CITI Technical Report 93–6

XNetMod: A Design Tool for Large-Scale Networks

Kerman Deboo
deboo@athena.mit.edu

ABSTRACT

The network modeling tool XNetMod (based on NetMod) uses simple analytical models to allow designers of interconnected local area networks to analyze and simulate the potential performance of networks. This tool can be used on university campuses, and in industrial and government enterprises. XNetMod is based on the X Window System (X11R5) and the OSF Motif widget set (1.2).

August 18, 1993

Center for Information Technology Integration
University of Michigan
519 West William Street
Ann Arbor, MI 48103-4943

XNetMod: A Design Tool for Large-Scale Networks

Kerman Deboo

August 18, 1993

1. Introduction

Local Area Networks (LANs) rapidly evolved and multiplied in the 1980s. The increasingly diverse infrastructure, complexity and flexibility of networks, and growing number of applications has made predicting workloads, stress points, and network growth more difficult.

There are several important performance considerations in the design of LANs, such as reliability, maintainability, and expandability. XNetMod, a network modeling tool for the X Window System, allows users to analyze the performance of a configuration before implementing it.

XNetMod (0.0.0) is based on the X Window System (X11R5) and the OSF Motif widget set (1.2). An attempt was made to conform to the OSF Motif style guide.

2. About XNetMod

Network performance evaluation includes modeling, analysis, and design. Three approaches to performance evaluation are analysis, simulation, and measurement. It is the common consensus among network managers that these three approaches towards network performance evaluation are complementary in nature and each has its own place in the design life cycle of a computer network. Hence, network managers are increasingly demanding a network strategy toolbox to replace stand-alone tools.

XNetMod addresses the needs of network managers. It is an integrated tool for modeling, analyzing, and designing LANs. The principal application of the tool is to assist network managers in configuring a potential user's data network using proposed network hardware and software components. XNetMod can be used in a university, industrial, or government networking environment consisting of thousands of computer sites.

Network designers can use the interface, called the Model Builder, to graphically display and manipulate network topology, specify the input and output parameters for various network components,[1] and view the results of any of the three types of analyses.

XNetMod caters to both novice and experienced designers. Designers don't need to be experts in any modeling discipline or familiar with the specifications of various network media and protocols. They merely need to know what devices are present in the networks and have a basic knowledge of how they are connected. However, XNetMod provides enough flexibility to allow experienced designers to override the default specifications of the network components.

1. The terms *component*, *element*, and *node* are used interchangeably in this paper.

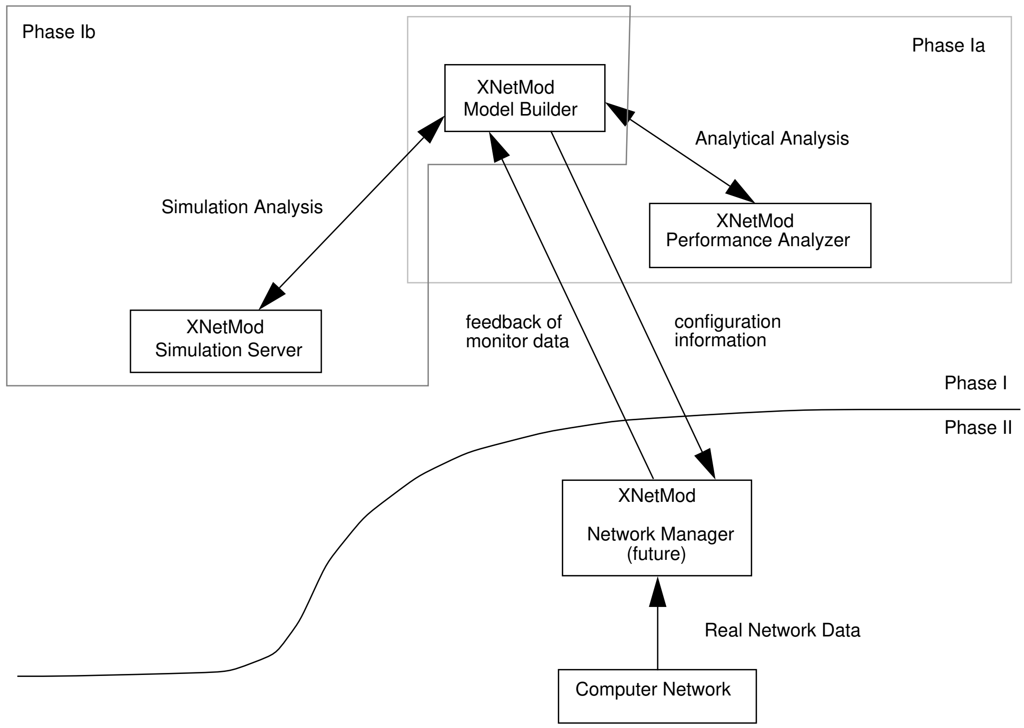

Figure 1: Architecture

XNetMod characteristics include:

