THE SIMULATION OF LOCAL AREA NETWORK PROTOCOLS

Anita Miller
Harris Corporation

Abstract. This paper discusses the use of simulation as a tool for evaluating the performance of local area network protocols. The design described includes a modular organization of protocol submodels and a user interface which allows the user to specify the configuration of both the network and the protocols. Results of experiments with several configurations of protocols at International Standards Organization (ISO) layers 1-4 are described.

INTRODUCTION

Local area networks communications software is organized in layers called protocols. Each protocol performs at least one well-defined function such as data packetizing, acknowledgement processing, time-out generation, addressing, or flow control to the higher layers. Interfaces between the protocols are carefully defined so that a minimal amount of information is passed to the adjacent layers and so that one protocol implementation may easily be replaced by another that provides the same interface. Typical local area network architectures are comprised of a hierarchy of as many as seven protocols.

This paper describes a discrete-event simulation model of a suite of local area network (LAN) protocols. The motivation for the model was the recognition of the complexity involved in assessing the performance of multiple layers of protocols under a variety of conditions. The model was constructed as a performance evaluation tool to allow experimentation with user-specified protocol and network configurations as well as different traffic loading and processor speeds. Feasibility and performance analysis using this tool was then conducted in support of a LAN Interoperability Study conducted by the Harris Government Information Systems Division for Rome Air Development Center* reported in Reference 1.

This research was sponsored in part by the Rome Air Development Center (United States Air Force) under contract #F30602-83-C-0108.

The model architecture was influenced by Didic and Wolfinger (Reference 2) who recommended that network protocol models be comprised of components that correspond to the layers of the International Standards Organization (ISO) reference model (shown in Figure 1) thus allowing the components to be defined and implemented as mutually independent.

Layer	ISO
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data link
1	Physical

Figure 1. ISO Reference Model for Network Protocols

Several features of the model are identified below because of their applicability to a wide range of simulation models:

1) Modular Organization

The model consists of several submodels that were written, verified, and validated independently. The user may specify through the user interface which combination of the submodels is to be operational for each experiment. The modular organization also permits the later addition of other protocols.

2) Flexibility

The model permits great flexibility because it is totally parameter driven. Examples of parameters to the model are

a. delay times such as the time to compute a CRC.
b. number of header bits added to a message to contain protocol data such as node addressing.

c. which protocols are to be modeled, for example; zero, one, or two protocols may be operational at layer 4.

d. network configuration data such as the number and location of nodes as well as their traffic characteristics.

3) User-friendly interface

The user interface is menu-driven and provides input error trapping and default values of parameters. The output statistics are collected at each protocol layer as well as network-wide.

4) Varying levels of detail

Some protocols are modeled in much greater detail than others because these layers were of more interest to the client. For example, the IP (Internet Protocol) is modeled as essentially one time delay in only a few lines of code while the TCP (Transport Control Protocol) submodel requires hundreds of lines to represent the desired detail.

5) Techniques for decreasing execution time

It is possible for LANs to be comprised of hundreds of nodes. If each protocol layer for every node were modeled in detail, the execution time would be enormous. In this model all the selected protocols were modeled for only a few representative nodes. The effects of the other nodes were modeled by only the media-access protocol in order to simulate realistic contention for the channel.

6) Report of model configuration

The typical usage of this simulation involves executing the model repeatedly with the variation of one parameter each time and comparing the results. In order to easily determine which configuration caused which results, it was important to produce a report of the parameter values along with the statistics collected.

SIMULATION PROCESS

A methodology for simulation projects must allow an organized approach but must also permit loopbacks as the model requirements are redefined and changed. The steps of the simulation process used are shown in flowchart form in Figure 2 and are detailed in the following sections.

