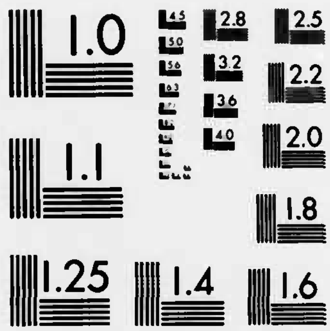
AD-A132 142 SPECIFICATIONS OF A SIMULATION MODEL FOR A LOCAL AREA NETWORK DESIGN IN S..(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA C P CHARPANTIDIS JUN 83 1/2

UNCLASSIFIED F/G 9/2 NL




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

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

SPECIFICATIONS OF A SIMULATION MODEL FOR A LOCAL AREA NETWORK DESIGN IN SUPPORT OF STOCK POINT LOGISTICS INTEGRATED COMMUNICATION ENVIRONMENT (SPLICE)

by

Cosmas P. Charpantidis

June 1983

Thesis Advisor: N.F. Schneidewind

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | |
|---|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. A132142 |
| 3. RECIPIENT'S CATALOG NUMBER | |
| 4. TITLE (and Subtitle) Specifications of a Simulation Model for a Local Area Network Design in Support of Stock Point Logistics Integrated Communication Environment (SPLICE) | 5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1983 |
| | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) Cosmas P. Charpantidis | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940 | 12. REPORT DATE June 1983 |
| | 13. NUMBER OF PAGES 100 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | 15. SECURITY CLASS. (of this report) |
| | 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

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Local Area Network, Simulation Model for SPLICE, Performance Evaluation for Delay Time and Throughput.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis provides the specifications of a simulation model, based on a given functional design for a Local Area Network (LAN) system, which implements functions of the Stock Point Logistics Integrated Communication Environment (SPLICE). First, today's LAN technologies and workload characterization of the SPLICE system are discussed in general. Then, the components of the LAN system and the model assumptions are

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 (continued)

identified in terms of an open network of queues. Finally, an initial approach for model implementation in GPSS is provided.

| Accession For | |
|---|---|
| NTIS GRA&I | [X] |
| DTIC TAB | [ ] |
| Unannounced | [ ] |
| Justification | |
| By | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A | |

# Specifications of a Simulation Model for a Local Area Network Design in Support of Stock Point Logistics Integrated Communication Environment (SPLICE)

by

Cosmas P. Charpantidis
Commander, Hellenic Navy
B.S., Hellenic Naval Academy, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1983

Author: ______________________________

Approved by: Norman F. Schneidewind
Thesis Advisor

Norman R. Lyons
Second Reader

______________________________
Chairman, Department of Computer Science

Kneale T. Marshall
Dean of Information and Policy Sciences

# ABSTRACT

This thesis provides the specifications of a simulation model, based on a given functional design for a Local Area Network (LAN) system, which implements functions of the Stock Point Logistics Integrated Communication Environment (SPLICE). First, today's LAN technologies and workload characterization of the SPLICE system are discussed in general. Then, the components of the LAN system and the model assumptions are identified in terms of an open network of queues. Finally, an initial approach for model implementation in GPSS is provided.

TABLE OF CONTENTS

LIST OF TABLES

## LIST OF FIGURES

# I. INTRODUCTION

## A. GENERAL

The development of resource-sharing network can facilitate the provision of a wide range of economic and reliable computer services. Computer-communication networks allow the sharing of specialized computer resources such as databases, programs, and hardware. Such a network consists of both the computer resources and a communications system interconnecting them and allows their full utilization to be achieved.

Within a restricted area such as a single building, or small cluster of buildings, high-speed (greater than 1 Mbit/sec) data transmission is available at a small fraction of the cost of obtaining comparable long-haul service from a tariffed common carrier. Local area networks use this low-cost, high-speed transmission capability as the basis for a general-purpose data transfer network. As the name implies, a Local Area Network (LAN) is a data communication network, typically a packet and message communication network, limited in geographic scope.

As a result of the growing demands for automated data processing at the Navy stock points and inventory control points, long range plans are being developed around the Stock Point Logistics Integrated Communication Environment

(SPLICE) concept. Developers of SPLICE have decided to employ a LAN at each site for integration of all computer resources. The future integration of this system has been considered [Ref. 1]. That is, as SPLICE requirements evolve and as technology changes, dissimilar devices, e.g., new host computers, mass storage devices, teleprocessing gateways, can be added to the LAN without having to redesign other parts of the SPLICE system.

There are two major objectives behind the development of SPLICE. First, there is the increased need for the use of CRT display terminals to interact with application logic and to fetch information from the system data base. Second, there is the need to standardize the multitude of interfaces currently existing across approximately sixty supply sites [Ref. 1].

The SPLICE project at the Naval Postgraduate School will produce specifications and recommendations for the design of the LAN to be implemented at stock points and inventory control points. The approach taken was to design first the logical or virtual LAN, specifying all the functional modules, their characteristics and the communication protocols, rather than focusing on the hardware characteristics first. The design and implementation strategy is based on a distributed architecture for LANs [Ref. 1].

The design of a computer communication system based on a functional description is a task requiring many interrelated decisions about hardware and software. The decision space for such a problem is so vast that it cannot, in general, be explored completely. Rather, design decisions are made sequentially, and the resulting system structure becomes progressively more constrained at each step. Use of discrete simulation can be a powerful tool in this process if its role is carefully planned.

The gross behavior of the proposed system can be studied, and the system design can sometimes be changed as a result of such study before these changes become expensive. The simulation model must be modified at each step of the refinement process when used in this manner. Care is required, however, so that the investment in simulation at any stage does not outweigh its utility.

Based on the functional specifications provided by Reference 1, this research will address the issue of designing an efficient simulation model in order to be used as a quantitative tool for performance evaluation of the particular LAN in the SPLICE system.

1. Scope of Research

Simulation models differ significantly in their construction and use. The analysis and development though, of any of these types. consists of three general phases:

conceptualization (specifications), implementation, and experimentation. Towards the development of a simulation model for the LAN implementing the SPLICE functions, this research covers the conceptualization and part of the implementation phase, by discussing the specifications and providing block diagrams for such a model.

### 2. Approach

The application of simulation to many types of systems, together with the different types of study which may be involved, result in many variations in the way a simulation study proceeds. Certain basic steps in the process can, however, be identified. The principal steps are considered to be [Ref. 2]:

- Definition of the problem
- Planning the study
- Formulation of the model
- Construction of a computer program for the model
- Validation of the model
- Design of experiments
- Execution of simulation run and analysis of the results

A brief discussion of each step taken in the development process follows:

a. Definition of the problem: Use the functional design specifications provided by Reference 1, as the basis for designing a simulation model, which will be used to estimate the performance of the LAN, in terms of response and transit time, in order to result in the best LAN technology and hardware configuration.

b. Planning the study: (1) A general study of system simulation was conducted for the purpose of obtaining a knowledge of system simulation and selecting an appropriate simulation technique to be used with the LAN simulation model. (2) Study LAN components and performance measures in general: LAN components and performance measurements were investigated in order to develop a knowledge and understanding of what actually composes a LAN and determine the different types of performance measures which could be made on computer networks. (3) Data analysis: Data provided by Reference 3 have been analyzed in order to define distributions and other parameter values, which will be used to drive the simulation model. (4) Study the functional specifications of the particular LAN: A very good understanding of the particular LAN design (functional logic) was necessary in order for the model to be suitable. (5) Study today's LAN technologies: Today's LAN technologies and their performance measurements were investigated in order to develop a knowledge of their advantages and disadvantages. (6) Provide specifications for a particular LAN simulation model: A detailed design of a LAN simulation model was provided to a level giving a valid representation of the system. (7) Study the GPSS simulation language: This final step in the development process was conducted in order to be able to draw the required block diagrams of the simulation model.

## B. OVERVIEW

Following the steps taken in the development process of the simulation model, this thesis discusses, first, in Chapter II today's LAN technologies. Next, in Chapter III, an analysis of the data provided by Reference 3 was conducted in order to provide workload characterization. Then, in Chapter IV, the specifications of a simulation model for the particular LAN are given. Finally, Chapter V is concerned with the implementation of the simulation model.

# II. LOCAL AREA NETWORK TECHNOLOGIES

## A. GENERAL

Designing a LAN that satisfies user demands is not a simple process. It requires choosing a configuration, a medium, access and link-control methods. Additionally, decisions have to be made about where to position the interface and which industry standards to observe. Such an approach in designing a LAN is shown in Reference 4.

By network technology, we mean the mechanism, both hardware and software, by which various computing facilities are interconnected for communication. Potential users have to select the appropriate technology for their intended applications based on their specific performance requirements and operating environment.

It is necessary to go through a brief comparison of the most popular LAN technologies in order that a robust configuration of a LAN may be provided. By the term robust we mean "under the most adverse conditions, specification requirements are met". An understanding of the basic principles of a LAN (such as those contained in Reference 5) is assumed.

## B. LOOP TOPOLOGIES

Several loop topologies for LANs have been proposed in the literature, some using centralized control, others using distributed control. The loop topology concept can be augmented with additional links that are provided from each node to improve performance and reliability. Any node or link failure disrupts communication unless a by-pass is provided. When N is large the delays may be excessive and interface overhead increases with N and may become a bottleneck. Multi-connected loop topologies help to overcome these problems. For higher reliability and relatively small maximum distances between node pairs, regular 2- and 3-connected loop networks can be constructed. Such multi-connected loop technologies can sustain several node and link failures.

The number of nodes directly influences the message delays in loop networks. A smaller number of nodes means the messages are relayed by fewer loop interfaces, and therefore the delays will be less. The maximum throughput under saturated conditions and the reliability of loop networks also depend on loop diameter, which is related to the number of nodes. Reference 6 shows that the throughput performance and the reliability of loop networks is better when diameters are small. A comparative study of the various multi-connected loop networks proposed in the literature, is also provided by Reference 6.

## C. IEEE-802 TOKEN RING

A token ring consists of a set of stations serially connected by a transmission medium. Information is transferred sequentially, bit by bit, from one station to the next. Each station regenerates and repeats each bit and serves as the means for attaching one or more devices (terminals, work-stations) to the network for the purpose of communicating with other devices on the network. A given station (the one that has access to the medium) transfers information onto the ring, where the information circulates from one station to the next. The addressed destination station(s) "copies" the information as it passes. As can be seen from Figure 2.1, given by Reference 7, the physical connectivity of the medium establishes the *logical connectivity* of active stations on the ring. A station gains the right to transmit its information onto the medium when it detects a token (free-token) passing on the medium. The token is a control signal comprised of a unique signalling sequence that circulates on the medium following each information transfer. Any station, upon detection of a token, may claim the token by modifying it to a start-of-frame sequence (busy-token) and appending appropriate address, information, frame check sequence fields and the end-of-frame delimiter.

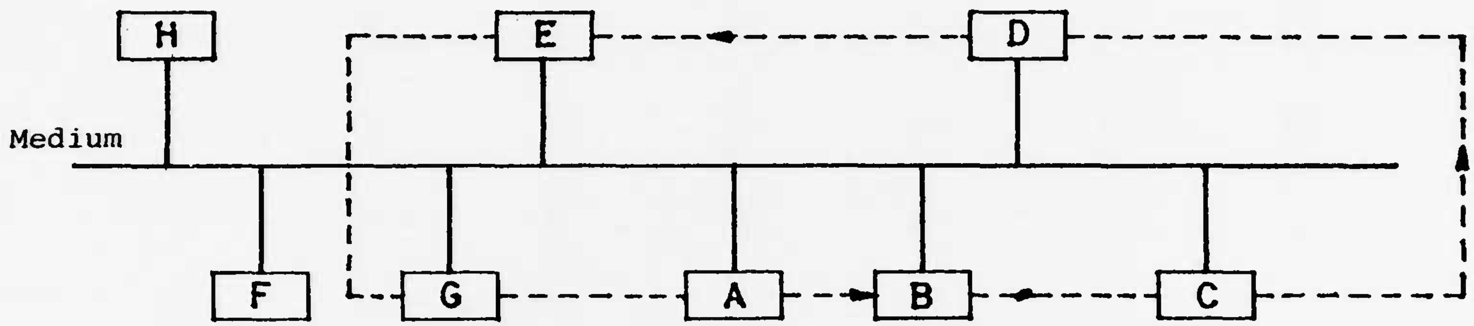



Figure 2.1. Logical Ring on a Physical Bus [Ref. 7]

At the completion of its information transfer and after checking to ensure that the proper operation has resulted, the station initiates a new token, which provides other stations the opportunity to gain access to the ring. A token-holding timer, started at the beginning of information transfer, controls the length of time a station may use the medium before passing the token.

The token ring medium access method specified is efficient in that the coordination of the attached stations requires only a small percentage of the medium's bandwidth capacity when the offered load is high, as stated in Reference 7. Each station's expected access delay to the medium increases no faster than the total load offered under overload conditions.

As stated in Reference 7, this access method is "fair" in the sense that it provides each attached station, with a given class of services (priority level), an equal share of the medium's bandwidth without requiring any station to use its full share at any particular access time. Multiple levels of priority are available for independent and dynamic assignment depending upon the relative class of service required for any given station. e.g., synchronous (real-time voice), asynchronous (interactive), immediate (network recovery). The worst case bounds for any station to gain access to the medium is computable in the absence of noise and with due consideration given to the priority level interaction.

Robust detection and recovery mechanisms are provided to restore network operation in an efficient and timely manner in the event that transmission errors or medium transients (e.g., those resulting from station insertion or removal) cause the access method to deviate from normal operation. Detection and recovery for these cases utilize a network monitoring function that may be distributed in all stations or optionally, centralized in a specific station with back-up capability in one or more alternate stations.

The token access method, as specified, does not place constraints on the station that has access to the medium relative to the logical link control or higher level protocols employed to effect data transfer. This access method does not preclude the possible use of other data link control protocols, e.g., ISO HDLC HRM, X.2S LAP-B, ISO basic mode, etc.

