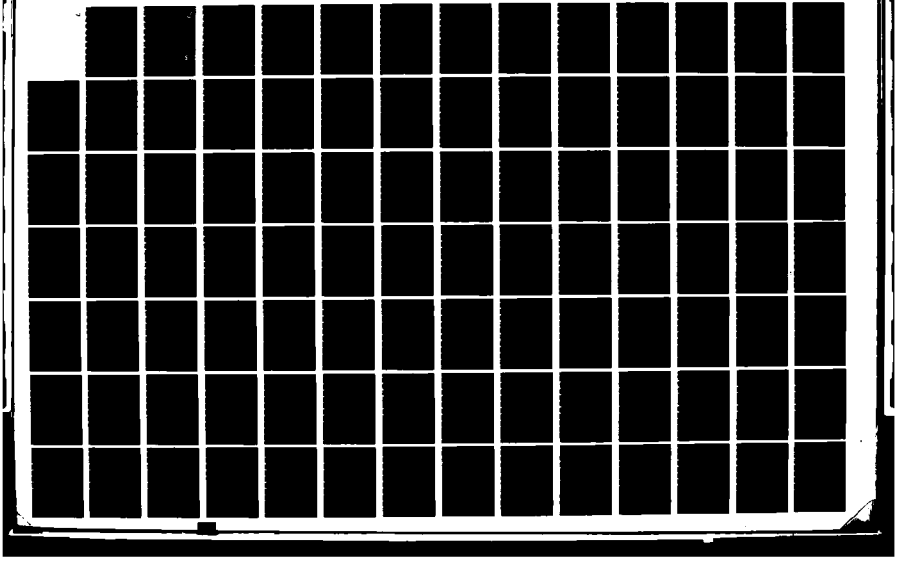
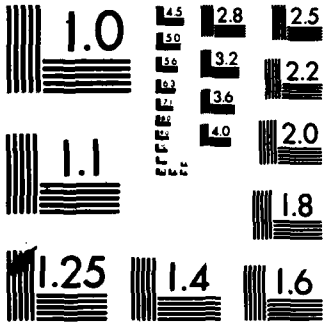
AD-A138 079 DESIGN OF A SECURE LOCAL NETWORK(U) AIR FORCE INST OF 1/2
TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
R G CUADROS DEC 83 AFIT/GCS/EE/83D-6

UNCLASSIFIED F/G 17/2 NL





DESIGN OF A SECURE LOCAL NETWORK

THESIS

AFIT/GCS/EE/83D-6 Ricardo G. Cuadros
Captain USAF

Approved for public release; distribution unlimited

AFIT/GCS/EE/83D-6

DESIGN OF A SECURE LOCAL NETWORK

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

Ricardo Cuadros, B. S., M. B. A.

Captain USAF

Graduate Computer Technology

December 1983

Acknowledgements

This thesis could not have been completed without the support and help of many individuals. Family, friends, and mentors, to those that gave me unwavering support and helped me, thank you.

Katherine Anne, my wife, deserves special mention, credit, and thanks for her loving forbearance and unflagging support. For their unwavering support, my parents, Miguel Angel and Rosa Ana, have my deepest thanks.

Among the others to whom I owe a debt of gratitude and must be mentioned are Timothy Mayberry, a friend who served as a backboard on whom I bounced ideas when I was stumped; Mrs. Linda Stoddard, AFIT/EN Research Librarian, who successfully tracked down copies of some key sources for me; and Major Walter D. Seward (PhD), my advisor, who was always optimistic about my chances of completing the program and was always willing to spend time helping me.

Additionally, aid from two groups of mentors and friends also helped make this thesis possible. While I was attending AFIT, Dr. Lee (Chairman, AFIT/MA), Dr. Potoczny, and Lt. Col. Bexfield (PhD) were invalu-

able in helping me overcome technical problems and in keeping AFIT-life in perspective. Then, while I was struggling to complete the thesis in Texas during 1983, four people who were instrumental in motivating me were Major Jim Sweeder (PhD), Dr. Al Roecks, Dr. John Romo, and Mr. Wilbur Hoelscher (MS).

