A LOCAL COMPUTER NETWORK SIMULATION

John M. McCoy
Stewart L. French
Razmik Abnous
M. J. Niccolai
University of Louisville

Abstract

Computer Networks are an important part of our society and they are quickly becoming an integral part of computer science basic curriculum. This paper describes the development of a computer simulation model for a local computer network and its use as a viable tool in computer science education.

Key Words

Computer Network, Local Computer Network, Modeling, Computer Simulation, Computer Network Simulation, Computer Performance.

I. Introduction

Computer networks are an important part of our society and directly or indirectly affect many people. Networks such as airline reservation systems, 24-hour banking systems, and the telephone switching system underlie many of our normal activities. For the purposes of this paper, a computer network will be defined as a system with multiple interconnected processing elements. Usually there are some global resources which all processing elements can utilize. A local computer network is a network where all processors are located in the same geographic location. This paper describes the development of a computer simulation model for a local computer network and its use as a viable tool in computer science education. Computer simulation of computer networks is a timely topic with much work currently being done. However, reference material is scarce and books dealing with this subject are just being introduced.

The development of a simulation model for our local network was desirable since the network architecture developed did not match exactly any existing system. One of the main uses of a computer simulation in computer architecture is to model the new system to determine if any major performance bottlenecks exist, and if any are found, possible techniques for eliminating them can be tested (1).

 0-89791-036-2/81/0200/0263 $00.75

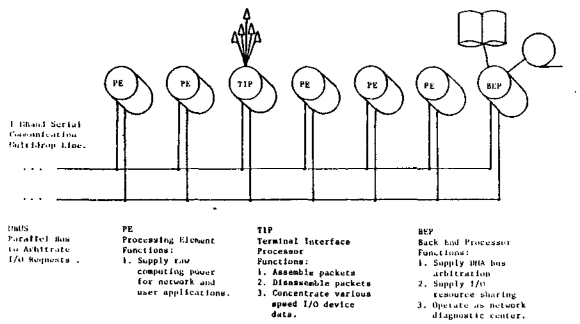
The system which was modeled is a multi-user system and simulation models have been effectively used in the past to evaluate hardware parameters such as the number of terminals, as well as software parameters, such as the optimal type of scheduling algorithm (2,3,4,5).

The simulation model described in this paper has several educational uses including:

1. Teaching simulation development including the modeling of complex systems, as well as implementation and statistical analysis;
2. The implementation of the model illustrates concurrent programming techniques using the DEC SYSTEM 10 interactive version of SIMULA, an ALGOL based simulation language (6,7); and
3. Teaching the concepts of computer networking. Although this project includes construction of the network, the simulation model introduces networking concepts without the need for a large amount of expensive equipment. The techniques used in this model could be used to study other types of computer networks.

II. Description of MANTRA System

The Multiple Access Networking and Teaching Research Arrangement (MANTRA) system is the network analyzed by this simulation model. Figure 1 contains a block diagram of the system. All computers in the network are LSI-11/03 microcomputers manufactured by Digital Equipment Corporation (DEC). The Terminal Interface Processor (TIP) is one processor dedicated to the function of interfacing terminating devices to the rest of the network. A terminating device is defined as a device capable of communicating using the RS232C or 20 ma loop standards which means that terminals or stand-alone computer systems such as APPLE microcomputers, PET microcomputers or even a large-scale computer such as IBM 370 will have the ability to communicate with the MANTRA system.



System Configuration

The tip can be functionally viewed as a data router or a front-end processor since it is an interrupt-driven high speed buffering device allowing data to flow in 3 directions including:

1. Device to Device
2. Device to Central LSI-11 Processor
3. Central LSI-11 Processor to Device

The central LSI-11 processors or processing elements (PE's) are LSI-11 processors capable of running user applications code. These processors also have the capability to communicate with each other via the MANTRA protocol and high speed DMA channels.

