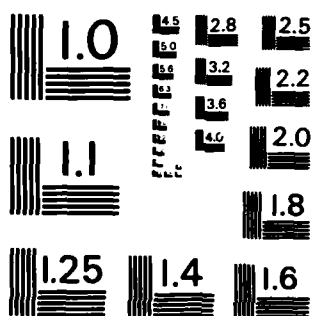




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

2

ESTIMATING COMPUTER COMMUNICATION NETWORK PERFORMANCE
USING NETWORK SIMULATIONS

Major(P) Albert B. Garcia
HQDA, MILPERCEN (DAPC-OPA-E)
200 Stovall Street
Alexandria, Virginia 22332

Final Report April 1985

AD-A162 580

Approved for Public Release, Distribution Unlimited



A dissertation submitted to the School of Engineering, University of Dayton,
Dayton, Ohio in partial fulfillment of the requirements for the degree
Doctor of Philosophy in Engineering, Major in Electrical Engineering.

NTIC FILE COPY

This document has been approved
for public release and sale; its
distribution is unlimited.

85 12 19 040

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A143 580	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Estimating Computer Communication Network Performance Using Network Simulations		5. TYPE OF REPORT & PERIOD COVERED Final April 1985
7. AUTHOR(s) Major(P) Albert B. Garcia		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Student, HQDA, MILPERCEN (DAPC-OPA-E) 200 Stovall Street Alexandria, Virginia 22332		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS HQDA, MILPERCEN ATTN: DAPC-OPA-E 200 Stovall Street, Alexandria, Virginia 22332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE April 1985
		13. NUMBER OF PAGES 142
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Dissertation submitted to the School of Engineering of the University of Dayton, Dayton, Ohio for the degree Doctor of Philosophy in Engineering.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) NETWORK SIMULATION, NETWORK PERFORMANCE, COMPUTER SIMULATION, COMMUNICATION NETWORKS, COMPUTER-COMMUNICATION NETWORKS, SLAM, SLAM SIMULATION, QUEUEING MODELS, QUEUEING MODEL SIMULATION, SIMULATION MODEL, MESSAGE DELAY MODEL		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A generalized queueing model simulation of store-and-forward computer communi- cation networks is developed and implemented using Simulation Language for Alternative Modeling (SLAM). The model includes an ACK/NAK data link protocol, four-level message precedence, finite queues and a response traffic scenario. Network performance, indicated by average message delay and message throughput, is estimated using the simulation model.		

ESTIMATING COMPUTER COMMUNICATION
NETWORK PERFORMANCE USING
NETWORK SIMULATIONS

Dissertation
Submitted to
The School of Engineering of the
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for
The Degree
Doctor of Philosophy in Engineering
Major in Electrical Engineering

by
Albert B. Garcia

UNIVERSITY OF DAYTON
Dayton, Ohio
April, 1985

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A-1	



ESTIMATING COMPUTER COMMUNICATION NETWORK PERFORMANCE
USING NETWORK SIMULATIONS

APPROVED BY:

Dana B Rogers

Dana B. Rogers, Ph.D.
Associate Professor
Department of Electrical
Engineering
Committee Chairperson

Bernhard M Schmidt

Bernhard M. Schmidt, Ph.D.
Chairman
Department of Electrical
Engineering
Committee Member

Anthony J Evers

Anthony J. Evers, M.S.E.E.
Associate Professor
Department of Electrical
Engineering
Committee Member

Stanley J. Back

Stanley J. Back, M.S.
Associate Professor
Department of Mathematics
Committee Member

Gary A. Thiele

Gary A. Thiele, Ph.D.
Associate Dean/Director
Graduate Studies & Research
School of Engineering

Russell A. Primrose

Russell A. Primrose, Ph.D.
Dean
School of Engineering

ABSTRACT

ESTIMATING COMPUTER COMMUNICATION NETWORK PERFORMANCE USING NETWORK SIMULATIONS

Albert B. Garcia, Ph.D.
University of Dayton, 1985

Major Professor: Dr. Dana B. Rogers

A generalized queueing model simulation of store-and-forward computer communication networks is developed and implemented using Simulation Language for Alternative Modeling (SLAM). A baseline simulation model is validated by comparison with published analytic models. The baseline model is expanded to include an ACK/NAK data link protocol, four-level message precedence, finite queues and a response traffic scenario. Network performance, as indicated by average message delay and message throughput, is estimated using the simulation model.

VITA

November 26, 1944	Born - Shirley, Massachusetts
1968	B.S.E.E., West Virginia University, Morgantown, West Virginia
1968 - Present	United States Army Officer
1970	M.S.E.E., West Virginia University, Morgantown, West Virginia
1980	M.B.A., Fairleigh Dickinson University, Rutheford, New Jersey
1985	Ph.D., University of Dayton, Dayton, Ohio

TABLE OF CONTENTS

ABSTRACT.	iii
TABLE OF CONTENTS	v
LIST OF ILLUSTRATIONS	vii
LIST OF TABLES.	ix
ACKNOWLEDGEMENTS.	x
CHAPTER	
I. INTRODUCTION	1
Motivation	
Purpose	
II. A CLASSICAL ANALYTIC NETWORK MODEL	4
Background	
The Network	
Network Performance Measurement	
Optimizing Network Performance	
A Classical Analytic Network Model	
Single-Server Queue Model	
Multiple Queue Model	
Kleinrock's Numerical Examples	
Star Network	
Fully-Connected Network	
Comments on Analytic Models	
III. THE SLAM SIMULATION MODEL.	23
Introduction	
The SLAM Language	
Simulation Program Organization	
Baseline Model	
Message Generation Module	
Node Processing Module	
Node-to-Node Transfer Module	
Statistics Collection Module	
Baseline Model Validation	

Baseline Model Expansion
 Full Simulation Model
 Message Generation Module
 Node Processing Module
 Node-to-Node Transfer Module
 Statistics Collection Module
 External Effects Module

IV. PERFORMANCE EVALUATION OF A NETWORK.	50
Introduction	
Network Description	
Star Network Performance	
Mesh Network Performance	
Mesh-Plus Network Performance	
Fully-Connected Network Performance	
Optimum Network Selection	
Four-Level Precedence	
Response Traffic	
ACK/NAK Protocol	
Finite Queue Capacity	
V. SUMMARY AND CONCLUSIONS.	69
Summary	
Discussion of Findings	
Conclusions	
Areas for Future Study	
APPENDIX	
A	75
B	80
BIBLIOGRAPHY.	140

LIST OF ILLUSTRATIONS

1. Structure of a Computer Communication Network. . .	5
2. ISO Protocol Reference Model	8
3. Single-Server Queue.	12
4. Multiple Queue Model of a Computer Communication .	14
5. Kleinrock's Star Network Example	18
6. Message Delay for Star Network Example	19
7. Kleinrock's Fully-Connected Network Example. . . .	20
8. Message Delay for Fully-Connected Network Example.	21
9. Message Flow in a Computer Communication Network .	27
10. Modular Structure of the Simulation Program. . . .	28
11. Baseline SLAM Simulation Network Diagram	31
12. Baseline SLAM Simulation Compared with Analytic. .	35
13. Fully-Connected SLAM Network Diagram	37
14. Fully-Connected SLAM Simulation Compared	38
15. Message Generation Module Segment	40
16. Node Processing Module Segment	42
17. Node-to-Node Transfer Module Segment	45
18. Statistics Collection Module Segment	47
19. External Effects Module Segment.	48
20. Simulation Network Topologies.	51
21. Message Delay for Star and Mesh Networks	54
22. Message Delay for Network Configurations	55

23.	Message Delay Compared with Network Cost	58
24.	Throughput Compared with Network Cost.	58
25.	Message Delay for Precedence Levels.	61
26.	Message Throughput for Response Traffic.	62
27.	Message Delay for Response Traffic	63
28.	Message Delay for Link Error Rates	66
29.	Rejected Messages Compared with Queue Capacity . .	68
30.	SLAM Network for a Single-Server Queue	76

LIST OF TABLES

1. Traffic Matrix.	15
2. Traffic Matrix Example.	18
3. Message Attribute Definitions	41
4. Message Response Traffic Matrix	59
5. SLAM Definitions for Single-Server Queue.	76
6. Comparison of SLAM Simulation with Analytic	79
7. List of User Specified Global Variables	81

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation for the assistance, encouragement and guidance provided by Dr. Dana B. Rogers throughout my graduate studies at the University of Dayton.

The computer resources for this study were provided by the US Air Force through the courtesy of the Air Force Institute of Technology, Wright-Patterson AFB, Ohio. The funding for this study was provided by the US Army.

Special thanks for the opportunity to pursue my graduate education are due Colonel Roger V. Sheffield, US Army Signal Corps. Skill, Endurance, Spirit!

CHAPTER I

INTRODUCTION

Motivation

The technological advances in the electronics industry during the past decade have spawned enormous growth in the computer and data communication fields. Capabilities formerly available only at large and expensive fixed computing centers are widely available due to two general trends--packaging of greater capabilities into smaller affordable minicomputers, and distributing the capabilities of fixed computing centers via data communications. The result has been the growth of integrated communication and computing networks. Some examples serve to illustrate this growth.

Remote public banking terminals and electronic funds transfer are now commonplace in the finance industry. Retail point-of-sale systems use intelligent terminals as cash registers to collect information on cash flow and inventory status. Large information banks with remote access are used by law-enforcement, medical, insurance, transportation and government agencies. Mobile computer terminals used by police, fire and medical personnel extend

computer systems to locations where services are rendered. Tactical military computer communication networks support mobile command and control systems.

All too often computer networks evolve in response to a current need without regard for future requirements. As the network grows in size, the complexity increases quickly beyond simple understanding. Unfortunately, sophisticated systems have been designed without a complete understanding of future workloads.[1] Sometimes computer networks are planned assuming that existing communication channels are adequate. This attractive assumption is made by computer network designers who have little or no control over communication resources. Adequate tools are needed to evaluate network performance, to determine if performance will meet objectives and to understand how performance can be improved.

A computer communication network design can be analyzed using a mathematical analytic model, a computer simulation model or by constructing a working model. All three approaches have been used. Unfortunately, present analytic models fail to capture all the characteristics of a mature network design. Work continues on developing useful analytic models.[4,17,21,26] The Department of Defense Advanced Research Projects Agency's ARPANET network [28] and the University of Hawaii's ALOHA network [18] are two examples of working research networks. These networks

and others provide a testbed for experimenting with specific design questions.

Computer simulation models are able to incorporate greater detail than analytic models. Simulation models cost less to implement than working models. Using a simulation model, a designer has the flexibility to vary the configuration and processing details of a network design.

Purpose

The purpose of this research is to develop and implement a generalized simulation model to evaluate computer communication networks by estimating network performance. The resulting simulation allows reasonably realistic and quantitative performance comparisons of network design alternatives.

In chapter II a classical analytic network model is presented. The analytic model is used to validate a baseline simulation model, forming the basis for a generalized full simulation model. Chapter III describes the baseline simulation, the validation results and the full simulation model. A hypothetical network is simulated in chapter IV to demonstrate the features of the full simulation. The final chapter summarizes the results of this research.

CHAPTER II

A CLASSICAL ANALYTIC NETWORK MODEL

Background

The Network

A computer communication network consists of a collection of nodes connected by data communication links. Each node contains computing resources which act on messages flowing in the network. The nodal structures of a computer communication network include input/output terminals, host computers and switching computers. The data communication links are two-way communication channels through which the messages pass. Figure 1, which is derived from Davies [6], shows the structure of a computer communication network partitioned into two separate subnetworks: the communication subnetwork and the user-resource subnetwork. The partitioning into two subnetworks is conceptual and may not represent physical boundaries for a specific computer communication network.

The user-resource subnetwork host computers provide the processing services and file storage for users of a computer facility and interface the user into the communication subnetwork. The communication subnetwork

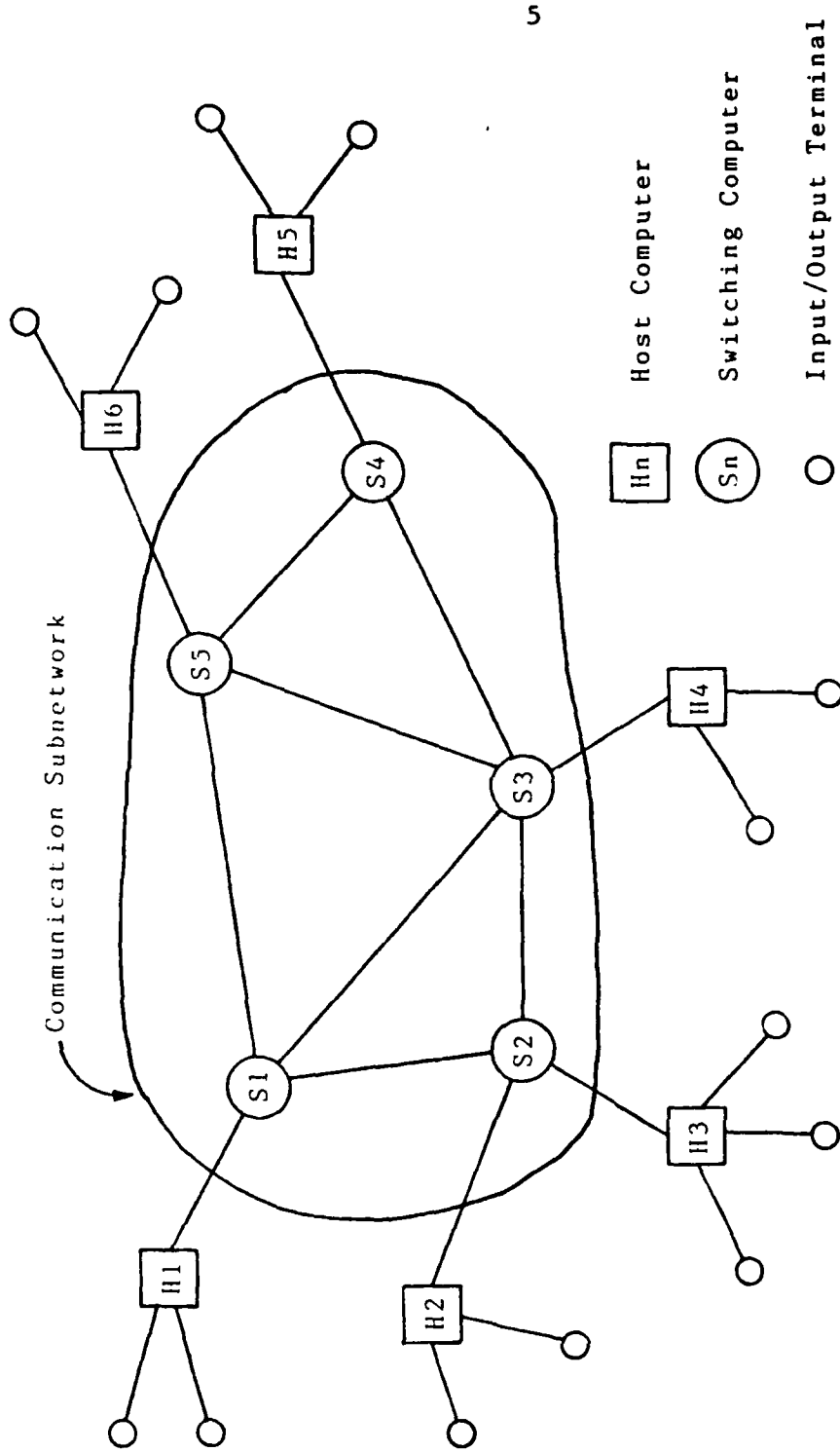


Figure 1: Structure of a computer communication network.

switching computers are responsible for establishing a path from source node to destination node for each message. This is accomplished by either circuit switching, message switching or packet switching.[15]

In circuit switching a complete path through the network is established from source to destination. After the path is established the message is transmitted. Circuit switching is similiar to the method used for a common telephone system.

In message switching the switching computers use a store-and-forward method of relaying messages. That is, messages are completely received at one node before they begin transmission to the next node. If the outgoing link from a node is busy, the message waits in a queue until the link becomes free. A message switched network was assumed in this study as it is well suited for computer data flow.[15]

Packet switching is basically the same as message switching except that messages are divided into packets, or smaller messages, before transmission. The individual packets are relayed through the network using the same store-and-forward method as message switching and are assembled at the destination to form the original message.

The activities of the communication subnetwork are usually transparent to the user. The activities include routing, acknowledging, detecting and correcting errors,

and scheduling messages for transmission. The emphasis of this study is the performance evaluation and design of the communication subnetwork. Unless otherwise stated, the term computer communication network in this paper refers to the communication subnetwork.

The messages flowing in the computer communication network are described by their source, destination, creation time, length and precedence. Messages may also contain additional information such as a type identification, serial number and requests for special facilities. Standards defining message architectures have been established by the International Telecommunications Union (ITU) and the International Standards Organization (ISO).[18] The message structure used in this study is compatible with ITU standards. In practice, each computer communication network uses a unique message structure.

The operating rules, or protocol, for the entire computer communication network are a description of the decision processes and conventions implemented within the network. Network protocol is usually organized in a series of layers or levels. One widely accepted protocol structure is the ISO reference model. This model, shown in figure 2 which is derived from Tannenbaum [28], has 7 layers. Layers 1, 2 and 3 are contained within the communication subnetwork. The physical layer is concerned with the mechanical, electrical and procedural aspects of

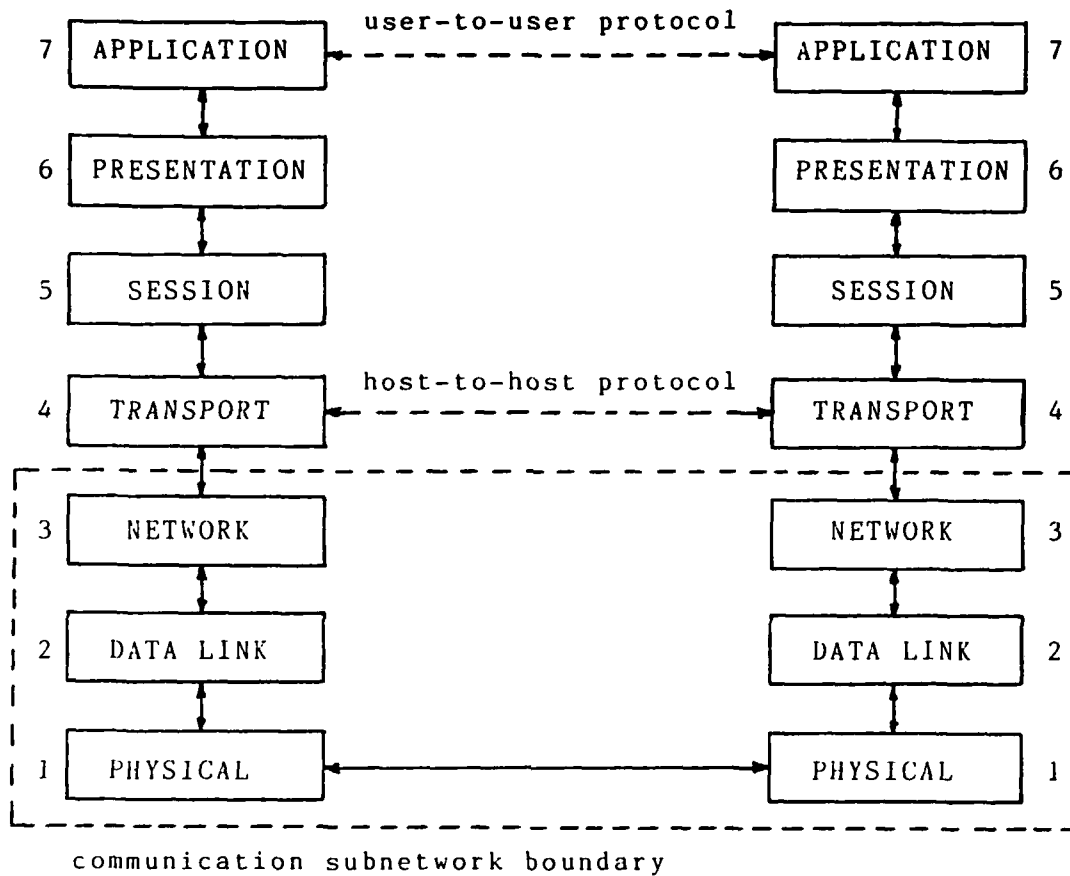


Figure 2: ISO protocol reference model.

communications equipment. Layers 2 and 3 describe how the communication subnetwork will function. These two layers are the principle focus of the computer communication simulation model implemented in this study. The structure of the network and data link protocol layers contain

parameters which directly influence overall computer communication network performance.

Network Performance Measurement

The performance of a computer communication network is usually measured in terms of message delay, throughput, cost and reliability.[15] Other parameters have also been explored such as a measure of active resources and a concept of fairness.[29,30]

Message delay is a measure of the time required for a message to travel from source to destination. Interactive users usually have short messages and are primarily interested in the amount of system delay. Throughput is a measure of the amount of information per unit time which can be passed through a network. Users transmitting long messages or files are interested in throughput performance. The total cost for construction and operation of a network can be allocated among the communication links as a function of link capacity. The function can contain both fixed and variable components. Linear, logarithmic and power-law cost functions have been investigated.[15] Considering the present intricate and changing tariff schedules for public communications, a true cost method yields better cost estimates for an actual network.

Network reliability is the ability of a network to continue to function in an acceptable manner after the loss

of one or more nodes or links. The number of nodes or links which may fail while still maintaining an acceptable level of network performance depends on the application.

Optimizing Network Performance

The design and analysis of a computer communication network involve many variables. The most significant design variables are topology, link capacity, node capacity, link protocol, routing procedure, precedence discipline and flow control.[5,6,12,15,18,27,28] Generally, the location of the host computers, the input traffic characteristics and the implementation costs are known. Optimum network performance is defined as achieving an acceptable level of message delay or throughput at minimum cost.

The design process is often partitioned into four optimization problems which differ only in the choice of design variables.[12,15,28] The four design cases are 1) link capacity assignment, 2) flow assignment, 3) capacity and flow assignment, and 4) topology, capacity and flow assignment. Since the large number of interrelated variables precludes an exhaustive search for the optimum design, a heuristic searching technique is used.

The search begins by selecting a starting topology. Flow and capacity assignments are made and the network performance and cost are determined. Slight modifications

are made to the network, and the network performance and cost are checked for improvement. The process is repeated until an acceptable network is found. The simulation model developed and implemented in this study is used to determine network performance at each iteration of the search.

A Classical Analytic Network Model

Kleinrock has developed an analytic model of a store-and-forward computer communication network which has been widely accepted.[12,15] He models a computer communication network as a system of single-server queues. This modeling approach has been applied to a variety of network situations using both analytic and simulation models.[8,9,19,25,30] In this section the analytic model is discussed, and the results for specific numerical examples using the model are reviewed.[12] These analytic results are applied in chapter III to validate the accuracy of a baseline computer simulation model.

Single-Server Queue Model

One unidirectional path through a switching computer at a network node is represented schematically in figure 3. Incoming messages are stored in a buffer queue until the outgoing transmission path, or server, is available. The total delay experienced by a message at the node consists primarily of the waiting time in the queue plus the

transmission time on the outgoing link. The most significant measure of steady-state performance for the node is given by the average delay experienced by a message passing through the node. Subject to certain assumptions, an analytic expression for average delay is well known.[15] The assumptions are introduced to simplify the mathematical model; thus, the model is an approximation of the characteristics of a real system.[12]

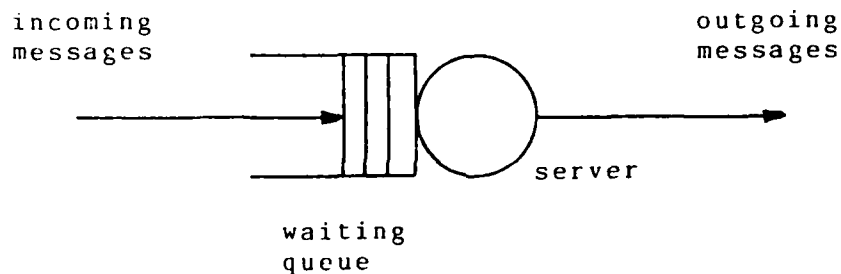


Figure 3: Single-Server queue.

The arrival of messages at the node is assumed to be a Poisson process with an average rate for message arrivals of λ messages/second. Message lengths have a negative exponential distribution with a mean of $1/\mu$ bits. The transmission speed of the outgoing link is C bits/second. For a message length of b bits the transmission time is b/C seconds. Messages are transmitted in order of arrival; that is, first-in-first-out (FIFO). Storage space for the queue is assumed infinite so that no message arrivals are

rejected. The values of λ and μ , which characterize the random distributions for message arrivals and message lengths, are constants. A fundamental result of queueing theory gives the average waiting time in the queue as

$$r = \frac{\rho}{\mu C(1-\rho)}$$

where $\rho = \lambda/\mu C$ is the utilization factor for the link. The total average delay in the system is the sum of the average waiting time and the average transmission time:

$$T = r + \frac{1}{\mu C} = \frac{1}{\mu C - \lambda}$$

Multiple Queue Model

Using the results for a single-server queue, Kleinrock extended the concept to an M-link N-node network. The computer communication network is modeled as a network of interconnected single-server queues. Figure 4 represents a segment of a network showing how the queues are interconnected. The input to a queue may come from more than one source. In constructing an analytic model for this network Kleinrock made some additional assumptions beyond those stated previously.

The M-links are assumed to be error-free, computer processing time at each node is assumed negligible, and electrical propagation time between nodes is ignored. The

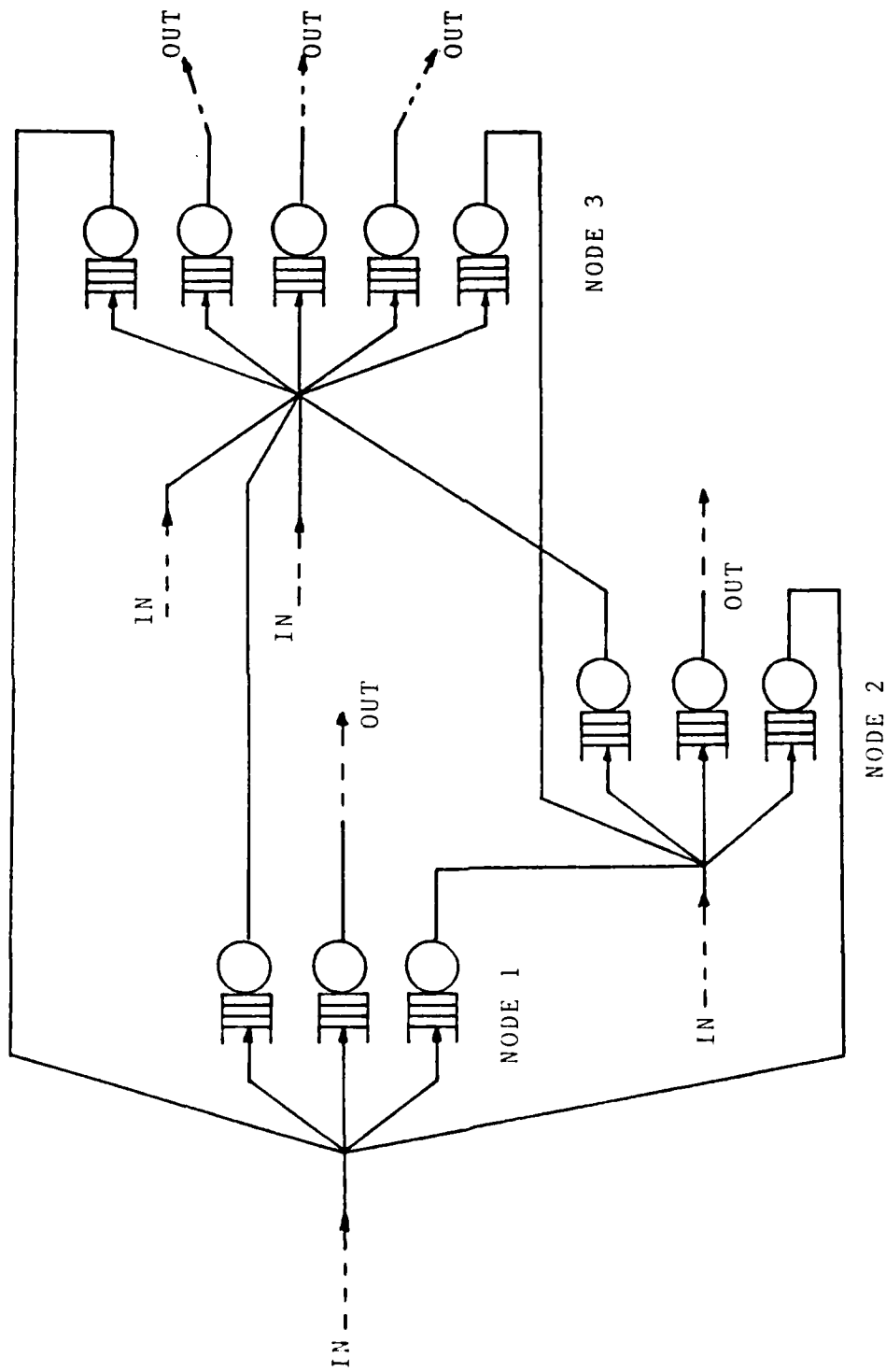


Figure 4: Multiple queue model of a computer communication network.

average message arrival rates are given by a traffic matrix in the form of table 1. The G_{jk} entries are the traffic intensities from source node j to destination node k given in messages/second. The total external traffic entering the net is

$$G = \sum_{j=1}^N \sum_{k=1}^N G_{jk}$$

TABLE 1
TRAFFIC MATRIX

node	1	2	3	...	k
1	-	12	13	...	1k
2	21	-	23	...	2k
3	31	32	-	...	3k
.	.	.	.		.
.	.	.	.		.
.	.	.	.		.
j	j1	j2	j3	...	-

Message routing through the network is fixed. The average number of messages which travel over the i th link of capacity C_i is λ_i , and the average delay for the i th link is $T_i = 1/(\mu C_i - \lambda_i)$. To avoid severe mathematical difficulties, Kleinrock assumes that a new message length

is chosen from a random distribution at each node. This independence assumption is contrary to reality, but it does produce a simple model which gives reliable results.[12] Based on these assumptions, the average message delay for a network is

$$T = \sum_{i=1}^M \frac{\lambda_i}{G} T_i \quad (1)$$

This expression for T , message delay averaged over the entire network, is a major performance measure for the network. The average message delay has a sharp threshold behavior as the rate of traffic input to the network is increased. The delay will rise sharply toward infinity whenever any single link in the network becomes saturated with messages. This occurs when message arrivals exceed the outgoing link transmission capacity.

Kleinrock also proposed a second analytic model in his original work.[12] In this model an average path length $\bar{n}$ is calculated based on an assumed fixed routing procedure. The average message delay T is

$$T = \frac{\bar{n} \left[\sum_{i=1}^N \sqrt{\lambda_i / \lambda} \right]^2}{\mu C (1 - \bar{n} \rho)} \quad (2)$$

This model also exhibits a sharp threshold behavior as the rate of traffic input to the network is increased. It does not, however, rise toward infinity when any single link becomes saturated. The message delay for the model becomes infinite as $\bar{n}\rho$ approaches unity. This model gives a more optimistic value for message delay and conceals the point where links in the real system become saturated.

Kleinrock's Numerical Examples

In this section two numerical examples published by Kleinrock are reviewed.[12] The results of these analytic models will be used to validate the computer simulation model developed in chapter III.