To all of those who helped me, a sincere thank you.

Preface

The purpose of this study was to design a multi-level secure local network for the U.S. Air Force's Electronic Security Command at Kelly Air Force Base, Texas. The resulting design was modeled with all traffic encrypted for secure point-to-point communications implementing a packet-switching store-and-forward scheme over a dual loop ring topology using frequency division multiplexed fiber optics. To analytically validate the design, Jackson's Theorem was applied to a simplified version of the model. The results were encouraging. To further evaluate the model, a simulation of the streamlined model was attempted on a microcomputer with 64K RAM. The language used for the simulation was PASCAL. Even though it appears to be feasible to validate a network model on a microcomputer, it was determined that this approach needs further research.

Table of Contents

Page Number

Page Number

List of Figures

List of Tables

List of Terms

access control: 1) network - strategy used to capture the network's transmission medium; 2) security - the process and procedures used to restrict entry into the system only to those who are authorized; these procedures implement the relevant discretionary and non-discretionary security policies

application node: for this thesis, a node, designated by an "A", which will respond to a job request from another node

available: a system that is operational and can provide service; an available system is characterized by long mean-time-between-failures and short time-to-repair, it is usually fault tolerant

backbone: the interconnection of interface message processors (IMPs); refer to topology

broadcast: a communication architecture with the following characteristics: 1) a single communication channel is shared by all IMPs; 2) all messages transmitted over the channel are received by all IMPs; 3) every message contains information to tell the IMPs if the message is for it, if it isn't it is ignored

block: 1) refer to packet; 2) "blocking" occurs when a message arrives from outside the system but cannot enter a node due to lack of buffer space

bulk data traffic: traffic composed of messages of more than 100,000 bits or, traffic which is not bursty

bursty traffic: traffic composed of messages of short duration; for this thesis, bursty messages will not exceed 16334 bits in length (excluding transmission overhead)

communication node: for this thesis, a node, designated by a "C", which can only generate job requests; the "C" nodes are gateways from/to other networks

CRC code: cyclic redundancy code, a polynomial checksum scheme which is used for the detection of transmission errors; for more information refer to Tannenbaum's Computer Networks

data base transfer traffic: for this thesis, messages which have a length of at least 100,000 bits

discretionary/non-discretionary security procedures: 1) discretionary security access procedures implement "need-to-know" protection that are established and may be changed by the organization which has cognizant authority over the resource to be accessed; 2) non-discretionary security access procedures implement mandatory access controls that require all users to be cleared to a security level and compartment equal to or exceeding the classification of the resource being accessed

error: a conditon that arises because of incorrect bits in a message as detected by a cyclic redundancy checksum (CRC)

encryption: a method useful for protection of data that must be transmitted over media that cannot be protected against unauthorized monitoring; two types of encryption: a) link: implies encryption and decryption by each network processor, is used for data flowing over a specific physical path (link); b) end-to-end: the message is enciphered once at the source and deciphered only at the final destination (LAN 83: 87)

fault: a condition that arises when a link is inoperable or a node fails

fault tolerant: a fault in one component does not bring the system to a halt; through redundancy in critical components and/or through the isolation of a fault to limiting the loss of service to a small fraction of the whole, a fault tolerant system displays "graceful degradation"

flexibility: that characteristic which permits growth and extension in functional capabilities, in number of nodes, and in geographic coverage

host: the computer system connected to an IMP or node

IMP: interface message processor; the basic communication component in a node, a communication support computer

interoperability: that characteristic which is the ability to communicate across different networks

intruder: an unauthorized agent or entity

multi-level secure network: for this thesis, a network which supports concurrent/simultaneous transmission of different security levels/categories; a multi-level secure network does not imply that the operating systems of hosts attached to its nodes are multi-level secure, each node's hosts may be operated at dedicated, system high, compartmented, and/or multiple secure levels

multiplexing: the process of achieving simultaneous transmissions of distinct signals over one channel of communication; there are two basic techniques: (1) frequency division and 2) time division (THO 71: 11-14)

