays occur when packets are sent to a transmitter which is already busy transmitting a packet. Queuing delays occur in both satellite nodes and in the hub node since both node types have transmitters.

After the students have analyzed the configuration, the model is expanded to add an additional star network. The two networks are connected by a point-to-point link between the hub nodes in each network. The hub nodes will be able to determine if a packet's destination address is within the local star network, or if the packet must be sent to the other hub node. The end-to-end delay for packets traversing the internet path will be greater since a third link is involved. The third link delay is also compounded by the fact that greater congestion at the internet transmitters results in more queuing delays.

## *Comparison of Aloha, CSMA and Ethernet Laboratory Exercise*

The students learn how to design a simple channel interface to a multi-tap bus, execute parametric simulations, and analyse the simulated results against theoretical predictions. The task is to design models which incorporate the Aloha and channel access method and the 1-persistent carrier sense multiple access (CSMA) method on a multi-tap bus link, where multiple nodes are connected through a shared channel. Each method's throughput performance is compared against the performance of the others, under various channel traffic loads.

## *Network Simulation Assignment*

After the above laboratories the students should have a solid enough understanding of the processes that are needed to set up a simulation of a network. In this assignment students choose a topic related to the physical characteristics of a CSMA/CD network, such as the maximum length of the cable, the maximum allowed packet (frame) size, or the data rate of the channel, and investigate how the performance is affected when one of these quantities is changed in various ways. They can also investigate how the arrangement of stations on the network affects the performance of the network and of the individual stations. Some possible topics for the assignment include:

- The sensitivity of throughput and delay to cable length.
- The sensitivity of throughput and delay to bit time (data rate).
- The sensitivity of throughput and delay to the mixture of packet lengths present on the network.
- The sensitivity of delay to the maximum allowed packet size.
- The effect on the performance of the network of introducing file servers.
- The effect of the introduction of point-to-point links on the performance of the network.

The students select one of these topics or come up with their own. After the student chose a topic, they design experiments to run in order to investigate the topic. Typically, the quality they are interested in investigating the behaviour of, will be affected by one or more aspects of the operation of the protocol. For example, they have already seen that the throughput of Ethernet is affected by the propagation delay of the network. Designing an experiment to investigate this phenomenon requires determining values of the propagation delay for which the throughput could provide some knowledge of the relationship between the two. Their designs will include the following information:

- A description of the basic network they plan to use (e.g. how many stations will there be, how long will the cable be, what packet lengths will be transmitted, etc.).

- A description of the dependent variable or variables they will investigate.
- A description of the independent variable or variables that will be investigated. For each, they will discuss the levels (or values) of the variable that they intend to simulate and why the particular set of levels have been chosen.
- A discussion of how they expect the experiments to clarify their understanding of the topic.

Results of the experiments are written up as a laboratory report and submitted.

### *Internetworking Laboratory Exercise*

The goal of this laboratory session is to provide the student with easy to use building blocks for the construction of internetwork models using TCP/IP. These building blocks take the form of node models for workstations and various gateways which can be interconnected with FDDI and Ethernet local area networks. The students are required to set up a simulation of an internetwork between two sites corresponding to a university department operating over two campuses.

### *Internetworking Assignment*

The aim of this assignment is to extend the students' understanding of how large communications networks are designed and implemented. Students need to learn good design techniques but these tend to be very difficult to teach unless students are able to see the impact of a poorly designed network on performance and be able to experiment with the design of the network to optimise performance. We use OPNET to help the students design and verify their network design for performance issues.

The students are required to design a detailed proposal for a fictitious company for which they are given floor plans, staffing details, computing requirements of the functional work areas, and minimum performance levels. The company is only willing to spend a certain amount of money and students are not able to spend beyond it. The students will be required to optimize the design of their network and this must be verified by OPNET simulations. They must then verify that what they have recommended in their design will meet all the requirements of the company.

The student's proposals are broken into three parts:

- The physical layout of the network including a specification of the planned equipment and cabling configuration with a justification for selections. This section will also include a breakdown of the costing of the network. Students will be provided with a list of various communication components and their costs to work from.
- OPNET simulation results demonstrating that they have met all the performance requirements of the company.
- A discussion of areas where performance can be improved given that they are restricted in expenditure. They should also include suggestions for future enhancements.

## Conclusion

In this paper we have presented an overview of how we integrated network simulations software into a data communications course to enhance student learning of data communications concepts. The integration involves lectures, laboratories and assignments. The project helped students to gain a better understanding of complex communications protocols and enable them to gain experience in designing and testing communication networks.

# References

Barnett, B.L. (1993) An Ethernet Performance Simulator for Undergraduate Networking. *SIGCSE Bulletin*, Vol 25(1) pp145-150.

Baxter N., H. Chien,, A. Loreen, K. Marshall, and S. Baraniuk, (2000) *OPNET user manuals*. Mill 3, Inc.

Daigle, J.N. (1992) *Queuing Theory for telecommunications*, Addison-Welsey Publishing Company Inc., 1992.

Ramsden, P. Context and strategy: Situational influences on learning. *In Schmeck, R.R. (ed) Learning Strategies and Learning Styles*. Plenum Press, NY, 1988.

Tanenbaum, A.S. *Computer networks*, Prentice-Hall International Inc., Englewood Cliffs, NJ, 1999.

# Bibliographies

**Michael Dixon** is a Senior Lecturer in, and program chair for, Telecommunications Management at Murdoch University. In 1998 he established Murdoch as one of the first university based Cisco Systems Networking Academies outside of the USA. He instructs in the CCNA and CCNP programs at both undergraduate and postgraduate levels and holds a PhD in communication engineering.

**Terry Koziniec** is a Lecturer in Telecommunications Management at Murdoch University. He instructs in the CCNA and CCNP programs at both undergraduate and postgraduate levels and has been engaged in CCNP curriculum development for Cisco Systems. As a doctoral student, he is examining explicit congestion notification in TCPIP networks.