Star Network

A 5-node star network has a traffic matrix as defined in table 2. The total network capacity is given as $C=38.33$ bits/second and message length is given as $1/\mu=0.1$ bits. The individual link capacity assignments are shown in figure 5. Using the analytic equations (1) and (2), the average message delay is shown in figure 6.

TABLE 2
TRAFFIC MATRIX EXAMPLE

node	messages/second				
	1	2	3	4	5
1	-	9.340	0.935	2.940	0.610
2	9.340	-	0.820	2.400	0.628
3	0.935	0.820	-	0.608	0.131
4	2.940	2.400	0.608	-	0.753
5	0.610	0.628	0.131	0.753	-

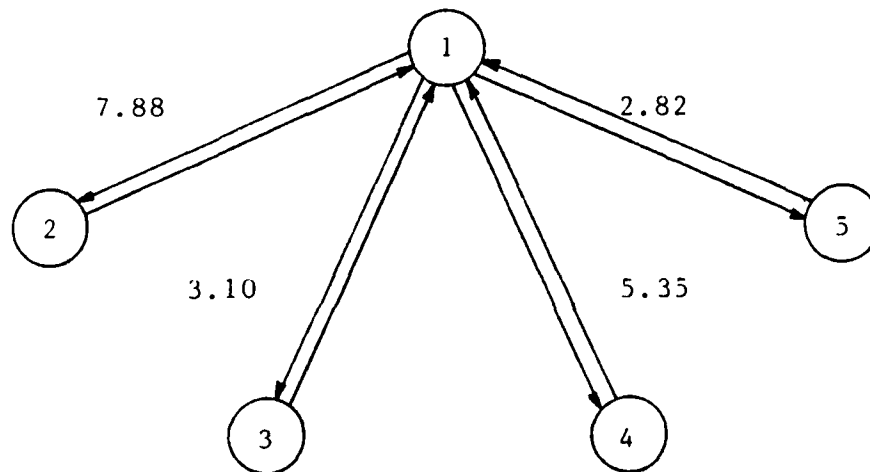


Figure 5: Kleinrock's star network example. Link capacities shown are in bits/second.

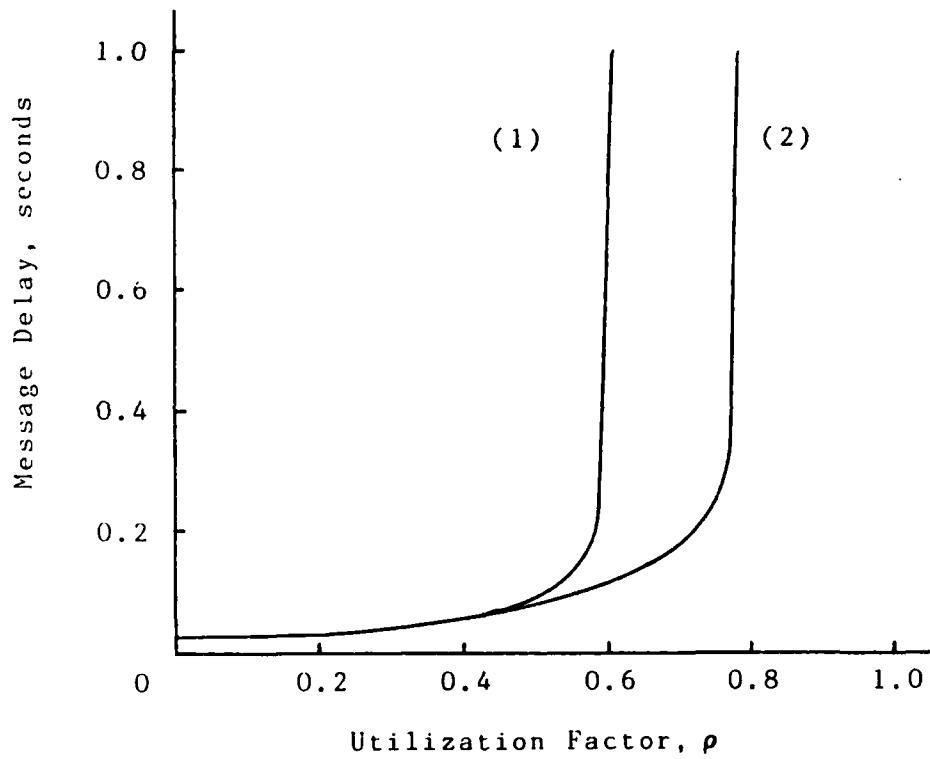


Figure 6: Message delay for star network example using analytic equations (1) and (2).

Fully-Connected Network

The 5-node fully-connected network in figure 7 has the same traffic matrix as defined in table 2. The total network capacity remains 38.33 bits/second. The average message delay for the fully-connected network calculated using analytic equations (1) and (2) is shown in figure 8.

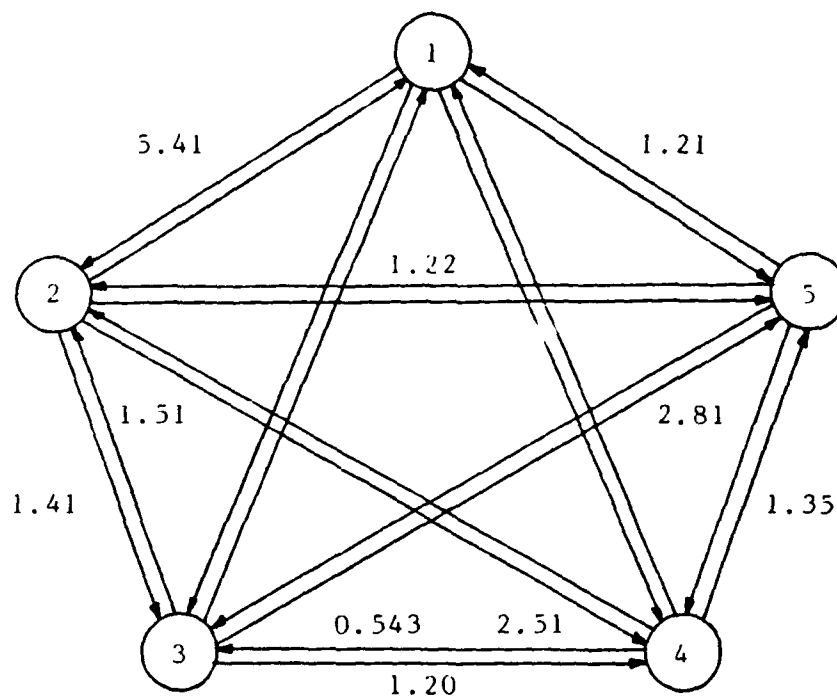


Figure 7: Kleinrock's fully-connected network example. Link capacities shown are in bits/second.

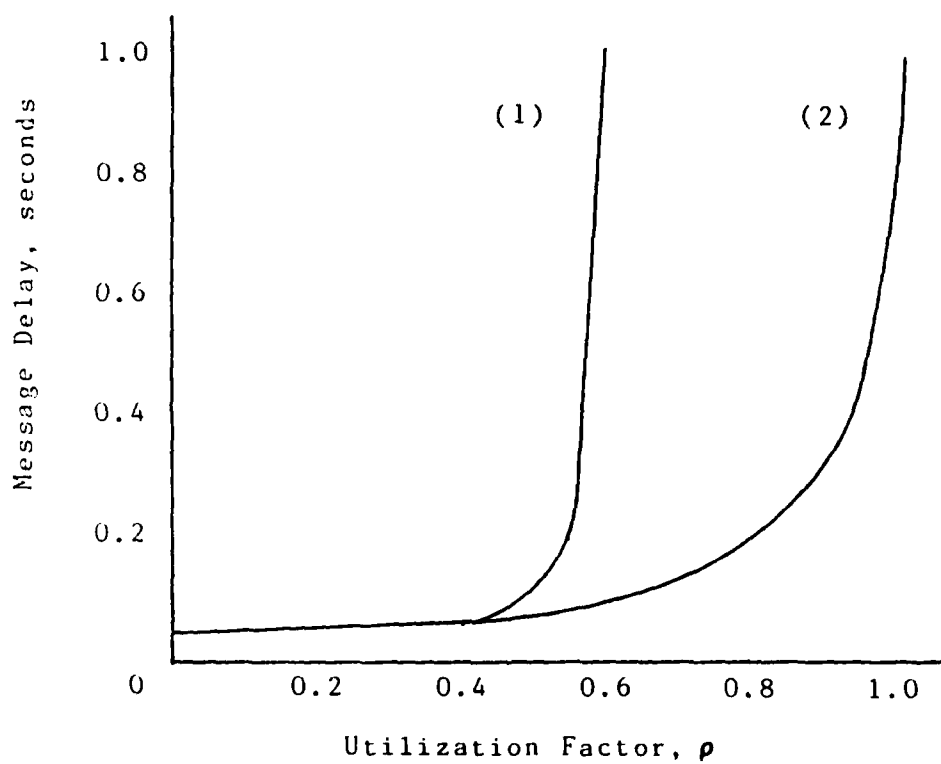


Figure 8: Message delay for fully-connected network example using analytic equations (1) and (2).

Comments on Analytic Models

Throughout the literature on analytic models of computer communication networks frequent mention is made of the intractability of mathematical solutions to these models.[5,15,19,25] Kleinrock states:

It is not hard to convince oneself that queueing theory is rather difficult and that exact results are hard to obtain; in fact, many of the interesting queueing phenomena have not yet yielded to exact analysis (and perhaps never will!). Moreover, in those simpler systems where exact results can be obtained, their form is sometimes so complex as to render them ineffectual for practical applications.[15]

While it is not reasonable to suggest forgoing analytic solutions, the urgency of solving present day computer communication network design problems drives one toward simulation models. When choosing between full scale network experimentation and simulation modeling, the latter clearly costs less to accomplish.

The two analytic models described in this chapter represent very rudimentary network designs. Adding detail to the analytic model to evaluate realistic alternatives is a formidable problem; however, adding detail to the computer simulation model is a reasonable goal. The following chapters present the details of a computer communication network simulation model for estimating average message delay and message throughput.

CHAPTER III

THE SLAM SIMULATION MODEL

Introduction

A computer simulation model is developed and implemented to investigate the performance of computer communication networks. The resulting model is a logical, mathematical representation of network activity and is used to estimate network behavior under a variety of hypothetical conditions. The simulation is a discrete-event stochastic model. It is written using a modular approach. In this chapter the simulation program development is described. Next, the modules of a baseline simulation of a computer communication network are described, and Kleinrock's 5-node network is used to validate the baseline simulation.[12] Finally, the full computer communication network simulation is presented.

The SLAM Language

Simulation Language for Alternative Modeling (SLAM) is a FORTRAN-based language distributed by Pritsker and Associates, Inc. of West Lafayette, Indiana.[24] This language was selected because its process oriented statements are suitable for modeling a computer communi-

cation network. The brief overview of the SLAM network language in this section will assist in understanding the models presented in this chapter. A detailed example of a SLAM simulation model of a single-server queue is presented in appendix A. A complete tutorial of the SLAM language is found in references [23,24].

A SLAM simulation includes control statements and a network description. The control statements initiate, modify and terminate the simulation and provide a means for selecting among options in the SLAM language. The network description is the unique portion of the program which the modeler writes to represent a real world process.

In the SLAM network description entities (messages) flow through a process (store-and-forward communication network). An entity can be assigned attribute values (message length, origin, destination) which distinguish it from other entities. Groupings of entities are called files. A process consists of a collection of actions (transmit, receive, check for errors) and structures (memory, channels) which correspond to the operation and configuration of the communication network being modeled.

Using a set of SLAM graphic network symbols, the communication network model is constructed. The SLAM graphic symbols form a shorthand notation for describing a model; and the simulation program is written by translating the graphic symbols into SLAM language statements. The

graphic symbols are an effective means of communicating the operation of a model and of documenting the simulation. Plain language comments inserted into the listing together with a SLAM network diagram form a complete documentation package for the simulation. There are 23 SLAM network statements available for constructing models. This makes SLAM an effective simulation language.

The output from a SLAM simulation can include an echo report, trace report, error messages or SLAM summary report. The first three reports are useful for debugging and validation of the model. The SLAM summary report displays the statistical results of the simulation.

A SLAM summary report is printed at the end of each simulation run or at intervals selected by control statements. The report includes statistics on files, activities or variables within a model. The SLAM summary report is the primary output of results from the simulation.

Simulation Program Organization

A top-down modular approach is used to construct the computer communication network simulation model. Dividing the network into functional modules facilitates writing the simulation in a logical and controlled manner. This method assists the validation process since each module can be checked for correctness individually. The modular

simulation allows changing one portion of the model without affecting others, thus permitting flexible experimentation and use of the simulation. With a top-down modular approach simulation details can be added or omitted according to the degree of detail desired in the simulation.

Message flow in a computer communication network simulation is described as shown in figure 9. Messages are created, queued, and transmitted, and performance statistics are gathered. These processes are categorized into four major simulation module types. Each module type is composed of a number of activities which correspond to the real world actions being simulated. A fifth module type, external effects, is included to provide additional flexibility in the simulation model. The activities included in each simulation module are shown in figure 10.

The SLAM simulation model is developed in two phases. First, a simulation model of a computer communication network was written using the same network description and assumptions as the Kleinrock star network analytic model presented in chapter II. This is done to validate a baseline SLAM simulation. The validated baseline simulation is used as a basis for all later simulations.

In the second phase additional details are added in a step-wise fashion to the baseline simulation, expanding to a full simulation model which includes all the details

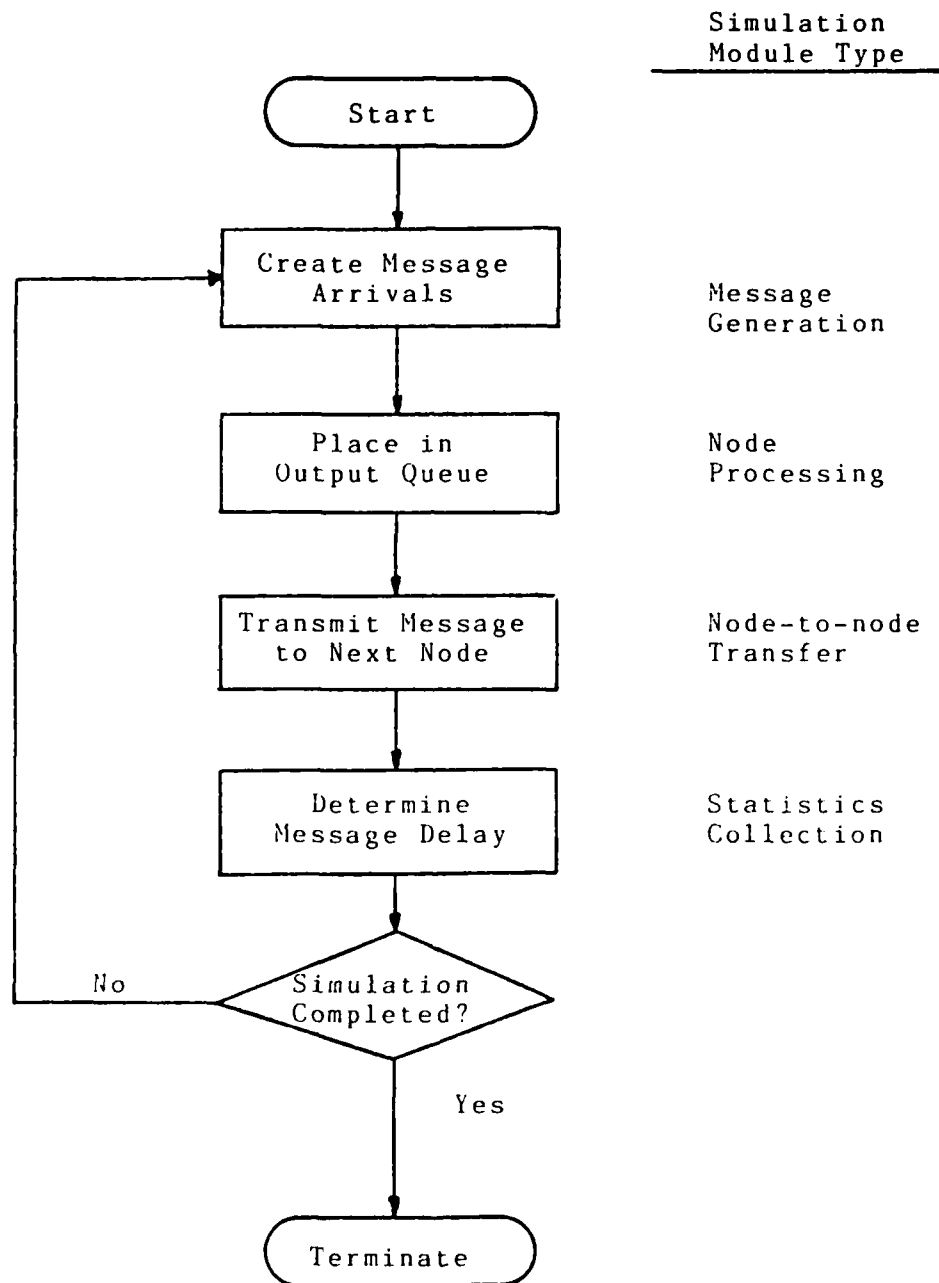


Figure 9: Message flow in a computer communication network simulation.

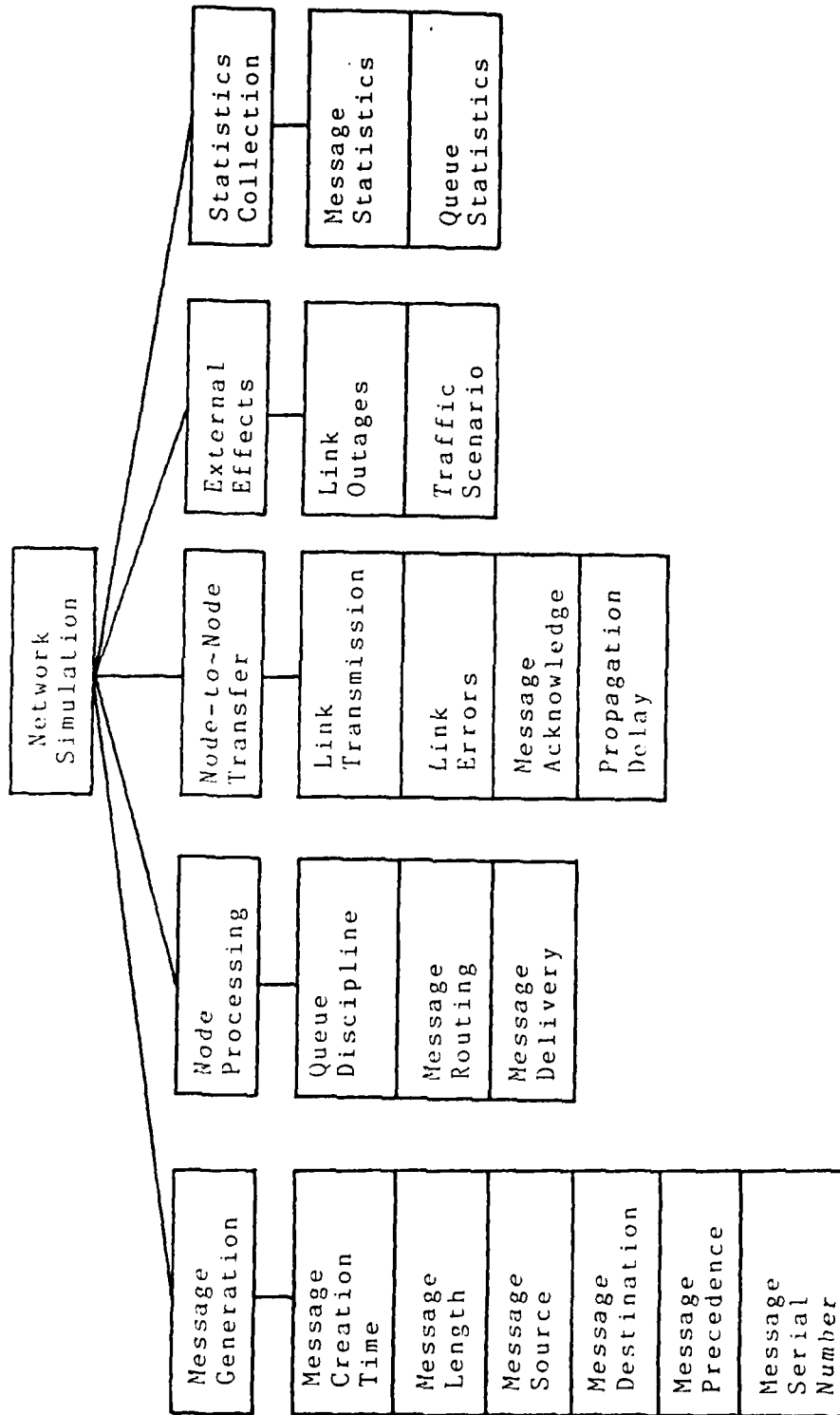


Figure 10: Modular structure of the simulation program.

shown in figure 10. The simulation is tested thoroughly at each step to insure that it produced consistent results. This procedure is used to insure the validity of the full simulation model. During the second phase, the full simulation model is used to analyze a hypothetical computer communication model.

Baseline Model

The baseline model for the star network is shown in figure 11 using SLAM network notation. The baseline model is used as the foundation to build and validate the full simulation model. The simulation is organized into four module types. SLAM language commands and labels are capitalized in this paper to permit easy reference to the network diagrams and program listings.

Message Generation Module

Message input to the network is specified using a traffic matrix as previously shown in table 1. Each G_{jk} entry in the matrix corresponds to a CREATE node having the time between creations selected from an exponential random distribution with a mean of $1/G_{jk}$ seconds. The traffic intensities were implemented as global variables $XX(1)$ thru $XX(10)$. Message attribute 1 contains the message creation time.

An ASSIGN node is used to place the destination node identifier into attribute 2. Traffic generated from

node 1, the central node in the star network, is placed in the appropriate outgoing queue according to the destination. Traffic generated from nodes 2 thru 5 is always transmitted to node 1 regardless of destination.

Node Processing Module

Node processing occurs only at the central node in this model. All other nodes can only send messages to the central node. At the central node each message destination, as indicated by attribute 2, is checked to determine message disposition. Messages terminating at the central node are sent to the statistics collection module represented by node TOT. All messages not terminating at node N1 are placed into the appropriate outgoing queue. Queue discipline is FIFO.

Node-to-Node Transfer Module

Transmitting a message is accomplished using a service ACTIVITY. The time for transmission on each link is $1/\mu C_i$. This value is used as the duration for each activity connecting the nodes in the network.

Statistics Collection Module

The time interval between message creation and message arrival at its destination is the message delay. This interval is determined at COLCT node TOT and reported in the SLAM summary report.

Baseline Model Validation

Repeated runs of the baseline model with varied traffic intensities are made to obtain a plot of average message delay against network utilization factor. The baseline SLAM simulation results are compared to

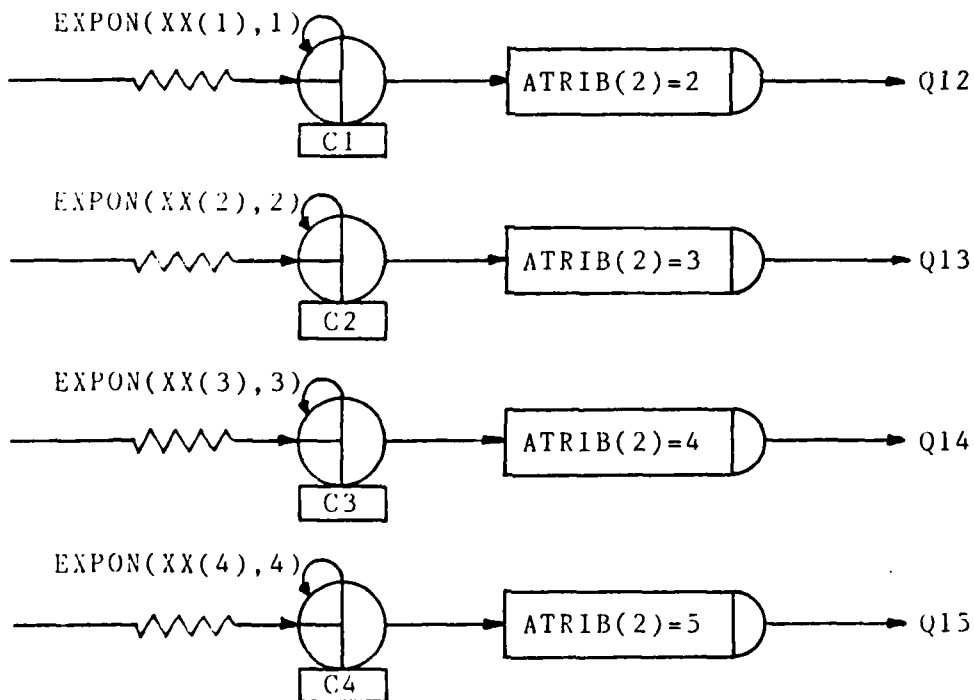


Figure 11: Baseline SLAM simulation network diagram,
a) message generation module, node 1.

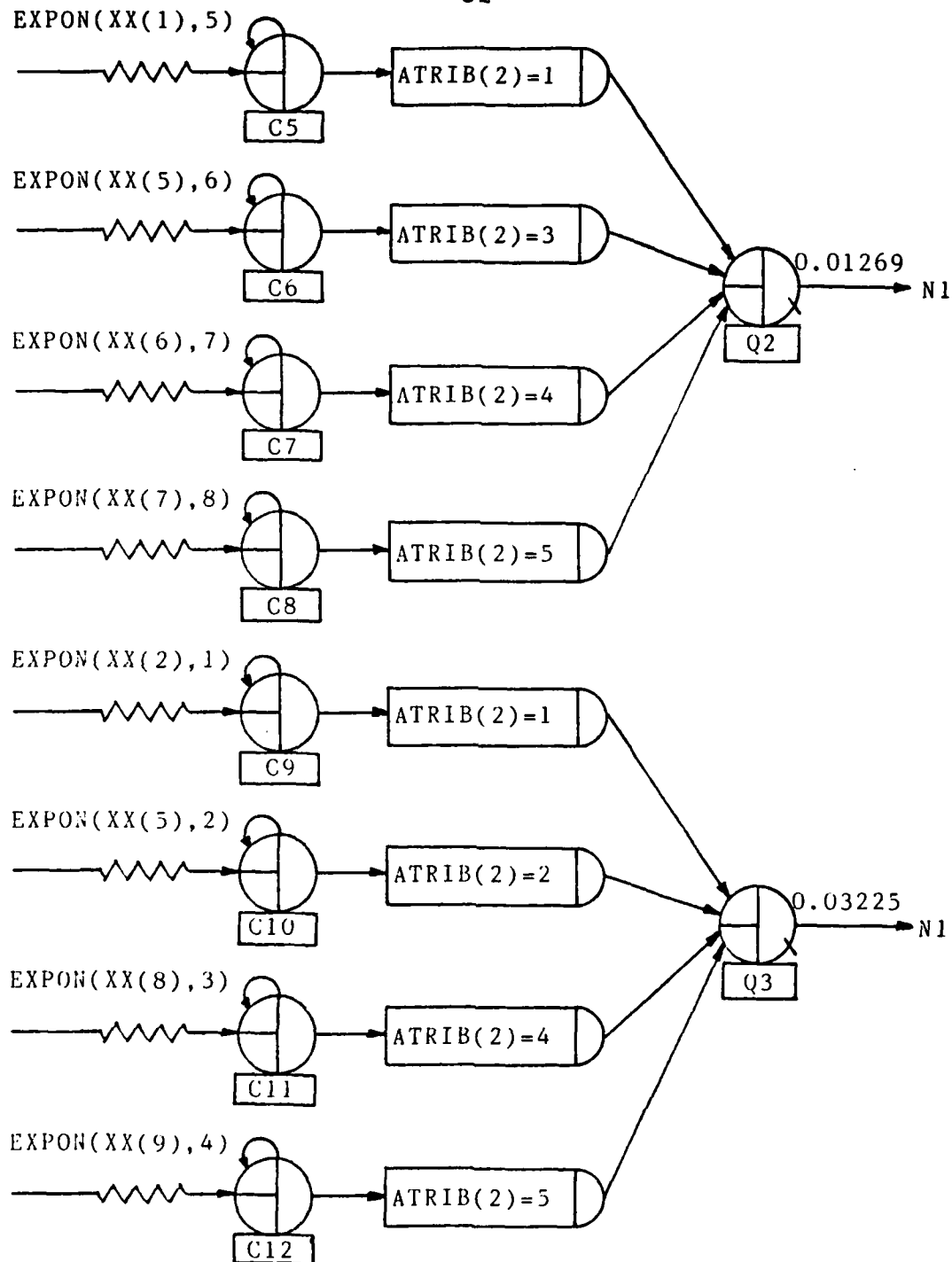


Figure 11 (con't): Baseline Slam simulation network diagram, b) message generation module and node transmission module for nodes 2 and 3.

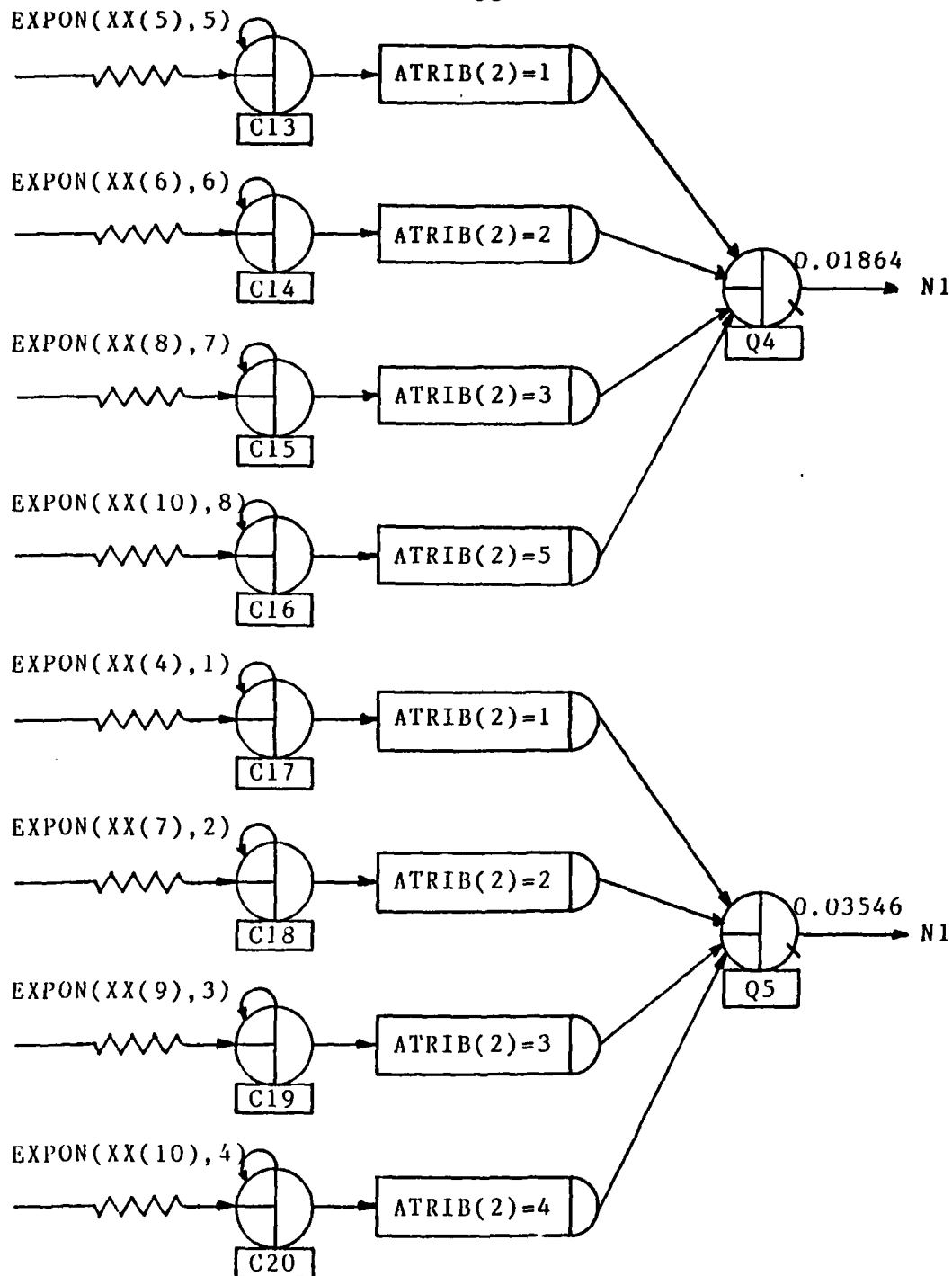


Figure 11 (con't): Baseline SLAM simulation network diagram, c) message generation module and node transmission module for nodes 4 and 5.