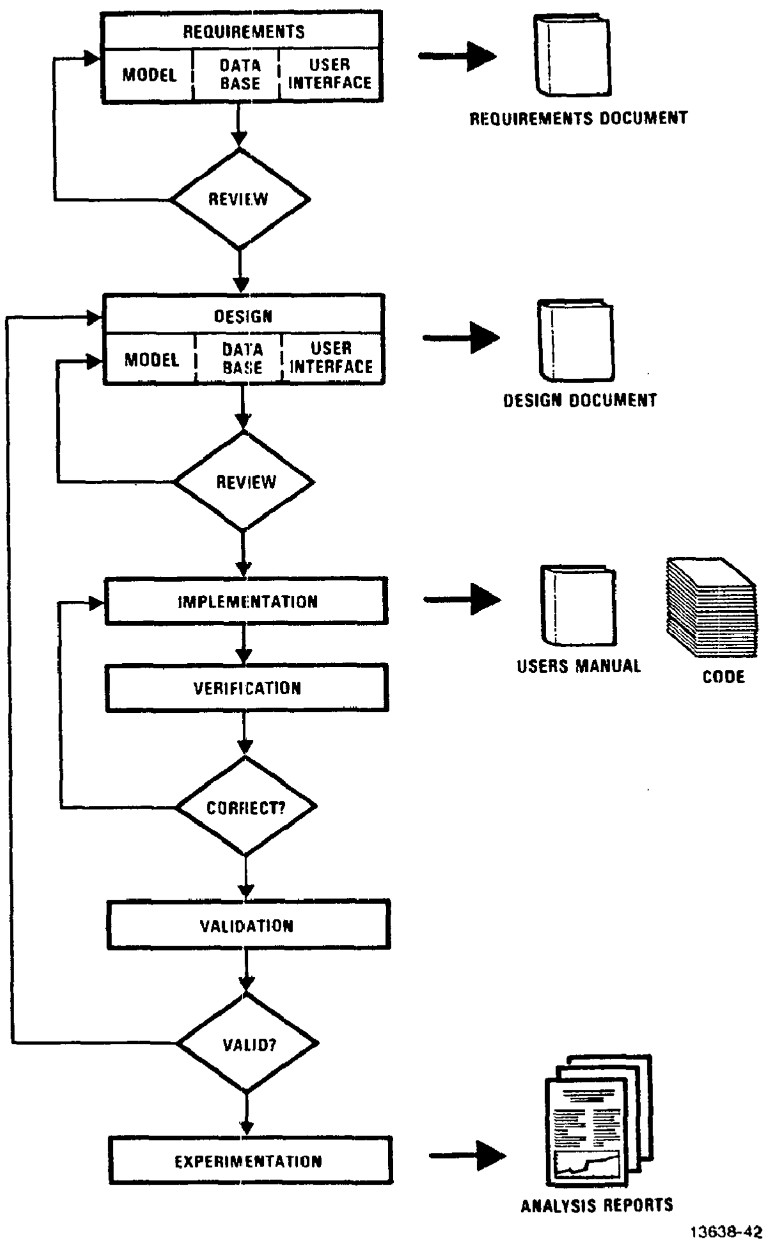

Figure 2. Software Development Methodology Tailored to Simulation

Requirements

To ensure that the goals of the modeling effort were accomplished, the model objectives were developed first. The objectives identified are summarized in the following list:

1) Represent operations of DOD/IEEE 802 hierarchy of layered protocols in multiple nodes attached to a single LAN. A future objective is also to represent an internet of multiple LANs.

2) Allow the user to vary the following system configuration aspects:

   a. protocol configuration
   b. protocol parameters
   c. node configurations
   d. message lengths
   e. message interarrival rates
   f. CPU loading from non-network tasks
   g. CPU processing speeds

3) Collect data on the following performance measures at each layer and for the entire network:

   a. throughput
   b. delay time
   c. channel utilization
   d. queue lengths
   e. retransmissions
   f. errors

4) Model the effect of loading on the channel by nodes other than those on which statistics were collected. The objective was to provide realistic contention and utilization of the channel.

5) Perform experiments to quantitatively evaluate:

   a. Media resource utilization
   b. Protocol overhead effects
   c. Processor resource requirements
   d. End-to-end performance

Figure 3 shows the specific protocol submodels selected to be implemented first. The protocol specifications used for each are documented on References 5-7. Plans are to extend the model in the future by adding other media-access protocols as alternatives to the Ethernet protocol and by adding a suite of protocols above TCP and UDP (User Datagram protocol).