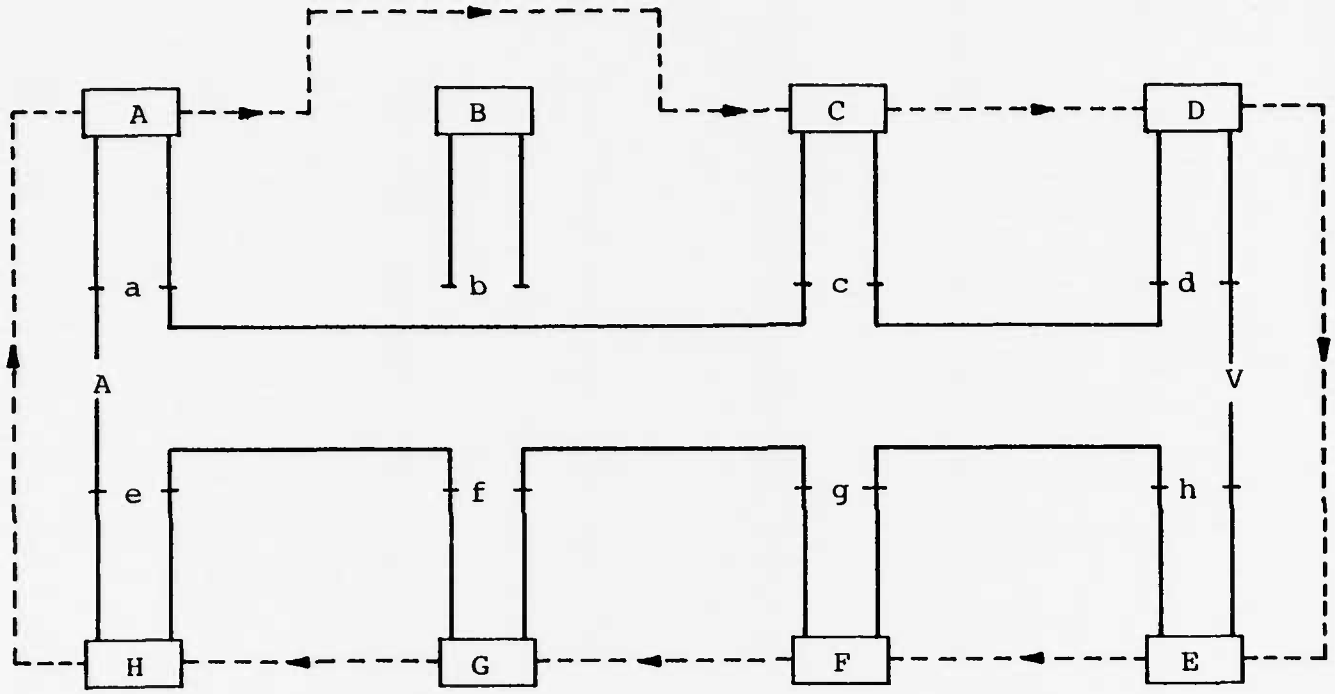
D. IEEE-802 TOKEN BUS

A shared medium can be generally categorized into two major types "BROADCAST" and "SEQUENTIAL". On a broadcast medium, each node will receive all signals transmitted and media of this type are most often associated with the bus configuration. On a sequential type, the right to access the medium passes from node to node in a logical or physical sense.

In IEEE-802 token bus, as it is stated in Reference 7, the token medium access method is always sequential in a logical sense as depicted in Figure 2.2. That is, during normal, steady state operation. the right to access the medium passes from node to node and the physical connectivity has little impact on the order of the logical ring. Nodes can respond to a query from the token holder even without being part of the logical ring (e.g., in Figure 2.2, nodes H and F can respond to polls and receive frames but cannot initiate a transmission since they will never be sent the token).

The *token* (*right* to transmit) is passed from node to node in numerically descending node address order. After each node has completed transmitting any logical link sublayer (LLC) data frames it may have, and has completed other maintenance functions, the node passes the token to its successor by sending a "token" frame.

After sending the token frame, the node listens to make sure that its successor hears the token frame and is active. If the sender hears a valid frame following the token, it assumes that its successor has the token and is transmitting. If the sender does not hear a valid frame within one




Physical Medium A, B, C, D, E, F, G, H - Ring Stations
Logical Data Flow a, b, c, d, e, f, g, h - By-Pass Relays

All stations are active except B (relay b illustrated in bypass mode).

Figure 2.2. Token Ring Configuration
[Ref. 7]

network slot time it assumes the successor did not hear the frame and resends the token.

If the successor does not respond to a second token frame, the sender assumes the successor has failed. The sender now sends a "who follows" frame with its successor's address in the data field of the frame. All nodes compare the value of the data field of a "who follows" frame with the address of their predecessor (the node that sends them the token). The node whose predecessor is the successor (failed node) of the sending node responds to the "who follows" frame by sending its address. The node holding the token establishes a new successor, bridging the failed node from the logical ring.

If the sending node hears no response to a "who follows" frame, it repeats the frame a second time. If there is still no response, the node tries a third strategy to re-establish the logical ring. The node now sends a "solicit successor" frame asking any node in the system to respond to it. If there are any operational nodes that can hear the request. they respond and the logical ring is re-established using the response window process to add new nodes in the logical ring [Ref. 7].

If two attempts at soliciting a successor fail, the node assumes that a catastrophe has occurred. Either all other nodes have failed, the medium has broken, or the node's own receiver has failed so that it cannot hear other nodes who have been responding to its requests. In this case the node quits attempting to maintain the logical ring and simply listens for some indication of activity from other nodes. In summary, the token is normally passed from node to node using a short token pass frame. If a node fails to pick up the token, the sending node uses a series of recovery procedures, that get increasingly drastic as the node fails to find a successor node.

The features that make token-passing different from other access methods are as described in Reference 7.

1. The method is efficient in the sense that the coordination of the nodes requires only a small percentage of the medium's capacity when the offered load is high, and that each node's expected access delay grows no faster than the total offered load under overload conditions.

2. It works at all data rates and distances considered in the IEEE-802 functional requirements, and has the potential for growth in both data rate and distance.

3. The method is fair in the sense that is offers each node an equal share of the medium's capacity, without requiring any node to take its full share.

4. It permits multiple classes of service.

5. It coordinates the node's transmissions so that they minimize and control their interference with each other.

6. The method imposes no additional requirements on the medium and the modem capabilities over those necessary for transmission and reception of multi-bit, multi-frame sequences at the specified mean bit error rate.

7. In the absence of system noise, the method provides computable, deterministic, worst-case bounds on access delay for any given network and loading configuration.

8. The method permits the presence of low-cost reduced-function nodes. It is assumed that at least one full-function node is needed to make the system operational. An example of a reduced function node is one that can "receive only" and therefore does not contain access control logic.

9. Minimal constraints are placed on how a node which momentarily controls the medium may use its share of the medium's capacity. In particular, the access method does not prohibit any node from using other specialized access methods (such as poll/response) during that node's access period, provided only that those specialized methods do not confuse the other nodes on the network as to the state of the overall access mechanism.

## E. PERFORMANCE ANALYSIS

1. The main advantage of the IEEE-802 technologies is that they provide standardization in the following sense: There is a large scale-separation of the system into three parts, the Logical Link Control (LLC) sublayer, the Media

Access sublayer, and the Physical Layer. These layers are intended to correspond closely to the lowest layers of the ISO Model for Open systems interconnection. The LLC and Media Access sublayers together encompass the functions intended for the Data Link Layer as defined in the OSI model. Such an architectural organization has two main advantages:

Clarity: A clean overall division of the design along architectural lines makes the standard cleaner.

Flexibility: Segregation of medium-dependent aspects in the Physical Layer allows the Logical Link Control and Media Access sublayers to apply to a family of transmission media. Partitioning the Data Link Layer allows various Media Access methods within a single family of local network standards.

2. A performance comparison made by Reference 8 through simulation shows the following results:

a. Contention Bus

The mean delay time remains low until the load is close to the maximum channel capacity. The contention time to gain control of the channel is independent of the packet size and the transmission time is directly proportional to the packet size. Therefore, the absolute delay per packet is less for smaller packets while the average delay per bit is smaller in the case of larger packets.

The contention time increases as the network load increases until the network becomes saturated. Therefore, with increasing network load, the present increase in absolute delay is higher for smaller packets.

An advantage of the contention bus technology is this: for a given load, and particularly for large packet size, the mean delay time is not very sensitive to the number of active stations on the bus. Even with only four active stations, the result is very similar to that of 128 stations [Ref. 8].

b. Token Ring

Under the assumption that a station is allowed to put all waiting packets onto the ring when the token arrives, the mean absolute delay time for a given normalized load is independent of the packet size. Thus, there is no need to talk about normalized delay time.

It has been shown [Ref. 8] that as the number of active stations increases, the delay characteristics get poorer. For example, when the number of active stations increases. the mean delay time goes up by approximately a factor of 2 at all load levels. This is because an additional station adds to the walk time delay (i.e., the time required for a bit to go once around the idle ring), thus increasing the propagation delay. Contention but technologies do not suffer from this disadvantage.

c. Slotted Ring

The model used by Reference 8 was based on the assumption that the packet size coincides with the slot size and that the gaps between slots are negligible, for simplicity.

If we keep the slot size constant while varying the number of slots n, then walk time increases with number of slots. This explains the increase in absolute delay time as n increases when the load is light. However, for a given load, the probability of finding an empty slot increases as n is increased. At higher loads, this wait time dominates the delay time and the mean delay time actually decreases as n goes up. More load can be accommodated before the system saturates, as well.

If we keep the walk time constant, varying the number of slots, then the slot size decreases as n increases. The normalized delay is shorter for larger slot (and thus packet) sizes.

It has been shown [Ref. 8] that for a given slot size and number of slots in the ring, there is a optimal number of active stations N which minimizes the mean delay time and maximizes the saturation load. For a given load, the larger the value of N, the smaller the mean number of packets waiting at each station. Thus the mean queue wait time monotonically decreases as N increases. However, because the number of stations ready to transmit has increased, the mean wait time to acquire an empty slot monotonically increases. Furthermore, the walk time increases monotonically with N. The opposing effects on the mean delay time as N varies imply that there is a value of N which minimizes the mean delay time for a given load. It has been shown [Ref. 8] that this value remains constant for all load levels. Knowledge of this information allows us to estimate how close an existing ring is operating from its theoritical optimum. It also allows an estimate of how many stations should be attached to the ring.

d. Reliability

The basic ring topology has often been criticized as being unreliable on the grounds that an open circuit anywhere or the failure of any repeater will disrupt the entire network. This is certainly a problem with a large number of repeaters strung together. However, the "star-shaped ring" design with a wire center at the hub of a star-configured network has greatly alleviated the problem.

The contention bus is essentially a passive device, thus its reliability is much higher than that of the basic ring design. However, with the use of repeaters in more complex bus systems, the reliability decreases.

e. Fairness

Due to the nature of the two technologies, the ring seems superior to the contention bus technology. Due to the back-off formula used for contention busses, the collided packets are discriminated against in favor of the freshly arrived ones. Thus a packet that has just arrived has a higher probability of getting to its destination before a packet that may arrive earlier and which has suffered one or more collisions.

f. Maintainability - Extensibility

Both have been improved in ring technologies and are comparable to the passive bus.

g. Summary of Performance Comparison

In general, the CSMA/CD bus technology should have lower mean delay time than the ring technologies when the load is light (probability of collision small). This is because, unlike the ring technologies where a station must wait for the arrival of the token or an empty slot, a station can transmit the packet at once. It is interesting to note that the acknowledgement of the arrival of a packet in the ring technologies is essentially the original packet itself. Thus the packet cannot be removed from the channel until it has completed at least one round trip. If the packet is very large, then we waste channel bandwidth.

Broadcast contention bus technology suffers from the fact that the upper bound of delay time is non-deterministic. Thus it is not appropriate for real time applications. Furthermore, the theoretical maximum network length is generally shorter than the ring technologies (e.g., it is 500 meters per line, 2.5 km between any two stations in a hierarchial network for Ethernet).

As a result of the previous discussion, we believe that the best way to select the appropriate technology for the SPLICE LAN, is the development of a simulation model, based on the functional specifications of the SPLICE LAN. Such a model will provide a performance evaluation for different technologies in accordance with the SPLICE requirements.

## III. WORKLOAD CHARACTERIZATION

### A. GENERAL

It is crucial to the performance evaluation of the SPLICE LAN, that an accurate characterization and projection be made of the Automated Data Processing (ADP) workload to be supported by the system through the 1980's and early 1990's.

The SPLICE system is designed to provide a telecommunications "Front End" to the existing Stock Point computer complexes and to support the implementation of interactive transaction processing. The processing of batch applications on the SPLICE configuration is to be held to a minimum, since the existing Stock Point computer complex is to continue to provide batch processing support. Thus the workload of most interest are interactive transactions (whether for processing within a SPLICE complex or for forwarding to a remote Stock Point computer complex).

### B. BACKGROUND

Reference 3 defines the SPLICE configuration requirements by projecting:

- the arrival of units of work at SPLICE processing facilities (workload analysis).
- the amount of processing resources comsumed in the satisfaction of the workload demand (process load).

- the magnitude of system resources available in each configuration (configuration sizing).

Before we go any further, it was felt that some definitions have to be provided for better understanding.

MESSAGE: A single transmission of a user's data between two points.

TRANSACTION: A series of MESSAGES, in one or both directions, which together achieve a unit of work, as defined by the particular application.

WORKLOAD: The arrival of TRANSACTIONS for processing at the SPLICE complex.

PROCESS LOAD: The individual demand for ADP system service (by component) caused by the arrival of a TRANSACTION.

COMPONENT: A segment of the PROCESS LOAD (i.e., edit, validation, etc.).

The eight major components of a transaction life cycle are provided below and a simplified transaction processing algorithm is depicted in Figure 3.1.

- Input/Edit: The editing of the input MESSAGE without reference to files.

- Validation Read: The acquisition of records for further MESSAGE validation.

- Validation: The final checking of the MESSAGE against the records.

- Error Messages: The output if any of the previous steps fail.

- Process Read: The acquisition of records for processing.

- Process: The actual information transformation/process.

- File Write: The modification/addition of records.

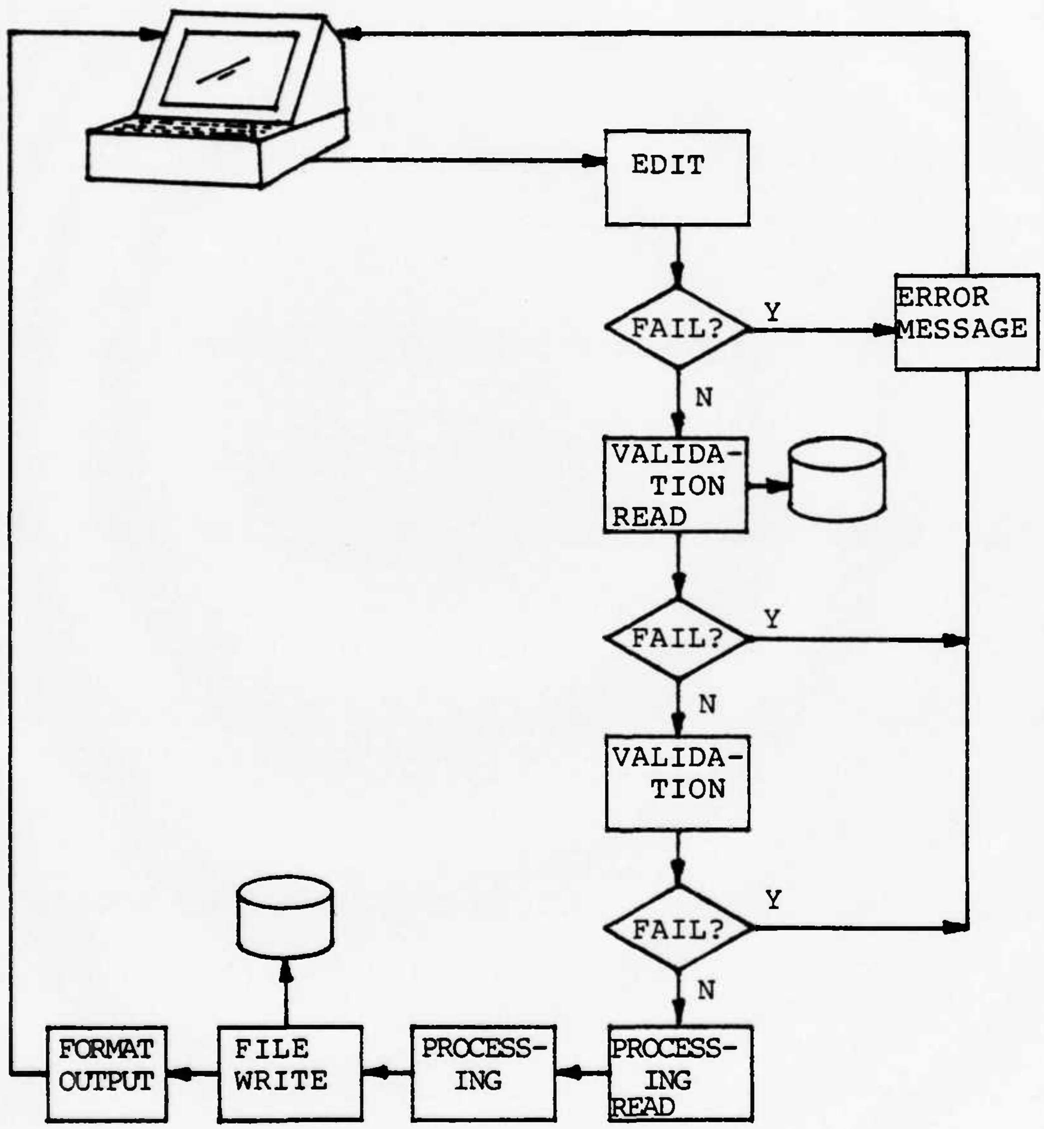



Figure 3.1. Simplified Processing Architecture [Ref. 3]

- Output: The generation of a MESSAGE in response to the input.