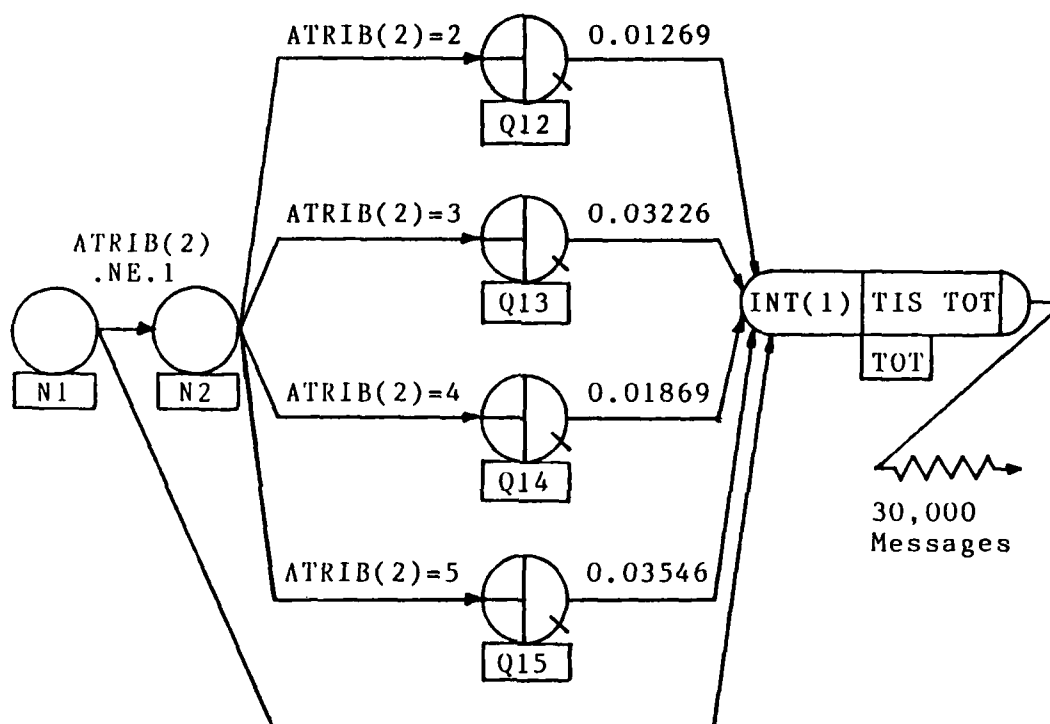


Figure 11 (con't): Baseline SLAM simulation network diagram, d) node processing, node-to-node transfer, and statistics collection modules.

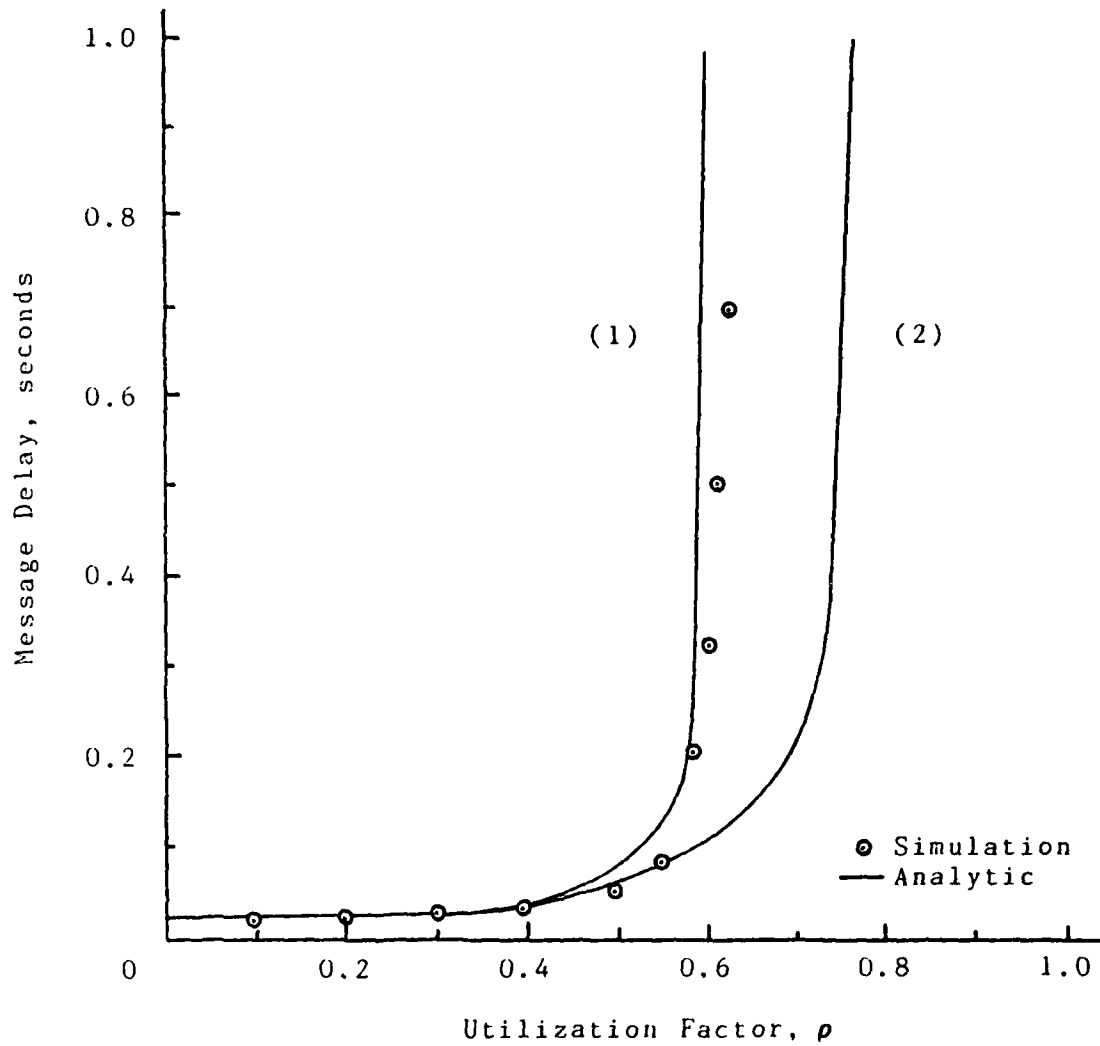


Figure 12: Baseline SLAM simulation compared with analytic equations (1) and (2).

Kleinrock's analytic model in figure 12. A sharp threshold behavior is strongly evident.

The SLAM trace report is used to verify the logical correctness of the SLAM program. The flow of messages through the simulation is checked by hand against the trace report output and verified correct. This provides a high degree of confidence that the simulation reproduces the desired model.

To verify that the simulation has reached a steady-state condition prior to computing the performance statistics, the following procedure is used. The simulation is run at 6 different rates of traffic input to transmit 20,000 messages. During each run a SLAM summary report is printed at intervals of 2,000 messages. The statistical arrays are cleared after each report. This provided a series of snapshots in time from the simulation. At each traffic input rate the message delays for each snapshot are approximately equal indicating an equilibrium condition. Queue lengths in the summary report are essentially constant for each snapshot. If the simulation is not at steady-state, the average message delay and the queue lengths increase with time.

Baseline Model Expansion

The baseline SLAM simulation of the star network is expanded to model a fully-connected network. This is done

by replacing all node processing modules with new modules similar to figure 3. This module implements the fully-connected routing scheme and provides an outgoing queue for each link in the network.

The traffic intensities are varied to obtain a plot of average message delay against network utilization factor. The results of this simulation are compared to Kleinrock's analytic models in figure 14. The simulation results compare well to analytic model equation (1).

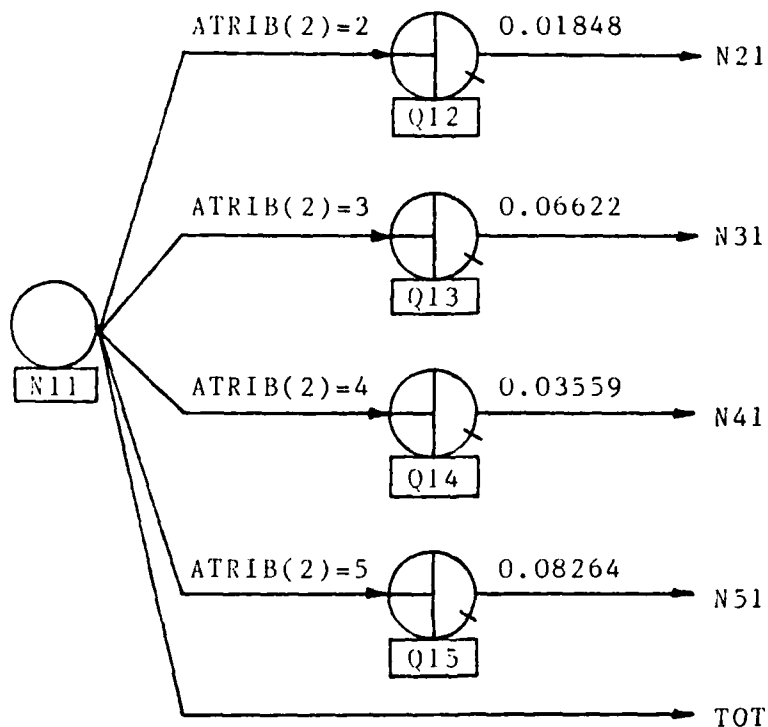


Figure 13: Fully-Connected SLAM network diagram, node processing and node-to-node transfer modules.

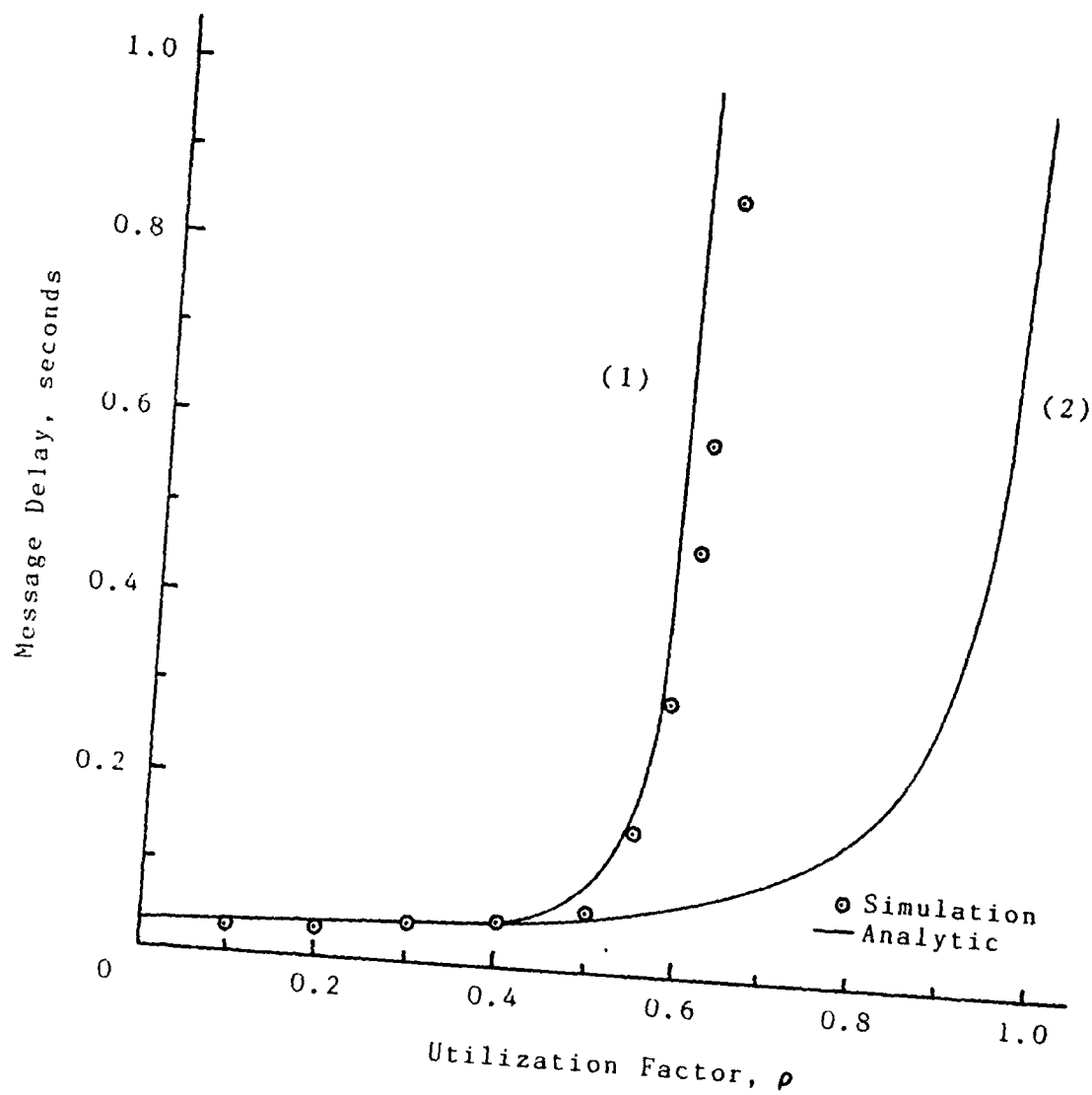


Figure 14: Fully-Connected SLAM simulation compared with analytic equations (1) and (2).

Comparing the simulation results for the star network with the fully-connected network shows that adding 6 additional links to the star network does not make a significant improvement in overall performance. The message traffic and link capacity between nodes 1 and 2 are the primary factors affecting average message delay. The simulation results show that this link is the first to saturate with increasing message flow. The fixed routing scheme does not take any advantage of the additional paths provided by the fully-connected network.

Full Simulation Model

The validated baseline simulation is expanded to include all the activities shown in figure 10. Each module is described in this section. A complete program listing and sample output is in appendix B.

Message Generation Module

All messages in the network contain the attributes listed in table 3. For each entry in the traffic matrix a module segment like that shown in figure 15 is required. This module segment randomly generates messages at the specified traffic intensity, marks all message attributes according to the probability distributions defined in the simulation control statements and places the messages in the appropriate outgoing queue. In an N-node network there are $N(N-1)$ of these message generation module segments.

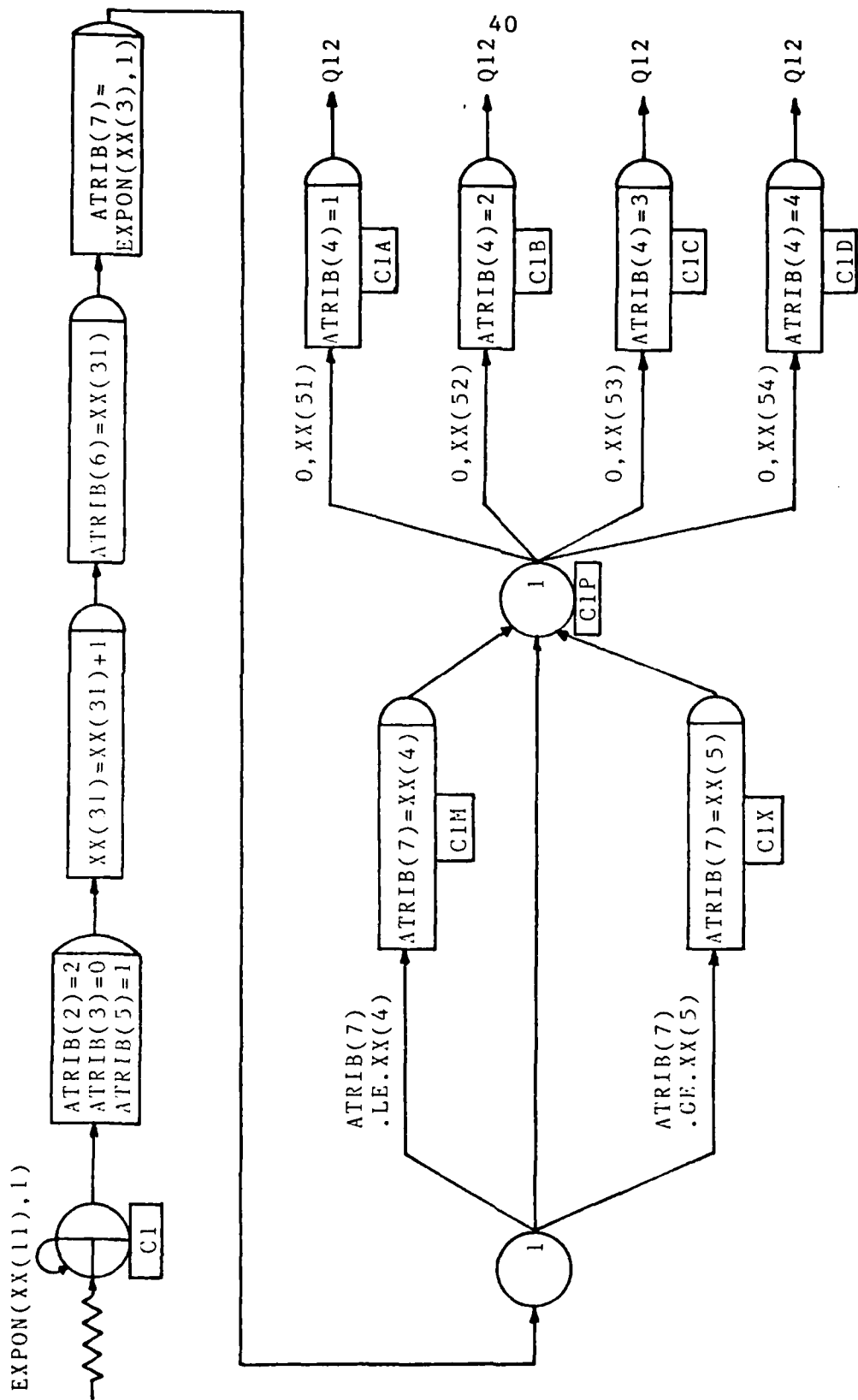


Figure 15: Message generation module segment for node 1 to node 2 traffic.

TABLE 3

MESSAGE ATTRIBUTE DEFINITIONS

Attribute	Definition	Units
1	Creation time	seconds
2	Destination node	integer
3	Type	integer
4	Precedence	integer
5	Origin node	integer
6	Serial Number	integer
7	Length	bits

The message type attribute is used to distinguish original messages from response messages. A 4-level precedence scheme is used. The serial number attribute is used to provide each message with a unique identification marking.

Node Processing Module

At each node in the computer communication network a switching computer performs node processing tasks. In the simulation these activities are modeled as shown in figure 16. One module segment like figure 16 is required for each network node.

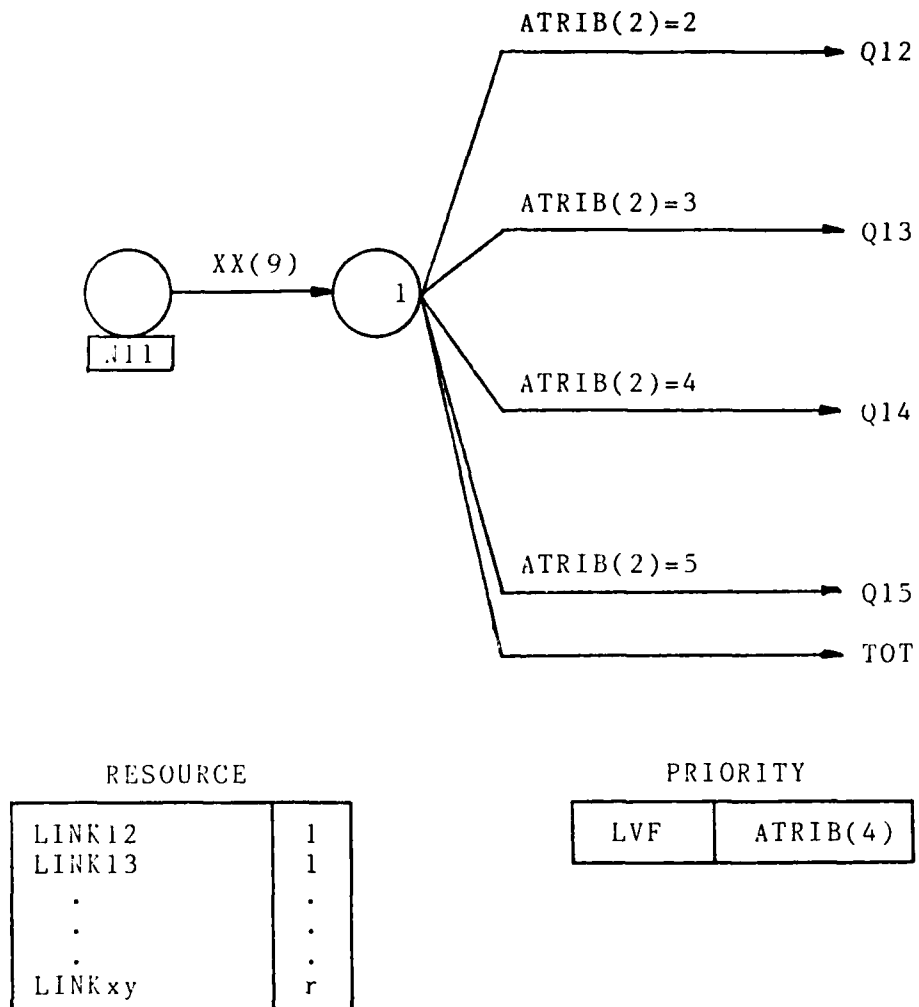


Figure 16: Node processing module segment for node 1.

Not all of the node processing details can be included in this module due to the characteristics of the SLAM language. Control statements outside of the SLAM network description as well as an initialization module are required to accomplish the node processing activities.

A decision is made concerning the degree of detail of the node processing tasks to be modeled. Kleinrock assumed that node processing time is negligible.[12] This assumption is only valid for long messages or low link capacities. As link capacity increases and as node processing tasks increase in complexity, the computer processing time at a node cannot be ignored. No switching computer hardware specifications are assumed in this model; thus, the node processing overhead time is lumped into a single variable at each node.

All messages arriving to a node are sorted by destination. Messages that have arrived at their destination are sent to the statistics collection module.

Queue discipline is implemented using a SLAM PRIORITY control statement. The queue discipline is highest-precedence-first based on message attribute 4.

Fixed message routing is implemented in the simulation using node labels at the time the simulation is written. Messages are placed in an outgoing queue according to a routing table that depends on the network topology.

The network topology is defined at the beginning of the network description using RESOURCE statements. One RESOURCE statement is required for each communication link in the network, and it is initially defined with a value of zero. At the start of the simulation each link is ALTERed to the required network connectivity based on global variables indicating the number of lines per link.

Node-to-Node Transfer Module

Node-to-node transfer activities include simulating link transmission time, link errors and a message acknowledgement protocol. An ACK/NAK data link protocol is simulated.[6] Figure 17 shows a node-to-node transfer module segment for link 12. One module segment is required for each link in the network.

Link transmission errors are simulated using probability branching. The error rate is specified for each link using a global variable.

Messages are queued for transmission on the outgoing link at an AWAIT node. If the queue capacity is exceeded, messages overflow to the statistics collection module. When the link is available, the highest precedence message is transmitted with or without errors as determined by the link error rate. Messages free of errors are acknowledged (ACK), and the link is released to allow the next message to be transmitted. Messages with errors are negative

acknowledged (NAK) and retransmitted until they are received at the next node without errors.

The duration of the service ACTIVITY representing link transmission time is calculated for each message using the link capacity and message length.

Statistics Collection Module

The SLAM summary report provides all the statistical output information for the simulation. A SLAM COLCT node is used to collect statistics that are related to the time a message arrives at the node or on a variable at the message arrival time. Estimates of the mean and standard deviation are computed for each COLCT node.

The average length, maximum length and the average waiting time for each queue are computed. The average utilization factor for each communication link is computed.

The statistics collection module is shown in figure 18. Message delay is determined for all messages. After sorting on precedence the message delay is determined for each precedence level. All messages which overflow the queues are sent to a separate COLCT node for each queue.

External Effects Module

The external effects module is the portion of the simulation used to model events external to the computer communication network. The external effect module included in the full simulation model is shown in figure 19. This

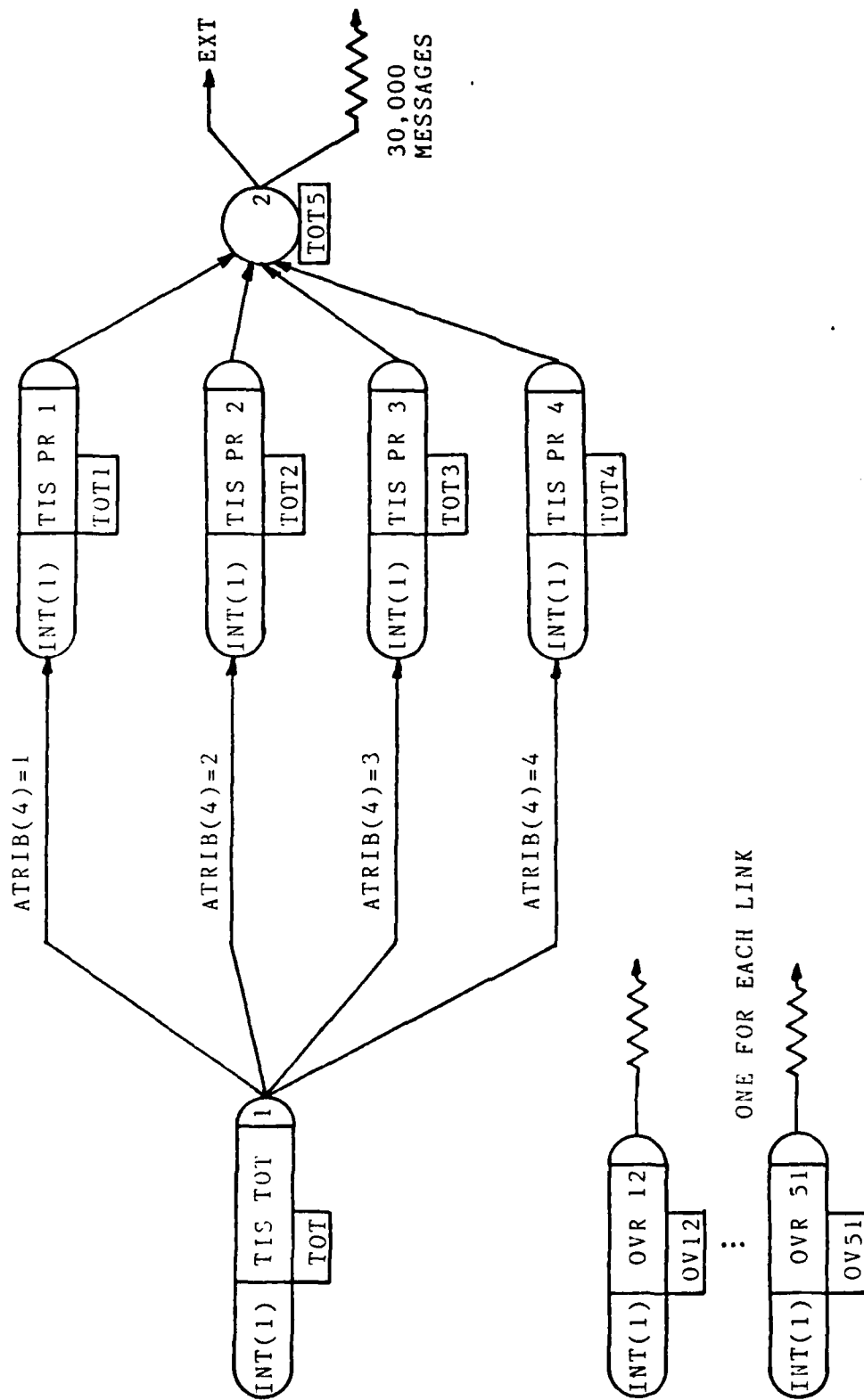


Figure 18: Statistics collection module segment.

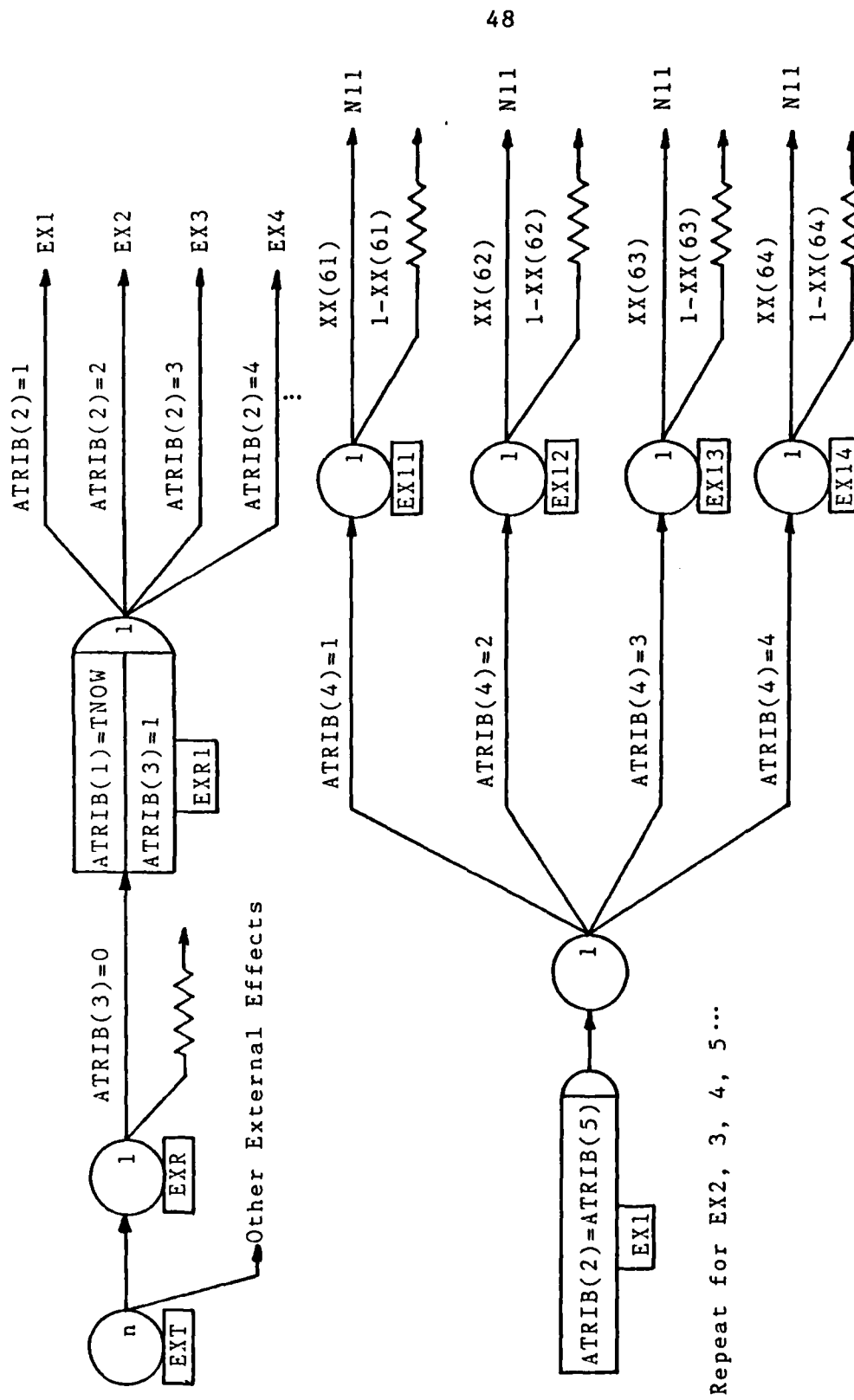


Figure 19: External effects module segment for response traffic.

module generates response messages when original messages arrive at their destination. The amount of response messages generated depends on the precedence of the arriving messages and on a response probability factor. Higher precedence messages generate more responses than lower precedence messages. Global variables are used as the response probability factors. The factors can be changed to observe the effect on network performance.