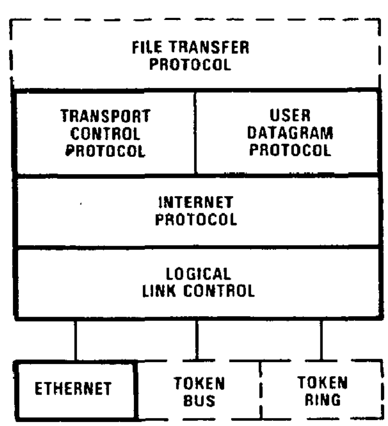

Figure 3. Implemented and Future Protocols Diagram

Design

The second step involved designing a model which represented the system under study and provided the detail necessary to satisfy the objectives. The areas which were of the most interest and received the greatest detail were the contention for the channel in the Ethernet protocol and the handshakes and state changes in the TCP protocol. The design was documented in a simulation-oriented psuedo code and was reviewed to ensure its accuracy.

Figure 4 illustrates the division of software into submodels. A control flow diagram is shown in Figure 5.

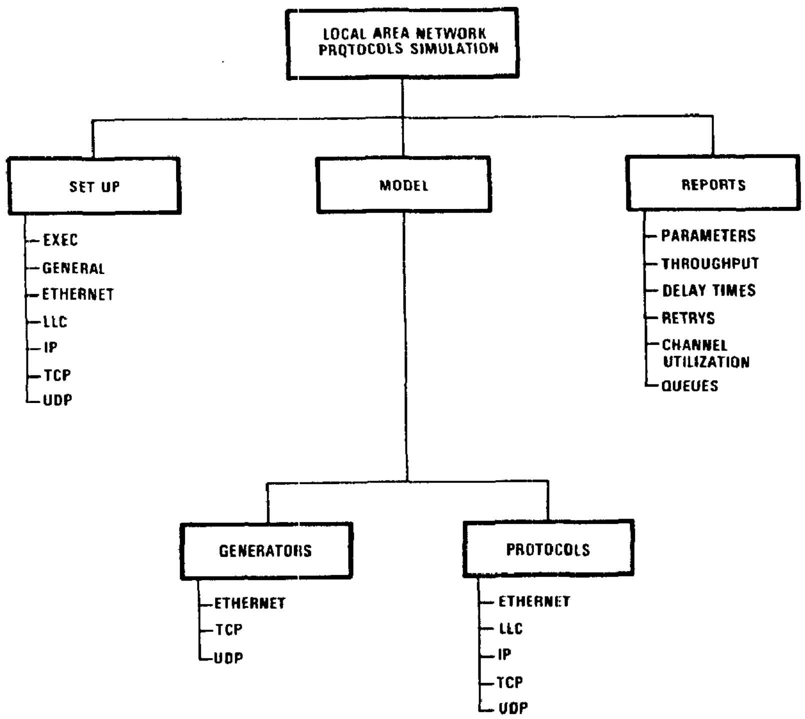

Figure 4. Model Partitioning Diagram

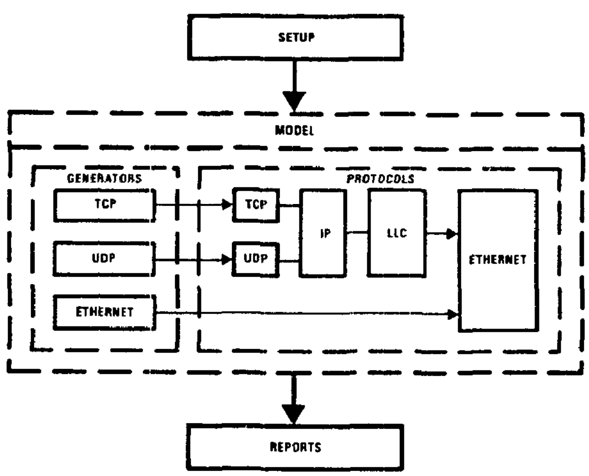

Figure 5. Model Flow Control Diagram

The purpose of the Setup model was to interface with the user to allow specification of the general network configuration parameters and the protocol specific parameters.