- Ease of subnetwork definition
- Flexibility to easily add or delete subnetworks
- Ability to provide substantial user interaction for either simple parameter changes or major reconfiguration of LANs and backbones
- Hierarchical modeling capability for extremely large models.

The XNetMod project was divided into two phases (see Figure 1). The goal of Phase I was to incorporate both analytical (Phase 1a) and simulation techniques (Phase 1b) into XNetMod. The goal of Phase II will be to incorporate monitoring techniques into XNetMod.

For the current version of XNetMod (0.0.0), only Phase I of the project has been fully implemented.

3. Features

3.1 Input Parameters

XNetMod provides customized forms to enable you to specify the input parameters for the elements. The format of each form depends on the element's group and type.

XNetMod classifies network components in a pseudo object-oriented manner. See Figure 2.

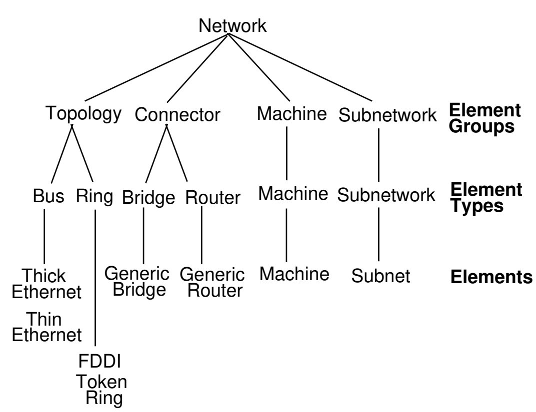

Figure 2: XNetMod Component Hierarchy

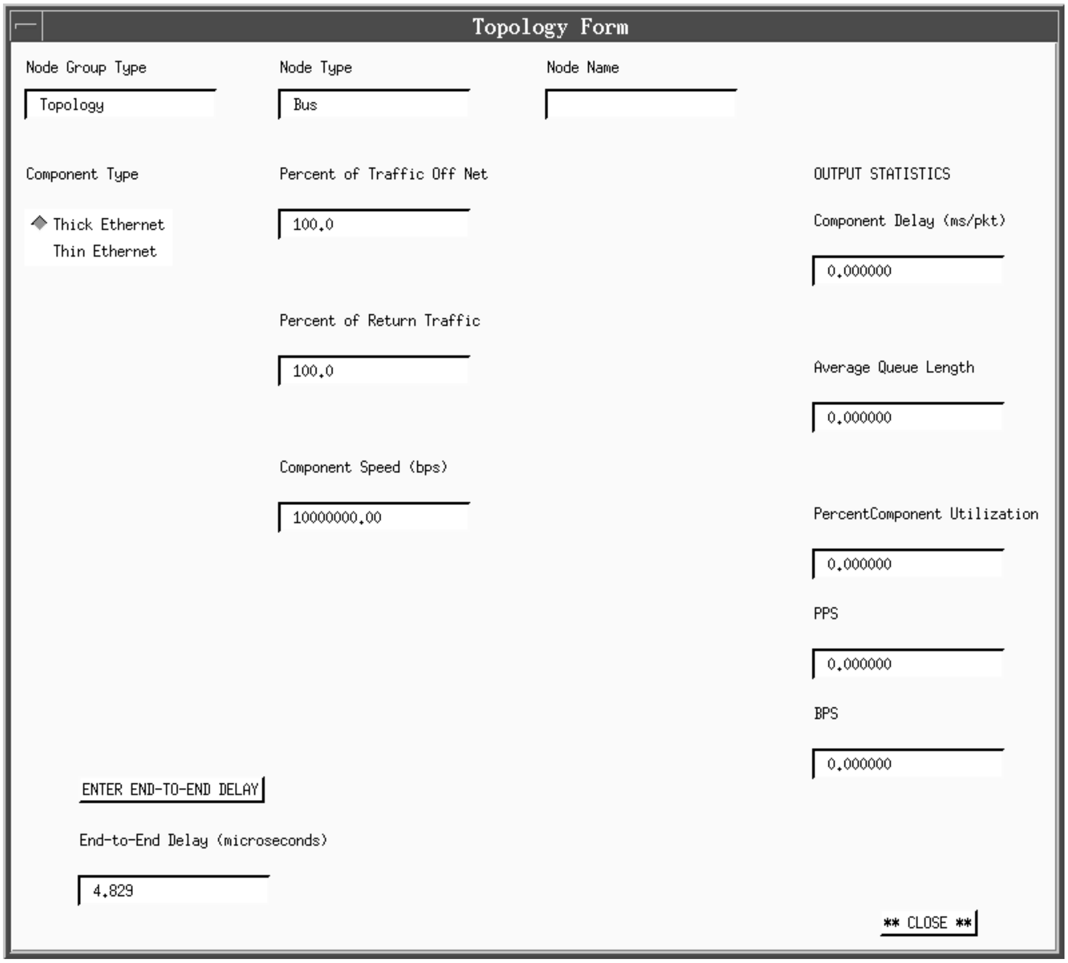

Figure 3. Topology Form

3.1.1 Topology

Topology shows how a LAN is laid out. At present the Topology element group is divided into two element types: Bus, and Ring.

When the elements are created, the only information specified is the element group and element type. Forms allow you to specify the exact element. See Figure 3.