Messages are first sorted by type. Only original messages are used to construct a response message. The original message arrival time is used as the response message creation time. Messages are sorted by original destination so that the response message is sent back to the origin node. The response messages are then sorted by precedence. The response probability for each precedence level is used to determine if the response message is transmitted or discarded. The module segment in figure 19 is repeated for each node in the network.

CHAPTER IV

PERFORMANCE EVALUATION OF A NETWORK

Introduction

To apply the SLAM simulation model and demonstrate the method of optimizing network performance, a hypothetical 5-node network is assumed. First, the selection of topology and assignment of link capacities are addressed. The baseline simulation is used to determine average message delay for a series of progressively more costly network alternatives. One alternative network is selected, and the full simulation model is used to investigate network performance. Each simulation detail included in the full model is varied to demonstrate the effect on network performance.

Network Description

The minimum connectivity requirement for the hypothetical network is shown in figure 20a. Each one-way link has a capacity of 2400 bits/second, and additional capacity may be added only in increments of 2400 bits/second. Average message lengths are equivalent to one-half of a common video screen display in ASCII code or 6400 bits. A minimum message length of 100 bits and

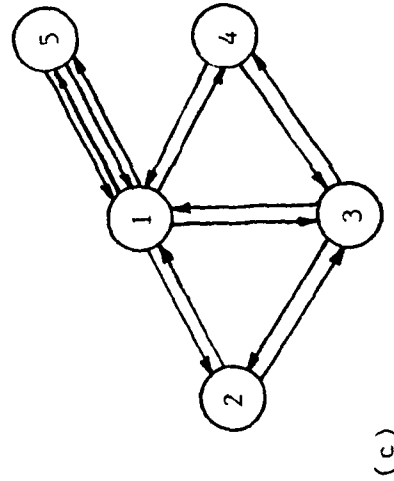
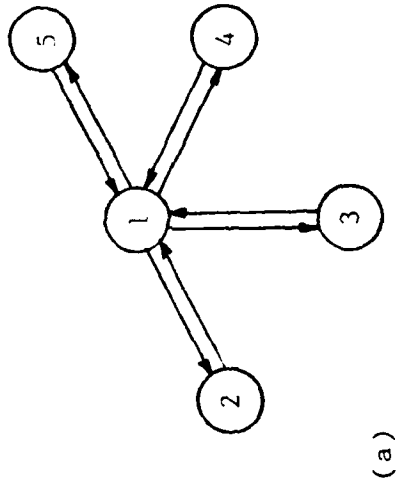
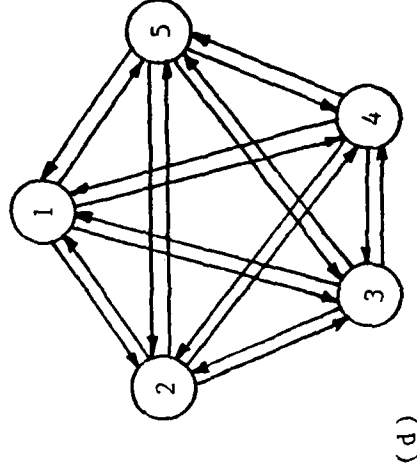
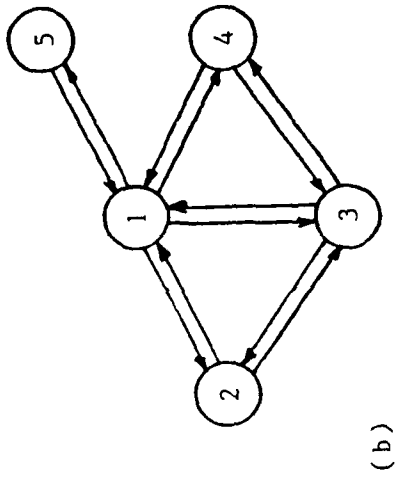


Figure 20: Simulation network topologies, a) star, b) mesh, c) mesh-plus, d) fully-connected.

maximum of 25,600 bits is imposed to confine the traffic input to realistic values. Traffic input to the network is evenly distributed over all nodes and destinations for this analysis, but the simulation can accommodate any distribution of traffic among the nodes. A 4-level precedence system can be activated. An ACK/NAK node-to-node protocol with a selectable link error rate can be activated, and response traffic may be introduced into the network.

Beginning with the star topology of figure 20a, network connectivity is increased to the mesh configuration of figure 20b, the mesh-plus configuration of figure 20c and the fully-connected network of figure 20d. The average message delay for each topology and link capacity assignment is estimated using the simulation model.

Star Network Performance

The message delay performance for the star network is plotted in figure 21. A sharp threshold behavior is evident.

The length of simulated time at low traffic input rates is approximately 11 hours, and the simulated time is approximately 1 hour at the highest rate of traffic input. The number of messages transmitted is 30,000. This is well in excess of the time required for the simulation to come to steady-state and was selected as a good compromise

between real simulation run time, computational accuracy and simulation file capacity.

Mesh Network Performance

The network topology is changed by adding two additional links between nodes 2, 3 and 4 to form the mesh network shown in figure 20b. A shortest path message routing scheme is used. The simulation results are shown in figure 21. Message delay is reduced in comparison with the star network. The value of improvement to the user is evaluated by considering the cost of adding the additional links. At low traffic intensities the improvement is negligible, but at traffic intensities above 100 messages/minute the delay is roughly halved by adding the additional links.

Mesh-Plus Network Performance

The links 1-5 and 5-1 are expected to be the primary factors limiting the mesh network performance since node 5 has only a single connection to the network. The simulation results show this to be true. At high traffic intensities the queue lengths for these 2 links are 10 times longer than for the other queues. Assuming that node 5 can only be connected to node 1, message delay performance can only be improved by increasing the link capacity. The link capacities between nodes 1 and 5 are doubled to form the next alternative network shown in

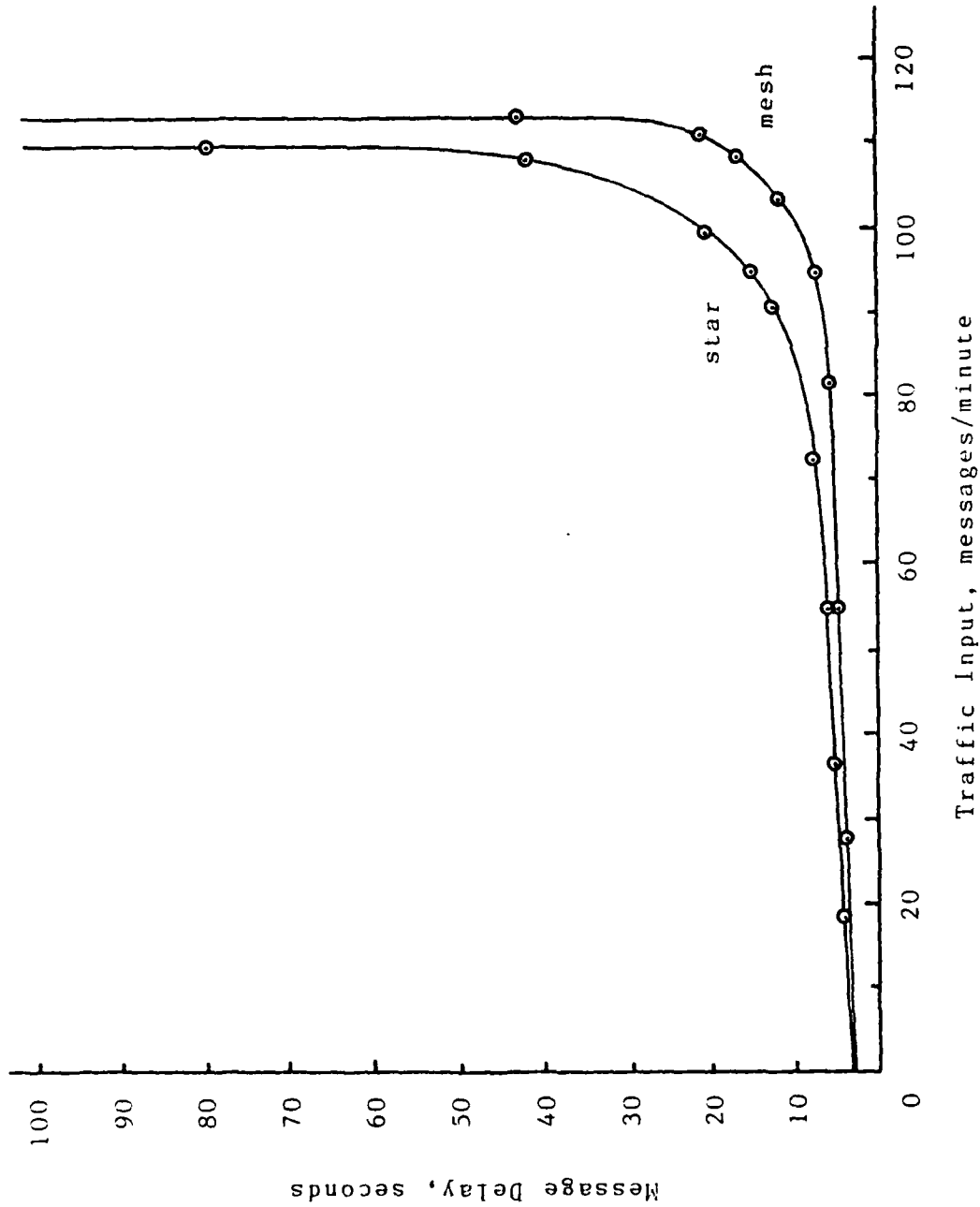


Figure 21: Message delay for star and mesh networks.

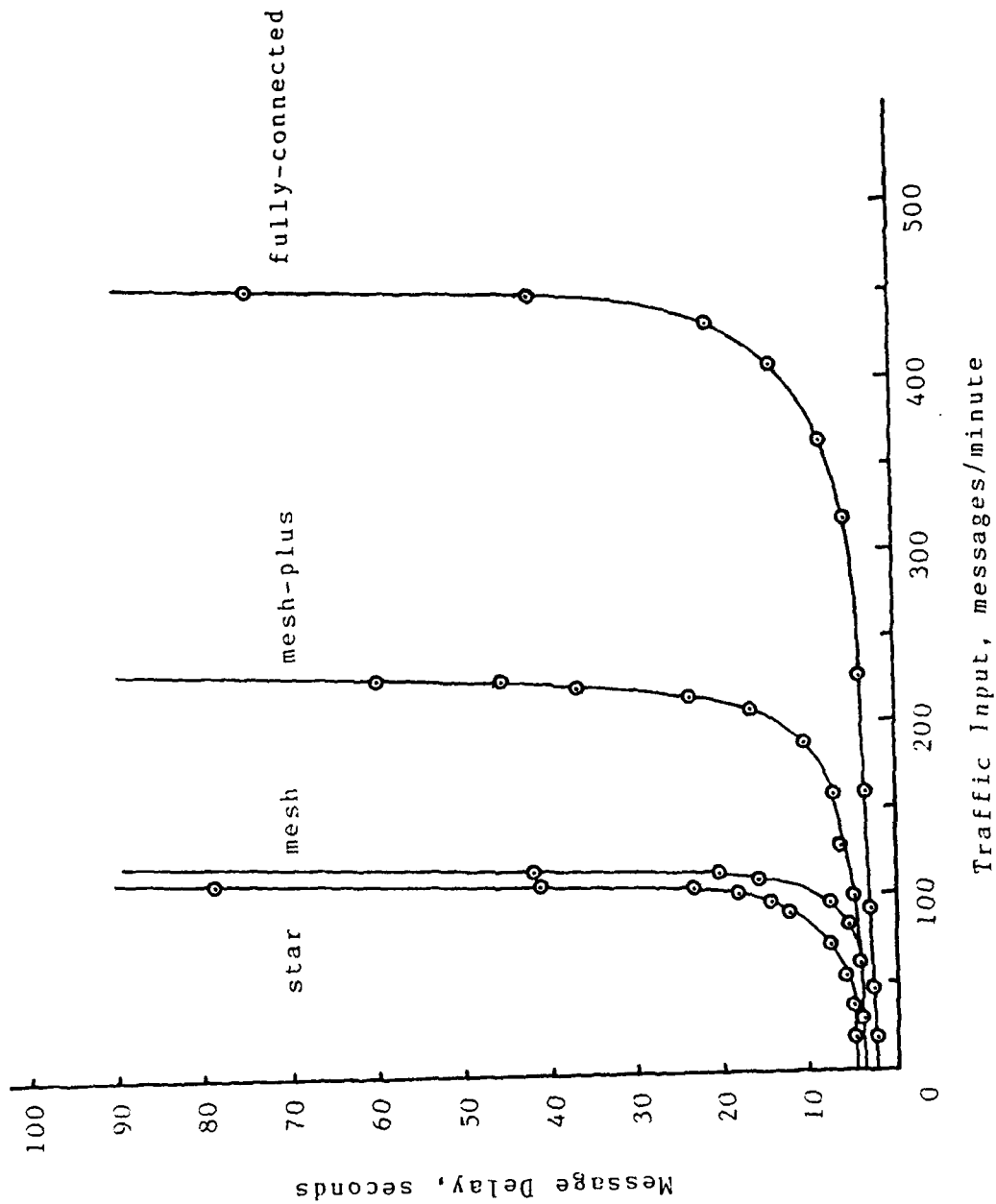


Figure 22: Message delay for network configurations.

figure 20c. As expected, average message delay performance is improved as shown by the curve labeled mesh-plus in figure 22.

Fully-Connected Network

Finally, the fully-connected network of figure 20d is simulated. This network has the maximum single link connectivity possible. The average delay performance curve, shown in figure 22, has the lowest delay of the 4 networks simulated. Other network topologies can also be simulated. The performance curves for other networks lie between the curves for the star network and the fully-connected network, assuming only single link capacity assignments. Improvement in performance over the fully-connected network requires increasing the single link capacities.

Optimum Network Selection

As the connectivity and link capacities are increased from the star network to the fully-connected network, the message delay performance improves, but the cost of building the networks also increases. By plotting network cost against message delay or message throughput, an indication of possible trade-offs are seen. For the purpose of presenting a hypothetical cost curve an assumption is made that each link costs 1 unit and that each additional line on a link costs 0.1 units.

In figure 23 message delay is compared with the cost of implementing the 4 alternative networks. For this example the throughput requirement is held constant at 12 Kbits/second (113 messages/second). The message delay for each alternative network is plotted with its cost. A dashed trend line connects the points for each network to aid in visualizing the cost/delay trade-off. The shape of the trend line reflects the relationship between the cost of adding link capacity and the resulting change in message delay. The pronounced discontinuity at the mesh-plus point occurs because of the large cost of adding additional links to form a fully-connected network. The added capacity does not significantly reduce message delay at the stated throughput requirement of 12 Kbits/second. From this limited set of alternatives the mesh-plus network is considered most cost effective in reducing system delay. A follow-on analysis of network cost proceeds with increasing the lines per link and finding the new network cost. This procedure continues until the time or budget for the analysis is exhausted.

Similarly, throughput performance is compared with network cost as shown in figure 24. The throughput values shown are for an average message delay of 5.2 seconds. The trend line again helps visualize the cost/throughput trade-off. The mesh-plus network appears to provide the best compromise between cost and throughput.

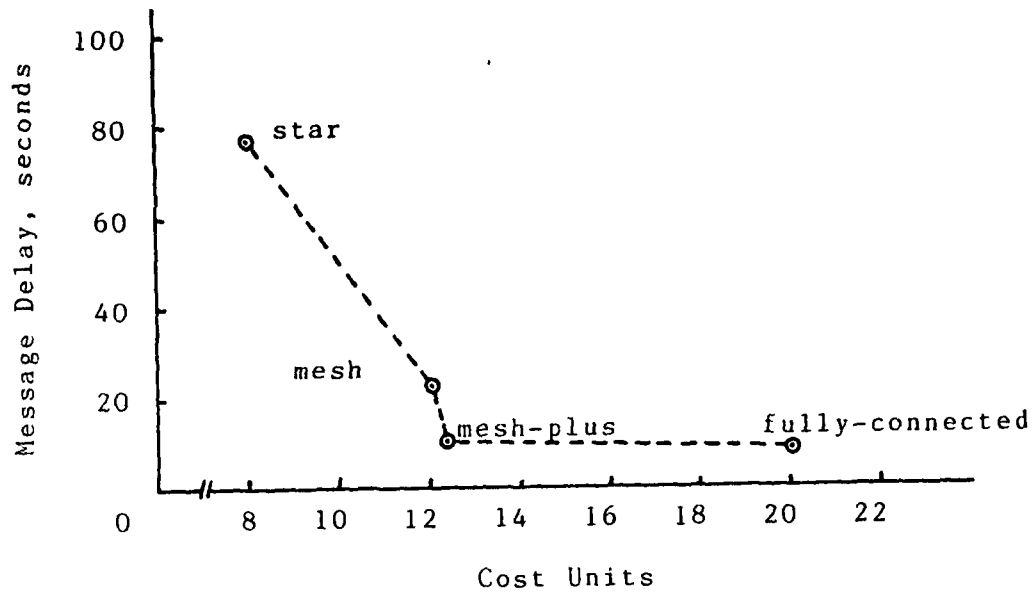


Figure 23: Message delay compared with network cost for a throughput requirement of 12 Kbits/second.

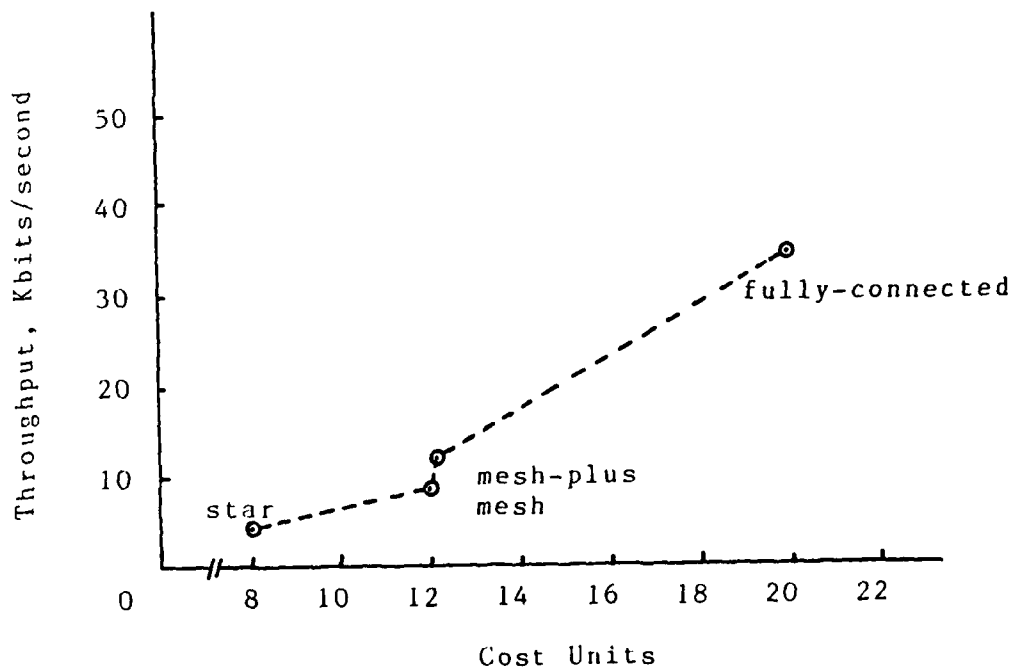


Figure 24: Throughput compared with network cost for a constant message delay of 5.2 seconds.

Four-Level Precedence

The mesh-plus network is used to study a 4-level precedence scheme. Each message is marked with a precedence level which indicates its priority for processing at each queue in the network. Within each precedence level the order of processing is FIFO. The precedence scheme simulated is non-preempting; that is, once a message starts transmission from a node, the arrival of a higher precedence message does not interrupt transmission. Message attribute 4 is marked with an integer from 1, highest, to 4, lowest, to indicate the precedence level. The percentage of messages marked for each level may be selected.

Using a message precedence distribution of 1-10%, 2-10%, 3-30% and 4-50%, message delay performance for the mesh-plus network is shown in figure 25. The highest precedence messages experience the least delay through the network. The lowest precedence messages take longer to pass through the network since they must wait in each queue while all higher precedence traffic is transmitted first.

At low traffic intensities there are fewer messages in the network and the difference in delay between the precedence levels is negligible. As traffic intensity increases, the lowest precedence traffic shows the first sign of increasing delay. Above approximately 200 messages/minute no level 4 messages are processed.

These messages remain in the node queues while higher precedence messages are processed first.

Response Traffic

The effect on network performance due to response traffic is demonstrated using the mesh-plus network. The response probability factors chosen to represent 4 hypothetical situations are shown in table 4. The simulation results for these conditions are shown in figures 26 and 27. Message throughput is plotted against message delay, and message delay is plotted against traffic input.

TABLE 4
MESSAGE RESPONSE TRAFFIC MATRIX

Precedence Level	Probability of a Response			
	Run 1	Run 2	Run 3	Run 4
1	1.0	1.0	1.0	1.0
2	0.5	0.5	0.75	0.9
3	0.1	0.3	0.5	0.6
4	0.0	0.1	0.2	0.5

In each run the original traffic input characteristics remain constant. Only the response traffic assumptions are varied. Throughput performance decreases as response

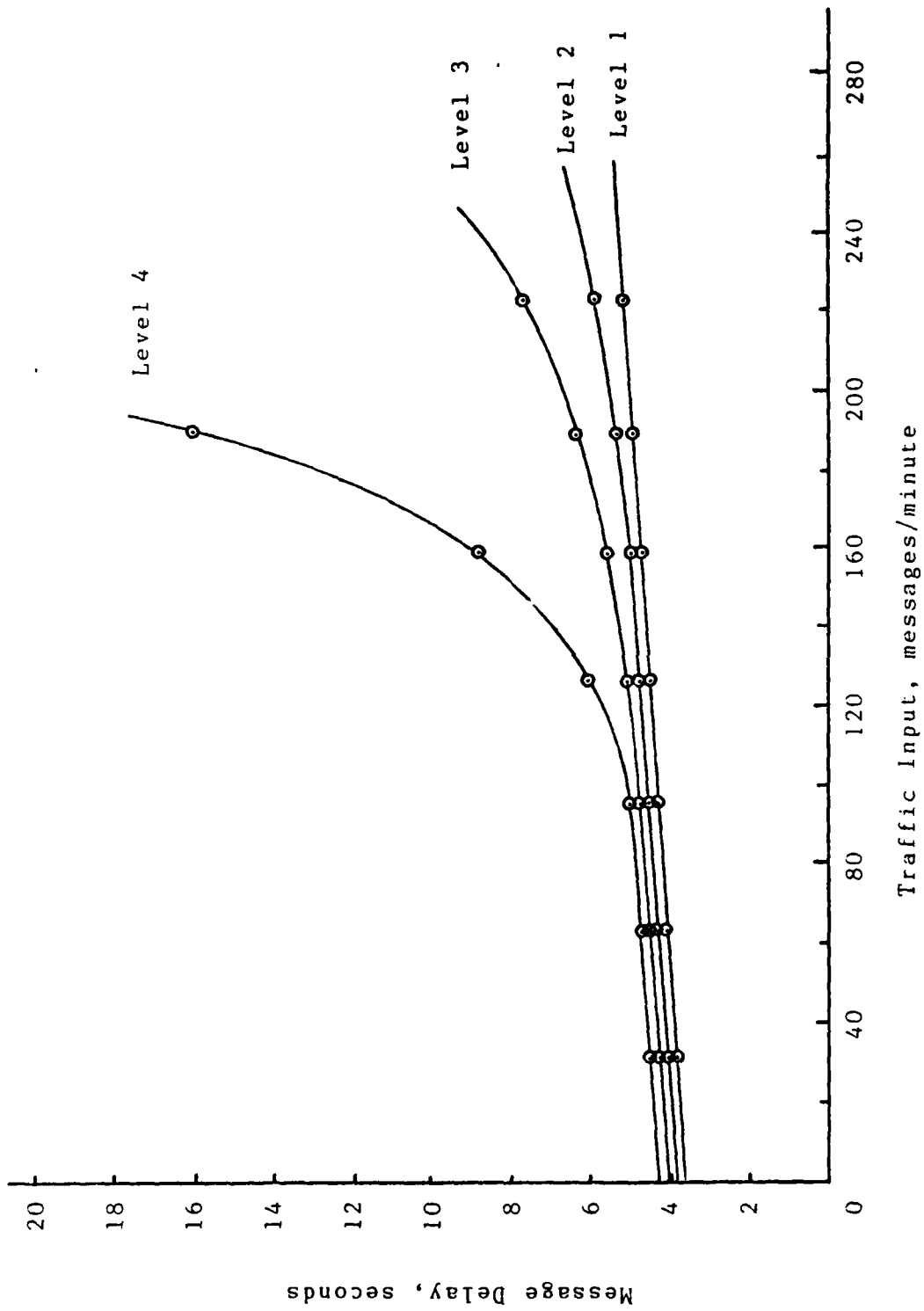


Figure 25: Message delay for precedence levels 1, 2, 3, 4.

traffic increases. Message delay increases as the amount of response traffic increases; however, the change in message delay is only significant at high traffic

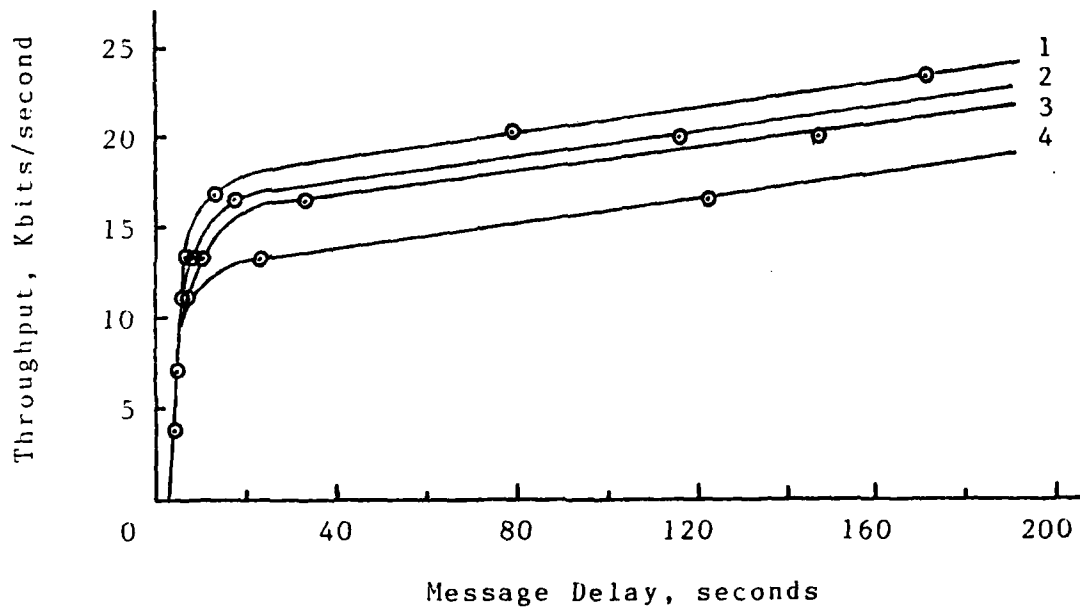


Figure 26: Message throughput for response traffic runs 1, 2, 3, 4 listed in table 4.

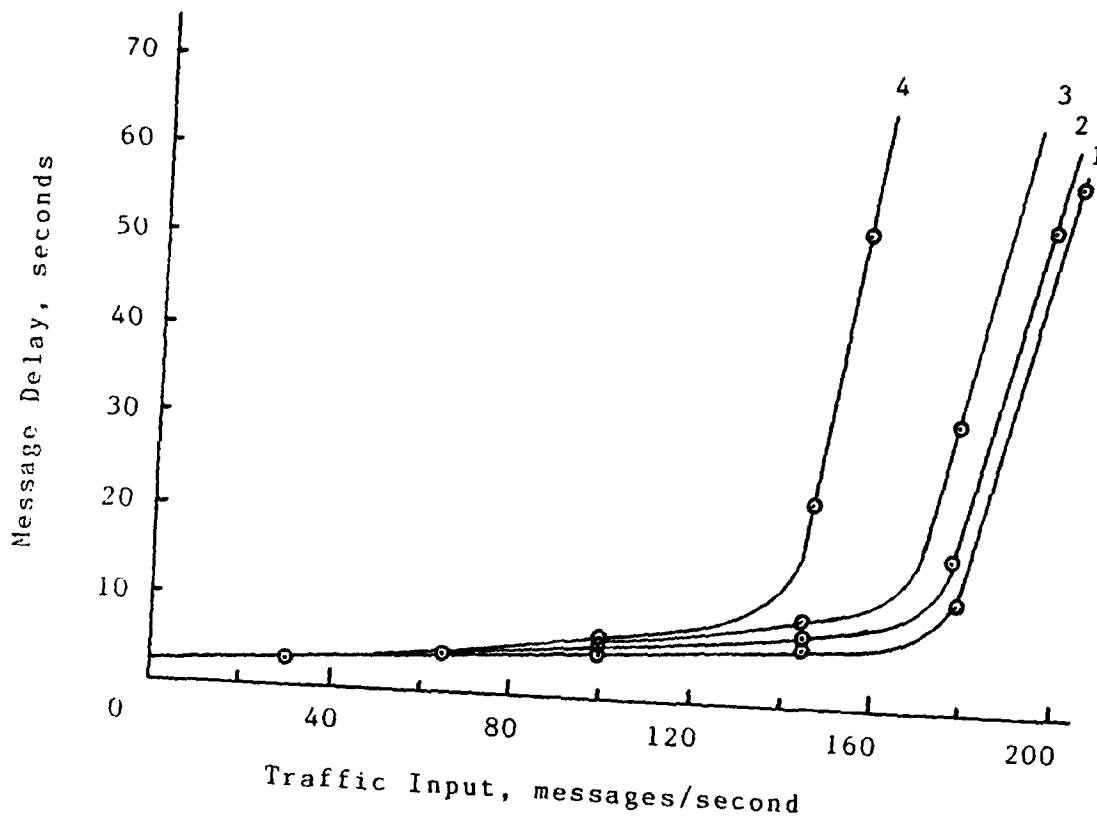


Figure 27: Message delay for the response traffic runs 1, 2, 3, 4 listed in table 4.

intensities. At low traffic intensities the network has sufficient unused capacity to handle the response traffic without causing a significant overall message delay.