All central LSI-11 processors are connected to a specialized processor used to control a variety of peripheral devices, including: floppy discs, hard discs, line printers and other devices. The combination of these shared devices is called the "Peripheral Pool". The computer which controls the Peripheral Pool is called the Back End Processor (BEP). Any central LSI-11 can connect to a free device in the Pool. The pooled device is then considered connected to the particular device until explicitly or implicitly released.

III. Description of the Model

The simulation model was designed to answer performance questions of the interrupt capabilities and DMA transfer rates of the TIP. Questions concerning the peripheral pool's dedicated processor, in regard to the number of requests that appear in a given time interval, could be resolved to find the optimum buffer length in this processor.

Specifically, two questions are addressed:

1. What is the maximum number of terminating devices and LSI-11 processors that the terminal Interface Processor can handle given:

   A. Various hardware response times
   B. Various software speeds of response.

The next question is much more subjective and is a function of many parameters including sophistication of program and operating system overhead.

2. What queue size should be allocated within the Peripheral Pool control processor to efficiently handle the I/O requests.

The model provided has several capabilities including:

1. Real-Time Application

   A. Gather data on user response time distribution;
   B. Gather data on system response time distribution;
   C. Gather LSI-11 commands to use later in automatic mode.

2. Automatic Mode
   Allows for selection of parameters, then the program will automatically select transactions from a direct access file of real-time collected transactions.

3. Various Configurations

   A. In real-time, a user can select any combination of terminating devices and/or any combination of processing elements to communicate with.
   B. In automatic mode, a configuration will be selected by the user and an associated interconnection matrix will be read from a file of stored configurations.

4. Real-Time Disc Accesses
   The disc accesses will be performed in real time when in the automatic mode. This will support the following capabilities:

   A. Supply data to the system of a varying length.
   B. Supply meaningful data for an LSI-11 to LSI-11 communication configuration.

In the initial set of experiments using the model, three parameters were selected to test including:

1. Number of LSI-11 central processors and number of terminating devices.
   Since in the initial configuration of the MANTRA system one terminal would communicate with one LSI-11, the configurations for this study were limited to 5 cases, all being on a one-to-one basis. The cases examined ranged from two terminating devices communicating with two LSI-11's to thirty-two terminating devices communicating with thirty-two LSI-11's.

2. Speed of the DMA Handling
   One of the five potential values could be selected with each value including software, hardware, and transmission delays.

3. Speed on interrupt responses
   One of five potential values could be selected with each value contained in the interval of typical hardware interrupt response time (8,9).

The output variables generated by the model are:

1. Length of Terminal Interface Processor's queue per time interval.
   Ideally the queue length will be zero until the number of devices and LSI-11's with the response of the TIP exceed its capability to respond quickly enough.

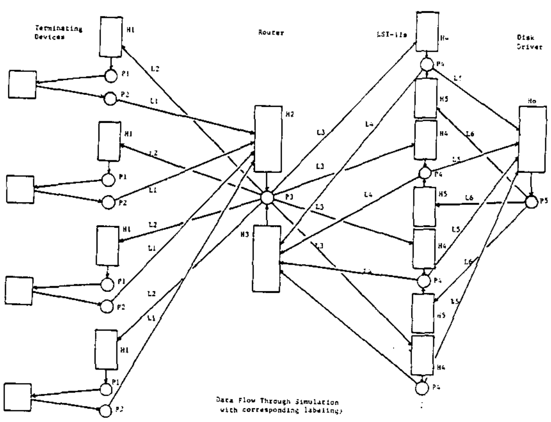
2. Number of characters serviced per time interval within TIP in two regards:

   A. DMA
   B. Interrupts to and from terminating devices

3. Length of disk request queue within Peripheral Pool Processor

IV. Use of a Model in Computer Science Education

The model of the MANTRA network is directly useful in illustrating the major steps necessary for the development of a computer simulation, as well as for the development of the local computer network. It illustrates the analysis step in the overall approach used to design the model. Figure 2 is a diagram which illustrates the flow of data through the simulation. It illustrates the complexity of queue representation for any computer network.