The Topology form has two fields that users can customize: Percent of Traffic Off Net and Percent of Return Traffic. The Percent of Traffic Off Net specifies the percent of the traffic on that component that will be forwarded to other parts of the network. The Percent of Return Traffic specifies the amount of traffic that is forwarded to this component from other parts of the network.

3.1.2 Connector

A connector connects two or more topologies. At present the Connector element group is further divided into two element types: Router, and Bridge.

XNetMod may not allow two elements to be connected together. For example, two elements of the Topology element group can only be connected to each other with an appropriate element from the Connector element group. If components of a network are represented by nodes, and their interconnections by edges, the network would be a connected graph.

The Connector form is similar to the Topology form.

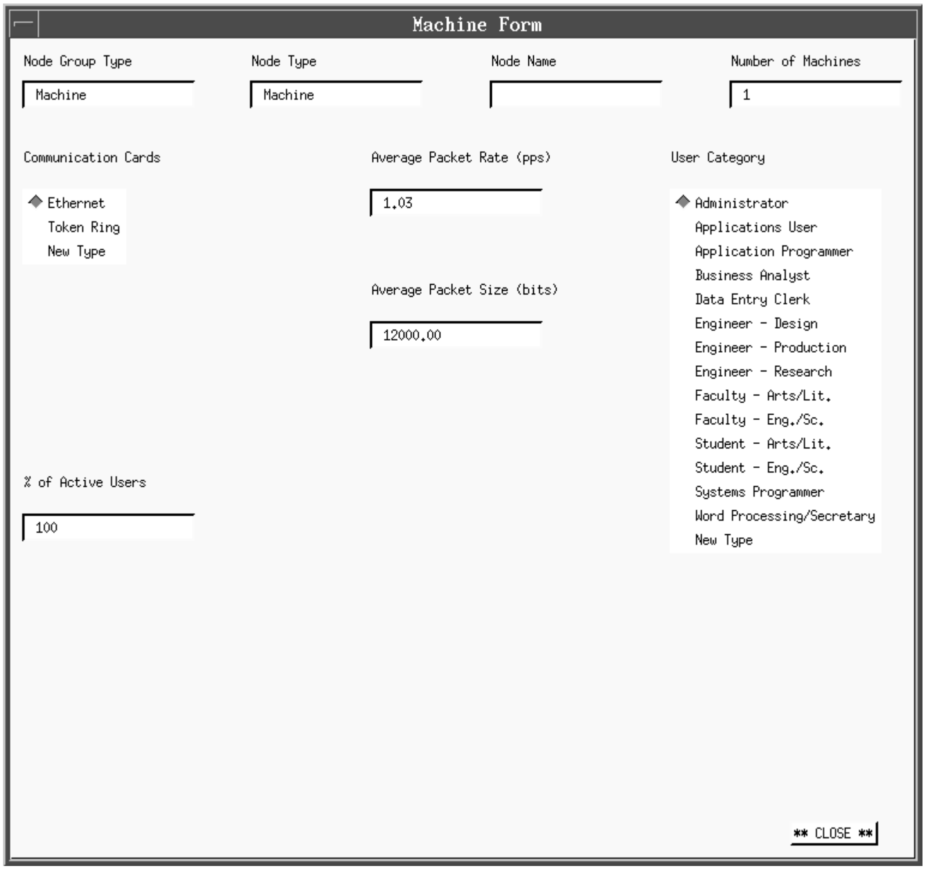

Figure 4. Machine Form

3.1.3 Machine

Because a machine is a source of data packets in a network, the Machine form allows you to specify the input parameters for a cluster of workstations, such as Macintosh's or PCs. See Figure 4.