The Generators submodels were designed to generate requests for the highest layer protocols being simulated. This modeled the requests from the client, above the highest layer protocol in the nodes modeled in detail. For example, if the user specified that only the Ethernet procotol was to be modeled then only that generator would be triggered. However, if the user specified that both the TCP and UDP protocols were to be simulated then both generators would be triggered. In addition, if only part of the simulated nodes were to be simulated in detail, the media-access protocol (Ethernet) would be triggered to model the channel traffic caused by the non-detailed nodes. Generators were not included for the IP and LLC protocols as these protocols always function in connection with a higher layer protocol.

The Protocols submodels simulate the operations of user-selected protocols and their adjacent layer interfaces. Message entities are passed from one protocol submodel to another as specified in Setup. Note that the flexible design permitted different configurations of protocols. For example, if the user selected to model the IP protocol but did not select to model the LLC (Logical Link Control) protocol, message entities would be routed from the IP protocol to the media-access protocol (Ethernet).

The Reports submodel was designed to produce reports of the statistics collected during model execution.

Implementation

The third step was to convert the design to computer executable code. The model was implemented on a Harris H-800 computer in SLAM II (Simulation Language for Alternative Modeling) and the user interface was coded in FORTRAN 77. SLAM provided the high level simulation constructs useful for modeling the concurrency and FORTRAN provided the capability to have an interactive user interface. The model was implemented in a top-down fashion employing a very modular framework to later allow additional protocols. Also, the model had several hundred parameters to provide the user great flexibility in configuring the system for an experiment.

Verification

The objective of the verification step was to determine that the model executed as designed. Techniques used for verification included traces, examination of the summary report, and structured walkthroughs.

Each protocol submodel was first tested with only one message entity then with multiple entities. Each submodel was also tested separately before being combined and tested with other submodels.

Event traces were used to verify each submodel. They provided a step by step report of the location of an entity and the simulated time at every point in the program where simulated time passes. The various conditions to be tested were forced in order to trace that entities followed the correct routes through the model, that the correct amount of simulated time passed, and that entities had the correct attributes.

The summary report also provided other data used in debugging the program. Data was provided on the minimum, maximum, and current status of resources, files, and activities. Additional debugging information was available from the statistics collected on collisions, retrys, interarrival times, message lengths, time in layers, and contents of the TCP Transmission Control Block.

A structured walkthrough at which several programmers reviewed the implementation and evaluated the flow of logic was used to verify all submodels.

Validation

Validation is the process of determining that the model is an accurate representation of the actual system. The validation techniques used were hand calculations, comparison with published results and face validity checks with experts.

For example, the expected performance of the model could be hand calculated for the case where only one node was transmitting. The performance of the model was also compared to published results from other studies. The Ethernet submodel, for example, was compared with results released by the IEEE Project 802 Traffic Handling Characteristics Committee Report (Reference 8) and the TCP submodel was tuned to results reported by MITRE (Reference 9).

Experimentation

The protocols of greatest interest during this study were the Ethernet and TCP protocols. They were modeled in more detail and more experimentation was done to determine their performance than with the other protocols. A sampling of the experimentation results are included in this paper as Figures 6-10. Further details can be found in Reference 1.

ACKNOWLEDGEMENTS

I would like to thank Walter Elden for his instruction, guidance, and patience; Susan West and Sheila Kasprzak for their technical assistance in the design and implementation of the model and user interface; as well as Steve Morris and Racquel Crowley for text editing and entry.