Definitions for each of the parameters involved in the above eight major components are as follows:

Transaction Class: A numeric transaction class identifier.

Input Message Length: The average length of an input message in characters including all protocol and/or control characters.

Input Edit Instructions: The number of instructions executed to edit the input message without or before accessing secondary storage files.

Input Edit Failure: The percentage of input messages that fail to pass the preliminary input edits.

Validation Reads: The number of records read for the purposes of validating the input message.

Validation Read Instructions: The number of instructions necessary to prepare for and execute a read operation.

Validation Record Length: The average number of characters read per validation record including any overhead characters.

Validation Read Failures: The percentage of messages for which the validation read operation fails.

Validation Instructions: The number of instructions executed in the process of final input message/transaction validation.

Validation Failures: The percentage of messages for which the validation process fails.

Error Message Format Instructions: The number of instructions executed to format and write back to the terminal an error message following a validation failure. The proaessing of the message is assumed to continue with the input of the revised message.

Error Message Length: The number of characters in the error message including any formal and/or control characters.

Processing Reads: The number of records read in anticipation of processing (in addition to the records read for validation).

Processing Read Record Length: The average number of characters in the records read for processing.

Processing Read Instructions: The number of instructions executed to prepare for an to execute the reading of records for processing.

Processing: The number of instructions executed during the processing phase of the transaction.

File Modifies: The number of records written to files that do not require any structural maintenance to index or directory files.

Modified Record Length: The average number of characters written during modify operations.

File Adds: The number of records written which require structural maintenance to all associated index or directory files.

Added Record Length: The average number of characters written during add operations.

Modify/Add Instructions: The number of instructions executed to prepare for an to execute the modification/add of records.

Output Format Instructions: The average number of instructions executed to format the output message.

Output Message Length: The average number of characters in the output message including any format and/or control characters.

Output Records: The lumber of output messages written.

## C. WORKLOAD FORECAST

The workload forecast has been prepared in Reference 3 for each of the 21 SPLICE main processing complexes and the process load is described in terms of twelve transaction archetypes which will not change over the life of the SPLICE contract. Instead of determining the resource requirements for each individual transaction, the above set of twelve transaction classes has been defined.

The workload characteristics of the most representative site (NSC NORFOLK) are provided in Appendices A and B, which will be discussed in this chapter.

The way data were provided by Reference 3 did not help much for obtaining an accurate characterization of the transaction arrivals in the LAN system, but since these were the only data available, they are utilized to the extent possible.

We analyzed the given data in many ways in order to reach the most reasonable conclusions about the distributions and the frequencies that characterize the transaction arrivals into the LAN system.

First, the maximum peak rate of transactions per six month period is considered to be an insufficient amount of data for defining distributions of arrivals in a computing system. Thus, lacking definitive information, we assume uniformly distributed transaction arrivals within each six month period in order to cover the peak rates.

Applying the goodness of fit test on data per each of the twelve transaction classes over the total period (1982-1993), we found that the data are uniformly distributed with an exponential growth over time (Appendices E through P). This result seemed highly unlikely; therefore, the data in Reference 3 are suspect.

Examining the data independently of transaction classes over the total period (1982-1993) and applying the goodness of fit test, we found that the distribution was very close to a logarithmic one as it is shown in Appendix C. It is known though that logarithmic distributions are not representative of arrival data for computer systems. Thus, we could not support such a conclusion for that particular site (NSC Norfolk).

Since our analysis casts doubts on the validity of the forecast data in Reference 3 and, in addition, a subsequent study [Ref. 9], refuted these data, and lacking real data (SPLICE has not been implemented), we concluded that the most reasonable assumption about transaction arrival rate is that it is Poisson.

In order to estimate the frequency of occurrence of transaction classes, we calculated horizontally the percentage of each transaction class per six month period and then we computed the overall percentages per transaction class in order to check consistency of the results. That was necessary in order to define the frequency of occurence for each particular transaction class to be used later in the model for transaction class selectimn. Results of the above computations are given in Appendix D.

Results for the goodness of fit tests ($X^2$) for distributions per transaction class over the total period are shown in Appendices E through P.

## D. OVERVIEW

In general, there is a feeling that the data provided by Reference 3 are not valid: (wrong summations for the total volume of transactions per six month period, data for every transaction class are uniformly distributed over the ten year period, logarithmic distribution is not generally acceptable for real data, etc.). Since there are no other data available, we used the frequency table in Appendix D as the best approximation available at the moment.

# IV. SPECIFICATIONS OF THE LAN SIMULATION MODEL

## A. INTRODUCTION

A definition given by Reference 10 describes a simulation model as "a logical-mathematical representation of a concept, system, or operation programmed for solution on a high-speed electronic computer." A more general one given by Reference 11 defines a simulation as the answer to the question "What if ...?".

A simulation model can be constructed and applied during any phase of system design or predesign conceptualization. It depends upon the answers desired about the system as to when a model should be constructed. According to Reference 10, a simulation can be constructed:

1. Before the system is designed in order to determine parameter sensitivity and to optimize or evaluate the system design.

2. During the system design phase, in order to test and experiment with system design concepts.

3. After a system has been designed and built in order to supplement system test results and to evaluate overall system effectiveness.

In simulation of a computer facility, a critical issue is the level of detail at which the simulation is to operate. Reference 12 distinguishes two extreme levels of detail; in practice of course, there exists between the two extremes a whole spectrum of levels. We will call the extremely fine level of detail the "micro level" and the extremely aggregated level the "macro level".

At the micro level, the effects of each individual machine-language instruction are simulated. The transactions are machine-language instructions and, perhaps, input-data sets. The state of the system includes the contents of main memory, auxilliary storage, and other output-data sets. The processor and the channels are treated as servicing entities. Thus, micro-level simulations are expensive both in terms of programming time and in terms of running time. Furthermore, simulation at the micro-level requires an extremely detailed understanding of the functions of the operating system, because these details must be incorporated into the logic of the model. Micro-level simulations are useful for study of computer-design problems. They can give the designer insight into the consequences of modifying the instruction set or the hardware and software capabilities. However, in a simulation intended to determine how to process a user's workload, this level of detail seems to be unnecessary and probably undesirable.

At the macro level, the effects of processing complete jobs are simulated, with each transaction representing a total job. The state of memory is described, not in terms of its contents, but rather in terms of the number of storage locations that are busy or idle. This is the level most often used in simulation of computer systems in order to evaluate a system's performance more effectively.

This thesis, as part of the SPLICE project at the Naval Postgraduate School, intends to support the functional design of a LAN implementing the SPLICE functions, by discussing in this chapter the specifications of a macro level simulation model which will help in the LAN design, optimization and evaluation.

### 1. Background

Today's computer systems have evolved a great deal. They are now many devices capable of independent operation; jobs can follow complex routing paths, returning to any device many times; different kinds of jobs can place different demands on different devices. New system features include concurrent programs, multiprogramming, multiprocessing, distributed processing, timesharing and virtual memory. It is no longer obvious how to use a job's input, execution, and output time to compute throughputs and response times in the system, or how to evaluate a computer system's performance.

Queueing network models have proved to be cost effective tools for analyzing modern computer systems. The increasing popularity of queueing network models for computer systems has three bases according to Reference 13:

a. These models capture the most important features of actual systems, e.g., many independent devices with queues and jobs moving from one device to the next. Experience shows that performance measures are much more sensitive to parameters such as mean service time per job at a device, or mean number of visits per job to a device, than to many of the details of policies and mechanisms throughout the operating system (which are difficult to represent concisely).

b. General service time distributions can be handled at many devices; load-dependent devices can be modeled; multiply classes of jobs can be accommodated.

c. The algorithms that solve the equations of the model are available as highly efficient queuing network evaluation packages.

The use of discrete simulation, employing a queueing network model, can be a powerful tool in the process of modeling the SPLICE LAN if its role is carefully planned.

2. Overview

In our case we will describe a LAN simulation model, based on the functional specifications provided by Reference 1, as an open queueing network. First, the simulation model resources will be described in terms of the components which compose the model LAN, i.e., the functional modules described in Reference 1, processing unit, memory unit and disc unit. Next, how these components are modeled as an open queueing network is defined assuming an on-line environment with different classes of transactions. Finally, the transaction flow through the LAN simulation model and the selection of transaction classes are described in terms of the stochastic processes representing the flow through the queueing network.

## B. SIMULATION MODEL COMPONENTS

The simulation model resources are described below in terms of components which compose the LAN model. For this particular design the resources which serve user transactions and contribute to delay, according to Reference 1, are the following:

1. Local Communication Module (LC)
   - Bus arbitration, i.e., traffic management.
   - Message transmission and reception including buffer management.
   - Message control (e.g., error detection, correction and acknowledgement).
   - Administration (message accounting, lost or
   - Misdirected message handling, LAN recovery and shutdown).

2. National Communication Module (NC)
   - Conversion of Defense Data Network protocol to LAN protocol and vise versa.
   - Message assembly/disassembly.

3. Front-End Processing (FEP)
   - Terminal and communication line buffering.
   - Code conversion.
   - Byte/word assembly/disassembly.

4. Terminal Management (TM)
   - Message editing.
   - Screen management.
   - Virtual terminal operations.

5. Data Base Management (DBM)
   - File creation.
   - File update.
   - Query processing and data retrieval.
   - Data dictionary creation and maintenance.
   - File catalog creation and maintenance.

6. Session Services (SS)
   - Establish and maintain local and remote sessions:
     * within the LAN.
     * with local host(s).
     * with remote host(s).
   - Provide logical and physical network addresses based on value of services request code.

7. Peripheral Management (PM)
   - Management of unit record input/output
     * read a card.
     * print a line, etc.
     * spool files for input and output.
   - Optical Character Recognition or Mark Sense Equipment.

8. Resources Allocation (RA)
   - Allocation of shared resources to functional modules.
     * Record keeping concerning allocation of shared resources.
     * Locating, accessing and making shared resources (e.g., memory, disk) available to functional modules.

In addition to the above model LAN components, which are considered as individual user transaction servers in the queueing network, the following physical resources will also be considered as model components representing individual servers, but instead of servicing user transactions, they will service functional modules (they will be servers of servers).

9. Processing units.

10. Disk storage.

11. Memory storage.

Industry standards for seek time, rotational delay time and transfer time will be used to estimate the mean service time per functional module for each transaction class.

## C. MODEL ASSUMPTIONS

1. The One-Line, One-Server Queueing system is assumed for each particular server in the system.

2. Since data provided by Reference 3 do not fit any theoretical distribution at an acceptable level of significance, it is assumed that the transaction arrivals are random, characterized by a Poisson distribution with mean interarrival time equal to the reciprocal of the total peak rate per six month period, given in Appendix A.

3. The interarrival-time random variable is assumed to be exponentially distributed and integer valued.

4. Like interarrival time, service time for each particular class of transactions per functional module (server) is assumed to be exponentially distributed and integer valued. The mean service time in our case will be computed based on current industry standards.

5. A random-number generator is assumed to be available. A description of such a generator will be given later in this chapter.

6. It will be assumed that all arriving transactions will remain for service up to a maximum quantum of time defined by an operating system's timer.

7. When the simulation begins, the system will be assumed to be "empty and idle". That is, there are no transactions in the queue initially, and the servers are idle.

8. After the simulation has been started, it should continue until a length of simulated time, which is provided as a parameter, has elapsed. In general, then, when the simulation is stopped, the servers may be in the process of providing service, and there may be one or more transactions in the queue.

9. A priority scheme for each server in the system (functional module or physical resource) will be applied, defining the following levels of priority among the transactions (from highest to lowest).

a. Control Output from FM.

b. Control Input to a FM.

c. Control Processing for a FM.

d. Data Output from FM.

e. Data Input to a FM.

f. Data Processing for a FM.

10. The processing sequence per transaction class through the functional modules is assumed to be given. Such a sequence can be determined by mapping the process load components to subcomponents of the functional modules.

11. As the simulation proceeds, information on the maximum queue lengths per server should be recorded. Enter and exit times per transaction should be recorded also, in order to compute the delay time. Interarrival time and service time distributions in effect and total simulation time should also be recorded.

## D. RANDOM NUMBER GENERATOR

When a Poisson arrival process is to be simulated, it is not the arrival rate which is of direct interest; instead, it is the corresponding interarrival times which must be known. This is consistent with computing the time of the next transaction's arrival by adding to a copy of the clock's current reading a value drawn from an interarrival-time distribution. When arrival rates are Poisson distributed, then, the corresponding interarrival times are exponentially distributed.

Given a value drawn from a 0-1 uniform random number generator described in Reference 14, the corresponding interarrival time can be directly computed by the following equation:

$$IAT_{sample} = (IAT_{avg})[-\log_e(1-RN_j)]$$

where $IAT_{sample}$ stands for the sampled interarrival time value; $IAT_{avg}$ is the average interarrival time in effect; $RN_j$ is a uniformly distributed random number between 01, and $\sqrt{\log_e}$ represents the natural logarithm operation. To draw a sample from the exponential distribution whose average value is $IAT_{avg}$, then, the sequence indicated by the above equation has to be followed:

a. Draw a value from a 0-1, uniform distribution.

b. Compute the natural log of 1 minus the random number.

c. Multiply the negative of this natural logarithm by $IAT_{avg}$.

Recalling that the values of $RN_j$ range over the closed interval from .000000 to .999999, we note that $\log_e(1-RN_j)$ is either 0 (for an $RN_j$ value of .000000), or negative (for $RN_j$ values greater than .000000). The quantity $-\log_e(1-RN_j)$ is consequently non-negative; and, because $IAT_{avg}$ must be non-negative as well, the value of $IAT_{sample}$ computed from the equation given above is also non-negative.

## E. TRANSACTION FLOW

A transaction flow through the LAN system simulation model is a flow through the network of queues, each modeling one functional module. The queueing network which represents the LAN system, as it was discussed earlier, is not considered and modeled as a whole, but it is decomposed into modules which are analyzed in isolation. The focus in this approach is on stochastic processes representing the flow through each module where the output of one module represents the input to a subsequent module.