At present there is only one element type, Machine, in the Machine element group.

A cluster of identical machines is represented by using just one element and then setting the Number of Machines field to the number of machines in the cluster. The Percent of Active Users field helps denote the actual number of machines in use at any given time. A percentage greater than 100% denotes multiple users on some machines.

3.1.4 Subnetworks

To facilitate the concept of bottom-up design, XNetMod allows users to create subnetworks. At present there is only one element type, Subnet, in the Subnetwork element group.

To create a subnetwork, it must already exist as a model. It can then be connected to the network by making a connection to an element inside the subnetwork. Subnetworks are saved separately from the rest of the model as though they are a single-level model. See Figure 5.

Subnetworks can be moved and destroyed like any other element, and viewed by choosing the View Subnetwork option from the pop up menu.

3.2 Output, Analysis, and Simulation

Output statistics provided by XNetMod are the component delay, average queue length, and percent utilization. The component speed field in the Topology form is updated whenever there is a change in the choice of element.

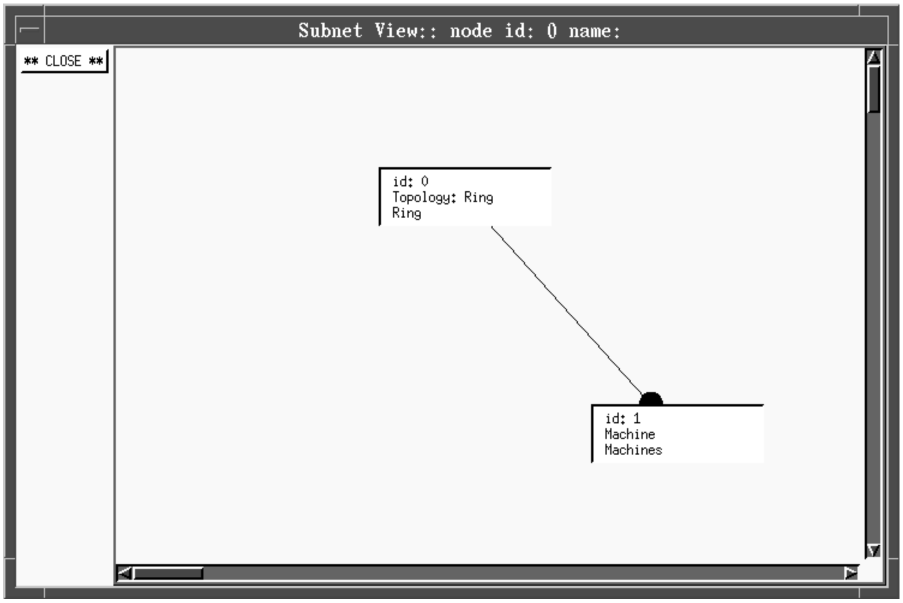

Figure 5. Subnet

XNetMod supports analytic and simulation techniques. It can summarize the entire results of an analysis or simulation, or just the results for a particular node.

4. Architecture

XNetMod consists of four major components: the Model Builder, the Performance Analyzer, the Simulation Server, and the Network Manager. Figure 1 shows the overall architecture of XNetMod.

4.1 Model Builder

The Model Builder is the central component of XNetMod. It provides facilities to:

- Graphically display and manipulate network topology
- Specify the input and output parameters for the various network components
- Display the results of any of the three types of analyses supported[2]
- Query and generate reports based on the currently displayed model.

2. Version 0.0.0 of XNetMod only supports analytic and simulation techniques.

Figure 6 shows the Model Builder. The interface includes: a menu bar to hold all the menus, a text area to display messages, an area to display the current model name, a scrollable drawing area, a button to display the component information panel, and a scrollable panel to display icons representing the available group types.

4.2 Performance Analyzer

The Performance Analyzer is responsible for all the analytical calculations. It gets as its input the network definition from the Model Builder and, in turn, returns to it the values of the output parameters to be displayed, such as component utilization, packet delay times, and average packet queue lengths at the components. See Figure 7.