node: an IMP and the equipment/machines connected to it; for this thesis, only one host is associated with each node

packet: a data transfer unit which is exchanged between nodes, one or more units make up a complete message; for this thesis, each packet will have a fixed length of 102,400 (100K) bits, this length includes holding up to 100,000 bits of data plus 2,400 bits of header and trailer information

point-to-point: also known as "store-and-forward", this is a communication technique whereby a message or packet is sent from one IMP to its destination IMP; when the source and destination IMPs are not directly adjacent or connected to one another, the transmission is via one or more intermediate IMPs, at each intermediate IMP the message is received in its entirety and temporarily stored there until it can be transmitted "forward" towards its final destination

protocol: the rules and conventions used to control network functions; logical abstractions of the physical process of communication; protocols perform three tasks: a) establish standard data elements, b) establish conventions, c) establish standard communications paths (MCQ 78: 1); refer to Figure II-2 for the seven layer ISO reference model

reliability: a) that characteristic which refers to the freedom from loss of service due to random failures in the equipment or facilities (STO 80: 1472-1473), often referred to as "availability"; b) freedom from random transmission errors

security reference monitor: a set of trusted hardware and software that establishes and enforces network security access controls to include all discretionary and non-discretionary policies and provide complete mediation

SLN: secure local network

survivability: that characteristic which is the ability to survive enemy actions; to Stover, the three aspects of monitorability, self-diagnosis, and maintainability are related to survivability (STO 80: 1241-1242)

switching methods: techniques used to affect how different users share the transmission medium (refer to Table II-3)

TCP/IP: Transmission Control Protocol/Internetwork Protocol; developed on the ARPANET, the protocol set adopted by the USAF as standard for all networks; refer to DOD 82, USAF 82, and USAF 83 sources for more information

topology: the physical layout of a network; there are two levels: 1) backbone - the interconnection of IMPs; 2) local access - the interconnection of hosts, terminals, and peripherals to a specific IMP

trusted: a component comprised of hardware and/or software that can be relied on to enforce the relevant security policy; a " 'trusted computing base' is ... the totality of protecting mechanisms within a ... system ... the combination of which are responsible for enforcing a security policy." (LAN 83: 88); a trusted component is correct (i.e., it operates according to its specifications) and incorruptible (i.e., it cannot be modified by an intruder) (NES 83: 1059)

Abstract

This research sponsored by the USAF's HQ ESC/AD develops a multilevel secure host-to-host computer local area network. The design process is presented. The resulting network uses a ring topology with packetized point-to-point switching over fiber optics communication links. For transmission security, packets are source host-to-destination host encrypted as well as encapsulated with link-to-link encryption. Message transmission is controlled with message acknowledgements and credits within a non-preemptive three priority class queue. A simplified version of the resulting network was validated by applying Jackson's Theorem. Additionally, the simplified view was modeled with a PASCAL simulation program executed on a 64K microcomputer. Unfortunately, the comparison of the simulation against the analytical results that were obtained using Jackson's Theorem was not possible due to problems modeling the network on the microcomputer. Follow-on work in the area of simulation is needed to successfully complete the simulation and compare results.

Chapter I: Introduction

Overview.

General Requirements. This thesis was sponsored by the U.S. Air Force's Electronic Security Command at Kelly A.F.B., Texas (HQ ESC/AD Bldg 2000 San Antonio, TX 78243). It develops a multi-level secure host-to-host local computer network model. Mr. Hoelscher (Chief, Executive System Software Branch and Technical Advisor, Directorate of Systems Technology) served as the point of contact at HQ ESC/AD. He provided the constraints and requirements which influenced the network's design (HOE 82; HOE 83).

There were two major ESC requirements that had to be met for a successful design. The first one was that the network had to efficiently process traffic that would be primarily bulk in nature.

The second major requirement was the most important and restrictive; the network had to be secure and provide concurrent multi-level security. The security aspects were pervasive because the network was required to receive, transmit, and process classified and compartmentalized information that, if compromised,

could damage national security.