The flow in the simulation model can be portrayed as a series of discrete events as it is shown in Figure 4.1 for transaction class-1. The occurence or timing of these events is on a next event scheduled basis; details will be discussed later in the implementation phase. Also, the occurence of the events is governed by the various statistical distributions of the requirements which are placed on individual system modules and physical resources.

1. Transaction Class Selection

The selection and processing of a transaction involves the determination of its class and consequently the module visitation sequence, the amount of physical resources required, and the mean service time per module for this particular transaction class. The visitation sequence can be defined by mapping the process load demands to the subcomponents of the functional modules. The physical resources required and the mean service time can be estimated according to the current industry standards.

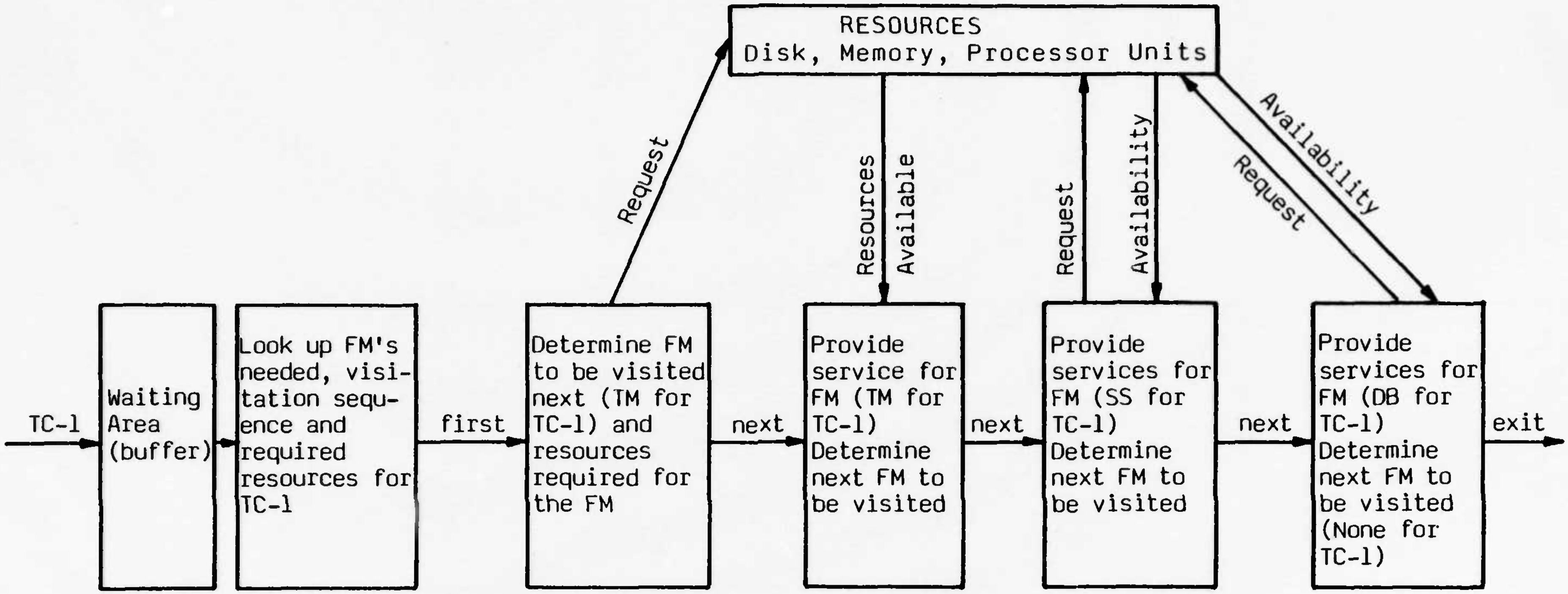



Figure 4-1. Transaction Flow in the Simulation Model

The arrival of transactions at their initial input into the Front-End processor has been characterized by a Poisson process, assuming independent and random inputs. Then it can be shown [Ref. 15] that the transaction inter-arrival times are exponentially distributed over a mean interarrival time. The latter can be computed through the total peak rate of transactions for each six month period, by taking the reciprocal of this peak rate. For example, in Appendix A the total volume for the first six-month period for transaction class-1 is 27.52411 transactions per second, then, the mean interarrival time is 1/27.52411 = .036332 seconds. Using a random number generator, we will take a sample value from the exponentially distributed interarrival times, which is the interarrival time for the next arrival. This sample value will be added to the copy of the simulation clock in order to determine the time of the next arrival.

The class of the transaction input to the system is determined by referring to cummulative probability of occurrance (Table IV-I) according to the frequency of occurrance per transaction class, given in Appendix D. A uniform random number generator is used drawing a sample value between .0000 and .9999 in order to define the probability and then the transaction class.

TABLE IV-I

TRANSACTION CLASS SELECTION (FIRST SIX-MONTH PERIOD 1982.0)

| Transaction Class | Frequency of Occurrence (%) | Probability Occurrence | Cummulative Probability | Visitation Sequence (Hypothetical) | Resources (Hypothetical) Disk | Mem | Proces |
|---|---|---|---|---|---|---|---|
| TC1 | 12.94 | .1294 | .1294 | 4-5-6 | 50 | 10 | 1 |
| TC2 | 3.73 | .0373 | .1667 | 4-6-8 | 80 | 5 | 2 |
| TC3 | 2.63 | .0263 | .1930 | 4-6 | 30 | 5 | 1 |
| TC4 | 1.50 | .0150 | .2080 | 2-4-6 | 15 | 8 | 3 |
| TC5 | 4.70 | .0470 | .2550 | 1-4-6-5 | 40 | 6 | 2 |
| TC6 | 8.08 | .0808 | .3358 | 1-4 | 0 | 0 | 1 |
| TC7 | 2.06 | .0206 | .3564 | 4-6-7 | 25 | 5 | 1 |
| TC8 | 2.98 | .0298 | .3862 | 4-6-3 | 20 | 10 | 4 |
| TC9 | 1.01 | .0101 | .3963 | 4-6-3-5 | 10 | 4 | 3 |
| TC10 | 5.97 | .0597 | .4560 | 2-4-3 | 0 | 0 | 1 |
| TC11 | 1.32 | .0132 | .4692 | 4-6-2 | 0 | 0 | 1 |
| TC12 | 53.07 | .5307 | .9999 | 4-6-5-2 | 40 | 5 | 2 |

NOTE: Disk and Memory units are in Kbytes, processing capacity is expressed in FLOP's (floating point operations per unit of time). Numbers in the visitation sequence column are taken from Table V-I and they represent the corresponding modules to be visited in the indicated sequences.

After a transaction has been assigned a class, the module visitation sequence and the amount of the required physical resources are determined. The availability of physical resources, then, is tested one by one and the transaction processing begins.

An additional random number generator is used in order to obtain sample values for service time for each module per transaction class, based on the given mean service time.

Various statistics associated with the transaction and the LAN system model are accumulated and updated and the transaction exits the model.

Congestion points can be identified and buffer sizing can be rearranged for better performance results.

Hardware configuration sizing can also be estimated according to the workload characterization over the total time of the SPLICE contract.

# V. IMPLEMENTATION OF THE SIMULATION MODEL

## A. GENERAL

In this chapter we will implement the simulation model using GPSS as an appropriate programming language for discrete-event simulations.

### 1. Primary and Secondary Events

By definition, a primary event is one whose time of occurrence is scheduled in advance of its actual occurrence. Any event which is not primary is, by definition, secondary. Secondary events, then, are not scheduled in advance. They occur when primary events do, but in "dependent" fashion, as a direct result of primary-event occurence.

### 2. The Simulation Clock

Simulation time elapses as events occur in a simulation, one by one. It is natural, then, to use a "simulated clock" as part of a queueing-system model. A variable has to be introduced to represent the "simulated clock" and this variable is then used to record what simulated time it is.

A "bootstrapping" technique is used to establish transaction arrival times. That is, when an arrival occurs a procedure is set up for determining the time of the next arrival. The time of the first arrival must be scheduled as one of the steps taken to initialize the simulation. Assume that when a simulation begins, the simulated clock itself is initialized with a value of zero. Then, to schedule the first arrival. a sample is drawn from the interarrival-time distribution at this first clock reading. The sampled value equals the future time at which the first arrival will occur. For example, if a value of 15 is drawn from an interarrival-time distribution. then. the first arrival will occur at 15.

GPSS uses an integer clock and this is appropriate for the One-Line, One-Server Queueing system, because interarrival times and service times are assumed to be integer valued.

The unit of time can be any time interval used as time unit. In practice, the unit of simulated time must be small enough to realistically reflect the time spans which occur in the system being modeled. Since we are modeling a computer system, the time unit should be one millisecond.

Now suppose that the simulation is in progress, and the state of the system has just been updated at the current point in simulated time. The next logical step is to "advance the clock". There are two alternative ways to find the value at which the clock should be advanced.

a. Advance the clock by exactly one time unit. Then scan the system to determine whether any events have been scheduled to occur at this new clock reading. If so, update the system by performing the logic for these events, then advance the clock again by one time unit, and so on. When testing indicates no events have been scheduled to occur at the clock's new reading, simply advance the clock immediately to its next value. The logic of this approach, uses a fixed time-increment clock. This approach causes a very high CPU overhead in simulation model execution.

b. The second approach to clock maintenance uses a variable time-increment clock. In this approach, when conditions call for advancing the clock, it is advanced to the time of the "imminent event". The imminent event is the one which has been scheduled to occur at the next earliest point in simulated time. In general, then, the amount by which the clock is advanced differs from advance to advance, giving rise to the phrase "variable time-increment clock".

The apparent advantage of the variable-time increment clock seems to be that intermediate points in time when nothing has been scheduled to occur anyway are skipped over, thereby probably saving computer time. This is not always true though depending upon the number of primary events in the system. In our case we will use the variable time-increment clock.

Finally, care should be taken to distinguish between simulated time and real time. When the simulation clock is advanced to a next reading, that reading remains fixed while the model is updated. Nevertheless. real time passes as the updating occurs. It may require hours of real time to move models of some systems through only minutes of simulated time. One the other hand, experiments equivalent to weeks. months, or even years of simulated time can often be conducted in only seconds of real time in the computer.

## B. APPROACH TAKEN IN BUILDING THE MODEL

It would be relatively easy to model the SPLICE LAN model by using twelve model segments, one for each class of transactions. When a job-transaction entered the model, its transaction class could then be routed to the appropriate model segment. There, it would move through a straight sequence of blocks. consisting of an ENTER-ADVANCE-LEAVE combination for each functional module (server) being visited. Throughout the model, block operands would be specified by using random number generators and exponental interarrival and service time distributions.

The disadvantages in taking the approach outlined above are that (1) a relatively large number of blocks would be required, and (2) a relatively inflexible model would result. Instead of taking such an approach. Matrix save values will be used to build a compact model. The principal part of the required model is a single ENTER-ADVANCE-LEAVE sequence. This single sequence can be used to simulate use of consecutive modules (servers) by all the transaction classes, providing that the following provisions are made:

1. The pertinent module number must be supplied as the A operand at the ENTER and LEAVE blocks.

2. The pertinent mean service time must be provided as the A operand at the ADVANCE block.

3. Each transaction must move through the single ENTER-ADVANCE-LEAVE block sequence the proper number of times (one time for each module to be visited).

4. Availability of physical resources must be tested before entering the ENTER-ADVANCE-LEAVE block sequence.

Table V-I enumerates the functional modules in order to use the corresponded numbers in the matrices instead of the actual names. Table V-II provides the total number of modules to be visited, the module visitation sequence and the mean service time for each module per transaction class.

The means for making these provisions will now be considered.

As for provisions (1) and (2), the pertinent module numbers can be stored in the right order in a visitation-sequence Matrix (Table V-III); and the pertinent mean service times can be correspondingly stored in a mean-service-time Matrix (Table V-IV). At the ENTER block, a transaction then simply needs to index into the proper cell of the visitation-sequence Matrix to obtain the number of the module it must visit next; similarly, when a module has

TABLE V-I

MODULE (SERVER) ENUMERATION

| | |
|---|---|
| 1 | Local Communication (LC) |
| 2 | National Communication (NC) |
| 3 | Front-End Processing (FEP) |
| 4 | Terminal Management (TM) |
| 5 | Data Base Management (DB) |
| 6 | Session Services (SS) |
| 7 | Peripheral Management (PM) |
| 8 | Recource Allocation (RA) |

TABLE V-II

VISITATION SEQUENCES AND MEAN SERVICE TIMES FOR THE TWELVE CLASSES OF TRANSACTIONS (HYPOTHETICAL DATA)

| Transaction Class | Total Number of Modules to be Visited | Module Visitation Sequence and Mean Service time in milliseconds |
|---|---|---|
| 1 | 3 | TM(40), SS(45), DB(50) |
| 2 | 3 | TM(55), SS(60), RA(75) |
| 3 | 2 | TM(80), SS(92) |
| 4 | 3 | NC(88), TM(75), SS(67) |
| 5 | 4 | LC(66), TM(54), SS(66), DB(80) |
| 6 | 2 | LC(65), TM(85) |
| 7 | 3 | TM(64), SS(58), PM(100) |
| 8 | 3 | TM(43), SS(58), FEP(55) |
| 9 | 4 | TM(45), SS(74), FEP(63), DB(87) |
| 10 | 3 | NC(85), TM(88), FEP(90) |
| 11 | 3 | TM(35), SS(60), NC(100) |
| 12 | 4 | RM(55), SS(65), DB(83), NC(85) |

TABLE V-III

VISITATION SEQUENCE MATRIX
(Hypothetical Data)

| ROWS (Trans-Class) | COLUMNS (Number of modules yet to be visited) 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 6 | 4 | | | | | |
| 2 | 8 | 6 | 4 | | | | | |
| 3 | 6 | 4 | | | | | | |
| 4 | 6 | 4 | 2 | | | | | |
| 5 | 5 | 6 | 4 | 1 | | | | |
| 6 | 4 | 1 | | | | | | |
| 7 | 7 | 6 | 4 | | | | | |
| 8 | 3 | 6 | 4 | | | | | |
| 9 | 5 | 3 | 6 | 4 | | | | |
| 10 | 3 | 4 | 2 | | | | | |
| 11 | 2 | 6 | 4 | | | | | |
| 12 | 2 | 5 | 6 | 4 | | | | |

NOTE: The cells represent the identification of the modules to be visited next.