Currently, all the network components are modeled using simple closed form analytic expressions, considering the effects of the lowest two OSI layers, namely the physical and data link (MAC sublayer only) layers. Future versions will analyze the logical link control (LLC) sublayer and network and transport layers. The entire network is modeled using a simple queueing network model.

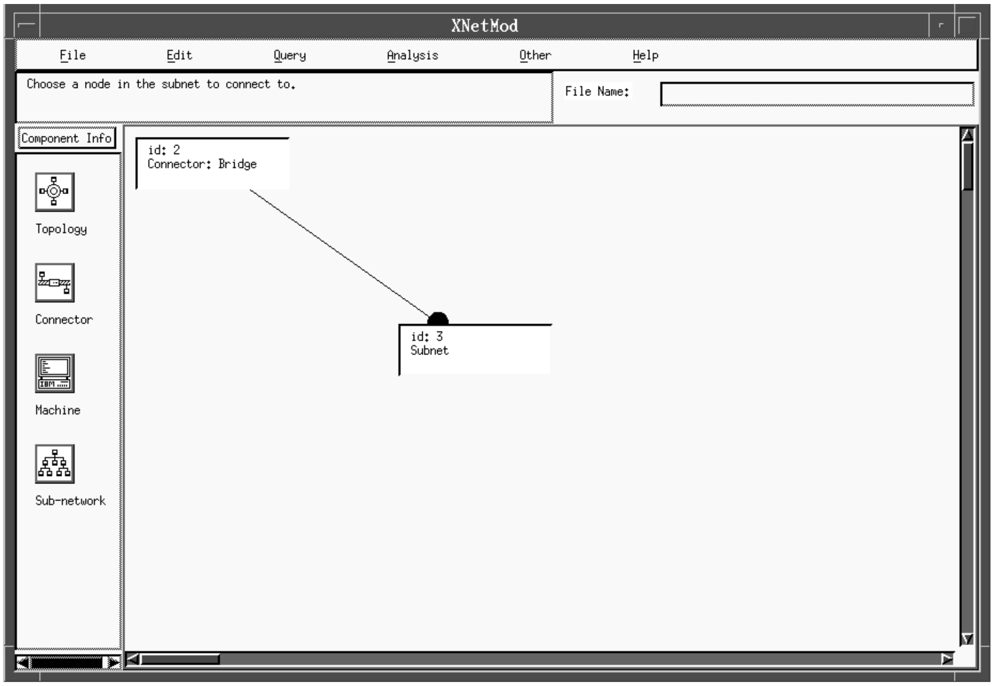

Figure 6. Model Builder

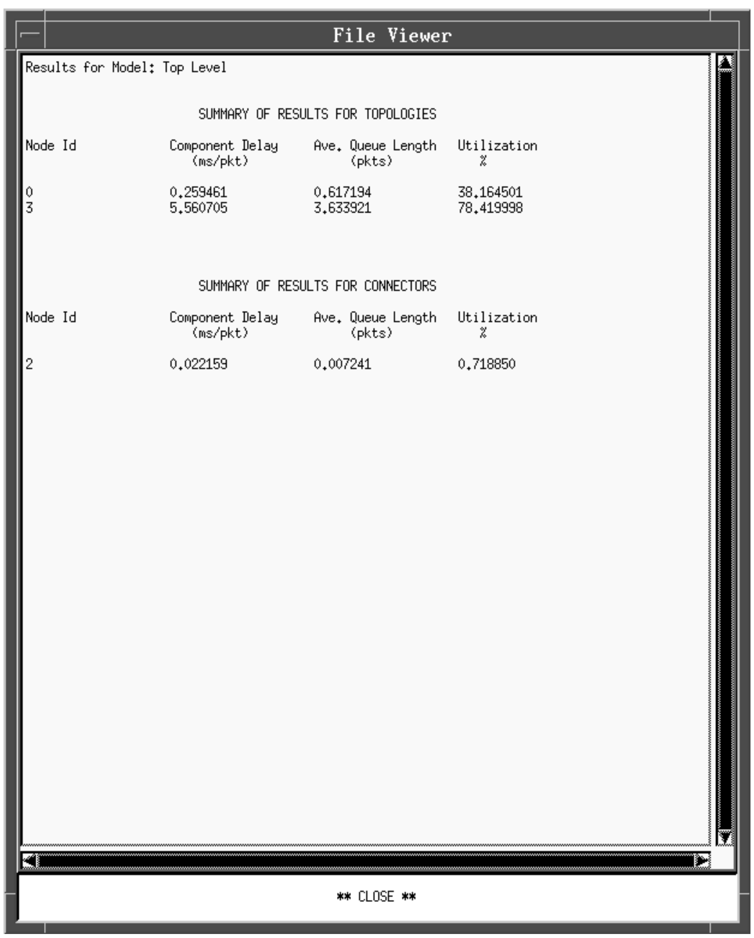

Figure 7. Performance Analyzer

4.3 Simulation Server

The Simulation Server is responsible for the simulation analysis. In this version of XNetMod, the simulation server is based on the NEST package [4]. The Simulation Server receives as its input a network definition from the Model Builder, performs the desired simulation, and returns the simulation results to be displayed. See Figure 8.

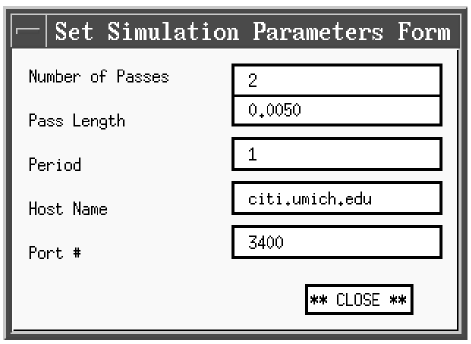

Figure 8. Simulation Server

4.4 Network Manager

The Model Builder supplies the Network Manager with the network configuration information. It also provides facilities to query and generate reports based on the data gathered by the Network Manager. The data gathered by the Network Manager can also be used to validate and adjust the estimates of input parameters for the analytic and simulation techniques. This ability is very useful because the accuracy of any analytic or simulation model is dependent on the accuracy of the values for the input parameters.

The Network Manager component will be implemented in a future release.

5. XNetMod versus NetMod

XNetMod is based on NetMod [2,3], an analytical modeling tool, and the NetMod-NEST interface [1]. Both were developed at the University of Michigan. NEST is a network simulation tool developed at Columbia University.

XNetMod has objectives that are similar to its predecessors, NetMod and the NetMod-NEST interface, but nevertheless differs from them in the design and implementation details. The objectives of XNetMod actually subsume those of NetMod and the NetMod-NEST interface.