The model illustrates the statistical analysis steps necessary in any computer simulation. The two major types of statistical analyses required were:

1. Gathering necessary statistical distributions

Our model required two statistical distributions so that it could run automatically and gather data. The first was a user think-time distribution which is used to create transactions from the simulated terminating devices. This distribution was obtained by running the simulation in a real-time mode with real users and fitting the gathered data to an exponential distribution. An exponential distribution was used because of its past success in modeling terminal requests in multi-user computer systems (10,11). Figure 3 defines the interval of time that was considered user think time as used by the model.

```
  /  Time n   ---------------  User hits return
A {
  \  Time n+1 ---------------  Output begins
B {
  \  Time.n+2 ---------------  User hits return
```

The time interval A represents the computer response time.
The time interval B represents the user think time.

FIGURE 3 - RESPONSE TIME CALCULATIONS

Figure 4 is a graph of the calculated user think time distribution used by the model.

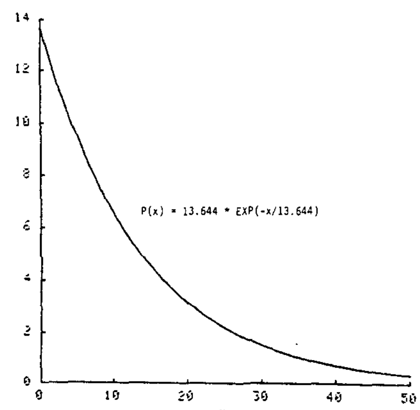


FIGURE 4 - User Think Time Distribution

The next required distribution was a computer response time distribution which was obtained by fitting empirical data to an exponential distribution. Figure 5 is a graph of the computer response time distribution. The curve was divided into three intervals including:

1. LSI Response Time
   This interval was used for simulating commands that did not require disc usage.

2. Disc-command Response Time
   This interval was used to simulate commands that did require disc usage.

3. Application Code
   This interval was viewed as representing the response times of user application code and was not used in the model.

The Kolmogorov-Smirnov test was used to verify the obtained distribution at a 99% confidence level.

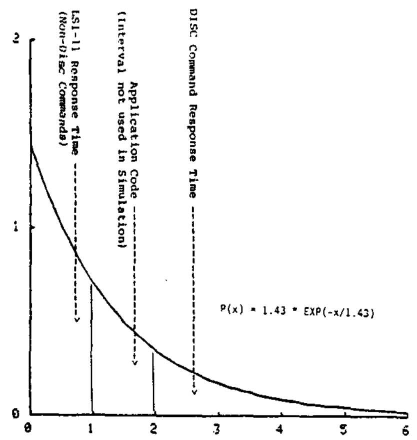
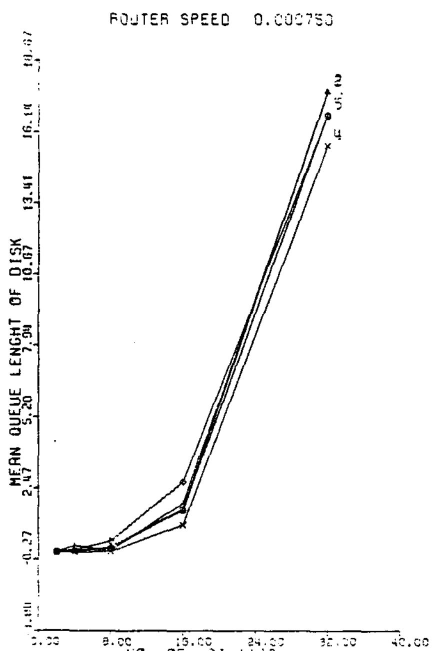