TABLE V-IV

MEAN SERVICE TIME MATRIX
(μsec)

| ROWS (Trans-Class) | COLUMNS (Number of modules yet to be visited) 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 40 | 45 | 50 | | | | | |
| 2 | 55 | 60 | 75 | | | | | |
| 3 | 80 | 92 | | | | | | |
| 4 | 88 | 75 | 67 | | | | | |
| 5 | 66 | 54 | 66 | 80 | | | | |
| 6 | 65 | 85 | | | | | | |
| 7 | 64 | 58 | 100 | | | | | |
| 8 | 43 | 58 | 55 | | | | | |
| 9 | 45 | 74 | 63 | 87 | | | | |
| 10 | 85 | 88 | 90 | | | | | |
| 11 | 35 | 60 | 100 | | | | | |
| 12 | 55 | 65 | 83 | 85 | | | | |

been captured, the transaction can index into the corresponding cell in the mean service-time Matrix to obtain the pertinent value of the A operand at the ADVANCE block. With respect to provision (3), the number of modules a transaction must visit can be stored in a parameter of the transaction when it first enters the model. After each service has been performed, the parameters can be decremented by 1. then tested to determine whether there are yet more services to be performed (that is, tested to determine whether the parameter has yet been decremented to a value of 0). If at least one additional service is indicated, the transaction can be routed back through the ENTER-ADVANCE-LEAVE sequence. Then the parameter can be decremented and tested again. etc. Finally, with respect to provision (4). the physical resources that a transaction needs can be stored in parameters of the transaction. one for each resource, when the transaction first enters the model. The availability of the resources required by the transaction can be tested by comparing the parameter value for each particular physical resource which is stored in a table and updated for every entering transaction in the ENTER-ADVANCE-LEAVE block sequence and for every transaction completion (That is, the value of the required physical resource per transaction is subtracted from the current value of the corresponded value in the table and if the result is a negative number, then the transaction does not enter the ENTER-ADVANCE-LEAVE block sequence, creating a queue line at that particular resource; if the result is zero or a positive number, then the other resources are tested in the same way and if the results are zero or positive numbers, then the contents of the current values in the table are changed to the new current values (reduced) and the transaction enters the ENTER-ADVANCE-LEAVE block sequence. When the transaction completes its service and exits the model, then, the released physical resources are added to the current values of the table becoming available for other transactions).

1. Set-Up and Use of Matrices

We now consider more closely how the visitation sequence and mean service time matrices can be set-up and used. Table V-III shows the appearance of the visitation sequence matrix, assuming that GPSS storages 1 through 8 are used to simulate functional modules (servers) 1 through 8 respectively. There are twelve rows in the matrix, one for each transaction class. There are eight columns in the matrix, corresponding to the maximum number of modules (servers) that any one transaction class must visit. The column numbers are interpreted as the number of modules to be visited by a given transaction class. Entries in the body of the matrix are interpreted as the numbers of the modules (servers) to be visited next. For example, as indicated in Table V-III, a transaction of class 1 must visit three modules. When it first arrives into the system, the number of modules to be visited is three. In column 3, row 1 of Table V-III. the number of the module to be visited next (that is. to be visited first) is found to be "4". Module "4" is TM (as indicated in Table V-I), and as Table V-II shows, the first service on transaction class 1 is performed in module TM. Row 1. column 3 of the visitation sequence matrix therefore contains the pertinent module number. When the terminal management operation has been performed, a class 1 transaction has only two modules yet to be visited. Row 1. column 2 of the visitation sequence matrix should consequently contain the number of the module to be visited next. The number in that cell is a "6". Module "6" is session services and, as Table V-II shows, the second operation on transaction class 1 is indeed performed in SS. and so on.

Table V-IV shows the mean service time matrix. Row and column indices have the same significance and interpretations as in Table V-III. Entries in the body of the matrix are mean service times. expressed in milliseconds (time unit). The entries in the given cells in Table V-IV are linked directly to entries in the corresponding cells in Table V-III. For example, when transaction class 1 visits module 4 (row 1. column 3, Table V-III), its mean service time there is 30 milliseconds (row 1. column 3, Table V-IV). Then. when transaction class 1 visits module 6 (row 1. column 2. Table V-III), its mean service time there is 15 milliseconds (row 1. column 2, Table V-IV), and so on.

It is a simple matter for a transaction to index into the proper cell of the visitation sequence and mean service time matrices. Assume that when a transaction enters the model, its transaction class is coded in Parameter 1 as a 1, or 2, or ... 12. Parameter 1 can then be used as a matrix row-index. Assume further that when a transaction arrives. the number of modules it must visit (yet to be visited) is copied to Parameter 2. Parameter 2 can then be used as a matrix column-index. When the first operation (service) has been performed, Parameter 2 can be decremented by 1, meaning that (1) its value can be interpreted as the number of modules yet to be visited, and (2) its value can continue to be used as the appropriate martix column-index. Suppose that halfword matrix save value 1 is used for the visitation sequence. Then "MH1(P1.P2)" indirectly specifies the proper current module number for the scheme just described. Suppose further that halfword matrix save value 2 is used for the mean service times. Then, because of the cell-to-cell correspondence between the two matrices. "MH2(P1,P2)" indirectly specifies the corresponding mean service time.

Model Segment 1, shown in Figure 5-2, should appear brief and straightforward. Location names have been supplied for the ASSIGN blocks, the ADVANCE block and the TEST block. At any given time, all jobs - transactions in the system - are either at these ASSIGN blocks (waiting for disk or memory or processing unit, in order to move into the ENTER block), or the ADVANCE block (because they are on the future events chain), or at the TEST block (waiting to move into the ENTER block again). Hence, the sum of the current counts at these blocks equals the total number of transactions in the model. The variable COUNT has this sum of current counts in its value. It is this variable that is evaluated at the end of each simulated period (six months) to determine how many transactions are currently in the system.

A timer-transaction enters Model Segment 2 at the end of each six month period to record the value of the variable COUNT in the Table T Jobs. After the processor resets the model to eliminate statistics accumulated during this period, the simulation for the next period is started.

Sample format for the output of the entire eleven years (1982-1993) simulation period is shown in Table V-VI. This figure is provided only to show format. It does not show actual results from the simulation.

| Operant | Significance | Default Value |
|---|---|---|
| A | Name of the matrix in which an element is to be modified | Error |
| B | Row subscript | Error |
| C | Column subscript | Error |
| D | Data to be used in the modification process | Error |
| E | The character H indicates that the matrix is on the halfword type | Matrix is of the fullword type |

Figure 5-1. The MSAVEVALUE Block and its A, B, C, D, E Operands

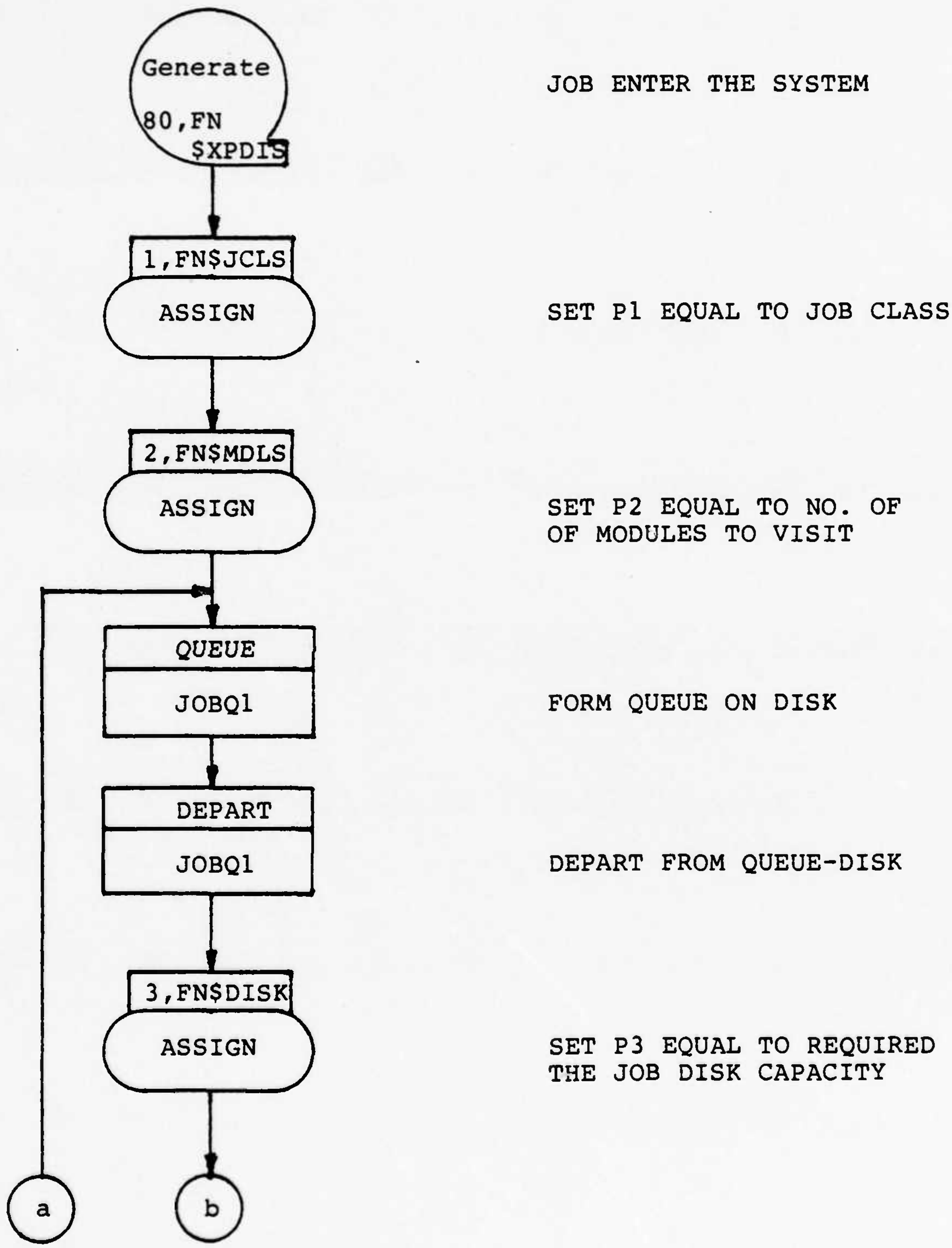
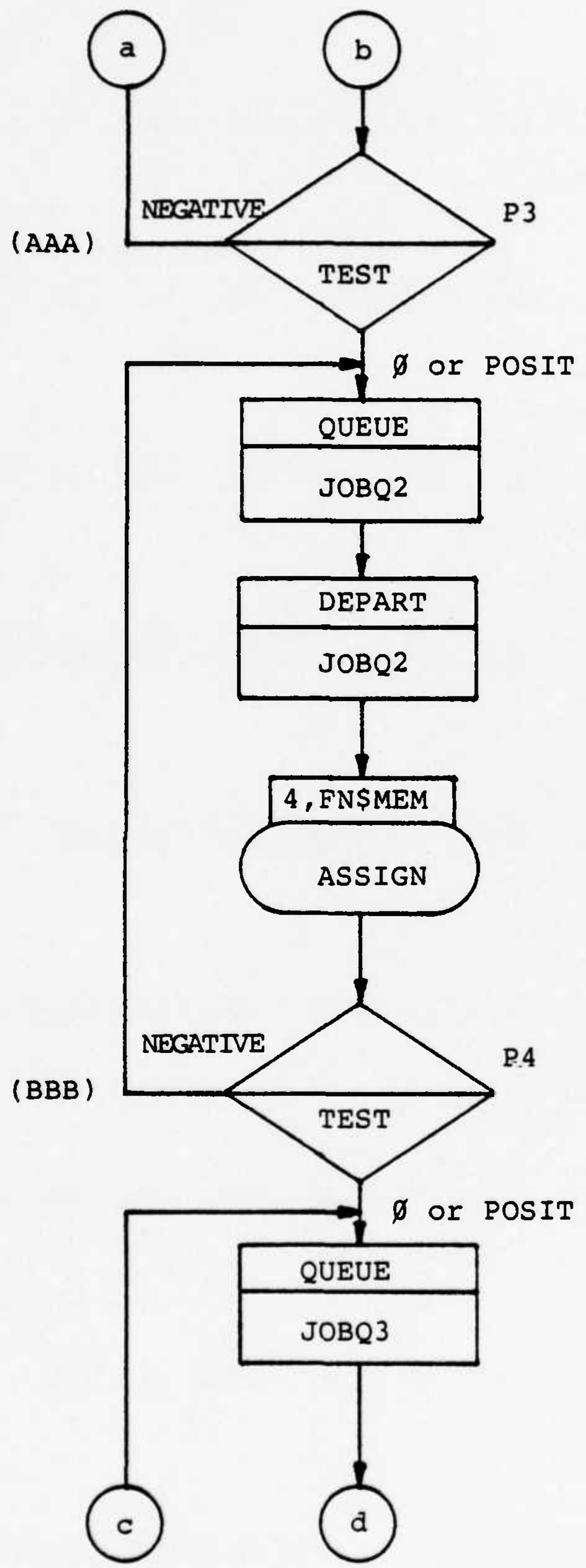



Figure 5.2. Block Diagram for GPSS Program (Hypothetical Data)




IS DISK AVAILABLE?
IF NOT GO BACK TO QUEUE LINE.

FORM QUEUE ON MEMORY STORAGE

DEPART FROM MEMORY QUEUE

SET P4 EQUAL TO MEMORY
CAPACITY REQUIRED BY THE JOB

IS MEMORY CAPACITY AVAILABLE?
IF NOT GO BACK TO QUEUE LINE.

FORM QUEUE ON PROCESSING UNITS

Figure 5.2 (continued)