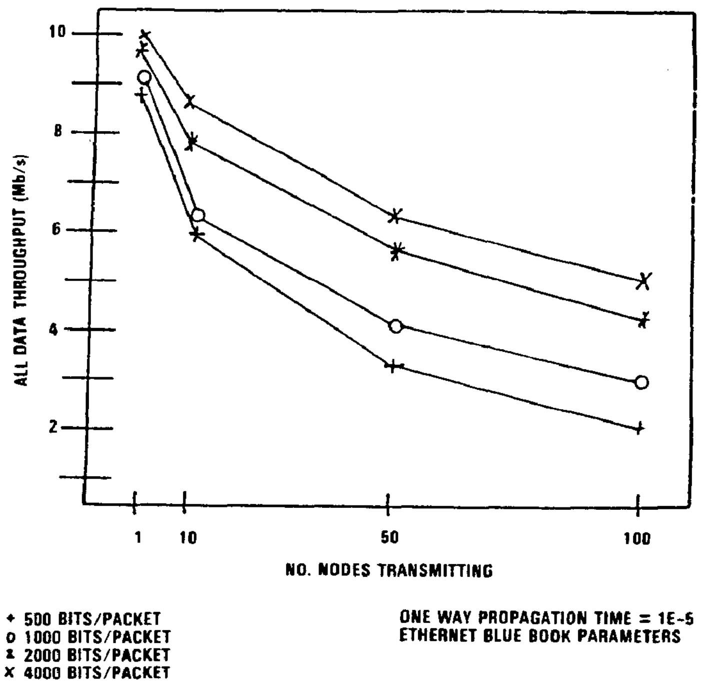

Figure 6. This graph shows the maximum throughput (with Ethernet only) for 1, 10, 50, and 100 nodes with messages having fixed lengths of 500, 1000, 2000, and 4000 bits.

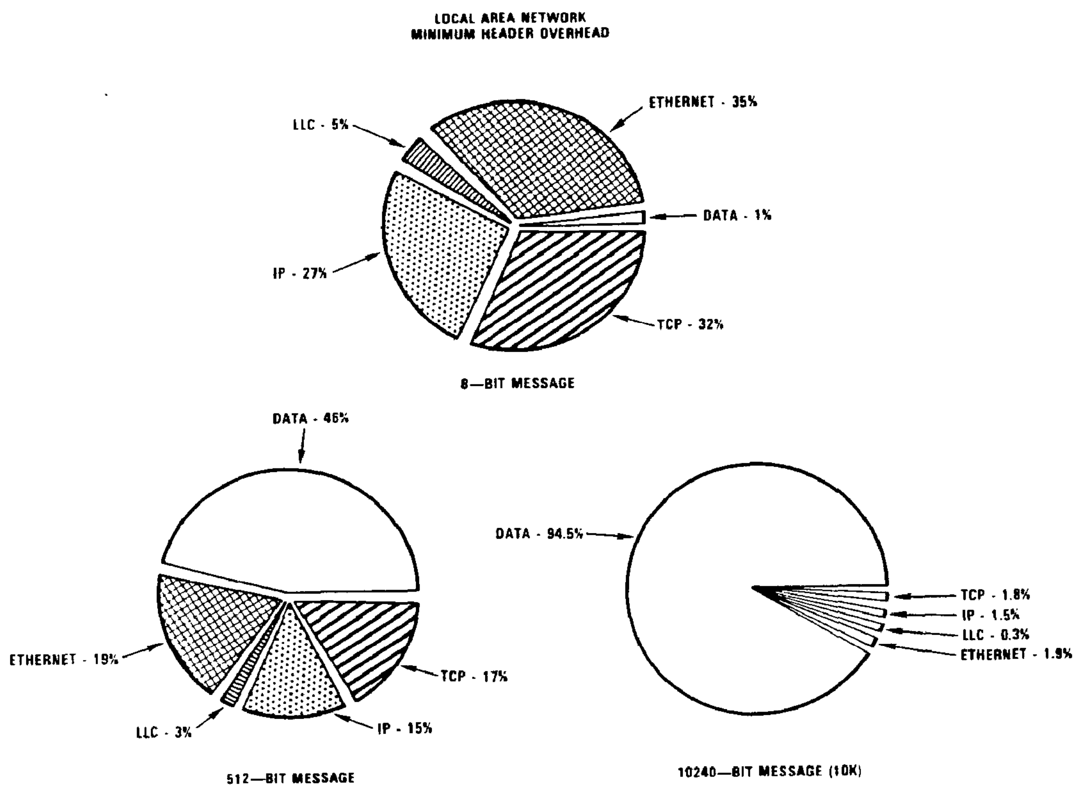

Figure 7. These pie charts illustrate the dramatic impact of the header bits appended to the message by the Ethernet, LLC, IP, and TCP protocols. Note that for 8-bit messages, only 1% of the traffic is user data.