5.1 Implementation Differences

XNetMod was developed using C and X/Motif in the UNIX environment, whereas NetMod was developed in the HyperCard environment on the Macintosh. This difference has resulted in XNetMod having several advantages over NetMod, such as the ability to use colors, a bigger working area, higher portability, access to greater computing power for more detailed models, and greater potential for extensions.

5.2 Design Differences

There are several design differences between XNetMod and NetMod. XNetMod has a well-structured, pseudo object-oriented view of the network, that is not present in NetMod. NetMod has a notion of parent-child relationship among network components. In XNetMod all network components are peers, as in actual networks. At present NetMod supports models of a more diverse range of network components.

References

1. U. Amin, D.W. Bachmann, K. D. Deboo, and T. J. Teorey, "NESTMOD: The NetMod–NEST Interface," pp 239-254 in *Proceedings of the 1991 CAS Conference*, Toronto, Ontario, Canada, IBM Canada Ltd. Laboratory (October 1991).
2. D. W. Bachmann, *NetMod User's Guide Version 0.16*, University of Michigan, Ann Arbor, MI (December 1990).
3. D. W. Bachmann, M. E. Segal, M. M. Srinivasan, and T. J. Teorey, "NetMod: A Design Tool for Large-Scale Heterogeneous Campus Networks," pp 15-24 in *IEEE Journal on Selected Areas in Communications*, Vol. 9 No. 1 (January 1991).
4. A. Dupuy, J. Schwartz, Y. Yemini, and D. Bacon, "NEST: A Network Simulation and Prototyping Testbed," *Communications of the ACM*, Vol. 33 No. 10 (October 1990).

Acknowledgments

Thanks to Toby Teorey and Dave Bachmann for their help and guidance during the project. Thanks to Dr. Jacob Slonim et al. at the Centre of Advanced Studies, IBM Canada Laboratories, Ltd., for their kind support. Thanks to Gerald Winters for his help.

The XNetMod project is funded by IBM. It is a part of the Consortium on Research in Distributed Systems (CORDS) project involving several U.S. and Canadian Universities along with the Centre for Advanced Studies, IBM Canada Laboratories, Ltd.