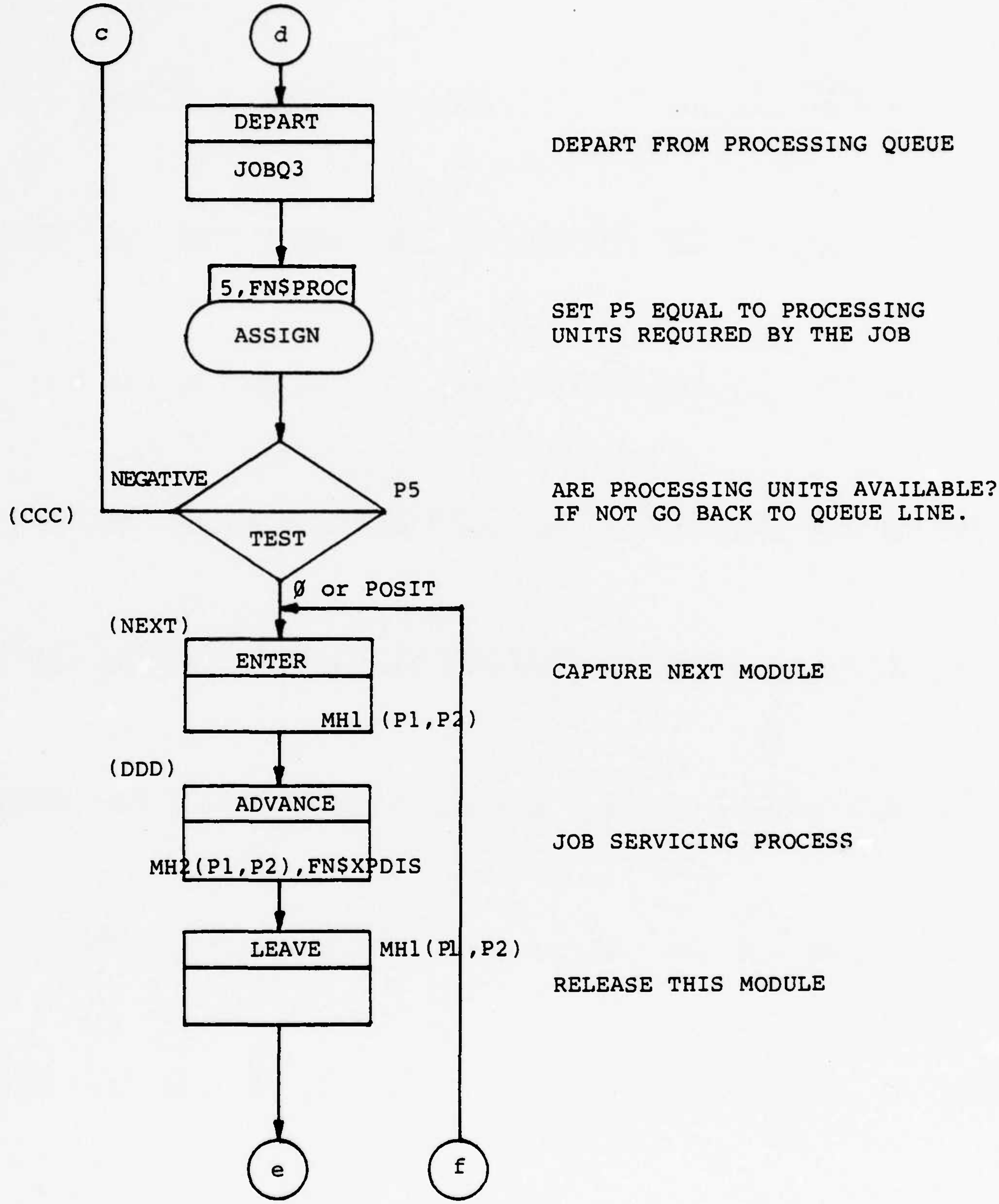
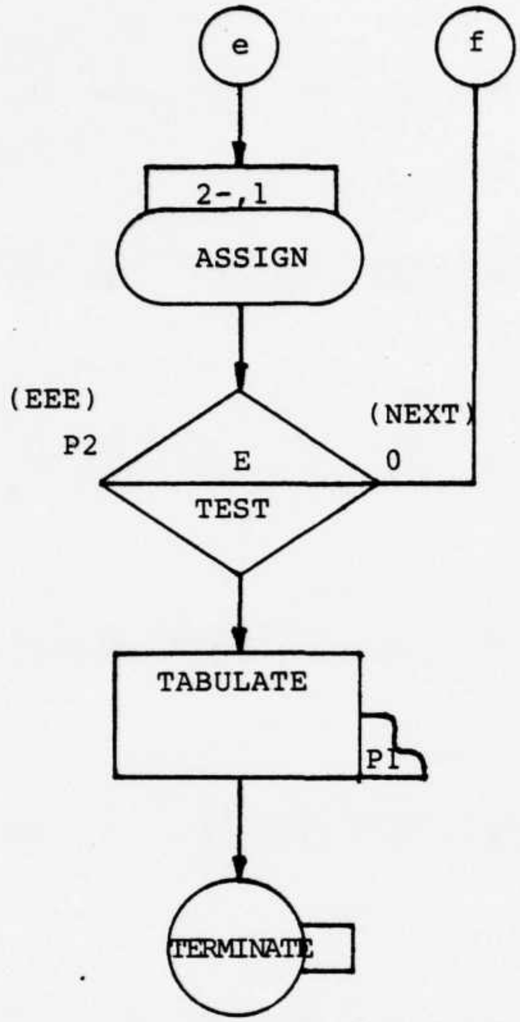



Figure 5.2 (continued)




UPDATE NO. OF MODULES YET TO BE VISITED

JOB DONE? IF NOT GO TO NEXT MODULE.

YES, RECORD TIME SPENT IN THE SYSTEM

LEAVE THE SYSTEM

MODEL SEGMENT 1

Figure 5.2 (continued)

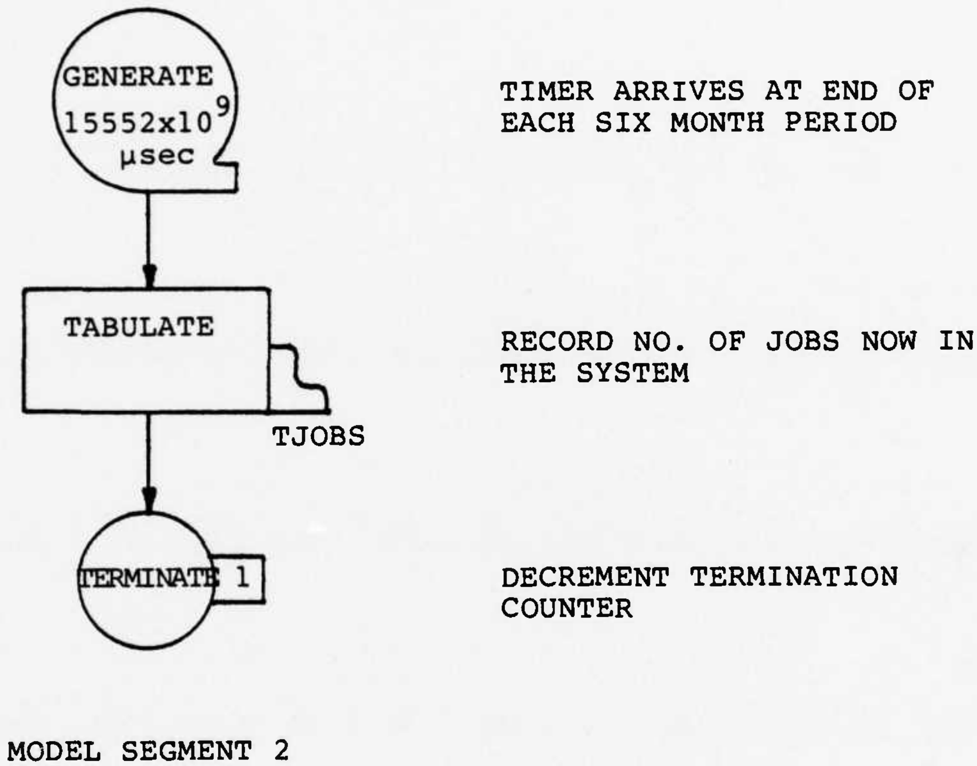



Figure 5.3. Model Segment 2 Diagram

TABLE V-V

TABLE OF DEFINITIONS

| GPSS ENTITY | INTERPRETATION |
|---|---|
| <u>Transactions</u> | |
| Model Segment 1 | A transaction |
| | P1: Parameter 1 values of 1,2...12 indicate transactions of class 1 through 12 respectively. |
| | P2: Parameter 2 indicates the total number of modules "yet to be visited" by a transaction. |
| | P3: Parameter 3 indicates the required transaction class disk storage. |
| | P4: Parameter 4 indicates the required transaction class memory storage. |
| | P5: Parameter 5 indicates the required transaction class processing units. |
| Model Segment 2 | A timer-transaction |
| <u>Functions</u> | |
| MDLS | A function describing the total number of modules each transaction class must visit. |
| TRCLS | A function describing the distribution of transaction-classes within the stream of arriving transactions. |
| XPDIS | Exponential distribution function. |
| <u>Matrix Savevalues</u> (Halfword) | |
| 1 | Visitation sequence matrix. |
| 2 | Mean service time matrix. |
| Storages | |
| 1,2,3,4,5,6,7,8 | Storages used to simulate modules 1 through 8, respectively. |
| Tables 1,2,...12 | Tables in which the system residence times of transaction class 1 through 12, respectively, are recorded. |
| TIOBS | Table used to record the total number of transactions in the system at the end of each six-month period. |
| Variables COUNT | A variable whose value equals the total number of transactions in the system. |

TABLE V-VI

PROGRAM OUTPUT
(Hypothetical Data)

| Average Number of Transactions in System | | Average System Residence Time (μsecs, by transaction class) 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1982.0 | 1100 | 250 | 110 | 50 | 300 | 220 | 180 | 100 | 40 | 30 | 20 | 10 | 400 |
| .5 | 1180 | 300 | 120 | 58 | 350 | 230 | 185 | 120 | 50 | 35 | 28 | 20 | 450 |
| 1983.0 | 1200 | 320 | 140 | 65 | 370 | 250 | 198 | 180 | 68 | 40 | 36 | 30 | 480 |
| .5 | 1240 | | | | | | | | | | | | |
| 1984.0 | 1290 | | | | | | | | | | | | |
| .5 | 1340 | | | | | | | | | | | | |
| 1985.0 | 1380 | | | | | | | | | | | | |
| .5 | 1400 | | | | | | | | | | | | |
| 1986.0 | 1450 | | | | | | | | | | | | |
| .5 | 1520 | | | | | | | | | | | | |
| 1987.0 | 1600 | | | | | | | | | | | | |
| .5 | 1640 | | | | | | | | | | | | |
| 1988.0 | 1700 | | | | | | | | | | | | |
| .5 | 1740 | | | | | | | | | | | | |
| 1989.0 | 1800 | | | | | | | | | | | | |
| .5 | 1830 | | | | | | | | | | | | |
| 1990.0 | 1880 | | | | | | | | | | | | |
| .5 | 1900 | | | | | | | | | | | | |
| 1991.0 | 1950 | | | | | | | | | | | | |
| .5 | 2000 | | | | | | | | | | | | |
| 1992.0 | 2050 | | | | | | | | | | | | |
| .5 | 2150 | | | | | | | | | | | | |
| 1993.0 | 2200 | | | | | | | | | | | | |

# VI. CONCLUSIONS

Simulation is a technique of growing importance in many fields, theoretical and applied. Distributed computer systems of any kind are too complex to predict their performance without the aid of a tool, such as simulation.

The generation of a representative job stream is one of the most important considerations in developing a simulation model for a computer system. First, parameter values must reflect the characteristics of individual jobs that are processed by the computer that is being simulated. Second, a set of individual jobs must be selected to represent the total workload of the computer. The third consideration involves generation of jobs with a pattern of interarrival times that matches the actual workload on the computer system.

The specifications of a Local Area Network (LAN) simulation model have been given in this thesis based on a SPLICE-LAN functional design. The model is specified to allow the evaluation of alternative LAN technologies operating under the forecasting workload of the SPLICE system.

The resolution and amount of detail to be presented in the model depends on the questions to be asked of the model. The more specialized a model becomes. the less able it is to answer new and unexpected questions. For the SPLICE LAN design in its present state of development, the simulation model appears to be adequate for obtaining a basic understanding of network performance.

It should be emphasized though that the construction of a simulation model is an iterative process. The LAN model specified in this thesis can be considered as a first generation model which allows for future growth by proceeding to a more complex and sophisticated level in future generation models.

# APPENDIX A

## WORKLOAD FORECASTS FOR NSC NORFOLK [Ref. 3]

| YEAR | PEAK HOUR TRANSACTION ARRIVAL RATE BY TRANSACTION CLASS (TRANS/SEC) | | | | | | | | | | | | TOTAL VOLUME |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TC1 | TC2 | TC3 | TC4 | TC5 | TC6 | TC7 | TC8 | TC9 | TC10 | TC11 | TC12 | |
| 1982.0 | 3.53861 | 1.02111 | .71837 | .40943 | 1.29135 | 2.21056 | .56324 | .81519 | .27623 | 1.62948 | .36155 | 14.51316 | 27.52411 |
| 1982.5 | 3.61110 | 1.04579 | .73664 | .42317 | 1.34343 | 2.25778 | .58842 | .84599 | .28476 | 1.66388 | .37191 | 14.78802 | 28.14027 |
| 1983.0 | 3.68512 | 1.07115 | .75545 | .43747 | 1.29783 | 2.30610 | .61479 | .87810 | .29362 | 1.69906 | .38264 | 15.06872 | 28.77300 |
| 1983.5 | 3.76071 | 1.09721 | .77482 | .45233 | 1.45466 | 2.35554 | .64241 | .91159 | .30281 | 1.73503 | .39374 | 15.35540 | 29.42285 |
| 1984.0 | 3.83790 | 1.12399 | .79476 | .46779 | 1.51402 | 2.40613 | .67134 | .94652 | .31236 | 1.77181 | .40523 | 15.64820 | 30.09038 |
| 1984.5 | 3.91672 | 1.15151 | .81530 | .48387 | 1.57606 | 2.45792 | .70164 | .98295 | .32226 | 1.80942 | .41713 | 15.94726 | 30.77619 |
| 1985.0 | 3.99722 | 1.17980 | .83646 | .50060 | 1.64088 | 2.51092 | .73338 | 1.02097 | .33255 | 1.84789 | .42945 | 16.25274 | 31.48088 |
| 1985.5 | 4.07943 | 1.20888 | .85826 | .51801 | 1.70862 | 2.56517 | .76663 | 1.06064 | .34324 | 1.88723 | .44222 | 16.56479 | 32.20510 |
| 1986.0 | 4.16339 | 1.23878 | .88072 | .53613 | 1.77942 | 2.62071 | .80146 | 1.10203 | .35434 | 1.92746 | .45544 | 16.88356 | 32.94950 |
| 1986.5 | 4.24914 | 1.26953 | .90387 | .55499 | 1.85343 | 2.67758 | .83796 | 1.14524 | .36588 | 1.96862 | .46915 | 17.20923 | 33.71478 |
| 1987.0 | 4.33673 | 1.30116 | .92773 | .57463 | 1.93080 | 2.73580 | .87619 | 1.19034 | .37787 | 2.01072 | .48336 | 17.54196 | 34.50164 |
| 1987.5 | 4.42619 | 1.33369 | .95233 | .59508 | 2.01169 | 2.79542 | .91626 | 1.23742 | .39033 | 2.05378 | .49808 | 17.88191 | 35.31083 |
| 1988.0 | 4.51757 | 1.36715 | .97770 | .61637 | 2.09626 | 2.85648 | .95824 | 1.28657 | .40329 | 2.09784 | .51336 | 18.22927 | 36.14313 |
| 1988.5 | 4.61092 | 1.40159 | 1.00386 | .63855 | 2.18469 | 2.91902 | 1.00223 | 1.33790 | .41676 | 2.14293 | .52919 | 18.58421 | 36.99934 |
| 1989.0 | 4.70627 | 1.43702 | 1.03085 | .66166 | 2.27717 | 2.98308 | 1.04833 | 1.29151 | .43077 | 2.18905 | .54562 | 18.94693 | 37.88029 |
| 1989.5 | 4.80368 | 1.47349 | 1.05869 | .68574 | 2.37389 | 3.04870 | 1.09665 | 1.44749 | .44535 | 2.23626 | .56267 | 19.31760 | 38.78688 |
| 1990.0 | 4.90319 | 1.51103 | 1.08741 | .71083 | 2.47506 | 3.11594 | 1.14728 | 1.50598 | .46052 | 2.28457 | .58036 | 19.69642 | 39.71999 |
| 1990.5 | 5.00486 | 1.54968 | 1.11706 | .73698 | 2.58088 | 3.18483 | 1.20035 | 1.56707 | .47630 | 2.33401 | .59872 | 20.08360 | 40.68057 |
| 1991.0 | 5.10874 | 1.58948 | 1.14766 | .76424 | 2.69158 | 3.25543 | 1.25597 | 1.63090 | .49273 | 2.38462 | .61778 | 20.47934 | 41.66962 |
| 1991.5 | 5.21487 | 1.63047 | 1.17925 | .79266 | 2.80740 | 3.32778 | 1.31427 | 1.69759 | .50983 | 2.43642 | .63758 | 20.88385 | 42.68815 |
| 1992.0 | 5.32331 | 1.67269 | 1.21187 | .82229 | 2.92858 | 3.40195 | 1.37538 | 1.76729 | .52764 | 2.48945 | .65313 | 21.29735 | 43.73723 |
| 1992.5 | 5.43413 | 1.71618 | 1.24556 | .85319 | 3.05537 | 3.47798 | 1.43944 | 1.84012 | .54618 | 2.54375 | .67948 | 21.72005 | 44.81798 |
| 1993.0 | 5.54736 | 1.76100 | 1.28036 | .88542 | 3.18806 | 3.55593 | 1.50660 | 1.91626 | .56550 | 2.59934 | .70167 | 22.15220 | 45.93155 |
| 1993.5 | 5.66308 | 1.80718 | 1.31631 | .91905 | 3.32692 | 3.63585 | 1.57700 | 1.99584 | .58561 | 2.65627 | .72472 | 22.59402 | 47.07915 |
| TOTAL SUM | 104.82914 | 32.95951 | 23.61129 | 15.04048 | 51.48805 | 68.66260 | 23.63546 | 31.52150 | 9.81613 | 50.39889 | 12.45918 | 437.63979 | 862.06262 |
| FREQ | 12.16 | 3.82 | 2.74 | 1.74 | 5.97 | 7.96 | 2.74 | 3.66 | 1.14 | 5.85 | 1.45 | 50.77 | 100% |