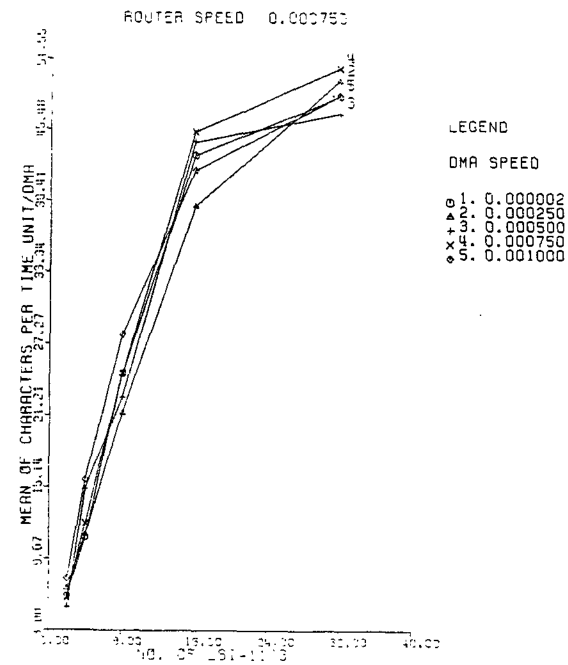
FIGURE 5 - Computor and Disc Response Time Distribution

2. Statistical Analysis of Output

The large amount of data produced by the simulation was reduced to plots. Figure 6 contains a plot of the mean disc request queue length versus number of LSI-11's. Since this is a one to one correspondence between terminating devices and LSI-11's, this graph also shows how increasing the number of terminating devices affects the system.



This graph seems to indicate that the network cannot support more than 16 computers or terminating devices since the mean disc request queue size increases rapidly as the number of computers goes beyond 16. Figure 7, which contains a graph of the mean number of characters handled by the DMA Unit, seems to support this conclusion since it levels off at about 16 computers indicating the DMA unit cannot support more than 16 computers or terminating devices.



V. Conclusions

This type of modeling effort has also been shown to be concurrent programming. The major problem in teaching concurrent programming concepts is the difficulty in representing and implementing such programs. SIMULA is an appropriate language to use to implement concurrent programs and is one of the few high level languages which provides such facilities. Our work was directed toward using SIMULA to model a local computer network; however, with extensions, it could be used to model multi-user operating systems and study the use of various scheduling, device and memory allocation schemes.

The model described in this paper allows schools with limited budgets to enter the expanding field of computer networking and to experiment with different network configurations without the large amount of money necessary to actually build such a system. Feasibility studies of new architectures by a simulation model might increase the chance of funding so that a system can be built.

In conclusion, it is likely computer networks will become more common and the use of computer simulation of networks will also expand since the complexity of networks is growing at a much faster rate than the mathematical theory necessary to analyze these systems using analytical methods.

A complete manual with examples and source listings can be found in reference 12. In addition, the general SIMULA model described above is available on tape for use on DEC-10 equipment.

VI. References

1. Stone, H., Introduction to Computer Architecture, Science Research Associates, 1975, pp. 474-541.

2. ibid.

3. Martin, James, Design of Real-Time Computer Systems, Prentice-Hall, 1967.

4. Hellerman, Herbert, Computer System Performance, McGraw-Hill, 1975.

5. Yourdon, Edward, Design of ON-LINE Computer Systems, Prentice-Hall, 1972.

6. Franta, W. R., The Process View of Simulation, Amsterdam, North-Holland, 1977.

7. Swedish National Defense Research Institute, SIMULA Language Handbooks I, II, III, Dec System 10 Documentation, Digital Equipment Corporation.

8. Micro-computer Processors, Digital Equipment Corporation, 1978, pp. B-4, B-5.

9. Micro-computer Processor Handbook, Digital Equipment Corporation, 1979, p. 556.

10. Martin, pp. 372-438.

11. Hellerman

12. University of Louisville Class Project Report, AMCS 522 Discrete System Simulation, "A Simulation of the MANTRA Local Computer Network", Razmik Abnous, Stewart French, John McCoy, Unpublished.