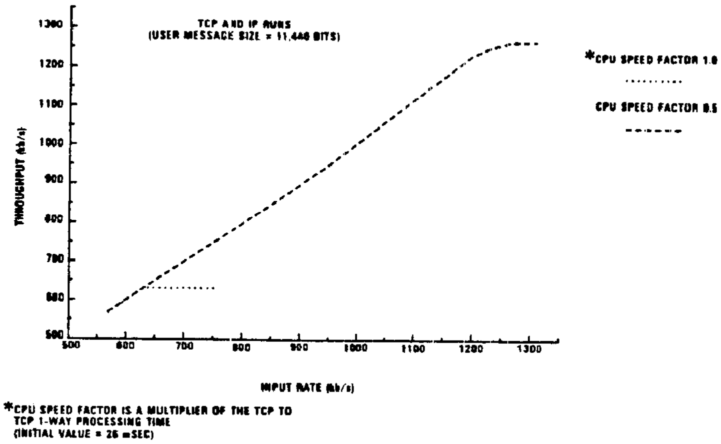

Figure 8. The maximum throughput with TCP and IP is greatly affected by the speed of the CPUs at the network nodes.

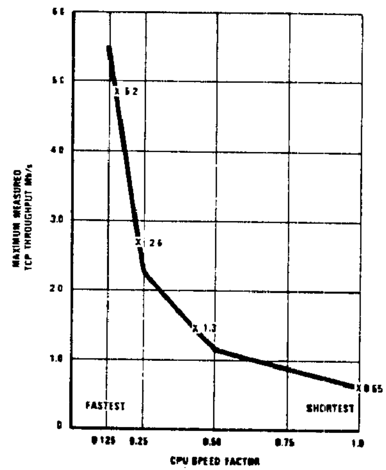

Figure 9. This graph plots the maximum measured TCP throughput for various CPU speeds.

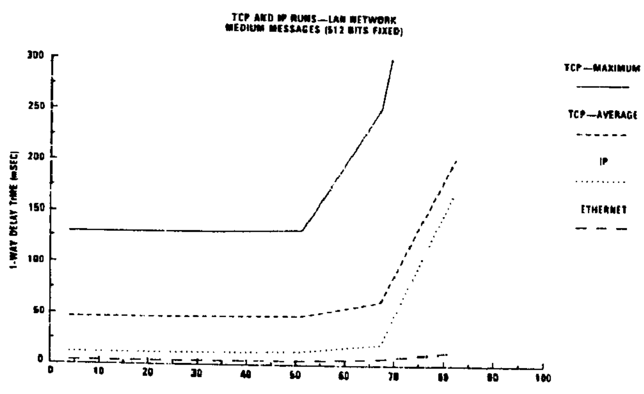

Figure 10. This graph shows % channel utilization versus one-way delay time for TCP, IP, and Ethernet. The data for this type of graph was compiled from many executions of the model.

REFERENCES

1. Elden, W. L. and Miller, A., et al., "LAN Interoperability Study Report," RADC Report No. RADC-TR-84, August, 1984.

2. Didic, M. and Wolfinger, B., "Simulation of a Local Computer Network Architecture Applying a Unified Modeling System," Computer Networks, May, 1982.

3. "The Ethernet - a Local Area Network, Data Link Layer and Physical Layer Specification," Digital, Intel and Xerox, Version 1.0, September, 1980.

4. "Internet Protocol Military Standard," MIL-STD-1777, August, 1983.

5. "Transmission Control Protocol Military Standard," MIL-STD-1778, August, 1983.

6. "IEEE Project 802 Local Area Network Standards, Logical Link Control," Draft E, September, 1983.

7. "IEEE Project 802 Local Area Network Standards, CSMA/CD Access Method and Physical Layer Specifications," December, 1982.

8. Stuck, B., "Analyzing Congestion in Local Area Networks: IEEE Computer Society Project 802 Local Area Network Standards"

9. Holmgren, S., "Evaluation of TCP/IP in a Local Network," The MITRE Corp., WP81W00568, September, 1981.