# APPENDIX B

## PROCESS LOAD FORECASTS FOR NSC NORFOLK [Ref. 3]

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EDIT | MESSAGE LENGTH | 2 | 200 | 200 | 50 | 175 | 800 | 175 | 175 | 30 | 800 | 80 | 80 |
| | NO. OF INSTRUCTIONS | 40 | 50 | 50 | 100 | 100 | 250 | 100 | 100 | 300 | 300 | 30 | 10 |
| | % OF FAILURE | 1 | 1 | 1 | 1 | 5 | 8 | 1 | 5 | 2 | 10 | 2 | 1 |
| VALIDATION READ | NO. OF RECORDS | 1 | 10 | 18 | 1 | 8 | 20 | 1 | 8 | 5 | 20 | 5 | 0 |
| | RECORD LENGTH | 1500 | 250 | 350 | 100 | 250 | 350 | 200 | 250 | 150 | 350 | 150 | 0 |
| | NO. OF INSTRUCTIONS PER ACCESS | 5 | 20 | 20 | 10 | 20 | 30 | 10 | 10 | 20 | 30 | 20 | 0 |
| | % FAIL | 0 | 1 | 1 | 1 | 2 | . 3 | 1 | 2 | 3 | 3 | 3 | 0 |
| VAL-IDA-TION | NO. OF INSTRUCTIONS | 0 | 0 | 0 | 50 | 150 | 500 | 50 | 130 | 300 | 500 | 500 | 0 |
| | % FAIL | 0 | 0 | 0 | 1 | 1 | 2 | 1 | 2 | 2 | . 2 | 2 | 0 |
| ERROR MSG | NO. OF INSTRUCTIONS | 5 | 5 | 5 | 30 | 50 | 50 | 30 | 40 | 100 | 50 | 100 | 35 |
| | MESSAGE LENGTH | 80 | 80 | 80 | 500 | 600 | 1500 | 500 | 800 | 80 | 1500 | 80 | 150 |
| PROCESS-ING READ | NO. OF RECORDS | 0 | 0 | 0 | 0 | 5 | 10 | 0 | 4 | 100 | 10 | 25 | 1 |
| | RECORD LENGTH | 0 | 0 | 0 | 0 | 200 | 350 | 0 | 200 | 300 | 350 | 150 | 80 |
| | NO. OF INSTRUCTIONS | 0 | 0 | 0 | 0 | 10 | 20 | 0 | 15 | 5 | 20 | 15 | 10 |
| PROC. | NO. OF INSTRUCTIONS | 0 | 0 | 0 | 0 | 175 | 250 | 0 | 175 | 500 | 250 | 2500 | 50 |
| FILE WRITE | NO. OF INSTRUCTIONS | 0 | 0 | 0 | 20 | 30 | 30 | 20 | 20 | 20 | 30 | 20 | 10 |
| | NO. OF MODIFIED RECORD | 0 | 0 | 0 | 0 | 5 | 15 | 1 | 10 | 200 | 20 | 5 | 0 |
| | LENGTH OF MODIFIED RECORD | 0 | 0 | 0 | 0 | 200 | 250 | 200 | 250 | 100 | 350 | 250 | 0 |
| | NO. OF ADDS | 0 | 0 | 0 | 1 | 2 | 15 | 0 | 0 | 100 | 0 | 2 | 1 |
| | LENGTH OF ADDED RECORD | 0 | 0 | 0 | 100 | 250 | 350 | 0 | 0 | 200 | 0 | 75 | 80 |
| | NO. OF INDICIES | 0 | 0 | 0 | 5 | 10 | 10 | 5 | 0 | 4 | 0 | 3 | 2 |
| FORMAT OUTPUT | NO. OF INSTRUCTIONS | 5 | 50 | 50 | 20 | 30 | 50 | 20 | 30 | 50 | 50 | 50 | 20 |
| | MESSAGE LENGTH | 1500 | 1000 | 1800 | 500 | 1000 | 1500 | 500 | 750 | 132 | 1500 | 80 | 80 |
| | NO. OF RECORDS | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 400 | 1 | 125 | 1 |

## APPENDIX C

### NSC NORFOLK
### GOODNESS OF FIT TEST FOR TOTAL DATA INDEPENDENTLY OF TRANSACTION CLASS

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $.2 \le x \le 1.8$ | 1.0 | 178 | .6687 | 192.59 |
| 2 | $1.8 \le x \le 3.4$ | 2.6 | 58 | .1666 | 47.98 |
| 3 | $3.4 \le x \le 5$ | 4.2 | 21 | .0570 | 16.42 |
| 4 | $5 \le x \le 6.6$ | 5.8 | 7 | .9243 | 6.99 |
| 5 | $6.6 \le x \le 8.2$ | 7.4 | 0 | .0118 | 3.40 |
| 6 | $8.2 \le x \le 9.8$ | 9 | 0 | .0067 | 1.93 |
| 7 | $9.8 \le x \le 11.4$ | 10.6 | 0 | .0038 | 1.10 |
| 8 | $11.4 \le x \le 13$ | 12.2 | 0 | .0022 | .63 |
| 9 | $13 \le x \le 14.6$ | 13.8 | 1 | .0015 | .43 |
| 10 | $14.6 \le x \le 16.2$ | 15.4 | 5 | .0010 | .29 |
| 11 | $16.2 \le x \le 17.8$ | 17 | 5 | .0007 | .20 |
| 12 | $17.8 \le x \le 19.4$ | 18.6 | 5 | .0005 | .15 |
| 13 | $19.4 \le x \le 21$ | 20.2 | 4 | .0003 | .09 |
| 14 | $21 \le x \le 22.6$ | 21.8 | 4 | .0003 | .09 |
| | | | 288 | | 272.3 |

To find the probability that a random variable having the log-normal distribution assumes a value between a and b $(0<a<b)$, we apply the following formula:

$$\rho(a \le X \le b) = \int_a^b \frac{1}{\sqrt{27\ell}} X^{-1} e^{-(\ell_n X-\alpha)^2/2f^2} \, dX$$

Changing variables by letting $y = \ell_n X$ and hence, $dy = X^{-1} dX$, we obtain:

$$\rho(a \le X \le b) = \int_{\ell_n a}^{\ell_n b} \frac{1}{\sqrt{27\ell}} e^{-(y-\alpha)^2/2\beta^2} dy$$

and it can be seen that this probability equals the probability that a random variable having the normal distribution with $\mu=\alpha$ and $=\beta$ assumes a value between $\ell_n a$ and $\rho_\mu b$. Thus,

$$\rho(a \le X \le b) = F\left(\frac{\ell_n b-\alpha}{\beta}\right) - F\left(\frac{\ell_n a-\alpha}{\beta}\right)$$

where F(Z) is the probability that a random variable having the standard normal distribution assumes a value.

The long-normal distribution occurs in practice whenever we encounter a random variable which is such that its logarithm has a normal distribution.

Testing logarithmic distribution with parameter values $\alpha=0$, $\ell=1$ we have:

$$\chi^2 = \sum_{i=1}^{14} \frac{(n_i-t_i)^2}{t_i} = 1.1053 + 2.0926 + 1.2354 + 0 + 3.4 + 1.93 + 1.1 + .63 + .7556 + 76.49 + 115.19 + \ldots = 530.6$$

Taking the sum up to 9th interval we have $\chi^2 = 12.27$.

From the $\chi^2$ tables we have the following values:

a. Degree of freedom 13 $\chi^2 = .05 = 22.362$

b. Degree of freedom 8 $\chi^2 = .05 = 15.507$

Since $\chi^2 > \chi^2_{13,.05}$ the hypothesis that the distribution is logarithmic cannot be accepted. Other theoritical distributions have been tested without good results. Taking the intervals up to the 9th interval, we find that $\chi^2 < \chi^2_{8,.05}$ and consequently the hypothesis cannot be rejected for those particular data. Some data smoothing can be applied in order for the data to fit, but even so, the logarithmic distribution does not fit the data.

## APPENDIX D

## HORIZONTAL FREQUENCY TABLE PER TRANSACTION CLASS OVER SIX MONTH PERIOD

| | TC1 | TC2 | TC3 | TC4 | TC5 | TC6 | TC7 | TC8 | TC9 | TC10 | TC11 | TC12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1982.0 | 12.93905 | 3.73372 | 2.62674 | 1.49709 | 4.72186 | 8.08299 | 2.0595 | 2.98077 | 1.01004 | 5.95825 | 1.32202 | 53.0679 |
| .5 | 12.91482 | 3.74018 | 2.63453 | 1.51343 | 4.80467 | 8.07477 | 2.10443 | 3.02561 | 1.01842 | 5.95074 | 1.3301 | 52.88823 |
| 1983.0 | 12.88928 | 3.74651 | 2.6423 | 1.53012 | 4.88913 | 8.06594 | 2.15032 | 3.07129 | 1.02698 | 5.94273 | 1.33834 | 52.70522 |
| .5 | 12.86335 | 3.75296 | 2.64023 | 1.54717 | 4.9756 | 8.05702 | 2.19675 | 3.11805 | 1.03574 | 5.93459 | 1.34677 | 52.5225 |
| 1984.0 | 12.83576 | 3.34448 | 2.65805 | 1.56451 | 5.0636 | 8.04724 | 2.24528 | 3.16561 | 1.04468 | 5.92577 | 1.35528 | 52.33502 |
| .5 | 12.80725 | 3.76531 | 2.66594 | 1.5822 | 5.15354 | 8.03713 | 2.29428 | 3.21414 | 1.05375 | 5.9166 | 1.36397 | 52.14583 |
| 1985.0 | 12.77766 | 3.77139 | 2.67386 | 1.60023 | 5.2453 | 8.0265 | 2.34435 | 3.26367 | 1.06304 | 5.90703 | 1.37279 | 51.95413 |
| .5 | 12.74697 | 3.77738 | 2.6818 | 1.61862 | 5.33891 | 8.01537 | 2.39548 | 3.31417 | 1.07252 | 5.89701 | 1.3818 | 51.75992 |
| 1986.0 | 12.71518 | 3.78329 | 2.68076 | 1.63736 | 5.43443 | 8.00377 | 2.44769 | 3.36565 | 1.08217 | 5.88655 | 1.39093 | 51.56318 |
| .5 | 12.68225 | 3.78911 | 2.69774 | 1.65645 | 5.53186 | 7.99167 | 2.50102 | 3.41815 | 1.09202 | 5.87566 | 1.40025 | 51.36375 |
| 1987.0 | 12.64821 | 3.79487 | 2.70575 | 1.67592 | 5.63124 | 7.97904 | 2.55543 | 3.47166 | 1.10207 | 5.86433 | 1.40973 | 51.16169 |
| .5 | 12.61303 | 3.80053 | 2.71379 | 1.69576 | 5.73258 | 7.96593 | 2.611 | 3.52619 | 1.11229 | 5.85252 | 1.41934 | 50.95696 |
| 1988.0 | 12.57671 | 3.80608 | 2.72187 | 1.71594 | 5.83589 | 7.95231 | 2.66769 | 3.58175 | 1.12274 | 5.84029 | 1.42917 | 50.74949 |
| .5 | 12.53926 | 3.81158 | 2.72996 | 1.73651 | 5.9412 | 7.93819 | 2.72553 | 3.63838 | 1.13336 | 5.82763 | 1.43911 | 50.53923 |
| 1989.0 | 12.50063 | 3.81696 | 2.7381 | 1.75747 | 6.04853 | 7.92355 | 2.78453 | 3.69608 | 1.14419 | 5.81447 | 1.44925 | 50.32617 |
| .5 | 12.46084 | 3.82226 | 2.74626 | 1.77882 | 6.15791 | 7.90839 | 2.84473 | 3.75481 | 1.15524 | 5.8009 | 1.45957 | 50.11023 |
| 1990.0 | 12.41987 | 3.82746 | 2.75442 | 1.80054 | 6.26937 | 7.89273 | 2.90608 | 3.81467 | 1.1665 | 5.78685 | 1.47006 | 49.89139 |
| .5 | 12.37774 | 3.83258 | 2.76265 | 1.82265 | 6.38289 | 7.87654 | 2.96864 | 3.87559 | 1.17795 | 5.77234 | 1.48072 | 49.66966 |
| 1991.0 | 12.33444 | 3.83761 | 2.77088 | 1.84516 | 6.4985 | 7.85985 | 3.03239 | 3.93761 | 1.18963 | 5.75738 | 1.49155 | 49.44494 |
| .5 | 12.28995 | 3.84255 | 2.77915 | 1.86807 | 6.61623 | 7.84262 | 3.09735 | 4.00073 | 1.20152 | 5.74194 | 1.50247 | 49.21725 |
| 1992.0 | 12.24566 | 3.84783 | 2.48776 | 1.89158 | 6.73686 | 7.82579 | 3.1639 | 4.06544 | 1.21377 | 5.72669 | 1.50245 | 48.99209 |
| .5 | 12.19743 | 3.85213 | 2.79578 | 1.91506 | 6.85807 | 7.80666 | 3.23096 | 4.13032 | 1.22595 | 5.70969 | 1.52515 | 48.75275 |
| 1993.0 | 12.14935 | 3.85679 | 2.80413 | 1.93917 | 6.98221 | 7.78789 | 3.29962 | 4.19683 | 1.23851 | 5.69285 | 1.53673 | 48.51588 |
| .5 | 12.10007 | 3.86132 | 2.81212 | 1.96369 | 7.10849 | 7.76857 | 3.36951 | 4.26443 | 1.25124 | 5.67554 | 1.54847 | 48.27572 |

NOTE: The results are consistent with the frequency observed in the table in Appendix A.