ACK/NAK Protocol

The simulations described previously operate with the assumption that all transmissions are error-free. In a real network noise will introduce errors into the messages. Some errors may be corrected using coding schemes implemented in both hardware and software. The time required to accomplish encoding and decoding for error control is included in the simulation in the node processing overhead term. Excessive errors will require the message to be retransmitted.

The mesh-plus network simulation is used to simulate link error rates from 0 to 50 percent. The message performance results are shown in figure 28. As the error rate increases, message delay increases due to the number of retransmissions required. At low traffic intensities a 50 percent error rate results in increasing average message delay by 2-1/2 times.

Message throughput decreases at higher error rates. As the network begins to saturate with retransmitted messages, only the higher precedence messages are transmitted. The lower precedence messages remain in the node queues. For example, with a 50 percent error rate and

a traffic input of 95 messages/minute, precedence level 1 message delay is 18.6 seconds while precedence level 4 message delay under the same conditions is 35 minutes. The average queue length was 153 messages, and the maximum queue length is 701 messages. Clearly, under these conditions only precedence level 1 messages are experiencing an acceptable rate of throughput.

Finite Queue Capacity

The queue capacities in the previous simulations are essentially infinite since queue capacities are set to a large number. The SLAM summary report includes information on the average and maximum queue lengths. Using the queue length and message length, an estimate of the node memory capacity required for each link is made. Assuming a queue length of 20 messages each of 6400 bits, 128 Kbits of message storage is required per link. At node 1 in the mesh network 8 links requires 128 Kbytes of message storage in ASCII code.

To investigate the effects of a finite queue capacity, the maximum queue capacity in the simulation is varied for a series of runs while all other variables are held constant. When queue lengths reach the maximum level, all arriving messages are rejected. The statistics collected for each link on all rejected messages include

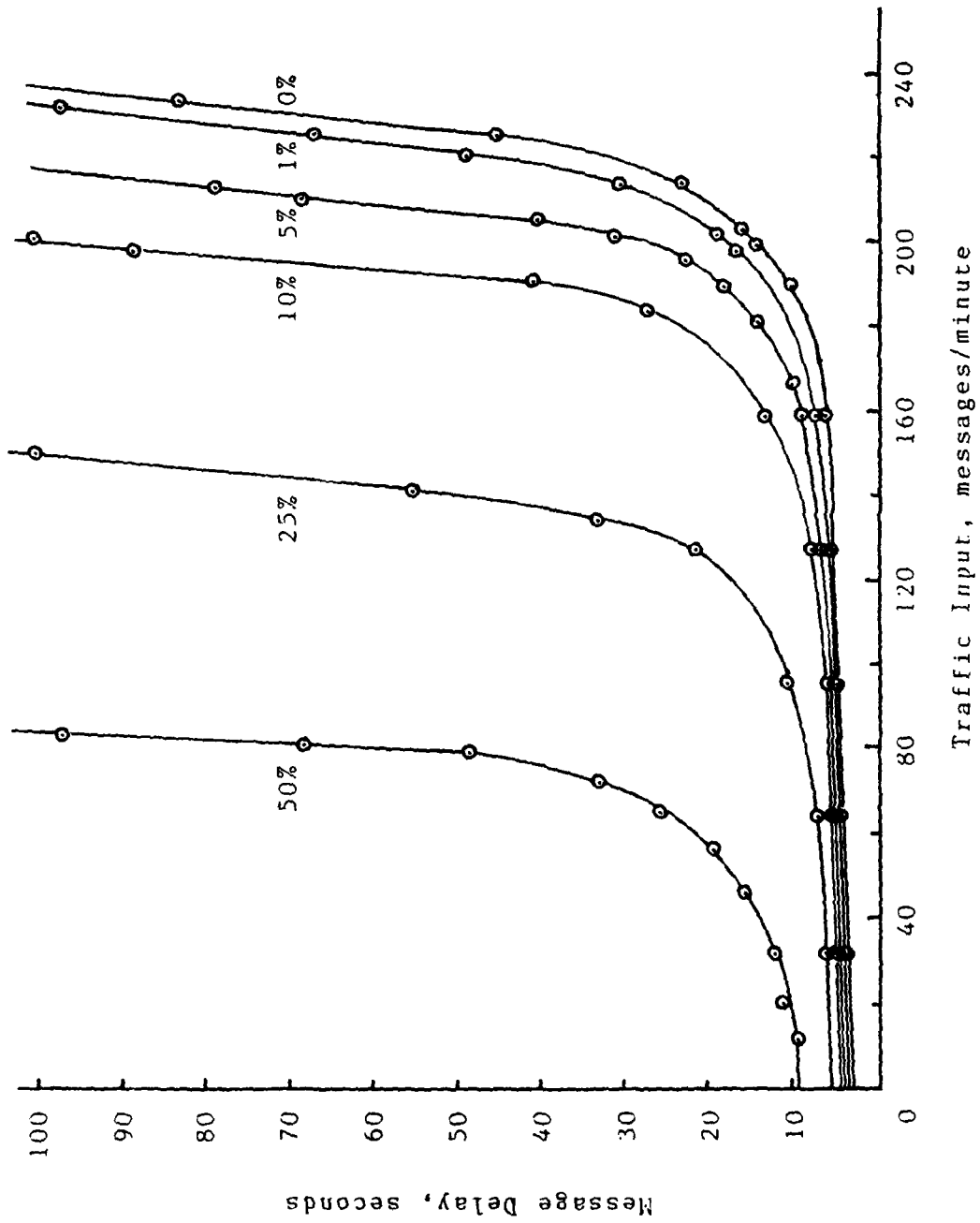


Figure 28: Message delay for link error rates.

the total number of rejections and the average time between message rejections.

The queue activity averaged over all 12 links of the mesh-plus network is shown in figure 29. For a traffic input rate of 189 messages/minute the average queue length is 7 messages. The number of rejected messages per 10,000 transmitted is plotted against queue capacity. For some acceptable level of message rejects, say 5 percent, the minimum queue capacity at this traffic input rate is 16 messages or 12.8 Kbytes of message storage per link.

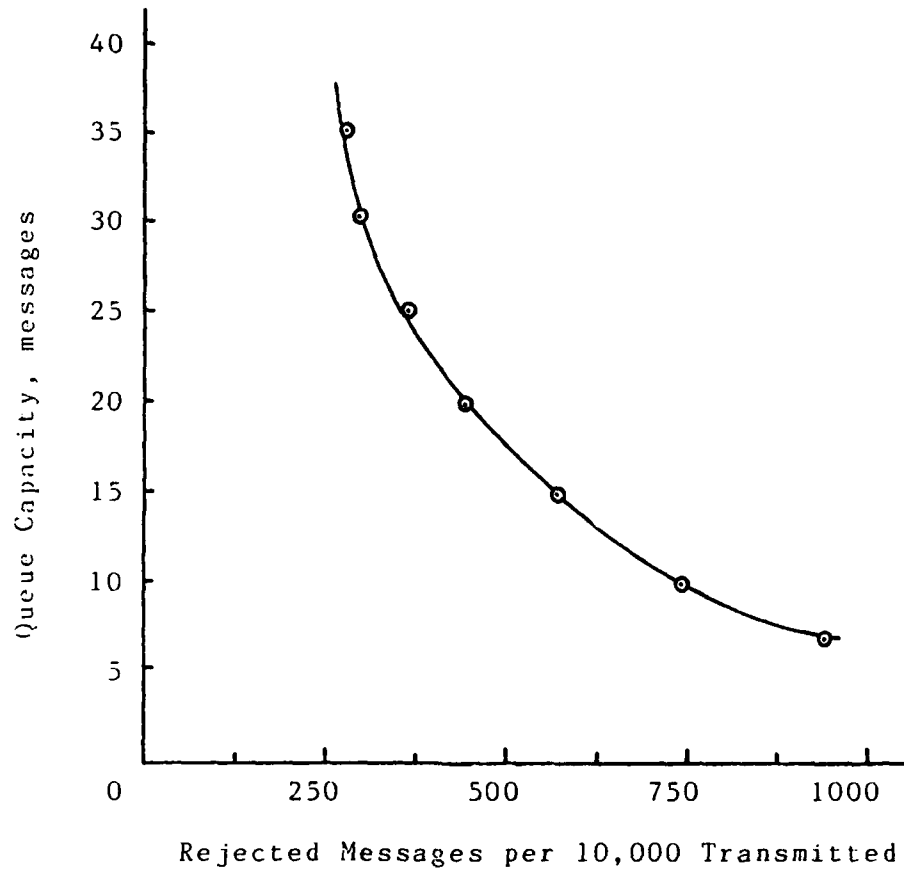


Figure 29: Rejected messages compared with queue capacity for a traffic intensity of 189 messages/second.

CHAPTER V

SUMMARY AND CONCLUSIONS

Summary

A generalized queueing model simulation of store-and-forward computer communication networks is developed and implemented. The simulation is used to provide realistic and quantitative estimates of network performance by predicting message delay and throughput. This simulation is an effective tool for making comparisons of alternative network designs. The accuracy of the simulation is demonstrated by comparison with published analytic models. The simulation is written using Simulation Language for Alternative Modeling (SLAM). A generalized simulation is achieved by making maximum use of the global variable feature in SLAM.

Discussion of Findings

The simulation provides a clear indication of how message delay performance depends on the input traffic intensity and the network characteristics. The sharp threshold behavior of message delay (and throughput) is estimated for specific network configurations. The star, mesh, mesh-plus and fully-connected networks all show a

sharp threshold for message delay as input traffic increased. When comparing the message delay performance of these networks, there is little difference in the delay when input traffic intensity is low. As traffic increases, each network saturates at a different traffic intensity depending on the topology and link capacity. The simulation gives a performance curve for each network and predicts when network saturation will occur. The performance curves are used in a cost/performance trade-off analysis to optimize a network design.

The 4-level precedence scheme demonstrates that higher precedence traffic users are assured of low message delay even when network traffic intensities increase. At low traffic intensities all users experience approximately the same message delay since there is sufficient network capacity to handle all the traffic. As input traffic increases, the lower precedence users wait in queues while higher precedence traffic is transmitted first. A difference in message delay times of greater than 5 to 1 is observed for the networks simulated. The simulation demonstrates that a network may be shared by users with differing precedence levels and still guarantee the highest precedence users acceptable performance.

Accurate prediction of network performance depends on accurate simulation of the input traffic. The message generation module allows complete control of all input

traffic characteristics. The response traffic feature provides additional control over the input traffic. Simulating only original traffic requirements without considering that some messages may require the recipient to send a message in response can lead to network designs which will fail under high traffic loads. The simulation demonstrates this effect. At high traffic loads the response traffic decreased throughput by approximately 25 percent.

The effect of data link errors on performance is demonstrated using the ACK/NAK data link protocol feature. As link error rate increased, message delay also increased. The percentage increase in message delay is found to be greater than the error rate percentage. This occurs because there are two components to message delay on error prone links. The message is delayed by being retransmitted and further delayed by waiting for the ACK/NAK message to be transmitted. Consequently, a link error rate of 10 percent causes a reduction in throughput of 19 percent. A 25 percent link error rate reduces throughput by 50 percent. The simulation allows a network designer to identify the magnitude of the effects due to link transmission errors.

The effect of a finite queue capacity is predicted by counting messages that are rejected from the network for insufficient queue space. The simulation estimates the

number of rejected messages as a function of queue capacity. This allows a network designer to determine the memory requirements for the node switching computers.

Conclusions

The SLAM language is an adequate tool for implementing the simulation model. There is some lack of flexibility in the SLAM network language structure which precludes complete generalization of the simulation model. This is not uncommon for a higher order language. For example, the computer communication network topology must be written into the simulation model using node labels in the SLAM network description. Since node labels are constants, the network topology cannot be altered during a simulation run. Changing the network topology is easily accomplished by modifying the node labels in the program listing between simulation runs.

The size of the simulation model is limited in SLAM by the amount of memory space available in the computer used to run the simulation. There is a trade-off between topology, simulation details and the number of message attributes which can be included in a simulation. Each of these must be adjusted so that the available memory space is not exceeded. This is accomplished using the SLAM echo report and by trial and error. As a result, large

network topologies cannot be simulated with the same degree of detail as smaller topologies.

The finite queue capacities are specified in the SLAM language by constants. Since SLAM does not accommodate global variables as queue capacities, the queue specifications cannot be altered during a simulation run. The queue capacities must be modified directly in the program listing between simulation runs.

These limitations on the simulation are considered minor. The advantages of using a simulation language with highly visible and easily modified program statements outweigh any restrictions in the SLAM code. The SLAM language meets the requirement of writing a generalized computer communication network simulation.

Areas for Future Study

The modular structure of this simulation allows adding additional detail to the model without disturbing the existing simulation. For example, preemptive queue discipline or variable flow assignment can be added to the simulation. Other data link protocols, possibly including a time-out feature, can be included. The structure and clarity of the SLAM language allows tailoring the simulation to any specific network requirement.

The computer communication network architecture simulated in this study uses store-and-forward message

switching. The links between the nodes represent dedicated communication links such as wire or microwave radio. An alternative network architecture is to replace all the dedicated links with one shared link connecting all network nodes. The single shared link models a radio network accessible to all the nodes. The node-to-node transfer module in the SLAM simulation can be modified to model this network architecture.

APPENDIX A

Single-Server Queue Simulation in SLAM

An example of a single-server queue is programmed in SLAM to illustrate use of the language. A complete tutorial of the SLAM language is found in references [23,24].

In this example message arrivals to an infinite queue have a Poisson distribution and message lengths are exponentially distributed. There is one transmission link out of the queue. Messages are created, placed into the queue and removed one at a time on a first-in-first-out (FIFO) basis for transmission. The total time spent in the queue plus the transmission time is the system delay. The system delay will be estimated using the simulation.

The SLAM graphic symbols for this example are shown in figure 30. Each SLAM symbol is explained below. The attributes and global variable definitions for this example are in table 5.

The results produced by the simulation model are shown in table 6. The simulation model produces results which agree closely with the analytical model.

TABLE 5

SLAM DEFINITIONS FOR SINGLE-SERVER QUEUE

QUANTITY	DEFINITION	VALUE/UNITS
Attribute 1	Message creation time	seconds
Attribute 2	Message length	bits
Attribute 3	Message transmission time	seconds
Global Variable 1	Link capacity	2000 bits /sec
Global Variable 2	Mean message inter-arrival time	0.2 sec
Global Variable 3	Mean message length	100 bits

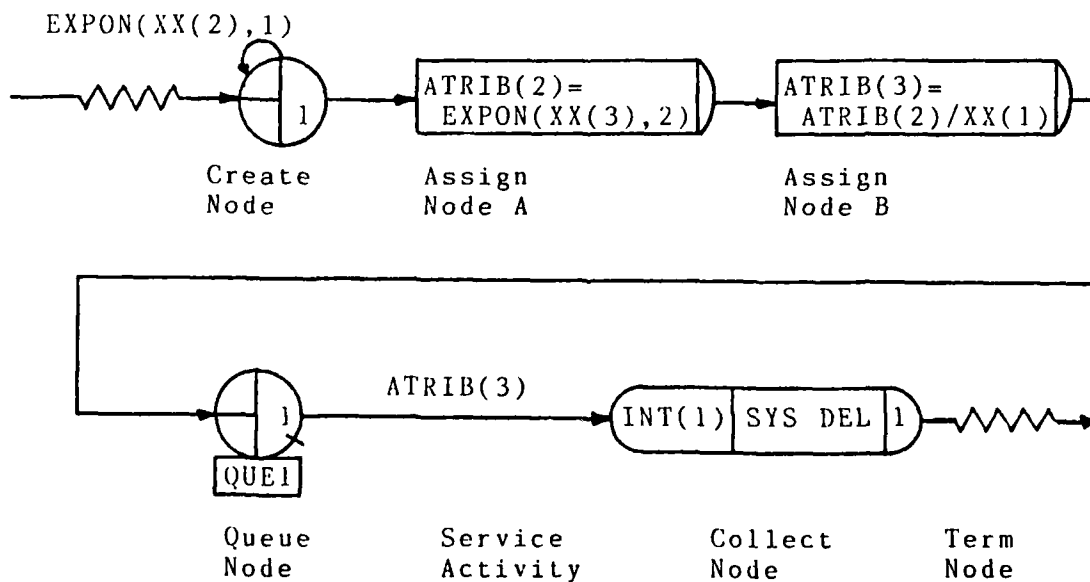


Figure 30: SLAM network for a single-server queue.

CREATE Node

The CREATE node generates messages. The first message is created at time zero. The time between message creations is a random process having an exponential distribution with a mean given by global variable XX(2). Random number stream 1 is used to generate the exponential distribution. For each message created attribute 1 will contain the time the message was created. Only one branch extends from this node.

ASSIGN Node

The ASSIGN node is used to place a value into an attribute of each message which passes through the node. At ASSIGN node A attribute 1 takes a value determined by a random process having an exponential distribution with a mean given by global variable XX(3). Random number stream 2 is used to generate the exponential distribution. Attribute 2 is the message length. At ASSIGN node B attribute 3 takes the value of attribute 2 divided by global variable XX(1). Attribute 3 is the transmission time for the message.

QUEUE Node

The QUEUE node is a location where messages await transmission. Initially, there are no messages in the queue. Queue capacity is infinite. File number 1 is used to store queued messages.

Service ACTIVITY

Service ACTIVITY 1 represents message transmission. It has a duration equal to attribute 3. Only one transmission path is available to the queue.

COLCT Node

A COLCT node is used to collect statistics on entities or variables in a simulation. In this example the interval between attribute 1, message creation time, and the current time is collected as a statistic labeled SYS DEL for system delay. These statistics will appear in the SLAM summary report at the end of the simulation run.

TERM Node

The TERM node is used to remove messages from the simulation. It can also be used to terminate a simulation after a specified number of entities have arrived at the termination node. In this example the termination count is infinite, and the run time for the simulation is determined by a SLAM control statement.

Single-Server Queue Results

Cravis [5] provides a numerical example of a single-server queue based on the Kleinrock model. The message arrival rate λ is 5 messages/second, link capacity C is 2,000 bits/second and the average message length $1/\mu$ is 100 bits. The system delay T is given by $T=1/(\mu C-\lambda)$, and

$T=0.06667$ seconds. Using these values in the SLAM simulation gives the results shown in table 6. For each of the 10 simulation runs the random number stream was initialized to a different value. The SLAM simulation results agree with the analytical model.

TABLE 6

COMPARISON OF SLAM SIMULATION WITH ANALYTIC MODEL
FOR SINGLE-SERVER QUEUE

MODEL	SYSTEM DELAY seconds
Analytic	0.06667
Simulation Run 1	0.06514
2	0.06986
3	0.06426
4	0.05101
5	0.05002
6	0.07521
7	0.07992
8	0.05332
9	0.08055
10	0.07555

APPENDIX B

Simulation Program Listing

The simulation programs were executed on a Control Data Corporation CYBER 845 computer using the operating system NOS 2.2-605/587. The Simulation Language for Alternative Programming was SLAM II version 2.0 available from Pritsker and Associates, Inc., P.O. Box 2413, West Lafayette, Indiana 47906.

Table 7 lists the user specified global variables included in the full simulation model. The full simulation model program listing with sample output is included in this appendix.

TABLE 7
LIST OF USER SPECIFIED GLOBAL VARIABLES

VARIABLE	DEFINITION
XX(1)	Link Transmission Rate is bits/second
XX(3)	Mean Message Length in bits
XX(4)	Minimum Message Length in bits
XX(5)	Maximum Message Length in bits
XX(8)	ACK/NAK Message Length in bits/second
XX(9)	Node Processing Overhead in seconds
XX(11)	Gamma12 and Gamma21 Mean Message Interarrival Time in seconds
XX(12)	Gamma13 and Gamma31 Mean Message Interarrival Time in seconds
XX(13)	Gamma14 and Gamma41 Mean Message Interarrival Time in seconds
XX(14)	Gamma15 and Gamma51 Mean Message Interarrival Time in seconds
XX(15)	Gamma23 and Gamma32 Mean Message Interarrival Time in seconds
XX(16)	Gamma24 and Gamma42 Mean Message Interarrival Time in seconds
XX(17)	Gamma25 and Gamma52 Mean Message Interarrival Time in seconds
XX(18)	Gamma34 and Gamma43 Mean Message Interarrival Time in seconds
XX(19)	Gamma35 and Gamma53 Mean Message Interarrival Time in seconds
XX(20)	Gamma45 and Gamma54 Mean Message Interarrival Time in seconds
XX(31)	Node 1 Message Serial Number Counter
XX(32)	Node 2 Message Serial Number Counter
XX(33)	Node 3 Message Serial Number Counter
XX(34)	Node 4 Message Serial Number Counter
XX(35)	Node 5 Message Serial Number Counter

VARIABLE	DEFINITION
XX(38)	Link12 Error Rate in percent/100
XX(39)	Link13 Error Rate in percent/100
XX(40)	Link14 Error Rate in percent/100
XX(41)	Link15 Error Rate in percent/100
XX(42)	Link21 Error Rate in percent/100
XX(43)	Link23 Error Rate in percent/100
XX(44)	Link31 Error Rate in percent/100
XX(45)	Link32 Error Rate in percent/100
XX(46)	Link34 Error Rate in percent/100
XX(47)	Link41 Error Rate in percent/100
XX(48)	Link43 Error Rate in percent/100
XX(49)	Link51 Error Rate in percent/100
XX(51)	Precedence 1 Traffic Rate in percent/100
XX(52)	Precedence 2 Traffic Rate in percent/100
XX(53)	Precedence 3 Traffic Rate in percent/100
XX(54)	Precedence 4 Traffic Rate in percent/100
XX(61)	Response Rate for Precedence 1 Traffic in percent/100
XX(62)	Response Rate for Precedence 2 Traffic in percent/100
XX(63)	Response Rate for Precedence 3 Traffic in percent/100
XX(64)	Response Rate for Precedence 4 Traffic in percent/100
XX(71)	Number of Lines for Link12
XX(72)	Number of Lines for Link13
XX(73)	Number of Lines for Link14
XX(74)	Number of Lines for Link15
XX(75)	Number of Lines for Link21
XX(76)	Number of Lines for Link23
XX(77)	Number of Lines for Link31
XX(78)	Number of Lines for Link32
XX(79)	Number of Lines for Link34
XX(80)	Number of Lines for Link41
XX(81)	Number of Lines for Link43
XX(82)	Number of Lines for Link51

1 GEN, BERT GARCIA, FULL MODEL P24B, 02/25/85, 5, Y, N, Y, N, Y;
2 LIMITS, 12, 7, 4000;
3 ;
4 ;
5 ; SIMULATES A COMPUTER COMMUNICATIONS NETWORK WITH FOLLOWING PROPERTIES:
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;
18 ;
19 ;
20 ;
21 ;
22 ;
23 ;
24 ;
25 ;
26 ;
27 ;
28 ;
29 ;
30 ;
31 ;
32 ;
33 ;
34 ;
35 ;

TOPOLOGY: MESH NETWORK WITH CONNECTIVITY AS SPECIFIED
IN MESSAGE ROUTING TABLE. LINK CAPACITY IS
SELECTABLE.

MESSAGE INPUT: POISSON PROCESS. INTERARRIVAL TIMES HAVE
EXPONENTIAL RANDOM DISTRIBUTION WITH RATES
SPECIFIED IN TRAFFIC MATRIX.

MESSAGE LENGTH: EXPONENTIAL RANDOM DISTRIBUTION WITH SELECTABLE
MEAN, MAXIMUM AND MINIMUM.

QUEUE DISCIPLINE: FOUR-LEVEL PRECEDENCE, HIGHEST PRECEDENCE FIRST.

QUEUE CAPACITY: FINITE AS SPECIFIED IN AWAIT NODES.

LINK PROTOCOL: ACK/NAK WITH ADJUSTABLE LINK ERROR RATE.

GLOBAL VARIABLE DEFINITIONS:

XX(1) = LINK TRANSMIT RATE IN BITS/SEC
XX(3) = MEAN MESSAGE LENGTH IN BITS
XX(4) = MINIMUM MESSAGE LENGTH IN BITS
XX(5) = MAXIMUM MESSAGE LENGTH IN BITS
XX(8) = ACK/NAK MESSAGE LENGTH IN BITS/SEC
XX(9) = NODE PROCESSING OVERHEAD IN SEC

XX(11) = GAWPA12 AND GAWPA21 MEAN MESSAGE INTERARRIVAL TIME
XX(12) = GAWPA13 AND GAWPA31 MEAN MESSAGE INTERARRIVAL TIME

AD-A162 580 ESTIMATING COMPUTER COMMUNICATION NETWORK PERFORMANCE
USING NETWORK SIMULATIONS(U) ARMY MILITARY PERSONNEL
CENTER ALEXANDRIA VA A B GARCIA APR 85

AD-A162 580 ESTIMATING COMPUTER COMMUNICATION NETWORK PERFORMANCE
USING NETWORK SIMULATIONS(U) ARMY MILITARY PERSONNEL
CENTER ALEXANDRIA VA A B GARCIA APR 85

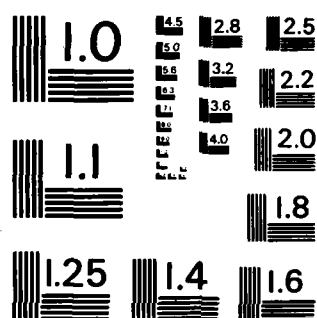
AD-A162 580 ESTIMATING COMPUTER COMMUNICATION NETWORK PERFORMANCE 2/2
USING NETWORK SIMULATIONS(U) ARMY MILITARY PERSONNEL
CENTER ALEXANDRIA VA A B GARCIA APR 85

UNCLASSIFIED CENTER ALEXANDRIA VA H B GARCIA APR 85 F/G 17/2 NL

UNCLASSIFIED CENTER ALEXANDRIA VA H B GARCIA APR 85 F/G 17/2 NL

UNCLASSIFIED CENTER ALEXANDRIA VA H B GARCIA APR 85 F/G 17/2 NL

[illegible]



36 ;	XX(13) = GANNA14 AND GANNA41 MEAN MESSAGE INTERARRIVAL TIME
37 ;	XX(14) = GANNA15 AND GANNA51 MEAN MESSAGE INTERARRIVAL TIME
38 ;	XX(15) = GANNA23 AND GANNA32 MEAN MESSAGE INTERARRIVAL TIME
39 ;	XX(16) = GANNA24 AND GANNA42 MEAN MESSAGE INTERARRIVAL TIME
40 ;	XX(17) = GANNA25 AND GANNA52 MEAN MESSAGE INTERARRIVAL TIME
41 ;	XX(18) = GANNA34 AND GANNA43 MEAN MESSAGE INTERARRIVAL TIME
42 ;	XX(19) = GANNA35 AND GANNA53 MEAN MESSAGE INTERARRIVAL TIME
43 ;	XX(20) = GANNA45 AND GANNA54 MEAN MESSAGE INTERARRIVAL TIME
44 ;	
45 ;	XX(31) = NODE 1 MESSAGE SERIAL NUMBER COUNTER
46 ;	XX(32) = NODE 2 MESSAGE SERIAL NUMBER COUNTER
47 ;	XX(33) = NODE 3 MESSAGE SERIAL NUMBER COUNTER
48 ;	XX(34) = NODE 4 MESSAGE SERIAL NUMBER COUNTER
49 ;	XX(35) = NODE 5 MESSAGE SERIAL NUMBER COUNTER
50 ;	
51 ;	XX(38) = LINK12 ERROR RATE IN PERCENT/100
52 ;	XX(39) = LINK13 ERROR RATE IN PERCENT/100
53 ;	XX(40) = LINK14 ERROR RATE IN PERCENT/100
54 ;	XX(41) = LINK15 ERROR RATE IN PERCENT/100
55 ;	XX(42) = LINK21 ERROR RATE IN PERCENT/100
56 ;	XX(43) = LINK23 ERROR RATE IN PERCENT/100
57 ;	XX(44) = LINK31 ERROR RATE IN PERCENT/100
58 ;	XX(45) = LINK32 ERROR RATE IN PERCENT/100
59 ;	XX(46) = LINK34 ERROR RATE IN PERCENT/100
60 ;	XX(47) = LINK41 ERROR RATE IN PERCENT/100
61 ;	XX(48) = LINK43 ERROR RATE IN PERCENT/100
62 ;	XX(49) = LINK51 ERROR RATE IN PERCENT/100
63 ;	
64 ;	XX(51) = PERCENTAGE/100 OF PRECEDENCE 1 TRAFFIC
65 ;	XX(52) = PERCENTAGE/100 OF PRECEDENCE 2 TRAFFIC
66 ;	XX(53) = PERCENTAGE/100 OF PRECEDENCE 3 TRAFFIC
67 ;	XX(54) = PERCENTAGE/100 OF PRECEDENCE 4 TRAFFIC
68 ;	
69 ;	XX(61) = PRECEDENCE 1 RESPONSE RATE
70 ;	XX(62) = PRECEDENCE 2 RESPONSE RATE
71 ;	XX(63) = PRECEDENCE 3 RESPONSE RATE

```

72 ;
73 ;
74 ;
75 ;
76 ;
77 ;
78 ;
79 ;
80 ;
81 ;
82 ;
83 ;
84 ;
85 ;
86 ;
87 ;
88 ; GLOBAL VARIABLE VALUES:
89 ;
90 INTLC,
91   XX(1)=2400,
92   XX(3)=6400,
93   XX(4)=100,
94   XX(5)=25600,
95   XX(8)=3300,
96   XX(9)=0.001;
97 ;
98 INTLC,
99   XX(11)=38.095238,
100  XX(12)=38.095238,
101  XX(13)=38.095238,
102  XX(14)=38.095238,
103  XX(15)=38.095238,
104  XX(16)=38.095238,
105  XX(17)=38.095238,
106  XX(18)=38.095238,
107  XX(19)=38.095238,

```

```

XX(64) = FREQUENCY 4 RESPONSE RATE
XX(71) = NUMBER OF LINES FOR LINK12
XX(72) = NUMBER OF LINES FOR LINK13
XX(73) = NUMBER OF LINES FOR LINK14
XX(74) = NUMBER OF LINES FOR LINK15
XX(75) = NUMBER OF LINES FOR LINK21
XX(76) = NUMBER OF LINES FOR LINK23
XX(77) = NUMBER OF LINES FOR LINK31
XX(78) = NUMBER OF LINES FOR LINK32
XX(79) = NUMBER OF LINES FOR LINK34
XX(80) = NUMBER OF LINES FOR LINK41
XX(81) = NUMBER OF LINES FOR LINK43
XX(82) = NUMBER OF LINES FOR LINK51

```

```

108
109 ;
110 INTLC,
111
112 XX(31)=0,
113 XX(32)=0,
114 XX(33)=0,
115 XX(34)=0,
116 XX(35)=0;
117 ;
118 INTLC,
119
120 XX(38)=0.01,
121 XX(39)=0.01,
122 XX(40)=0.01,
123 XX(41)=0.01,
124 XX(42)=0.01,
125 XX(43)=0.01,
126 XX(44)=0.01,
127 XX(45)=0.01,
128 XX(46)=0.01,
129 XX(47)=0.01,
130 XX(48)=0.01,
131 XX(49)=0.01;
132 ;
133 INTLC,
134
135 XX(51)=0.1,
136 XX(52)=0.1,
137 XX(53)=0.3,
138 XX(54)=0.5,
139 XX(61)=1.0,
140 XX(62)=0.5,
141 XX(63)=0.3,
142 XX(64)=0.0;
143 ;
144 INTLC,
145
146 XX(71)=1,
147 XX(72)=1,

```

XX(20)=38.095238;

144 XX(73)=1,
 145 XX(74)=2,
 146 XX(75)=1,
 147 XX(76)=1,
 148 XX(77)=1,
 149 XX(78)=1,
 150 XX(79)=1,
 151 XX(80)=1,
 152 XX(81)=1,
 153 XX(82)=2;

154 ; MESSAGE ATTRIBUTES:

155 ; ATRIB(1) = MESSAGE CREATION TIME
 156 ; ATRIB(2) = MESSAGE DESTINATION NODE
 157 ; ATRIB(3) = MESSAGE TYPE
 158 ; ATRIB(4) = MESSAGE PRECEDENCE
 159 ; ATRIB(5) = MESSAGE ORIGIN NODE
 160 ; ATRIB(6) = MESSAGE SERIAL NUMBER
 161 ; ATRIB(7) = MESSAGE LENGTH

162 ; INPUT TRAFFIC MATRIX:
 163 ; VALUES SHOWN ARE FOR NETWORK UTILIZATION FACTOR RHO=0.1.
 164 ; MATRIX IS USUALLY SYMMETRICAL. ENTRIES ARE 1/G.K.