Additionally, the resulting model had to be verified. A simplified version of the model was analytically evaluated by applying Jackson's Theorem. Additionally, a limited simulation written in PASCAL was attempted on the streamlined model. The simulation was executed on a 64K microcomputer. Unfortunately, this part of the verification was not completed to form a part of the model's analysis.

These issues were refined during the development of the thesis. But the dominant requirement throughout the design process was security.

Multi-level security requirements and the protocols and architecture required to support them are areas that have received increased interest as illustrated by the bibliography of this thesis. The many favorable characteristics of computer networks have been well documented by authors such as Booth, the Dennings, Donaldson, Kent, Kline, Kuo, Popek, Stelte, Tanenbaum, Tropper, and Weitzman. However, primarily due to a fear of compromise, the military has not taken full advantage of computer networks (STI 80: 1472).

Recently, with the advent of applications such as electronic fund transfers, security problems have been receiving greater scrutiny by the business and academic communities (KEN 76: 8; KON 81: 761; KUO 81: xi; TAN 81b: 480). Many experts feel that even with safeguards such as access controls, flow controls, data encryption, and inference controls, "absolute" security is impossible (DEN 79: 227-228, 246; POP 79: 355). But what degree of security is attainable?

Organization. Prior to performing any analysis which would lead to a model for a secure network, an approach was required. A series of principles were reviewed and those deemed appropriate were adopted. These principles formed the foundation of the methodology that was adopted to develop the network. This methodology is covered in Chapter I.

The next chapter is a discussion of some of the major constraints and requirements that apply to the model, those of security. The final section of the second chapter presents several safeguards and assumptions on the model's security and its environment.

The third chapter discusses how and why this

particular model was developed. It describes in detail the design process. The decisions made concerning topology, network control, and protocols are presented here with the ever present influence of security. Whenever possible, while examining the model's various features, comparisons are made among the advantages and disadvantages of other network designs.

In the fourth chapter, the analysis and verification are discussed. The simplifying assumptions and the results of applying Jackson's Theorem are analyzed.

The final chapter presents conclusions, recommendations, and further areas of study generated by this thesis.

Methodology

Background. The methodology adopted for this study rests on two distinct but related sets of principles. The overriding set of principles are security related. However, the network could not be developed strictly with security in view if it was to perform any useful applications with any reasonable degree of efficiency. Therefore, the overall approach was to develop a network

with the additional principles of simplicity, and reliability. The goal was a network which was as simple as possible (to ease implementation, review, maintenance, and future growth) and as available (fault tolerant, with long mean-time-between-failures, and with short time-to-repair) as possible while not over complicating the design aspects which would make it impossible to provide adequate security.

The principles followed to analyze, develop, and maintain security were adapted from Dr. Stephen B. Kent's "Protocols and Techniques for Data Communication Networks". Kent delineates eight specific principles of design.

Kent's Principles. Kent's first principle is probably the most important. The design should be simple. A simple design simplifies the tasks of implementation, verification, and maintenance.

The next two principles, that of fail-safe defaults and of complete mediation, are constraints that help attain a secure system. These principles are directed not at exclusion (or "why not" permit access) but at "why" should access by allowed. This

positive approach constrains the set of who may access the system and its resources in a manner which permits greater restriction and hence less chance of an intruder penetrating through oversight. Thus, access will only be permitted if specifically, instead of tacitly, granted. The default will be to deny access. In this manner, the person seeking access must go through some human (security officer) control prior to the system getting his "name" in the system's access roster. Therefore, all users are required to comply with non-discretionary (mandatory) security rules which serve as an overall barrier to the intruder. But discretionary control should also be provided. This control can be specified at the option of the user who can further constrain what he does for a particular application, session, and/or transaction (AME 83a: 15). With users conscientiously applying discretionary security rules, unnecessary security risks are avoided.