APPENDIX E

NCS NORFOLK
GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-1
OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $3.5<x\leq 3.9$ | 3.7 | 5 | .182 | 4.368 |
| 2 | $3.9<x\leq 4.3$ | 4.1 | 5 | .182 | 4.368 |
| 3 | $4.3<x\leq 4.7$ | 4.5 | 4 | .182 | 4.368 |
| 4 | $4.7<x\leq 5.1$ | 4.9 | 4 | .182 | 4.368 |
| 5 | $5.1<x\leq 5.5$ | 5.3 | 4 | .182 | 4.368 |
| 6 | $5.5<x\leq 5.7$ | 5.6 | 2 | .090 | 2.160 |
| | | | 24 | | 24 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{5} \frac{(n_i-t_i)^2}{t_i} = .091 + .091 + .091 + .091 + .091 + 1.376$$

$$= 1.831$$

With degree of freedom 5 and level of significance = .05 we have from the tables $\chi^2_{.05} = 11.070$.

Since $\chi^2<\chi^2_{5;.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

## APPENDIX F

NSC NORFOLK
GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-2
OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $1.0<x\leq 1.15$ | 1.075 | 5 | .167 | 4.008 |
| 2 | $1.15<x\leq 1.3$ | 1.225 | 5 | .167 | 4.008 |
| 3 | $1.3<x\leq 1.45$ | 1.375 | 5 | .167 | 4.008 |
| 4 | $1.45<x\leq 1.6$ | 1.525 | 4 | .167 | 4.008 |
| 5 | $1.6<x\leq 1.75$ | 1.675 | 3 | .166 | 3.984 |
| 6 | $1.75<x\leq 1.83$ | 1.825 | 2 | .165 | 3.96 |
| | | | 24 | | 23.976 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{6} \frac{(n_i - t_i)^2}{t_i} = .246 + .246 + .246 + 0 + .243 + .970$$

$$= 1.951$$

With degree of freedom 5 and level of significance = .05 we have from the tables $\chi^2_{.05} = 11.070$.

Since $\chi^2 < \chi^2_{5,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

APPENDIX G

NSC NORFOLK
GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-3
OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $.7<x\leq.824$ | .762 | 6 | .201 | 4.824 |
| 2 | $.824<x\leq.948$ | .886 | 5 | .201 | 4.824 |
| 3 | $.948<x\leq1.072$ | 1.010 | 5 | .201 | 4.824 |
| 4 | $1.072<x\leq1.196$ | 1.134 | 4 | .200 | 4.8 |
| 5 | $1.196<x\leq1.317$ | 1.258 | 4 | .197 | 4.728 |
| | | | 24 | | 24 |

*Testing for uniform* distribution we have:

$$\chi^2 = \sum_{i=1}^{6} \frac{(n_i - t_i)^2}{t_i} = .287 + .006 + .006 + .133 + .112 = .544$$

With degree of freedom 4 and level of significance = .05

we have from the tables $\chi^2_{.05} = 9.488$.

Since $\chi^2 < \chi^2_{5,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

## APPENDIX H

### NCS NORFOLK
### GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-4
### OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $.4<x\leq.465$ | .40325 | 4 | .125 | 3 |
| 2 | $.465<x\leq.530$ | .46825 | 4 | .125 | 3 |
| 3 | $.530<x\leq.595$ | .53325 | 3 | .125 | 3 |
| 4 | $.595<x\leq.66$ | .59825 | 3 | .125 | 3 |
| 5 | $.66<x\leq.725$ | .66325 | 3 | .125 | 3 |
| 6 | $.725<x\leq.79$ | .72825 | 2 | .125 | 3 |
| 7 | $.79<x\leq.855$ | .79325 | 3 | .125 | 3 |
| 8 | $.855<x\leq.92$ | .85825 | 2 | .125 | 3 |
| | | | 24 | 1.00 | 24 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{8} \frac{(n_i-t_i)^2}{t_i} = .333 + .333 + 0 + 0 + 0 + .333 + 0 + .333 = 1.333$$

With degree of freedom 7 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 14.067$.

Since $\chi^2<\chi^2_{4,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

APPENDIX I

NSC NORFOLK
GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-5
OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $1.29<x\leq 1.545$ | 1.4175 | 5 | .125 | 3 |
| 2 | $1.545<x\leq 1.8$ | 1.6725 | 4 | .125 | 3 |
| 3 | $1.8<x\leq 2.055$ | 1.9275 | 3 | .125 | 3 |
| 4 | $2.055<x\leq 2.310$ | 2.1825 | 3 | .125 | 3 |
| 5 | $2.310<x\leq 2.565$ | 2.4375 | 2 | .125 | 3 |
| 6 | $2.565<x\leq 2.820$ | 2.6925 | 3 | .125 | 3 |
| 7 | $2.820<x\leq 3.075$ | 2.9475 | 2 | .125 | 3 |
| 8 | $3.075<x\leq 3.327$ | 3.201 | 2 | .125 | 2.98 |
| | | | 24 | 1 | 23.98 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{8} \frac{(n_i - t_i)^2}{t_i} = 1.333 + .333 + 0 + 0 + .333 + 0 + .333 + .320 = 2.652$$

With degree of freedom 7 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 14.067$.

Since $\chi^2 < \chi^2_{7,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

## APPENDIX J

### NSC NORFOLK
### GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-6
### OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $2.21<x\leq 2.496$ | 2.353 | 6 | .2 | 4.8 |
| 2 | $2.496<x\leq 2.782$ | 2.639 | 5 | .2 | 4.8 |
| 3 | $2.782<x\leq 3.068$ | 2.925 | 5 | .2 | 4.8 |
| 4 | $3.068<x\leq 3.354$ | 3.211 | 4 | .2 | 4.8 |
| 5 | $3.354<x<3.64$ | 3.497 | 4 | .2 | 4.8 |
| | | | 24 | 1.0 | 24 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{5} \frac{(n_i - t_i)^2}{t_i} = .299 + .008 + .008 + .219 + .219 = .753$$

With degree of freedom 4 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 9.488$.

Since $\chi^2 < \chi^2_{7,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

## APPENDIX K

### NSC NORFOLK
### GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-7 OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $.56<x\leq.73$ | .645 | 6 | .166 | 3.984 |
| 2 | $.73<x\leq.90$ | .815 | 5 | .166 | 3.984 |
| 3 | $.90<x\leq1.07$ | .985 | 4 | .166 | 3.984 |
| 4 | $1.07<x\leq1.24$ | 1.155 | 3 | .166 | 3.984 |
| 5 | $1.24<x\leq1.41$ | 1.325 | 3 | .166 | 3.984 |
| 6 | $1.41<x\leq1.578$ | 1.495 | 3 | .165 | 3.96 |
| | | | 24 | .995 | 23.88 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{6} \frac{(n_i-t_i)^2}{t_i} = 1.02 + .25 + .00006 + .24 + .24 + .23$$

$$= 1.99$$

With degree of freedom 5 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 11.070$.

Since $\chi^2<\chi^2_{7,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

## APPENDIX L

NSC NORFOLK
GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-8
OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $.8<x\le 1.0$ | .9 | 6 | .167 | 4.008 |
| 2 | $1.0<x\le 1.2$ | 1.1 | 5 | .167 | 4.008 |
| 3 | $1.2<x\le 1.4$ | 1.3 | 4 | .167 | 4.008 |
| 4 | $1.4<x\le 1.6$ | 1.5 | 3 | .167 | 4.008 |
| 5 | $1.6<x\le 1.8$ | 1.7 | 3 | .167 | 4.008 |
| 6 | $1.8<x\le 2.0$ | 1.9 | 3 | .165 | 3.96 |
| | | | 24 | 1.0 | 24 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{6} \frac{(n_i - t_i)^2}{t_i} = .99 + .246 + .0 + .254 + .233 = 1.723$$

With degree of freedom 5 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 11.07$.

Since $\chi^2<\chi^2_{7,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

## APPENDIX M

### NSC NORFOLK
### GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-9
### OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $.2<x\leq.28$ | .24 | 1 | .2 | 4.8 |
| 2 | $.28<x\leq.36$ | .32 | 8 | .2 | 4.8 |
| 3 | $.36<x\leq.44$ | .4 | 6 | .2 | 4.8 |
| 4 | $.44<x\leq.52$ | .48 | 5 | .2 | 4.8 |
| 5 | $.52<x\leq.6$ | .56 | 4 | .2 | 4.8 |
| | | | 24 | 1.0 | 24 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{6} \frac{(n_i-t_i)^2}{t_i} = 3.008 + 2.133 + .299 + .008 + .134 = 5.582$$

With degree of freedom 4 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 9.488$.

Since $\chi^2<\chi^2_{7,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

APPENDIX N

NSC NORFOLK
GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-10
OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $1.62<x\leq 1.75$ | 1.685 | 4 | .125 | 3 |
| 2 | $1.75<x\leq 1.88$ | 1.815 | 3 | .125 | 3 |
| 3 | $1.88<x\leq 2.01$ | 1.945 | 3 | .125 | 3 |
| 4 | $2.01<x\leq 2.14$ | 2.075 | 3 | .125 | 3 |
| 5 | $2.14<x\leq 2.27$ | 2.205 | 3 | .125 | 3 |
| 6 | $2.27<x\leq 2.4$ | 2.335 | 3 | .125 | 3 |
| 7 | $2.4<x\leq 2.53$ | 2.465 | 2 | .125 | 3 |
| 8 | $2.53<x\leq 2.66$ | 2.595 | 3 | .125 | 3 |
| | | | 24 | 1.0 | 24 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{8} \frac{(n_i - t_i)^2}{t_i} = .333 + 0 + 0 + 0 + 0 + 0 + .333 + 0$$

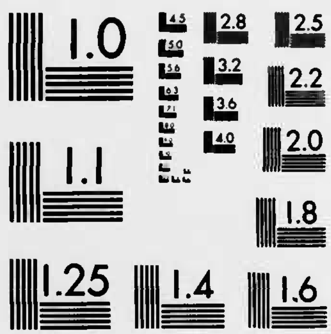
$$= .666$$

With degree of freedom 7 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 14.067$.

Since $\chi^2 < \chi^2_{7,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

AD-A132 142 SPECIFICATIONS OF A SIMULATION MODEL FOR A LOCAL AREA NETWORK DESIGN IN S..(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA C P CHARPANTIDIS JUN 83 2/2

UNCLASSIFIED F/G 9/2 NL




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

## APPENDIX O

NSC NORFOLK
GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-11
OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $.36<x\leq.433$ | .3965 | 7 | .2 | 4.8 |
| 2 | $.433<x\leq.506$ | .4695 | 5 | .2 | 4.8 |
| 3 | $.506<x\leq.579$ | .5425 | 4 | .2 | 4.8 |
| 4 | $.579<x\leq.652$ | .6155 | 4 | .2 | 4.8 |
| 5 | $.652<x\leq.725$ | .6885 | 4 | .2 | 4.8 |
| | | | 24 | 1.0 | 24 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{5} \frac{(n_i - t_i)^2}{t_i} = 1.009 + .008 + .133 + .133 + .133$$

$$= 1.416$$

With degree of freedom 4 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 9.488$.

Since $\chi^2 < \chi^2_{7,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

## APPENDIX P

### NSC NORFOLK
### GOODNESS OF FIT TEST ON DATA FOR TRANSACTION CLASS-12
### OVER 1982-1993

| Interval Number | Interval | Mean | Observed Frequency | Probability | Expected Frequency |
|---|---|---|---|---|---|
| $i$ | $I_i$ | $X_i$ | $n_i$ | $P_i$ | $t_i$ |
| 1 | $14.4<x\leq16.04$ | 15.22 | 6 | .2 | 4.8 |
| 2 | $16.04<x\leq17.68$ | 16.86 | 5 | .2 | 4.8 |
| 3 | $17.68<x\leq19.32$ | 18.50 | 5 | .2 | 4.8 |
| 4 | $19.32<x\leq20.96$ | 20.14 | 4 | .2 | 4.8 |
| 5 | $20.96<x\leq22.6$ | 21.78 | 4 | .2 | 4.8 |
| | | | 24 | 1.0 | 24 |

Testing for uniform distribution we have:

$$\chi^2 = \sum_{i=1}^{5} \frac{(n_i - t_i)^2}{t_i} = .299 + .008 + .008 + .133 + .133 = .581$$

With degree of freedom 4 and level of significance = .05 we have from the tables: $\chi^2_{.05} = 9.488$.

Since $\chi^2<\chi^2_{7,.05}$ the hypothesis cannot be rejected and the data fit in a uniform distribution.

## LIST OF REFERENCES


1. Schneidewind, Norman F., Functional Design of a Local Area Network for the Stock Point Logistics Integrated Communication Environment, December 1982.

2. Gordon, G., System Simulation, Prentice Hall, 1969.

3. Network Analysis Corporation. Revised Stock Point Integrated Communication Environment (SPLICE) Configuration Analysis. Washington Operations. 301 Tower Building, Vienna. VA 22180. 28 April 1980.

4. Karp, Peggy M. and Socher, Ivan D., Designing Local Area Networks, Amdax Corp., April 1982.

5. Tanenbaum. A.S., Computer Networks, Prentice Hall. Englewood Cliffs. NJ, 1981.

6. Raghavendra. C.S., Gerla, M., and Parker, D.S., Multiconnected Loop Topologies for LANs, Proceedings of INFOCOM 82. Las Vegas. March 1982 (1st joint conference of IEEE Computer Communication Society).

7. IEEE Project 802, Local Area Networks, Draft C, IEEE, May 17. 1982.

8. Nadkarni, Ashok V., Chanson, Samuel T., and Kumar. A., Performance of Some Local Area Network Technologies, COMPCON 83. March 1983.

9. U.S. Naval Supply Systems Command (Code 041). SPLICE Pseudo-Actual Transaction and Workload Comparison Report, 31 December 1982.

10. Martin, F.F., Computer Modeling and Simulation, Wiley. 1968.

11. Hamming, Richard W., How do You Know the Simulation is Relevant?, lecture. New Jersey, 1975.

12. Maisel. Herbet and Gnugnoli. Giuliano. Simulation of Discrete Stochastic Systems, 1975.

13. G. Scott Graham. Queueing Network Models of Computer System Performance, Computing Surveys. Vol. 10. No. 3. September 1978.

14. Schriber, Thomas J., Simulation Using GPSS, Copyright C. 1974.

15. Allen, A.D., Probability, Statistics, and Queueing Theory with Computer Science Applications, Academic Press, 1978.

# INITIAL DISTRIBUTION LIST

| | | No. Copies |
|---|---|---|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, Virginia 22314 | 2 |
| 2. | Library, Code 0142<br>Naval Postgraduate School<br>Monterey. California 93940 | 2 |
| 3. | Department Chairman. Code 52Hq<br>Department of Computer Science<br>Naval Postgraduate School<br>Monterey. California 93940 | 1 |
| 4. | Prof. Norman F. Schneidewind. Code 54Ss<br>Naval Postgraduate School<br>Monterey. California 93940 | 1 |
| 5. | Hellenic Navy General Staff<br>Stratopedon Papagou<br>Holargos, Athens, Greece | 2 |
| 6. | CDR Cosmas Charpantidis<br>Hellenic Navy General Staff<br>Stratopedon Papagou<br>Holargos, Athens. Greece | 4 |
| 7. | LCDR Ted Case<br>Fleet Material Support Office<br>Code 94L<br>Mechanicsburg, Pennsylvania 17055 | 1 |
| 8. | LCDR David Fuller<br>Commander. Naval Supply Systems Command<br>Code 0415A<br>Washington, DC 20376 | 1 |
| 9. | Computer Technologies Curricular Office<br>Code 37<br>Naval Postgraduate School<br>Monterey. California 93940 | 1 |

END

FILMED

9-83

DTIC