	FROM	TO				
		1	2	3	4	5
173 ;	1	-	38.095238	38.095238	38.095238	38.095238
174 ;	2	38.095238	-	38.095238	38.095238	38.095238
175 ;	3	38.095238	38.095238	-	38.095238	38.095238
176 ;	4	38.095238	38.095238	38.095238	-	38.095238
177 ;	5	38.095238	38.095238	38.095238	38.095238	-

178 ;
 179 ;

MESSAGE ROUTING TABLE:

	TO				
	1	2	3	4	5
FROM					
1	-	2	3	4	5
2	1	-	3	3	1
3	1	2	-	4	1
4	1	3	3	-	1
5	1	1	1	1	-

NODE QUEUES ARE IMPLEMENTED USING RESOURCE FILES REPRESENTING
COMMUNICATION LINKS. AVAILABLE LINKS FOR THIS NETWORK ARE:

NODE 1	NODE 2	NODE 3	NODE 4	NODE 5
LINK12	LINK21	LINK31	LINK41	LINK51
LINK13	LINK23	LINK32	LINK43	
LINK14		LINK34		
LINK15				

QUEUE CAPACITY: 25 MESSAGES.

***** NETWORK TOPOLOGY DESCRIPTION *****

NETWORK;

LINK RESOURCES DEFINE NETWORK CONNECTIVITY.

RESOURCE/LINK12(0),1/LINK13(0),2/LINK14(0),3/LINK15(0),4;
 RESOURCE/LINK21(0),5/LINK23(0),6;
 RESOURCE/LINK31(0),7/LINK32(0),8/LINK34(0),9;
 RESOURCE/LINK41(0),10/LINK43(0),11;
 RESOURCE/LINK51(0),12;

```

243 ;
244 ; ALTER NUMBER OF LINES FOR EACH LINK.
245 ; CREATE ONE DUMMY MESSAGE
246 ; TO ALTER EACH LINK WITH
247 ; REQUIRED NUMBER OF LINES.
248 ;
249 ; ALTER, LINK12/XX(71),1;
250 ; ALTER, LINK13/XX(72),1;
251 ; ALTER, LINK14/XX(73),1;
252 ; ALTER, LINK15/XX(74),1;
253 ; ALTER, LINK21/XX(75),1;
254 ; ALTER, LINK23/XX(76),1;
255 ; ALTER, LINK31/XX(77),1;
256 ; ALTER, LINK32/XX(78),1;
257 ; ALTER, LINK34/XX(79),1;
258 ; ALTER, LINK41/XX(80),1;
259 ; ALTER, LINK43/XX(81),1;
260 ; ALTER, LINK51/XX(82),1;
261 ; TERM;
262 ;
263 ; DISCARD DUMMY MESSAGE
264 ;
265 ;
266 ;
267 ; END OF NETWORK TOPOLOGY DESCRIPTION
268 ;
269 ; *****
270 ; *
271 ; * MESSAGE GENERATION
272 ; *
273 ;
274 ; EACH ORIGIN NODE HAS A SEPARATE TRAFFIC GENERATOR STATEMENT FOR EACH
275 ; DESTINATION NODE.
276 ;
277 ; ** NODE 1 **
278 ; CREATE MESSAGES AND STORE CREATION TIME IN ATTRIB(1). RANDOM TIME
279 ; BETWEEN MESSAGE CREATIONS HAS MEAN VALUE GIVEN BY GLOBAL VARIABLE.
280 ; MESSAGE PRECEDENCE LEVEL IS DETERMINED BY PROBABILITY BRANCHING AND
281 ; INDICATED IN ATTRIBUTE 4. CREATED MESSAGE IS PLACED IN THE
282 ; PROPER TRANSMIT QUEUE ACCORDING TO THE ROUTING TABLE.

```

```

277 ; TRAFFIC TO NODE 2:
278 C1 CREATE,EXPON(XX(11),1),,1;
279 ASSIGN,TRIB(2)=2,TRIB(3)=0,TRIB(5)=1,1;
280 ASSIGN,TRIB(7)=EXPON(XX(3),1),1;
281 ASSIGN,XX(31)=XX(31)+1,1;
282 ASSIGN,TRIB(6)=XX(31),1;
283 QUN,1;
284 ACT,,TRIB(7).LE..XX(4),C1M;
285 ACT,,TRIB(7).GE..XX(5),C1X;
286 ACT,,C1P;
287 C1M ASSIGN,TRIB(7)=XX(4);
288 ACT,,C1P;
289 C1X ASSIGN,TRIB(7)=XX(5);
290 ACT,,C1P;
291 C1P QUN,1;
292 ACT,,XX(51),C1A;
293 ACT,,XX(52),C1B;
294 ACT,,XX(53),C1C;
295 ACT,,XX(54),C1D;
296 C1A ASSIGN,TRIB(4)=1;
297 ACT,,Q12;
298 C1B ASSIGN,TRIB(4)=2;
299 ACT,,Q12;
300 C1C ASSIGN,TRIB(4)=3;
301 ACT,,Q12;
302 C1D ASSIGN,TRIB(4)=4;
303 ACT,,Q12;
304 ;
305 ; TRAFFIC TO NODE 3:
306 C2 CREATE,EXPON(XX(12),2),,1;
307 ASSIGN,TRIB(2)=3,TRIB(3)=0,TRIB(5)=1,1;
308 ASSIGN,TRIB(7)=EXPON(XX(3),2),1;
309 ASSIGN,XX(31)=XX(31)+1,1;
310 ASSIGN,TRIB(6)=XX(31),1;
311 QUN,1;
312 ACT,,TRIB(7).LE..XX(4),C2M;

```

CREATE MESSAGE
 ASSIGN DESTIN, TYPE, ORIGIN
 ASSIGN MESSAGE LENGTH
 INCREMENT SER NUM COUNTER
 ASSIGN SERIAL NUMBER
 CHECK FOR MINIMUM LENGTH
 CHECK FOR MAXIMUM LENGTH
 ASSIGN MINIMUM LENGTH
 ASSIGN MAXIMUM LENGTH
 DISTRIB MSGS BY PRECEDENCE
 ASSIGN PR 1
 PUT MSG IN OUTGOING QUEUE
 ASSIGN PR 2
 PUT MSG IN OUTGOING QUEUE
 ASSIGN PR 3
 PUT MSG IN OUTGOING QUEUE
 ASSIGN PR 4
 PUT MSG IN OUTGOING QUEUE
 CREATE MESSAGE
 ASSIGN DESTIN, TYPE, ORIGIN
 ASSIGN MESSAGE LENGTH
 INCREMENT SER NUM COUNTER
 ASSIGN SERIAL NUMBER
 CHECK FOR MINIMUM LENGTH

```

313      ACT, ATTRIB(7), GE, XX(5), C2X;
314      ACT, ,, C2P;
315      C2F1 ASSIGN, ATTRIB(7)=XX(4);
316      ACT, ,, C2P;
317      C2X ASSIGN, ATTRIB(7)=XX(5);
318      ACT, ,, C2P;
319      C2P GOON, 1;
320      ACT, XX(51), C2A;
321      ACT, XX(52), C2B;
322      ACT, XX(53), C2C;
323      ACT, XX(54), C2D;
324      C2A ASSIGN, ATTRIB(4)=1;
325      ACT, ,, Q13;
326      C2B ASSIGN, ATTRIB(4)=2;
327      ACT, ,, Q13;
328      C2C ASSIGN, ATTRIB(4)=3;
329      ACT, ,, Q13;
330      C2D ASSIGN, ATTRIB(4)=4;
331      ACT, ,, Q13;
332      ;
333      ; TRAFFIC TO NODE 4;
334      C3 CREATE, EXON(XX(12), 3), , 1;
335      ASSIGN, ATTRIB(2)=4, ATTRIB(3)=0, ATTRIB(5)=1, 1;
336      ASSIGN, ATTRIB(7)=EXON(XX(3), 3), 1;
337      ASSIGN, XX(31)=XX(31)+1, 1;
338      ASSIGN, ATTRIB(6)=XX(31), 1;
339      GOON, 1;
340      ACT, ATTRIB(7), LE, XX(4), C3M;
341      ACT, ATTRIB(7), GE, XX(5), C3X;
342      ACT, ,, C3P;
343      C3M ASSIGN, ATTRIB(7)=XX(4);
344      ACT, ,, C3P;
345      C3X ASSIGN, ATTRIB(7)=XX(5);
346      ACT, ,, C3P;
347      C3P GOON, 1;
348      ACT, XX(51), C3A;

```

CHECK FOR MAXIMUM LENGTH

ASSIGN MINIMUM LENGTH

ASSIGN MAXIMUM LENGTH

DISTRIB MSGS BY PRECEDENCE

ASSIGN PR 1

PUT MSG IN OUTGOING QUEUE

ASSIGN PR 2

PUT MSG IN OUTGOING QUEUE

ASSIGN PR 3

PUT MSG IN OUTGOING QUEUE

ASSIGN PR 4

PUT MSG IN OUTGOING QUEUE

CREATE MESSAGE

ASSIGN DESTIN, TYPE, ORIGIN

ASSIGN MESSAGE LENGTH

INCREMENT SER NUM COUNTER

ASSIGN SER NUM

CHECK FOR MINIMUM LENGTH

CHECK FOR MAXIMUM LENGTH

ASSIGN MINIMUM LENGTH

ASSIGN MAXIMUM LENGTH

DISTRIB MSGS BY PRECEDENCE

```

349 ACT,,XX(52),C3B;
350 ACT,,XX(53),C3;
351 ACT,,XX(54),C3D;
352 C3A ASSIGN,TRIB(4)=1;
353 ACT,,Q14;
354 C3B ASSIGN,TRIB(4)=2;
355 ACT,,Q14;
356 C3C ASSIGN,TRIB(4)=3;
357 ACT,,Q14;
358 C3D ASSIGN,TRIB(4)=4;
359 ACT,,Q14;
360 ;
361 ; TRAFFIC TO NODE 5:
362 C4 CREATE,EXPON(XX(14),4),,1;
363 ASSIGN,TRIB(2)=5,TRIB(3)=0,TRIB(5)=1,1;
364 ASSIGN,TRIB(7)=EXPON(XX(3),4),1;
365 ASSIGN,XX(31)=XX(31)+1,1;
366 ASSIGN,TRIB(6)=XX(31),1;
367 QCON,1;
368 ACT,,TRIB(7).LE.XX(4),C4N;
369 ACT,,TRIB(7).GE.XX(5),C4X;
370 ACT,,C4P;
371 C4N ASSIGN,TRIB(7)=XX(4);
372 ACT,,C4P;
373 C4X ASSIGN,TRIB(7)=XX(5);
374 ACT,,C4P;
375 QCON,1;
376 ACT,,XX(51),C4A;
377 ACT,,XX(52),C4B;
378 ACT,,XX(53),C4C;
379 ACT,,XX(54),C4D;
380 C4A ASSIGN,TRIB(4)=1;
381 ACT,,Q15;
382 C4B ASSIGN,TRIB(4)=2;
383 ACT,,Q15;
384 C4C ASSIGN,TRIB(4)=3;

```

ASSIGN PR 1
 PUT MSG IN OUTGOING QUEUE
 ASSIGN PR 2
 PUT MSG IN OUTGOING QUEUE
 ASSIGN PR 3
 PUT MSG IN OUTGOING QUEUE
 ASSIGN PR 4
 PUT MSG IN OUTGOING QUEUE

CREATE MESSAGE
 ASSIGN DESIN, TYPE, ORIGIN
 ASSIGN MESSAGE LENGTH
 INCREMENT SER NUM COUNTER
 ASSIGN SERIAL NUMBER

CHECK FOR MINIMUM LENGTH
 CHECK FOR MAXIMUM LENGTH
 ASSIGN MINIMUM LENGTH
 ASSIGN MAXIMUM LENGTH

DISTRIB MSGS BY PRECEDENCE

ASSIGN PR 1
 PUT MSG IN OUTGOING QUEUE
 ASSIGN PR 2
 PUT MSG IN OUTGOING QUEUE
 ASSIGN PR 3

```

385      ACT,,Q15;
386 C4D  ASSIGN,TRIB(4)=4;
387      ACT,,Q15;
388 ;
389 ;
390 ; ** NODE 2 **
391 ;
392 ; TRAFFIC TO NODE 1:
393 C5  CREATE,EXPON(X(11),5),,1;
394      ASSIGN,TRIB(2)=1,TRIB(3)=0,TRIB(5)=2,1;
395      ASSIGN,TRIB(7)=EXPON(X(3),5),1;
396      ASSIGN,XX(32)=XX(32)+1,1;
397      ASSIGN,TRIB(6)=XX(32),1;
398      QJUN,1;
399      ACT,,TRIB(7).LE.XX(4),C5F;
400      ACT,,TRIB(7).GE.XX(5),C5X;
401      ACT,,C5P;
402 C5F  ASSIGN,TRIB(7)=XX(4);
403      ACT,,C5P;
404 C5X  ASSIGN,TRIB(7)=XX(5);
405      ACT,,C5P;
406 C5P  QJUN,1;
407      ACT,XX(51),C5A;
408      ACT,XX(52),C5B;
409      ACT,XX(53),C5C;
410      ACT,XX(54),C5D;
411 C5A  ASSIGN,TRIB(4)=1;
412      ACT,,Q21;
413 C5B  ASSIGN,TRIB(4)=2;
414      ACT,,Q21;
415 C5C  ASSIGN,TRIB(4)=3;
416      ACT,,Q21;
417 C5D  ASSIGN,TRIB(4)=4;
418      ACT,,Q21;
419 ;
420 ; TRAFFIC TO NODE 3:

```

```

PUT MSG IN OUTGOING QUEUE
ASSIGN PR 4
PUT MSG IN OUTGOING QUEUE
END OF NODE 1 GENERATOR

```

```

421 C6 CREATE,EXPON(XY(15),6),,1;
422 ASSIGN,ATTRIB(2)=3,ATTRIB(3)=0,ATTRIB(5)=2,1;
423 ASSIGN,ATTRIB(7)=EXPON(XY(3),6),1;
424 ASSIGN,XY(32)=XY(32)+1,1;
425 ASSIGN,ATTRIB(6)=XY(32),1;
426 GOTO,1;
427 ACT,ATTRIB(7).LE..XY(4),C6H;
428 ACT,ATTRIB(7).GE..XY(5),C6X;
429 ACT,,,C6P;
430 C6H ASSIGN,ATTRIB(7)=XY(4);
431 ACT,,,C6P;
432 C6X ASSIGN,ATTRIB(7)=XY(5);
433 ACT,,,C6P;
434 C6P GOTO,1;
435 ACT,,XY(31),C6A;
436 ACT,,XY(52),C6B;
437 ACT,,XY(53),C6C;
438 ACT,,XY(54),C6D;
439 C6A ASSIGN,ATTRIB(4)=1;
440 ACT,,,Q23;
441 C6B ASSIGN,ATTRIB(4)=2;
442 ACT,,,Q23;
443 C6C ASSIGN,ATTRIB(4)=3;
444 ACT,,,Q23;
445 C6D ASSIGN,ATTRIB(4)=4;
446 ACT,,,Q23;
447 ;
448 ; TRAFFIC TO NODE 4;
449 C7 CREATE,EXPON(XY(16),7),,1;
450 ASSIGN,ATTRIB(2)=4,ATTRIB(3)=0,ATTRIB(5)=2,1;
451 ASSIGN,ATTRIB(7)=EXPON(XY(3),7),1;
452 ASSIGN,XY(32)=XY(32)+1,1;
453 ASSIGN,ATTRIB(6)=XY(32),1;
454 GOTO,1;
455 ACT,ATTRIB(7).LE..XY(4),C7M;
456 ACT,ATTRIB(7).GE..XY(5),C7X;

```

```

457      ACT,,,C7P;
458 C7M  ASSIGN,TRIB(7)=XX(4);
459      ACT,,,C7P;
460 C7N  ASSIGN,TRIB(7)=XX(5);
461      ACT,,,C7P;
462 C7P  GOON,1;
463      ACT,,XX(51),C7A;
464      ACT,,XX(52),C7B;
465      ACT,,XX(53),C7C;
466      ACT,,XX(54),C7D;
467 C7A  ASSIGN,TRIB(4)=1;
468      ACT,,,Q23;
469 C7B  ASSIGN,TRIB(4)=2;
470      ACT,,,Q23;
471 C7C  ASSIGN,TRIB(4)=3;
472      ACT,,,Q23;
473 C7D  ASSIGN,TRIB(4)=4;
474      ACT,,,Q23;
475 ;
476 ; TRAFFIC TO NODE 5;
477 C8  CREATE,EXPON(XX(17),8),,1;
478      ASSIGN,TRIB(2)=5,TRIB(3)=0,TRIB(5)=2,1;
479      ASSIGN,TRIB(7)=EXPON(XX(3),8),1;
480      ASSIGN,XX(32)=XX(32)+1,1;
481      ASSIGN,TRIB(6)=XX(32),1;
482      GOON,1;
483      ACT,,TRIB(7).LE.XX(4),C8M;
484      ACT,,TRIB(7).GE.XX(5),C8X;
485      ACT,,,C8P;
486 C8M  ASSIGN,TRIB(7)=XX(4);
487      ACT,,,C8P;
488 C8X  ASSIGN,TRIB(7)=XX(5);
489      ACT,,,C8P;
490 C8P  GOON,1;
491      ACT,,XX(51),C8A;
492      ACT,,XX(52),C8B;

```

```

493      ACT,,XX(53),C8C;
494      ACT,,XX(54),C8D;
495      ASSIGN,ATTRIB(4)=1;
496      ACT,,Q21;
497      ASSIGN,ATTRIB(4)=2;
498      ACT,,Q21;
499      ASSIGN,ATTRIB(4)=3;
500      ACT,,Q21;
501      ASSIGN,ATTRIB(4)=4;
502      ACT,,Q21;
503      ;
504      ; ** NODE 3 **
505      ;
506      ; TRAFFIC TO NODE 1:
507      C9  CREATE,EXPON(XX(12),1),,1;
508      ASSIGN,ATTRIB(2)=1,ATTRIB(3)=0,ATTRIB(5)=3,1;
509      ASSIGN,ATTRIB(7)=EXPON(XX(3),1),1;
510      ASSIGN,XX(33)=XX(33)+1,1;
511      ASSIGN,ATTRIB(6)=XX(33),1;
512      GOON,1;
513      ACT,,ATTRIB(7).LE.XX(4),C9M;
514      ACT,,ATTRIB(7).GE.XX(5),C9X;
515      ACT,,C9P;
516      C9M  ASSIGN,ATTRIB(7)=XX(4);
517      ACT,,C9M;
518      C9X  ASSIGN,ATTRIB(7)=XX(5);
519      ACT,,C9P;
520      C9P  GOON,1;
521      ACT,,XX(51),C9A;
522      ACT,,XX(52),C9B;
523      ACT,,XX(53),C9C;
524      ACT,,XX(54),C9D;
525      C9A  ASSIGN,ATTRIB(4)=1;
526      ACT,,Q31;
527      C9B  ASSIGN,ATTRIB(4)=2;
528      ACT,,Q31;

```

END OF NODE 2 GENERATOR

```

529 C9C ASSIGN,ATTRIB(4)=3;
530 ACT,,,Q31;
531 C9D ASSIGN,ATTRIB(4)=4;
532 ACT,,,Q31;
533 ;
534 ; TRAFFIC TO NODE 2;
535 C10 CREATE,EXPON(XX(15),2),,1;
536 ASSIGN,ATTRIB(2)=2,ATTRIB(3)=0,ATTRIB(5)=3,1;
537 ASSIGN,ATTRIB(7)=EXPON(XX(3),2),1;
538 ASSIGN,XX(33)=XX(33)+1,1;
539 ASSIGN,ATTRIB(6)=XX(33),1;
540 GOTO,1;
541 ACT,,ATTRIB(7).LE.XX(4),C10M;
542 ACT,,ATTRIB(7).GE.XX(5),C10X;
543 ACT,,,C10P;
544 C10M ASSIGN,ATTRIB(7)=XX(4);
545 ACT,,,C10P;
546 C10X ASSIGN,ATTRIB(7)=XX(5);
547 ACT,,,C10P;
548 C10P GOTO,1;
549 ACT,,XX(51),C10A;
550 ACT,,XX(52),C10B;
551 ACT,,XX(53),C10C;
552 ACT,,XX(54),C10D;
553 C10A ASSIGN,ATTRIB(4)=1;
554 ACT,,,Q32;
555 C10B ASSIGN,ATTRIB(4)=2;
556 ACT,,,Q32;
557 C10C ASSIGN,ATTRIB(4)=3;
558 ACT,,,Q32;
559 C10D ASSIGN,ATTRIB(4)=4;
560 ACT,,,Q32;
561 ;
562 ; TRAFFIC TO NODE 4;
563 C11 CREATE,EXPON(XX(18),3),,1;
564 ASSIGN,ATTRIB(2)=4,ATTRIB(3)=0,ATTRIB(5)=3,1;

```

```

365 ASSIGN,TRIB(7)=EXPON(XX(3),3),1;
366 ASSIGN,XX(33)=XX(33)+1,1;
367 ASSIGN,TRIB(6)=XX(33),1;
368 QOON,1;
369 ACT,,TRIB(7).LE.XX(4),C11M;
370 ACT,,TRIB(7).GE.XX(5),C11X;
371 ACT,,,C11P;
372 C11M ASSIGN,TRIB(7)=XX(4);
373 ACT,,,C11P;
374 C11X ASSIGN,TRIB(7)=XX(5);
375 ACT,,,C11P;
376 C11P QOON,1;
377 ACT,,X(51),C11A;
378 ACT,,X(52),C11B;
379 ACT,,X(53),C11C;
380 ACT,,X(54),C11D;
381 C11A ASSIGN,TRIB(4)=1;
382 ACT,,,Q34;
383 C11B ASSIGN,TRIB(4)=2;
384 ACT,,,Q34;
385 C11C ASSIGN,TRIB(4)=3;
386 ACT,,,Q34;
387 C11D ASSIGN,TRIB(4)=4;
388 ACT,,,Q34;
389 ;
390 ; TRAFFIC TO NODE 5;
391 C12 CREATE,EXPON(XX(19),4),,1;
392 ASSIGN,TRIB(2)=5,TRIB(3)=0,TRIB(5)=3,1;
393 ASSIGN,TRIB(7)=EXPON(XX(3),4),1;
394 ASSIGN,XX(33)=XX(33)+1,1;
395 ASSIGN,TRIB(6)=XX(33),1;
396 QOON,1;
397 ACT,,TRIB(7).LE.XX(4),C12M;
398 ACT,,TRIB(7).GE.XX(5),C12X;
399 ACT,,,C12P;
600 C12M ASSIGN,TRIB(7)=XX(4);

```

```

601 ACT,,,C12P;
602 C12X ASSIGN,ATLIB(7)=XX(5);
603 ACT,,,C12P;
604 C12P COON,1;
605 ACT,,XX(51),C12A;
606 ACT,,XX(52),C12B;
607 ACT,,XX(53),C12C;
608 ACT,,XX(54),C12D;
609 C12A ASSIGN,ATLIB(4)=1;
610 ACT,,,Q31;
611 C12B ASSIGN,ATLIB(4)=2;
612 ACT,,,Q31;
613 C12C ASSIGN,ATLIB(4)=3;
614 ACT,,,Q31;
615 C12D ASSIGN,ATLIB(4)=4;
616 ACT,,,Q31;
617 ; ** NODE 4 **
618 ;
619 ;
620 ; TRAFFIC TO NODE 1:
621 C13 CREATE,EXPON(XX(13),5),,1;
622 ASSIGN,ATLIB(2)=1,ATLIB(3)=0,ATLIB(5)=4,1;
623 ASSIGN,ATLIB(7)=EXPON(XX(3),5),1;
624 ASSIGN,XX(34)=XX(34)+1,1;
625 ASSIGN,ATLIB(6)=XX(34),1;
626 COON,1;
627 ACT,,ATLIB(7).LE.XX(4),C13M;
628 ACT,,ATLIB(7).GE.XX(5),C13X;
629 ACT,,,C13P;
630 C13M ASSIGN,ATLIB(7)=XX(4);
631 ACT,,,C13P;
632 C13X ASSIGN,ATLIB(7)=XX(5);
633 ACT,,,C13P;
634 C13P COON,1;
635 ACT,,XX(51),C13A;
636 ACT,,XX(52),C13B;

```

END OF NODE 3 GENERATOR

```

637 ACT,,XX(53),C13C;
638 ACT,,XX(54),C13D;
639 C13A ASSIGN,TRIB(4)=1;
640 ACT,,Q41;
641 C13B ASSIGN,TRIB(4)=2;
642 ACT,,Q41;
643 C13C ASSIGN,TRIB(4)=3;
644 ACT,,Q41;
645 C13D ASSIGN,TRIB(4)=4;
646 ACT,,Q41;
647 ;
648 ; TRAFFIC TO NODE 2;
649 C14 CREATE,EXPON(XX(16),6),,1;
650 ASSIGN,TRIB(2)=2,TRIB(3)=0,TRIB(5)=4,1;
651 ASSIGN,TRIB(7)=EXPON(XX(3),6),1;
652 ASSIGN,XX(34)=XX(34)+1,1;
653 ASSIGN,TRIB(6)=XX(34),1;
654 OCON,1;
655 ACT,,TRIB(7).LE.XX(4),C14M;
656 ACT,,TRIB(7).GE.XX(5),C14X;
657 ACT,,C14P;
658 C14M ASSIGN,TRIB(7)=XX(4);
659 ACT,,C14P;
660 C14X ASSIGN,TRIB(7)=XX(5);
661 ACT,,C14P;
662 C14P OCON,1;
663 ACT,,XX(51),C14A;
664 ACT,,XX(52),C14B;
665 ACT,,XX(53),C14C;
666 ACT,,XX(54),C14D;
667 C14A ASSIGN,TRIB(4)=1;
668 ACT,,Q43;
669 C14B ASSIGN,TRIB(4)=2;
670 ACT,,Q43;
671 C14C ASSIGN,TRIB(4)=3;
672 ACT,,Q43;

```

```

673 C14D ASSIGN,TRIB(4)=4;
674 ACT,,,Q43;
675 ;
676 ; TRAFFIC TO NODE 3:
677 C15 CREATE,EXPON(X(18),7),,1;
678 ASSIGN,TRIB(2)=3,TRIB(3)=0,TRIB(5)=4,1;
679 ASSIGN,TRIB(7)=EXPON(X(3),7),1;
680 ASSIGN,,X(34)=X(34)+1,1;
681 ASSIGN,TRIB(6)=X(34),1;
682 GOON,1;
683 ACT,,TRIB(7).LE.X(4),C154;
684 ACT,,TRIB(7).GE.X(5),C15X;
685 ACT,,,C15P;
686 C15I ASSIGN,TRIB(7)=X(4);
687 ACT,,,C15P;
688 C15X ASSIGN,TRIB(7)=X(5);
689 ACT,,,C15P;
690 C15P GOON,1;
691 ACT,,X(51),C15A;
692 ACT,,X(52),C15B;
693 ACT,,X(53),C15C;
694 ACT,,X(54),C15D;
695 C15A ASSIGN,TRIB(4)=1;
696 ACT,,,Q43;
697 C15B ASSIGN,TRIB(4)=2;
698 ACT,,,Q43;
699 C15C ASSIGN,TRIB(4)=3;
700 ACT,,,Q43;
701 C15D ASSIGN,TRIB(4)=4;
702 ACT,,,Q43;
703 ;
704 ; TRAFFIC TO NODE 5:
705 C16 CREATE,EXPON(X(20),8),,1;
706 ASSIGN,TRIB(2)=5,TRIB(3)=0,TRIB(5)=4,1;
707 ASSIGN,TRIB(7)=EXPON(X(3),8),1;
708 ASSIGN,,X(34)=X(34)+1,1;

```

```

709 ASSIGN,TRIB(6)=XX(34),1;
710 GOUN,1;
711 ACT,,TRIB(7).LE.XX(4),C16N;
712 ACT,,TRIB(7).GE.XX(5),C16X;
713 ACT,,,C16P;
714 C16N ASSIGN,TRIB(7)=XX(4);
715 ACT,,,C16P;
716 C16X ASSIGN,TRIB(7)=XX(5);
717 ACT,,,C16P;
718 GOUN,1;
719 ACT,,XX(51),C16A;
720 ACT,,XX(52),C16B;
721 ACT,,XX(53),C16C;
722 ACT,,XX(54),C16D;
723 C16A ASSIGN,TRIB(4)=1;
724 ACT,,,Q41;
725 C16B ASSIGN,TRIB(4)=2;
726 ACT,,,Q41;
727 C16C ASSIGN,TRIB(4)=3;
728 ACT,,,Q41;
729 C16D ASSIGN,TRIB(4)=4;
730 ACT,,,Q41;
731 ;
732 ; ** NODE 5 **
733 ;
734 ; TRAFFIC TO NODE 1:
735 C17 CREATE,EXPON(XX(14),1),1;
736 ASSIGN,TRIB(2)=1,TRIB(3)=0,TRIB(5)=5,1;
737 ASSIGN,TRIB(7)=EXPON(XX(3),1),1;
738 ASSIGN,XX(35)=XX(35)+1,1;
739 ASSIGN,TRIB(6)=XX(35),1;
740 GOUN,1;
741 ACT,,TRIB(7).LE.XX(4),C17N;
742 ACT,,TRIB(7).GE.XX(5),C17X;
743 ACT,,,C17P;
744 C17N ASSIGN,TRIB(7)=XX(4);