The fourth principle is not widely accepted by the military. It is the principle of open design. The argument against an open design is that "a secret design may have the additional advantage of

significantly raising the price of penetration, especially the risk of detection". But Kent argues that an open design is easier to review since there is no need to hide safeguards which should remain secret in a closed design (KEN 81b: 372). However, in light of the sensitivity of national security requirements, a closed design should be followed.

Separation of privilege and of least privilege are the fifth and sixth principles. These principles help limit damage from penetration. They enforce least access, ensure "need-to-know", and add the safeguard of multiple keys for access to any given level. Any security violation should have a limited scope of potential compromise/damage. Not only should there be separate access rosters for different security classifications, but each security classification should be compartmentalized to deny complete access to that level in case of penetration. This compartmentalization is created through separate rosters, passwords, and even hardware safeguards which will act as bulwarks and will not allow complete access to a level when one section

has been penetrated. This need to limit damage is further emphasized in the seventh principle.

The seventh principle is that of least common mechanism. By keeping to the very minimum mechanisms which are in common throughout the system, penetration can be more readily localized and subversion of the entire system is less likely to occur. This entails the use of separate rosters and different passwords for each system resource, as well as the use of other physical, software, hardware, and human safeguards to secure components of the system from a potential intrusion (the use of discretionary controls helps accomplish this endeavor). Thus rosters cannot be accessed by the same password and different passwords and security profiles are required for different resources located in separate physical locations (like vaults) to which access is restricted to different sets of users.

Because of these principles, different authorizations or permissions are required to access different components and compartments. By requiring an audit trail that tracks location of user, password(s),

location of resource(s) required, and time of system/resource call and release, a system can be implemented with multiple crosschecks which will reveal where a penetration has occurred, what has been subject to compromise, and the extent of the compromise. Knowing what has been compromised is a major goal in a security conscious environment.

Finally, the last principle is that of psychological acceptability. User friendliness is a concept often overlooked. But a safeguard which can not be easily and routinely used is often ignored. What is the use of passwords if the user has them written on a piece of paper in his wallet because they are so many and so long? This results in the elimination of a barrier for a potential intruder. Whenever and wherever possible, the safeguards and countermeasures should be automatic and should use only trusted system components.

The Approach. The approach taken to apply this methodology was to first read about networks and then analyze network designs in light of Kent's principles. The works of Clark, Kuo, McQuillan,

Tanenbaum, Thurber and Tropper were the most applicable during the initial stages of this study. Acceptable designs were earmarked for further comparison during which additional constraints caused by the environment were applied. Once the choices were narrowed to a few general options, a comparison of their respective advantages and disadvantages was made using tables derived from the previously mentioned sources (as well as from the works of Agrawala, Bux, Habara, Homayoun, Ikeda, Penney, Popek, Kent, Stillman, Stover, and Wolf) which summarized these characteristics. From these tables a choice of topology, network access controls, and protocols was made bearing in mind the need for simplicity and reliability.

The chosen options (discussed in Chapter III) were then combined into a design which could meet the desired characteristics for the secure network. It was then necessary to validate this design. To do so, Jackson's Theorem was applied to a simplified version of the model as a check. Then an attempt was made to perform a PASCAL simulation on a 64K RAM microcomputer of the streamlined model. This was done to achieve greater

confidence in the results and, also, to investigate how a network validation could be performed on a microcomputer. This, unfortunately, was not completed as part of this thesis. The choice of machine and the choice of language caused problems which were not resolved by the completion of this research. Thus, verification of the model was by way of Jackson's Theorem and only for a simplified version of it.

Before an analysis was feasible, a design was required. But what must the network to be designed safeguard against? An overview of security requirements is presented in the next chapter.

Chapter II: Security

Security Requirements: An Overview.

The Environment. The environment in which a network must operate constrains the topological options available for implementation. Additional restrictions occur when the network must be a secure local network (SLN).

According to Coviello and Lebow, "the essential distinctions" between military and non-military applications "can be summed up with the single catch-phrase 'survivability'" (COV 80: 1441). The military environment can range from peacetime to nuclear warfare. But many systems need not safeguard against all the conditions of the entire range of possibilities nor may they be able to do so. For example, this thesis's particular SLN is not expected to withstand overt physical attack. But survivability is possible only for a specific set of threats (COV 80: 1441), so what are the set of threats to be met by this thesis's SLN?