```

END OF NODE 4 GENERATOR

```

745      ACT,,C17P;
746 C17X ASSIGN,TRIB(7)=XX(5);
747      ACT,,C17P;
748 C17P  GUN,1;
749      ACT,,XX(51),C17A;
750      ACT,,XX(52),C17B;
751      ACT,,XX(53),C17C;
752      ACT,,XX(54),C17D;
753 C17A  ASSIGN,TRIB(4)=1;
754      ACT,,Q51;
755 C17B  ASSIGN,TRIB(4)=2;
756      ACT,,Q51;
757 C17C  ASSIGN,TRIB(4)=3;
758      ACT,,Q51;
759 C17D  ASSIGN,TRIB(4)=4;
760      ACT,,Q51;
761 ;
762 ; TRAFFIC TO NODE 2;
763 C18  CREATE,EXPON(XX(17),2),,1;
764      ASSIGN,TRIB(2)=2,TRIB(3)=0,TRIB(5)=5,1;
765      ASSIGN,TRIB(7)=EXPON(XX(3),2),1;
766      ASSIGN,XX(35)=XX(35)+1,1;
767      ASSIGN,TRIB(6)=XX(35),1;
768      GUN,1;
769      ACT,,TRIB(7).LE.XX(4),C18M;
770      ACT,,TRIB(7).GE.XX(5),C18X;
771      ACT,,C18P;
772 C18M  ASSIGN,TRIB(7)=XX(4);
773      ACT,,C18P;
774 C18X  ASSIGN,TRIB(7)=XX(5);
775      ACT,,C18P;
776 C18P  GUN,1;
777      ACT,,XX(51),C18A;
778      ACT,,XX(52),C18B;
779      ACT,,XX(53),C18C;
780      ACT,,XX(54),C18D;

```

```

781 C18A ASSIGN,ATTRIB(4)=1;
782 ACT,,,Q51;
783 C18B ASSIGN,ATTRIB(4)=2;
784 ACT,,,Q15;
785 C18C ASSIGN,ATTRIB(4)=3;
786 ACT,,,Q15;
787 C18D ASSIGN,ATTRIB(4)=4;
788 ACT,,,Q51;
789 ;
790 ; TRAFFIC TO NODE 3;
791 C19 CREATE,EXPON(XX(19),3),,1;
792 ASSIGN,ATTRIB(2)=3,ATTRIB(3)=0,ATTRIB(5)=5,1;
793 ASSIGN,ATTRIB(7)=EXPON(XX(3),3),1;
794 ASSIGN,XX(35)=XX(35)+1,1;
795 ASSIGN,ATTRIB(6)=XX(35),1;
796 GOON,1;
797 ACT,,ATTRIB(7).LE.XX(4),C19M;
798 ACT,,ATTRIB(7).GE.XX(5),C19X;
799 ACT,,,C19P;
800 C19M ASSIGN,ATTRIB(7)=XX(4);
801 ACT,,,C19P;
802 C19X ASSIGN,ATTRIB(7)=XX(5);
803 ACT,,,C19P;
804 C19P GOON,1;
805 ACT,,XX(51),C19A;
806 ACT,,XX(52),C19B;
807 ACT,,XX(53),C19C;
808 ACT,,XX(54),C19D;
809 C19A ASSIGN,ATTRIB(4)=1;
810 ACT,,,Q51;
811 C19B ASSIGN,ATTRIB(4)=2;
812 ACT,,,Q51;
813 C19C ASSIGN,ATTRIB(4)=3;
814 ACT,,,Q51;
815 C19D ASSIGN,ATTRIB(4)=4;
816 ACT,,,Q51;

```

```

817 ;
818 ; TRAFFIC TO NODE 4;
819 C20 CREATE,EXTON(XX(20),4),,1;
820 ASSIGN,TRIB(2)=4,TRIB(3)=0,TRIB(5)=5,1;
821 ASSIGN,TRIB(7)=EXON(XX(3),4),1;
822 ASSIGN,XX(35)=XX(35)+1,1;
823 ASSIGN,TRIB(6)=XX(35),1;
824 GOUN,1;
825 ACT,TRIB(7).LE,XX(4),C20H;
826 ACT,TRIB(7).GE,XX(5),C20X;
827 ACT,,,C20P;
828 C20H ASSIGN,TRIB(7)=XX(4);
829 ACT,,,C20P;
830 C20X ASSIGN,TRIB(7)=XX(5);
831 ACT,,,C20P;
832 C20P GOUN,1;
833 ACT,XX(51),C20A;
834 ACT,XX(52),C20B;
835 ACT,XX(53),C20C;
836 ACT,XX(54),C20D;
837 C20A ASSIGN,TRIB(4)=1;
838 ACT,,,Q51;
839 C20B ASSIGN,TRIB(4)=2;
840 ACT,,,Q51;
841 C20C ASSIGN,TRIB(4)=3;
842 ACT,,,Q51;
843 C20D ASSIGN,TRIB(4)=4;
844 ACT,,,Q51;
845 ;
846 ;

```

END OF NODE 5 GENERATOR

END OF MESSAGE GENERATION

```

847 ; *****
848 ; *
849 ; *
850 ; *
851 ; *
852 ;
853 ; ** NODE 1 **
854 ;
855 ; INCOMING MESSAGES ARE CHECKED FOR DESTINATION. IF THE DESTINATION
856 ; IS NODE 1 OR IF THERE ARE NO ROUTING INSTRUCTIONS, THE MESSAGE
857 ; IS PASSED TO STATISTICS COLLECTION. IF THE DESTINATION IS OTHER
858 ; THAN NODE 1, MESSAGE IS PUT INTO QUEUE FOR TRANSMISSION.
859 ; THERE IS A SEPARATE QUEUE FOR EACH OUTGOING LINK.
860 ;
861 ; QUN,1;
862 ; ACT,X(9);
863 ; QUN,1;
864 ; ACT,ATTRIB(2).EQ.2,Q12;
865 ; ACT,ATTRIB(2).EQ.3,Q13;
866 ; ACT,ATTRIB(2).EQ.4,Q14;
867 ; ACT,ATTRIB(2).EQ.5,Q15;
868 ; ACT,,,TOT;
869 ;
870 ;
871 ; ** NODE 2 **
872 ;
873 ; N21
874 ; QUN,1;
875 ; ACT,X(9);
876 ; QUN,1;
877 ; ACT,ATTRIB(2).EQ.1,Q21;
878 ; ACT,ATTRIB(2).EQ.3,Q23;
879 ; ACT,ATTRIB(2).EQ.4,Q23;
880 ; ACT,ATTRIB(2).EQ.5,Q21;
881 ; ACT,,,TOT;
882 ;

```

 *
 *
 *
 *
 ** NODE 1 **
 INCOMING MESSAGES ARE CHECKED FOR DESTINATION. IF THE DESTINATION
 IS NODE 1 OR IF THERE ARE NO ROUTING INSTRUCTIONS, THE MESSAGE
 IS PASSED TO STATISTICS COLLECTION. IF THE DESTINATION IS OTHER
 THAN NODE 1, MESSAGE IS PUT INTO QUEUE FOR TRANSMISSION.
 THERE IS A SEPARATE QUEUE FOR EACH OUTGOING LINK.
 QUN,1;
 ACT,X(9);
 QUN,1;
 ACT,ATTRIB(2).EQ.2,Q12;
 ACT,ATTRIB(2).EQ.3,Q13;
 ACT,ATTRIB(2).EQ.4,Q14;
 ACT,ATTRIB(2).EQ.5,Q15;
 ACT,,,TOT;
 END OF NODE 1
 ** NODE 2 **
 N21
 QUN,1;
 ACT,X(9);
 QUN,1;
 ACT,ATTRIB(2).EQ.1,Q21;
 ACT,ATTRIB(2).EQ.3,Q23;
 ACT,ATTRIB(2).EQ.4,Q23;
 ACT,ATTRIB(2).EQ.5,Q21;
 ACT,,,TOT;
 END OF NODE 2

```

883 ; ** NODE 3 **
884 ;
885 N31      COXN,1;
886      ACT,XX(9);
887      COXN,1;
888      ACT,,ATTRIB(2).EQ.1,Q31;
889      ACT,,ATTRIB(2).EQ.2,Q32;
890      ACT,,ATTRIB(2).EQ.4,Q34;
891      ACT,,ATTRIB(2).EQ.5,Q31;
892      ACT,,,TUF;
893 ;
894 ;
895 ; ** NODE 4 **
896 ;
897 N41      COXN,1;
898      ACT,XX(9);
899      COXN,1;
900      ACT,,ATTRIB(2).EQ.1,Q41;
901      ACT,,ATTRIB(2).EQ.2,Q43;
902      ACT,,ATTRIB(2).EQ.3,Q43;
903      ACT,,ATTRIB(2).EQ.5,Q41;
904      ACT,,,TUF;
905 ;
906 ;
907 ; ** NODE 5 **
908 ;
909 N51      COXN,1;
910      ACT,XX(9);
911      COXN,1;
912      ACT,,ATTRIB(2).EQ.1,Q51;
913      ACT,,ATTRIB(2).EQ.2,Q51;
914      ACT,,ATTRIB(2).EQ.3,Q51;
915      ACT,,ATTRIB(2).EQ.4,Q51;
916      ACT,,,TUF;
917 ;
918 ;

```

END OF NODE 3

END OF NODE 4

END OF NODE 5

```

919 ;
920 ;
921 ;
922 ; *****
923 ; *
924 ; *
925 ; *
926 ;
927 ;
928 ;
929 ;
930 ;
931 ;
932 ; ** NODE 1 **
933 ;
934 Q12 COUN,1;
935 ACT,,,CH12;
936 CH12 AWAIT(1/25),LNK12,BALK(OV12),1;
937 ACT,TRIB(7)/XX(1),1-XX(38),F12;
938 ACT,TRIB(7)/XX(1),XX(38),F12A;
939 F12A COUN,1;
940 ACT,XX(8)/XX(1),,F12B;
941 F12B COUN,1;
942 ACT,TRIB(7)/XX(1),1-XX(38),F12;
943 ACT,TRIB(7)/XX(1),XX(38),F12A;
944 F12 COUN,1;
945 ACT,XX(8)/XX(1),,F12F;
946 F12F FREE,LNK12,1;
947 ACT,,,N21;
948 ;
949 Q13 COUN,1;
950 ACT,,,CH13;
951 CH13 AWAIT(2/25),LNK13,BALK(OV13),1;
952 ACT,TRIB(7)/XX(1),1-XX(39),F13;
953 ACT,TRIB(7)/XX(1),XX(39),F13A;
954 F13A COUN,1;

```

 *
 *
 *

 NODE-TO-NODE TRANSFER

 *
 *
 *

 MESSAGES READY FOR TRANSMISSION ARE KEPT IN AN OUTGOING QUEUE AND
 REMOVED ONE AT A TIME FOR TRANSMISSION USING A FIFO DISCIPLINE.
 MESSAGE TRANSMISSION TIME DEPENDS ON MESSAGE LENGTH AND
 CHANNEL RATE.

 ** NODE 1 **

 Q12 COUN,1;
 ACT,,,CH12;
 CH12 AWAIT(1/25),LNK12,BALK(OV12),1;
 ACT,TRIB(7)/XX(1),1-XX(38),F12;
 ACT,TRIB(7)/XX(1),XX(38),F12A;
 F12A COUN,1;
 ACT,XX(8)/XX(1),,F12B;
 F12B COUN,1;
 ACT,TRIB(7)/XX(1),1-XX(38),F12;
 ACT,TRIB(7)/XX(1),XX(38),F12A;
 F12 COUN,1;
 ACT,XX(8)/XX(1),,F12F;
 F12F FREE,LNK12,1;
 ACT,,,N21;

 Q13 COUN,1;
 ACT,,,CH13;
 CH13 AWAIT(2/25),LNK13,BALK(OV13),1;
 ACT,TRIB(7)/XX(1),1-XX(39),F13;
 ACT,TRIB(7)/XX(1),XX(39),F13A;
 F13A COUN,1;

TRANSMIT WITH NO ERRORS
 RETRANSMIT WITH NO ERRORS
 TRANSMIT WITH ERRORS
 RETRANSMIT WITH ERRORS
 TRANSMIT ACK
 RELEASE THE LINK
 NODE LABEL N21 DEFINES TOPOLOGY

CAPTURE THE LINK
 TRANSMIT WITH NO ERRORS
 TRANSMIT WITH ERRORS

955	ACT,XX(8)/XX(1),,F13B;		TRANSMIT NAK
956	GOON,1;		
957	ACT,TRIB(7)/XX(1),1-XX(39),F13;		RETRANSMIT WITH NO ERRORS
958	ACT,TRIB(7)/XX(1),XX(39),F13A;		RETRANSMIT WITH ERRORS
959	GOON,1;		
960	ACT,XX(8)/XX(1),,F13F;		TRANSMIT ACK
961	FREE,LINK13,1;		RELEASE THE LINK
962	ACT,,,N31;		MODE LABEL NX1 DEFINES TOPOLOGY
963	;		
964	GOON,1;		
965	ACT,,,CH14;		
966	AWAIT(3/25),LINK14,BALK(OV14),1;		CAPTURE THE LINK
967	ACT,TRIB(7)/XX(1),1-XX(40),F14;		TRANSMIT WITH NO ERRORS
968	ACT,TRIB(7)/XX(1),XX(40),F14A;		TRANSMIT WITH ERRORS
969	GOON,1;		
970	ACT,XX(8)/XX(1),,F14B;		TRANSMIT NAK
971	GOON,1;		
972	ACT,TRIB(7)/XX(1),1-XX(40),F14;		RETRANSMIT WITH NO ERRORS
973	ACT,TRIB(7)/XX(1),XX(40),F14A;		RETRANSMIT WITH ERRORS
974	GOON,1;		
975	ACT,XX(8)/XX(1),,F14F;		TRANSMIT ACK
976	FREE,LINK14,1;		RELEASE THE LINK
977	ACT,,,N41;		MODE LABEL NX1 DEFINES TOPOLOGY
978	;		
979	GOON,1;		
980	ACT,,,CH15;		
981	AWAIT(4/25),LINK15,BALK(OV15),1;		CAPTURE THE LINK
982	ACT,TRIB(7)/XX(1),1-XX(41),F15;		TRANSMIT WITH NO ERRORS
983	ACT,TRIB(7)/XX(1),XX(41),F15A;		TRANSMIT WITH ERRORS
984	GOON,1;		
985	ACT,XX(8)/XX(1),,F15B;		TRANSMIT NAK
986	GOON,1;		
987	ACT,TRIB(7)/XX(1),1-XX(41),F15;		RETRANSMIT WITH NO ERRORS
988	ACT,TRIB(7)/XX(1),XX(41),F15A;		RETRANSMIT WITH ERRORS
989	GOON,1;		
990	ACT,XX(8)/XX(1),,F15F;		TRANSMIT ACK

RELEASE THE LINK
NODE LABEL NX1 DEFINES TOPOLOGY

END OF NODE 1

```

991 F15F FREE,LNK15,1;
992 ACT,,N51;
993 ;
994 ;
995 ;
996 ; ** NODE 2 **
997 ;
998 Q21 QCN,1;
999 ACT,,Q121;
1000 QH21 WAIT(5/25),LNK21,BULK(OV21),1;
1001 ACT,TRIB(7)/XX(1),1-XX(42),F21;
1002 ACT,TRIB(7)/XX(1),XX(42),F21A;
1003 F21A QCN,1;
1004 ACT,XX(8)/XX(1),,F21B;
1005 F21B QCN,1;
1006 ACT,TRIB(7)/XX(1),1-XX(42),F21;
1007 ACT,TRIB(7)/XX(1),XX(42),F21A;
1008 F21 QCN,1;
1009 ACT,XX(8)/XX(1),,F21F;
1010 F21F FREE,LNK21,1;
1011 ACT,,N11;
1012 ;
1013 Q23 QCN,1;
1014 ACT,,Q123;
1015 Q123 WAIT(6/25),LNK23,BULK(OV23),1;
1016 ACT,TRIB(7)/XX(1),1-XX(43),F23;
1017 ACT,TRIB(7)/XX(1),XX(43),F23A;
1018 F23A QCN,1;
1019 ACT,XX(8)/XX(1),,F23B;
1020 F23B QCN,1;
1021 ACT,TRIB(7)/XX(1),1-XX(43),F23;
1022 ACT,TRIB(7)/XX(1),XX(43),F23A;
1023 F23 QCN,1;
1024 ACT,XX(8)/XX(1),,F23F;
1025 F23F FREE,LNK23,1;
1026 ACT,,N31;

```

END OF NODE 2

```

1027 ;
1028 ;
1029 ;
1030 ; ** NODE 3 **
1031 ;
1032 Q31 GOON,1;
1033 ACT,,Q1B1;
1034 QH31 AWAIT(7/25),LINK31,BALK(OV31),1;
1035 ACT,TRIB(7)/XX(1),1-XX(44),F31;
1036 ACT,TRIB(7)/XX(1),XX(44),F31A;
1037 F31A GOON,1;
1038 ACT,XX(8)/XX(1),,F31B;
1039 F31B GOON,1;
1040 ACT,TRIB(7)/XX(1),1-XX(44),F31;
1041 ACT,TRIB(7)/XX(1),XX(44),F31A;
1042 F31 GOON,1;
1043 ACT,XX(8)/XX(1),,F31F;
1044 F31F FREE,LINK31,1;
1045 ACT,,N11;
1046 ;
1047 Q32 GOON,1;
1048 ACT,,Q1B2;
1049 QH32 AWAIT(8/25),LINK32,BALK(OV32),1;
1050 ACT,TRIB(7)/XX(1),1-XX(45),F32;
1051 ACT,TRIB(7)/XX(1),XX(45),F32A;
1052 F32A GOON,1;
1053 ACT,XX(8)/XX(1),,F32B;
1054 F32B GOON,1;
1055 ACT,TRIB(7)/XX(1),1-XX(45),F32;
1056 ACT,TRIB(7)/XX(1),XX(45),F32A;
1057 F32 GOON,1;
1058 ACT,XX(8)/XX(1),,F32F;
1059 F32F FREE,LINK32,1;
1060 ACT,,N21;
1061 ;
1062 Q34 GOON,1;

```

```

1063      ACT,,,Q134;
1064 Q134  WAIT(9/25),LINK34,BALK(OW34),1;
1065      ACT,TRIB(7)/XX(1),1-XX(46),F34;
1066      ACT,TRIB(7)/XX(1),XX(46),F34A;
1067 F34A  GOUN,1;
1068      ACT,XX(8)/XX(1),,F34B;
1069 F34B  GOUN,1;
1070      ACT,TRIB(7)/XX(1),1-XX(46),F34;
1071      ACT,TRIB(7)/XX(1),XX(46),F34A;
1072 F34   GOUN,1;
1073      ACT,XX(8)/XX(1),,F34F;
1074 F34F  FREE,LINK34,1;
1075      ACT,,,N41;
1076 ;
1077 ;
1078 ;
1079 ; ** NODE 4 **
1080 ;
1081 Q41   GOUN,1;
1082      ACT,,,Q141;
1083 CH41  WAIT(10/25),LINK41,BALK(OW41),1;
1084      ACT,TRIB(7)/XX(1),1-XX(47),F41;
1085      ACT,TRIB(7)/XX(1),XX(47),F41A;
1086 F41A  GOUN,1;
1087      ACT,XX(8)/XX(1),,F41B;
1088 F41B  GOUN,1;
1089      ACT,TRIB(7)/XX(1),1-XX(47),F41;
1090      ACT,TRIB(7)/XX(1),XX(47),F41A;
1091 F41   GOUN,1;
1092      ACT,XX(8)/XX(1),,F41F;
1093 F41F  FREE,LINK41,1;
1094      ACT,,,N11;
1095 ;
1096 Q43   GOUN,1;
1097      ACT,,,Q143;
1098 CH43  WAIT(11/25),LINK43,BALK(OW43),1;

```

END OF NODE 3

```

1099      ACT,ATRI(7)/XX(1),1-XX(48),F43;
1100      ACT,ATRI(7)/XX(1),XX(48),F43A;
1101      GOON,1;
1102      ACT,XX(8)/XX(1),F43B;
1103      GOON,1;
1104      ACT,ATRI(7)/XX(1),1-XX(48),F43;
1105      ACT,ATRI(7)/XX(1),XX(48),F43A;
1106      GOON,1;
1107      ACT,XX(8)/XX(1),F43F;
1108      FREE,LINK43,1;
1109      ACT,,,N31;
1110      ;
1111      ;
1112      ;
1113      ; ** NODE 5 **
1114      ;
1115      Q51      GOON,1;
1116      ACT,,,Q51;
1117      Q51      WAIT(12/25),LINK51,BALK(OV51),1;
1118      ACT,ATRI(7)/XX(1),1-XX(49),F51;
1119      ACT,ATRI(7)/XX(1),XX(49),F51A;
1120      F51A      GOON,1;
1121      ACT,XX(8)/XX(1),F51B;
1122      GOON,1;
1123      ACT,ATRI(7)/XX(1),1-XX(49),F51;
1124      ACT,ATRI(7)/XX(1),XX(49),F51A;
1125      F51      GOON,1;
1126      ACT,XX(8)/XX(1),F51F;
1127      F51F      FREE,LINK51,1;
1128      ACT,,,N11;
1129      ;
1130      ;
1131      ;
1132      ;

```

END OF NODE 4

END OF NODE 5

END OF NODE-TO-NODE TRANSFER

```

1133 ; *****
1134 ; *
1135 ; *
1136 ; *
1137 ; *
1138 ;
1139 ; STATISTICS ARE COLLECTED ON THE TOTAL TIME THAT A MESSAGE
1140 ; IS IN THE NETWORK. STATISTICS ARE COLLECTED AND REPORTED
1141 ; BY PRECEDENCE LEVEL..
1142 ;
1143 ; SLAM SUMMARY REPORT LABELS:
1144 ;
1145 ; TIS TOT = TOTAL TIME IN SYSTEM FOR ALL MESSAGES
1146 ; TIS PR 1 = TOTAL TIME IN SYSTEM FOR PRECEDENCE 1 MESSAGES ONLY
1147 ; TIS PR 2 = TOTAL TIME IN SYSTEM FOR PRECEDENCE 2 MESSAGES ONLY
1148 ; TIS PR 3 = TOTAL TIME IN SYSTEM FOR PRECEDENCE 3 MESSAGES ONLY
1149 ; TIS PR 4 = TOTAL TIME IN SYSTEM FOR PRECEDENCE 4 MESSAGES ONLY
1150 ; OVR 12 = MEAN TIME BETWEEN OVERFLOW ON LINK12
1151 ; OVR 13 = MEAN TIME BETWEEN OVERFLOW ON LINK13
1152 ; OVR 14 = MEAN TIME BETWEEN OVERFLOW ON LINK14
1153 ; OVR 15 = MEAN TIME BETWEEN OVERFLOW ON LINK15
1154 ; OVR 21 = MEAN TIME BETWEEN OVERFLOW ON LINK21
1155 ; OVR 23 = MEAN TIME BETWEEN OVERFLOW ON LINK23
1156 ; OVR 31 = MEAN TIME BETWEEN OVERFLOW ON LINK31
1157 ; OVR 32 = MEAN TIME BETWEEN OVERFLOW ON LINK32
1158 ; OVR 34 = MEAN TIME BETWEEN OVERFLOW ON LINK34
1159 ; OVR 41 = MEAN TIME BETWEEN OVERFLOW ON LINK41
1160 ; OVR 43 = MEAN TIME BETWEEN OVERFLOW ON LINK43
1161 ; OVR 51 = MEAN TIME BETWEEN OVERFLOW ON LINK51
1162 ;
1163 ; TOT COLCT,INT(1),TIS TOT,1;
1164 ; ACT,,ATTRIB(4).EQ.1,TOT1;
1165 ; ACT,,ATTRIB(4).EQ.2,TOT2;
1166 ; ACT,,ATTRIB(4).EQ.3,TOT3;
1167 ; ACT,,ATTRIB(4).EQ.4,TOT4;
1168 ; TOT1 COLCT,INT(1),TIS PR 1,1;

```

```

1169 ACT,...,IOL5;
1170 TUF2 COLCT,INT(1),TIS PR 2,,1;
1171 ACT,...,IOL5;
1172 TUF3 COLCT,INT(1),TIS PR 3,,1;
1173 ACT,...,IOL5;
1174 TUF4 COLCT,INT(1),TIS PR 4,,1;
1175 TUF5 COLCT,INT(1),TIS PR 4,,1;
1176 COLN,2;
1177 ACT,...,EXT;
1178 ACT;
1179 TERM,1000;
1180 ;
1181 ;
1182 ;
1183 ;
1184 OV12 COLCT,BETWEEN,OVR 12,,1;
1185 TERM;
1186 OV13 COLCT,BETWEEN,OVR 13,,1;
1187 TERM;
1188 OV14 COLCT,BETWEEN,OVR 14,,1;
1189 TERM;
1190 OV15 COLCT,BETWEEN,OVR 15,,1;
1191 TERM;
1192 OV21 COLCT,BETWEEN,OVR 21,,1;
1193 TERM;
1194 OV23 COLCT,BETWEEN,OVR 23,,1;
1195 TERM;
1196 OV31 COLCT,BETWEEN,OVR 31,,1;
1197 TERM;
1198 OV32 COLCT,BETWEEN,OVR 32,,1;
1199 TERM;
1200 OV34 COLCT,BETWEEN,OVR 34,,1;
1201 TERM;
1202 OV41 COLCT,BETWEEN,OVR 41,,1;
1203 TERM;
1204 OV43 COLCT,BETWEEN,OVR 43,,1;

```

PERFORM EXTERNAL EFFECTS

ALL MESSAGES ARE COUNTED

DISCARD OVERFLOW

STATISTICS ARE COLLECTED ON MESSAGES WHICH ARE REJECTED AT A NODE
FOR INSUFFICIENT QUEUE CAPACITY.

```

1205      TERM;
1206      COLCT,BEINEN,OWR 51,,1;
1207      TERM;
1208      ;
1209      ;
1210      ;
1211      ;
1212      ;
1213      ;
1214      ;
1215      ;
1216      EXI      COUN,1;
1217      ACT,,EXR;      PERFORM RESPONSE TRAFFIC
1218      ;
1219      ;
1220      ;
1221      ;
1222      ;
1223      ;
1224      ;
1225      EXR      COUN,1;
1226      ACT,,ATTRIB(3).EQ.0,EXR1;      PROCESS ONLY ORIGINAL MESSAGES
1227      ACT;
1228      TERM;
1229      ;
1230      ;
1231      ;
1232      ;
1233      EXR1      ASSIGN,ATTRIB(1)=TRON,ATTRIB(3)=1,1;
1234      ACT,,ATTRIB(2).EQ.1,EX1;
1235      ACT,,ATTRIB(2).EQ.2,EX2;
1236      ACT,,ATTRIB(2).EQ.3,EX3;
1237      ACT,,ATTRIB(2).EQ.4,EX4;
1238      ACT,,ATTRIB(2).EQ.5,EX5;

```

EXTERNAL EFFECTS

*** RESPONSE TRAFFIC ***

RESPONSE TRAFFIC IS GENERATED BASED ON PRIORITY LEVEL
OF ORIGINAL MESSAGE. ARRIVING RESPONSE MESSAGES DO NOT
GENERATE ANOTHER RESPONSE AND ARE DISCARDED.

```

1239 ;
1240 ; RESPONSE TRAFFIC IS RETURNED TO EACH NODE FOR TRANSMISSION.
1241 ; ORIGIN OF ORIGINAL MESSAGE IS USED AS DESTINATION.
1242 ; PREFERENCE OF MESSAGE DETERMINES HOW MUCH RESPONSE
1243 ; TRAFFIC IS GENERATED.
1244 ;
1245 ; ** NODE 1 **
1246 ;
1247 EX1 ASSIGN,ATTRIB(2)=ATTRIB(5);
1248 QUN,1;
1249 ACT,,ATTRIB(4).EQ.1,EX11;
1250 ACT,,ATTRIB(4).EQ.2,EX12;
1251 ACT,,ATTRIB(4).EQ.3,EX13;
1252 ACT,,ATTRIB(4).EQ.4,EX14;
1253 EX11 QUN,1;
1254 ACT,,XY(61),N11;
1255 ACT,,1-XX(61);
1256 TERM;
1257 EX12 QUN,1;
1258 ACT,,XY(62),N11;
1259 ACT,,1-XX(62);
1260 TERM;
1261 EX13 QUN,1;
1262 ACT,,XY(63),N11;
1263 ACT,,1-XX(63);
1264 TERM;
1265 EX14 QUN,1;
1266 ACT,,XY(64),N11;
1267 ACT,,1-XX(64);
1268 TERM;
1269 ;
1270 ;
1271 ; ** NODE 2 **
1272 ;
1273 EX2 ASSIGN,ATTRIB(2)=ATTRIB(5);
1274 QUN,1;

```

SORT ON PREFERENCE

END OF NODE 1

```

1275      ACT,,ATTRIB(4).EQ.1,EX21;
1276      ACT,,ATTRIB(4).EQ.2,EX22;
1277      ACT,,ATTRIB(4).EQ.3,EX23;
1278      ACT,,ATTRIB(4).EQ.4,EX24;
1279      GOON,1;
1280      ACT,,XX(61),N21;
1281      ACT,,1-XX(61);
1282      TERN;
1283      EX22 GOON,1;
1284      ACT,,XX(62),N21;
1285      ACT,,1-XX(62);
1286      TERN;
1287      EX23 GOON,1;
1288      ACT,,XX(63),N21;
1289      ACT,,1-XX(63);
1290      TERN;
1291      EX24 GOON,1;
1292      ACT,,XX(64),N21;
1293      ACT,,1-XX(64);
1294      TERN;
1295      ;
1296      ;
1297      ;
1298      ; ** NODE 3 **
1299      ;
1300      EX3  ASSIGN,ATTRIB(2)=ATTRIB(5);
1301      GOON,1;
1302      ACT,,ATTRIB(4).EQ.1,EX31;
1303      ACT,,ATTRIB(4).EQ.2,EX32;
1304      ACT,,ATTRIB(4).EQ.3,EX33;
1305      ACT,,ATTRIB(4).EQ.4,EX34;
1306      EX31 GOON,1;
1307      ACT,,XX(61),N31;
1308      ACT,,1-XX(61);
1309      TERN;
1310      EX32 GOON,1;

```

END OF NODE 2

```

1311 ACT,,XX(62),N31;
1312 ACT,,1-XX(62);
1313 TERM;
1314 EX33 GOUN,1;
1315 ACT,,XX(63),N31;
1316 ACT,,1-XX(63);
1317 TERM;
1318 EX34 GOUN,1;
1319 ACT,,XX(64),N31;
1320 ACT,,1-XX(64);
1321 TERM;
1322 ;
1323 ;
1324 ;
1325 ; ** NODE 4 **
1326 ;
1327 EX4 ASSIGN,ATTRIB(2)=ATTRIB(5);
1328 GOUN,1;
1329 ACT,,ATTRIB(4).EQ.1,EX41;
1330 ACT,,ATTRIB(4).EQ.2,EX42;
1331 ACT,,ATTRIB(4).EQ.3,EX43;
1332 ACT,,ATTRIB(4).EQ.4,EX44;
1333 EX41 GOUN,1;
1334 ACT,,XX(61),N41;
1335 ACT,,1-XX(61);
1336 TERM;
1337 EX42 GOUN,1;
1338 ACT,,XX(62),N41;
1339 ACT,,1-XX(62);
1340 TERM;
1341 EX43 GOUN,1;
1342 ACT,,XX(63),N41;
1343 ACT,,1-XX(63);
1344 TERM;
1345 EX44 GOUN,1;
1346 ACT,,XX(64),N41;

```

END OF NODE 3

```

1347 ACT,,1-XX(64);
1348 TERM;
1349 ;
1350 ;
1351 ;
1352 ; ** NODE 5 **
1353 ;
1354 EX5 ASSIGN,ATTRIB(2)=ATTRIB(5);
1355 GCON,1;
1356 ACT,,ATTRIB(4).EQ.1,EX51;
1357 ACT,,ATTRIB(4).EQ.2,EX52;
1358 ACT,,ATTRIB(4).EQ.3,EX53;
1359 ACT,,ATTRIB(4).EQ.4,EX54;
1360 GCON,1;
1361 ACT,,XX(61),N51;
1362 ACT,,1-XX(61);
1363 TERM;
1364 EX52 GCON,1;
1365 ACT,,XX(62),N51;
1366 ACT,,1-XX(62);
1367 TERM;
1368 EX53 GCON,1;
1369 ACT,,XX(63),N51;
1370 ACT,,1-XX(63);
1371 TERM;
1372 EX54 GCON,1;
1373 ACT,,XX(64),N51;
1374 ACT,,1-XX(64);
1375 TERM;
1376 ;
1377 ;
1378 ;
1379 ;
1380 ;
1381 ;
1382 ;

```

END OF NODE 4

END OF NODE 5

END OF RESPONSE TRAFFIC

END OF EXTERNAL EFFECTS

```

1383 ENNETWORK;
1384 ;
1385 ; END OF NETWORK DESCRIPTION;
1386 ;
1387 ;
1388 ;
1389 INIT,0; START AT TIME 0.0 SEC
1390 ;
1391 ; QUEUE PRELIMINARY IS BASED ON ATTRIBUTE 4. LOWEST VALUE OF
1392 ; ATTRIBUTE 4 QMS FIRST.
1393 PRIORITY/1,LVF(4)/2,LVF(4)/3,LVF(4)/4,LVF(4)/5,LVF(4);
1394 PRIORITY/6,LVF(4)/7,LVF(4)/8,LVF(4)/9,LVF(4)/10,LVF(4);
1395 PRIORITY/11,LVF(4)/12,LVF(4);
1396 ;
1397 ; RUN NUMBER ONE
1398 ; RND=0.1, 31.5 MSG/MIN
1399 NOWR,CLEAR,1500.; DISCARD WARM-UP PERIOD
1400 SIMULATE;

```

SLAM SUMMARY REPORT

SIMULATION PROJECT FULL MODEL P24B BY BERT GARCIA

DATE 2/25/1985 RUN NUMBER 1 OF 5

CURRENT TIME .1519E+05
STATISTICAL ARRAYS CLEARED AT TIME .1500E+04

***STATISTICS FOR VARIABLES BASED ON OBSERVATION**

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
TIS TOT	.6766E+01	.5261E+01	.7776E+00	.1418E+01	.4663E+02	9002
TIS PR 1	.6558E+01	.4879E+01	.7440E+00	.1418E+01	.3647E+02	1386
TIS PR 2	.6580E+01	.5124E+01	.7787E+00	.1418E+01	.3582E+02	1122
TIS PR 3	.6838E+01	.5293E+01	.7741E+00	.1418E+01	.4042E+02	2867
TIS PR 4	.6843E+01	.5414E+01	.7909E+00	.1418E+01	.4663E+02	3627
OWR 12		NO VALUES	RECORDED			
OWR 13		NO VALUES	RECORDED			
OWR 14		NO VALUES	RECORDED			
OWR 15		NO VALUES	RECORDED			
OWR 21		NO VALUES	RECORDED			
OWR 23		NO VALUES	RECORDED			
OWR 31		NO VALUES	RECORDED			
OWR 32		NO VALUES	RECORDED			
OWR 34		NO VALUES	RECORDED			
OWR 41		NO VALUES	RECORDED			
OWR 43		NO VALUES	RECORDED			
OWR 51		NO VALUES	RECORDED			

FILE STATISTICS

FILE NUMBER	ASSOCIATED NODE TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	AWAIT	.0644	.2980	4	0	.9547
2	AWAIT	.0573	.2777	4	0	.9121
3	AWAIT	.0516	.2499	3	0	.7791
4	AWAIT	.0408	.2436	3	0	.2880
5	AWAIT	.0463	.2576	4	0	.7289
6	AWAIT	.0574	.2681	4	0	.8738
7	AWAIT	.0691	.3294	6	0	1.0331
8	AWAIT	.0822	.3420	4	0	1.2843
9	AWAIT	.0652	.2984	4	0	.9976
10	AWAIT	.0595	.2748	3	0	.9135
11	AWAIT	.0815	.3385	3	0	1.1996
12	AWAIT	.0313	.2275	5	0	.2350
13	CALENDAR	23.7434	1.8069	32	23	1.5692

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
1	LINK12	1	.2669	.4423	1	0
2	LINK13	1	.2435	.4292	1	0
3	LINK14	1	.2715	.4448	1	0
4	LINK15	2	.5691	.7078	2	2
5	LINK21	1	.2585	.4378	1	0
6	LINK23	1	.2654	.4416	1	0
7	LINK31	1	.2601	.4387	1	0
8	LINK32	1	.2644	.4410	1	1
9	LINK34	1	.2664	.4421	1	0
10	LINK41	1	.2630	.4403	1	0
11	LINK43	1	.2834	.4507	1	0
12	LINK51	2	.5298	.6840	2	0

RESOURCE NUMBER	RESOURCE LABEL	CURRENT AVAILABLE	AVERAGE AVAILABLE	MINIMUM AVAILABLE	MAXIMUM AVAILABLE
1	LINK12	1	.7331	0	1
2	LINK13	1	.7565	0	1
3	LINK14	1	.7285	0	1
4	LINK15	0	1.4309	0	2
5	LINK21	1	.7415	0	1
6	LINK23	1	.7346	0	1
7	LINK31	1	.7399	0	1
8	LINK32	0	.7356	0	1
9	LINK34	1	.7336	0	1
10	LINK41	1	.7370	0	1
11	LINK43	1	.7166	0	1
12	LINK51	2	1.4702	0	2

1401 ;
1402 ; RUN NUMBER TWO
1403 ; RHO=0.2, 63 NSG/MIN
1404 INTLC,
1405 XX(11)=19.047619,
1406 XX(12)=19.047619,
1407 XX(13)=19.047619,
1408 XX(14)=19.047619,
1409 XX(15)=19.047619,
1410 XX(16)=19.047619,
1411 XX(17)=19.047619,
1412 XX(18)=19.047619,
1413 XX(19)=19.047619,
1414 XX(20)=19.047619;
1415 MONTR,CLEAR,750.;
1416 SIMULATE;

SLAN SUMMARY REPORT

SIMULATION PROJECT FULL MODEL P24B BY BERT GARCIA

DATE 2/25/1985 RUN NUMBER 2 OF 5

CURRENT TIME .7740E+04
STATISTICAL ARRAYS CLEARED AT TIME .7500E+03

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
TIS TOT	.9016E+01	.7594E+01	.8423E+00	.1418E+01	.1054E+03	9038
TIS PR 1	.7718E+01	.5290E+01	.6854E+00	.1418E+01	.3803E+02	1474
TIS PR 2	.8050E+01	.5802E+01	.7207E+00	.1418E+01	.4778E+02	1118
TIS PR 3	.8334E+01	.6225E+01	.7469E+00	.1418E+01	.4144E+02	2724
TIS PR 4	.1032E+02	.9355E+01	.9067E+00	.1418E+01	.1054E+03	3722
OWR 12		NO VALUES RECORDED				
OWR 13		NO VALUES RECORDED				
OWR 14		NO VALUES RECORDED				
OWR 15		NO VALUES RECORDED				
OWR 21		NO VALUES RECORDED				
OWR 23		NO VALUES RECORDED				
OWR 31		NO VALUES RECORDED				
OWR 32		NO VALUES RECORDED				
OWR 34		NO VALUES RECORDED				
OWR 41		NO VALUES RECORDED				
OWR 43		NO VALUES RECORDED				
OWR 51		NO VALUES RECORDED				

FILE STATISTICS

FILE NUMBER	ASSOCIATED MODE TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	AWAIT	.3366	.7091	5	0	2.5335
2	AWAIT	.2964	.7244	5	0	2.4345
3	AWAIT	.3781	.8352	8	1	2.8265
4	AWAIT	.3968	.9394	8	0	1.4066
5	AWAIT	.4109	.8530	6	0	3.0142
6	AWAIT	.3872	.8289	7	0	3.0244
7	AWAIT	.2516	.6545	7	0	2.1372
8	AWAIT	.3850	.7988	6	0	2.8566
9	AWAIT	.3424	.7987	6	5	2.6333
10	AWAIT	.4955	1.1244	7	0	3.8488
11	AWAIT	.3559	.8188	6	0	2.7890
12	AWAIT	.2557	.7897	8	1	1.0038
13	CALENDAR	27.3501	2.0079	35	27	.9197

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
1	LINK12	1	.5402	.4984	1	0
2	LINK13	1	.4868	.4998	1	0
3	LINK14	1	.5374	.4986	1	1
4	LINK15	2	1.1243	.8205	2	1
5	LINK21	1	.5441	.4981	1	1
6	LINK23	1	.5232	.4995	1	1
7	LINK31	1	.4704	.4991	1	0
8	LINK32	1	.5508	.4974	1	0
9	LINK34	1	.5103	.4999	1	1
10	LINK41	1	.5283	.4992	1	0
11	LINK43	1	.5069	.5000	1	0
12	LINK51	2	1.0253	.8134	2	2

RESOURCE NUMBER	RESOURCE LABEL	CURRENT AVAILABLE	AVERAGE AVAILABLE	MINIMUM AVAILABLE	MAXIMUM AVAILABLE
1	LINK12	1	.4598	0	1
2	LINK13	1	.5132	0	1
3	LINK14	0	.4626	0	1
4	LINK15	1	.8757	0	2
5	LINK21	0	.4559	0	1
6	LINK23	0	.4768	0	1
7	LINK31	1	.5296	0	1
8	LINK32	1	.4492	0	1
9	LINK34	0	.4897	0	1
10	LINK41	1	.4717	0	1
11	LINK43	1	.4931	0	1
12	LINK51	0	.9747	0	2

1

```

1417 ;
1418 ; RUN NUMBER THREE
1419 ; RHO=0.3, 94.5 MSG/IN
1420 INTLC,
1421 XX(11)=12.698413,
1422 XX(12)=12.698413,
1423 XX(13)=12.698413,
1424 XX(14)=12.698413,
1425 XX(15)=12.698413,
1426 XX(16)=12.698413,
1427 XX(17)=12.698413,
1428 XX(18)=12.698413,
1429 XX(19)=12.698413,
1430 XX(20)=12.698413;
1431 MONIR,CLEAR,500.;
1432 SIMULATE;

```

1

SLAM SUMMARY REPORT

SEULATION PROJECT FULL MODL P24B BY BERT GARCIA

DATE 2/25/1985 RUN NUMBER 3 OF 5

CURRENT TIME .5000E+04
STATISTICAL ARRAYS CLEARED AT TIME .5000E+03

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
TIS TOT	.1974E+02	.2361E+02	.1196E+01	.1418E+01	.1935E+03	9056
TIS PR 1	.9125E+01	.6044E+01	.6623E+00	.1418E+01	.3641E+02	1410
TIS PR 2	.1037E+02	.7139E+01	.6838E+00	.1418E+01	.5061E+02	1133
TIS PR 3	.1285E+02	.1007E+02	.7834E+00	.1418E+01	.6187E+02	2853
TIS PR 4	.3210E+02	.3178E+02	.9900E+00	.1418E+01	.1935E+03	3660
OWR 12		NO VALUES RECORDED				
OWR 13		NO VALUES RECORDED				
OWR 14		NO VALUES RECORDED				
OWR 15		NO VALUES RECORDED				
OWR 21		NO VALUES RECORDED				
OWR 23		NO VALUES RECORDED				
OWR 31		NO VALUES RECORDED				
OWR 32		NO VALUES RECORDED				
OWR 34		NO VALUES RECORDED				
OWR 41		NO VALUES RECORDED				
OWR 43		NO VALUES RECORDED				
OWR 51		NO VALUES RECORDED				

FILE STATISTICS

FILE NUMBER	ASSOCIATED NODE TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	AWAIT	3.4625	3.8357	16	0	16.8521
2	AWAIT	1.9199	2.6393	14	0	10.2106
3	AWAIT	1.9328	2.4967	12	0	9.7588
4	AWAIT	2.7726	3.7499	18	4	6.5798
5	AWAIT	2.4511	2.9506	14	1	12.2281
6	AWAIT	2.5016	2.7166	14	2	12.4527
7	AWAIT	1.3002	1.7604	10	0	6.9226
8	AWAIT	1.7902	2.2563	11	4	9.0888
9	AWAIT	2.4510	3.6211	16	2	12.6965
10	AWAIT	2.4364	2.7880	15	3	12.0498
11	AWAIT	2.1103	2.6652	11	0	10.8219
12	AWAIT	2.3948	3.1143	17	1	6.0693
13	CALENDAR	31.3906	1.6915	37	32	.6931

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
1	LINK12	1	.8289	.3766	1	0
2	LINK13	1	.7673	.4226	1	1
3	LINK14	1	.8165	.3871	1	1
4	LINK15	2	1.7018	.5958	2	2
5	LINK21	1	.8222	.3823	1	1
6	LINK23	1	.8471	.3599	1	1
7	LINK31	1	.7561	.4294	1	1
8	LINK32	1	.7894	.4077	1	1
9	LINK34	1	.7994	.4004	1	1
10	LINK41	1	.8255	.3796	1	1
11	LINK43	1	.7966	.4026	1	0
12	LINK51	2	1.6366	.6474	2	2

RESOURCE NUMBER	RESOURCE LABEL	CURRENT AVAILABLE	AVERAGE AVAILABLE	MINIMUM AVAILABLE	MAXIMUM AVAILABLE
1	LINK12	1	.1711	0	1
2	LINK13	0	.2327	0	1
3	LINK14	0	.1835	0	1
4	LINK15	0	.2982	0	2
5	LINK21	0	.1778	0	1
6	LINK23	0	.1529	0	1
7	LINK31	0	.2439	0	1
8	LINK32	0	.2106	0	1
9	LINK34	0	.2006	0	1
10	LINK41	0	.1745	0	1
11	LINK43	1	.2034	0	1
12	LINK51	0	.3634	0	2

1

1433 ;
 1434 ; RUN NUMBER FOUR
 1435 ; RID=0.4, 126 MSG/MIN
 1436 INTLC,
 1437 XX(11)=9.523810,
 1438 XX(12)=9.523810,
 1439 XX(13)=9.523810,
 1440 XX(14)=9.523810,
 1441 XX(15)=9.523810,
 1442 XX(16)=9.523810,
 1443 XX(17)=9.523810,
 1444 XX(18)=9.523810,
 1445 XX(19)=9.523810,
 1446 XX(20)=9.523810;
 1447 MONIR,CLEAR,325. ;
 1448 SIMULATE;

1

SLAM SUMMARY REPORT

SIMULATION PROJECT FULL MODEL P24B BY BERT GARCIA

DATE 2/25/1985 RUN NUMBER 4 OF 5

CURRENT TIME .4123E+04
STATISTICAL ARRAYS CLEARED AT TIME .3250E+03

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
TIS TOT	.7130E+02	.9119E+02	.1279E+01	.1419E+01	.5324E+03	9231
TIS PR 1	.9817E+01	.5904E+01	.6014E+00	.1419E+01	.3380E+02	1481
TIS PR 2	.1144E+02	.7792E+01	.6811E+00	.1435E+01	.7098E+02	1145
TIS PR 3	.1551E+02	.1104E+02	.7119E+00	.1596E+01	.8923E+02	2913
TIS PR 4	.1585E+03	.8926E+02	.5630E+00	.1518E+01	.5324E+03	3692
OVR 12	.2412E+02	.2426E+02	.1006E+01	.4288E+00	.5599E+02	4
OVR 13	.1096E+03	.3795E+03	.3462E+01	.5719E+00	.1531E+04	16
OVR 14	.5998E+02	.1444E+03	.2579E+01	.3929E+00	.5920E+03	17
OVR 15	.2747E+02	.1204E+03	.4383E+01	.3686E-02	.1078E+04	124
OVR 21	.6340E+02	.2263E+03	.3569E+01	.1322E+00	.1356E+04	38
OVR 23	.2865E+02	.6856E+02	.2393E+01	.4883E-01	.5911E+03	107
OVR 31	.6388E+02	.1876E+03	.2936E+01	.9643E-01	.9806E+03	35
OVR 32	.4928E+02	.1228E+03	.2493E+01	.4415E+00	.4275E+03	12
OVR 34		NO VALUES RETURNED				
OVR 41	.2793E+02	.6843E+02	.2450E+01	.4159E-01	.4305E+03	116
OVR 43		NO VALUES RETURNED				
OVR 51	.2204E+02	.4904E+02	.2225E+01	.9029E-03	.3409E+03	149

FILE STATISTICS

FILE NUMBER	ASSOCIATED NODE TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	AWAIT	10.2873	6.6839	25	17	40.4009
2	AWAIT	11.2751	6.6114	25	6	45.5524
3	AWAIT	11.9543	6.8004	25	23	49.7247
4	AWAIT	17.2197	5.1849	25	24	34.4546
5	AWAIT	13.3612	7.2497	25	12	53.4121
6	AWAIT	21.1265	3.0477	25	18	85.3529
7	AWAIT	15.6100	6.5324	25	14	61.5592
8	AWAIT	10.3123	7.4350	25	17	41.4419
9	AWAIT	8.1478	5.9228	23	11	33.7433
10	AWAIT	19.6647	4.3257	25	9	77.7917
11	AWAIT	7.3328	4.8642	22	10	29.5309
12	AWAIT	19.2712	5.0130	25	5	39.0532
13	CALENDAR	33.8154	.4449	37	34	.5934

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
1	LINK12	1	.9868	.1142	1	1
2	LINK13	1	.9791	.1431	1	1
3	LINK14	1	.9795	.1418	1	1
4	LINK15	2	2.0000	.0000	2	2
5	LINK21	1	.9828	.1300	1	1
6	LINK23	1	1.0000	.0000	1	1
7	LINK31	1	.9965	.0591	1	1
8	LINK32	1	.9735	.1607	1	1
9	LINK34	1	.9447	.2285	1	1
10	LINK41	1	1.0000	.0000	1	1
11	LINK43	1	.9686	.1744	1	1
12	LINK51	2	2.0000	.0000	2	2

RESOURCE NUMBER	RESOURCE LABEL	CURRENT AVAILABLE	AVERAGE AVAILABLE	MINIMUM AVAILABLE	MAXIMUM AVAILABLE
1	LINK12	0	.0132	0	1
2	LINK13	0	.0209	0	1
3	LINK14	0	.0205	0	1
4	LINK15	0	.0000	0	0
5	LINK21	0	.0172	0	1
6	LINK23	0	.0000	0	0
7	LINK31	0	.0035	0	1
8	LINK32	0	.0265	0	1
9	LINK34	0	.0553	0	1
10	LINK41	0	.0000	0	0
11	LINK43	0	.0314	0	1
12	LINK51	0	.0000	0	0

1

```

1449 ;
1450 ; RUN NUMBER FIVE
1451 ; RND=0.5, 158 MSG/MIN
1452 INILC,
1453 XX(11)=7.619048,
1454 XX(12)=7.619048,
1455 XX(13)=7.619048,
1456 XX(14)=7.619048,
1457 XX(15)=7.619048,
1458 XX(16)=7.619048,
1459 XX(17)=7.619048,
1460 XX(18)=7.619048,
1461 XX(19)=7.619048,
1462 XX(20)=7.619048;
1463 MUNTR,CLEAR,200.;
1464 SIMULATE;

```

1

SLAM SUMMARY REPORT

SIMULATION PROJECT FULL NAME, P24B BY BETT GARCIA

DATE 2/25/1985 RUN NUMBER 5 OF 5

CURRENT TIME .4116E+04

STATISTICAL ARRAYS CLEARED AT TIME .2000E+03

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
TIS TOT	.1023E+03	.1232E+03	.1202E+01	.1437E+01	.5973E+03	9531
TIS PR 1	.9759E+01	.6022E+01	.6171E+00	.1584E+01	.4197E+02	1471
TIS PR 2	.1051E+02	.6983E+01	.6645E+00	.1437E+01	.5617E+02	1147
TIS PR 3	.1441E+02	.1041E+02	.7222E+00	.1457E+01	.8769E+02	2932
TIS PR 4	.2281E+03	.9542E+02	.4184E+00	.1853E+01	.5973E+03	3981
OVR 12	.3159E+02	.1115E+03	.3531E+01	.2570E-01	.1006E+04	108
OVR 13	.2359E+02	.5392E+02	.2286E+01	.1047E-01	.3255E+03	154
OVR 14	.6059E+02	.2171E+03	.3584E+01	.4167E-01	.1645E+04	59
OVR 15	.1509E+02	.3458E+02	.2291E+01	.4462E-02	.2608E+03	252
OVR 21	.1764E+02	.6375E+02	.3614E+01	.6959E-02	.7537E+03	213
OVR 23	.1347E+02	.3035E+02	.2253E+01	.3135E-01	.2358E+03	280
OVR 31	.1773E+02	.3469E+02	.1957E+01	.2384E-01	.2512E+03	216
OVR 32	.2744E+02	.9426E+02	.3435E+01	.1930E-01	.8434E+03	105
OVR 34	.2336E+02	.6381E+02	.2732E+01	.9158E-03	.5543E+03	151
OVR 41	.1779E+02	.4502E+02	.2530E+01	.2927E-02	.3904E+03	220
OVR 43	.1443E+02	.4032E+02	.2785E+01	.8754E-02	.4619E+03	258
OVR 51	.9583E+01	.2416E+02	.2522E+01	.2315E-02	.2778E+03	409

FILE STATISTICS

FILE NUMBER	ASSOCIATED MODE TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	AWAIT	18.6933	5.7692	25	20	74.3945
2	AWAIT	21.5424	3.5896	25	22	87.6938
3	AWAIT	16.6497	6.5642	25	16	64.5538
4	AWAIT	21.1174	3.1953	25	24	42.3655
5	AWAIT	20.9416	4.1469	25	25	83.3425
6	AWAIT	22.3028	2.8177	25	24	89.9477
7	AWAIT	22.4283	2.2061	25	22	86.8754
8	AWAIT	18.0774	5.7808	25	9	73.9731
9	AWAIT	20.6124	4.1971	25	22	83.3879
10	AWAIT	21.9062	2.8993	25	25	86.2251
11	AWAIT	22.3862	2.8304	25	24	88.2841
12	AWAIT	21.7044	3.0831	25	25	43.1015
13	CALENDAR	33.9916	.1269	37	35	.5661

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
1	LINK12	1	1.0000	.0000	1	1
2	LINK13	1	1.0000	.0000	1	1
3	LINK14	1	.9877	.1104	1	1
4	LINK15	2	2.0000	.0000	2	2
5	LINK21	1	1.0000	.0000	1	1
6	LINK23	1	1.0000	.0000	1	1
7	LINK31	1	1.0000	.0000	1	1
8	LINK32	1	1.0000	.0000	1	1
9	LINK34	1	1.0000	.0000	1	1
10	LINK41	1	1.0000	.0000	1	1
11	LINK43	1	1.0000	.0000	1	1
12	LINK51	2	2.0000	.0000	2	2

RESOURCE NUMBER	RESOURCE LABEL	CURRENT AVAILABLE	AVERAGE AVAILABLE	MINIMUM AVAILABLE	MAXIMUM AVAILABLE
1	LINK12	0	.0000	0	0
2	LINK13	0	.0000	0	0
3	LINK14	0	.0123	0	1
4	LINK15	0	.0000	0	0
5	LINK21	0	.0000	0	0
6	LINK23	0	.0000	0	0
7	LINK31	0	.0000	0	0
8	LINK32	0	.0000	0	0
9	LINK34	0	.0000	0	0
10	LINK41	0	.0000	0	0
11	LINK43	0	.0000	0	0
12	LINK51	0	.0000	0	0

—FOR—

1 YBLR	(01) * ASD COMPUTER CENTER NOS CSB *	NOS 2.2-605/587.	19.56.47. 85/02/26.
--------	--------------------------------------	------------------	---------------------

19.48.12.GARCIA,STCSB.
 19.48.12.USER(T841239,)
 19.48.12.CHARGE,*
 19.48.12.* CHARGE(T841239,T841239)
 19.48.14.\$PROLOG,PROCL,...
 19.48.14.\$SEIFS,PROCL/FS=AD.
 19.48.14.PROCL.
 19.48.15.NOIE(OUTPUT,NR)+ WELCOME TO NOS 2.2
 19.48.15.IFE(\$\$.NE.\$\$,NOUP)
 19.48.15.ENDIF(NOUP)
 19.48.15.RETURN,PROCL.
 19.48.15.\$REVERT.OOL
 19.48.15.COPYBF,INPUT,TAPES.
 19.48.15.COPY COMPLETE.
 19.48.15.REKIND,TAPES.
 19.48.15.GET,SLAB.
 19.48.15.SETIL(5000).

```

19.48.16.FTN5, I=SLANB, ANSI=0, L=0.
19.48.16.      56100 CM STORAGE USED.
19.48.16.      0.041 CP SECONDS COMPILE TIME.
19.48.16.ATTACH, SLANLIB/UN=APPLIB.
19.48.17.LIBRARY, SLANLIB.
19.48.17.LGO.
19.56.46.      STOP
19.56.46.      376500 MAXIMUM EXECUTION FL.
19.56.46.      418.198 CP SECONDS EXECUTION TIME.
19.56.47.UFAD,      0.002KINS.
19.56.47.UEPF,      0.02KINS.
19.56.47.UENS,      29.978KINS.
19.56.47.UELP,      419.709SECS.
19.56.47.AESR,      995.373UNTS.
19.56.47.$OUT(*,OP=E)
19.56.47. NO FILES PROCESSED.
19.56.47.$DATAFILE(OUTPUT, JT=D)

```

BIBLIOGRAPHY

1. Bruell, S. C. and Balbo, G., Computational Algorithms for Closed Queueing Networks, New York, NY: North Holland, 1980.
2. Chlamtac, I. and Franta, W. R., "A generalized Simulator for Computer Networks," Simulation, October 1982, pp. 123-132.
3. Chou, W., Computer Communications Volume I: Principles, Englewood Cliffs, NJ: Prentice-Hall, 1983.
4. Chu, W. W., Fayolle, G. and Hibbits, D. G., "An Analysis of a Tandem Queueing System for Flow Control in Computer Networks," I.E.E.E. Transactions on Computers, Vol. C-30, No. 5, May 1981, pp. 318-323.
5. Cravis, H., Communications Network Analysis, Lexington, MA: Lexington Books, 1981.
6. Davies, D. W., Barber, D. L. A., Price, W. L. and Solomonides, C. M., Computer Networks and Their Protocols, New York, NY: John Wiley & Sons, 1979.
7. Foster, S. J., Design and Implementation of a Generic Computer Network Simulation System, Master's Thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1983.
8. Georganas, N. D., "Modeling and Analysis of Message Switched Computer-Communication Networks with Multilevel Flow Control," Computer Networks, Vol. 4, 1980, pp. 285-294.
9. Gopal, G. and Wong, J. W., "Delay Analysis of Broadcast Routing in Packet-Switching Networks," I.E.E.E. Transactions on Computers, Vol. C-30, No. 12, December 1981, pp. 915-922.
10. Houstis, C. E. and Leon, B. J., "Priority Queueing for the AUTODIN Store-and-Forward Network," Purdue University, West Lafayette, IN, to be published.

11. Kermani, P. and Kleinrock, L., "A Tradeoff Study on Switching Systems in Computer Communication Networks," I.E.E.E. Transactions on Computers, Vol. C-29, No. 12, December 1980, pp. 1053-1060.
12. Kleinrock, L., Communication Nets Stochastic Message Flow and Delay, New York, NY: McGraw-Hill Book Company, 1964, out of print, New York, NY: Dover, reprinted 1972.
13. -----, "Analytic and Simulation Methods in Computer Network Design," Proceedings of Spring Joint Computer Conference, 1970, pp. 569-579.
14. -----, Queueing Systems Volume I: Theory, New York, NY: John Wiley & Sons, 1975.
15. -----, Queueing Systems Volume II: Computer Applications, New York, NY: John Wiley & Sons, 1976.
16. -----, "On Communications and Networks," I.E.E.E. Transactions on Computers, Vol. C-25, No. 12, December 1976, pp. 1326-1335.
17. Konheim, A. G., "A Queueing Analysis of Two ARQ Protocols," I.E.E.E. Transactions on Communications, Vol. Com-28, No. 7, July 1980, pp. 1004-1014.
18. Kuo, F. F., ed., Protocols & Techniques for Data Communication, Englewood Cliffs, NJ: Prentice-Hall, 1981, pp. 186-189.
19. Lam, S. S., "Store-and-Forward Buffer Requirements in a Packet Switching Network," I.E.E.E. Transactions on Communications, Vol. Com-24, No. 4, April 1976, pp. 394-403.
20. Law, A. M. and Kelton, W. D., Simulation Modeling and Analysis, New York, NY: McGraw-Hill Book Company, 1982.
21. Lazar, A. A., "The Throughput Time Delay Function of an M/M/1 Queue," I.E.E.E. Transactions on Information Theory, Vol. IT-29, No. 6, November 1983, pp. 914-918.

22. Musselman, K. J. and Hannan, R. J., "A Network Simulation Model of a Computer Communication Systems," Pritsker & Associates, West Lafayette, IN, to be published.
23. O'Reilly, J. J., SLAM II Version 2 User's Manual, West Lafayette, IN: Pritsker & Associates, 1983.
24. Pritsker, A. A. B. and Pegden, C. D., Introduction to Simulation and SLAM, New York, NY: John Wiley & Sons, 1979.
25. Reiser, M., "A Queueing Network Analysis of Computer Communication Networks with Window Flow Control," I.E.E.E. Transactions on Communications, Vol. Com-27, No. 8, August 1979, 1199-1209.
26. Samari, N. K. and Schneider, G. M., "A Queueing Theory-Based Analytic Model of a Distributed Computer Network," I.E.E.E. Transactions on Computers, Vol. C-29, No. 11, November 1980, pp. 994-1001.
27. Schwartz, M., Computer-Communication Network Design and Analysis, Englewood Cliffs, NJ: Prentice-Hall, 1977.
28. Tanenbaum, A. S., Computer Networks, Englewood Cliffs, NJ: Prentice-Hall, 1981, pp. 67-70.
29. Uhr, L., "Comparing Serial Computers, Arrays, and Networks Using Measures of 'Active Resources'," I.E.E.E. Transactions on Computers, Vol. C-31, No. 10, October 1982.
30. Wong, J. W., Sauve, J. P. and Field, J. A., "A Study of Fairness in Packet-Switching Networks," I.E.E.E. Transactions on Communications, Vol. Com-30, No. 2, February 1982, pp. 346-353.
31. Yakubaitis, E. A., Network Architectures for Distributed Computing, New York, NY: Allerton Press, 1983.

END

FILMED

2-86

DTIC