Safeguards, Threats, and SLN Characteristics. The spectrum of safeguards and related threats which any SLN

should be able to survive are covered, among others, by Kent, Popek, and Stillman. The cited work of these authors does not cover the threat of war. Since the SLN developed for this thesis is not expected to survive in wartime, the safeguards and threats presented by them apply to the model. Unfortunately, not one of them gives a definite way of implementing any of these safeguards.

In pages 778-779 of his article "Security Requirements and Protocols for a Broadcast Scenario", Kent lists five major security requirements to counter potential threats. The first requirement is the need to prevent unauthorized release of message text. Then there is the need to prevent (or disrupt) traffic analysis by potential intruders. Wiretapping is one way that intruders can attempt to get the information they should be denied. Therefore, the need to safeguard against both active and passive wiretapping is critical. (Passive wiretapping is merely the listening of traffic without attempting to modify the transmission stream. Active wiretapping includes the insertion and/or deletion of traffic to modify the transmission stream.)

Kent also presents the need to verify message authenticity, integrity, and ordering as the fourth requirement. It is closely related to the need to prevent message stream modification, message deletion, and spurious or intentional message insertion (the fifth requirement).

Popek and Kline present many of the same requirements (POP 79: 332-334). They also mention the need to safeguard against the tapping of lines and the introduction of spurious messages. Additionally, they mention that safeguards are needed to prevent retransmission of a previously transmitted and acknowledged valid message and to detect and/or prevent disruption (or blockage) by malicious (intruder/interloper) acts or system failure(s).

The military's view of the threats is presented by Stillman and Defiore (STI 80: 1472-1473) who are technical advisors to the Air Force (USAF/SI). They reiterate the need to prevent unauthorized access to classified information, the need to assure information integrity, and the need to counter

wiretapping and analysis of traffic flow. Also they expand upon the need to guard against unauthorized access to physical facilities and communication links and against subversion by unauthorized users and authorized users not in their authorized "area". Furthermore, they present the need to protect the availability of resources for authorized use in three operational environments: routine, high traffic stress, and degraded operations which includes protection of authorized users from each other.

Stover presents safeguards and threats in a different way by defining six characteristics that any military SLN should have (STO 80: 1241-1242). These characteristics are desireable and pertinent to this SLN, too. They were used in helping reject options in Chapter III.

The first characteristic is that of survivability which Stover defines as the ability of the digital communications function to survive enemy actions. Stover presents the three related aspects of survivability: monitorability, self-diagnosis, and maintainability. To

Stover, monitorability, self-diagnosis, and maintainability mean that the network must be tolerant of failures; that failures must be detected, isolated, temporarily accommodated by operational procedures (which should be automatic whenever possible); and that failures must be repairable.

The second characteristic, reliability, refers to the freedom from loss of service due to random failures in the equipment or facilities, i.e. network operation ideally should not depend on the continued operation of any particular node or transmission link. A reliable system is dependable.

The next two characteristics, accuracy and stability, are related. Accuracy and stability refer to timing (message synchronization) and timing contributes to error detection and identification as well as to reliability. The key concept here is that the sending and receiving nodes agree when to send and expect messages and how these messages are being relayed. For example, if a message is expected and none is received in some given amount of time (a tolerance factor), then it

is safe to assume that some error has occurred. At this time, some error handling protocol gains control of the processing. As the percentage of errors that occur and are not detected decreases, the system reliability increases.

Flexibility is that characteristic which permits growth and extension in functional capabilities, in number of nodes, and/or geography. By their nature, networks tend to have the flexibility of incremental growth (BOO 81: 6-31; KUO 81: ix-xi; TAN 81a: 3-5).

The last characteristic is that of interoperability. Interfaces with other digital communication systems should be facilitated by having a timing which assures that the buffers will not have to be reset more frequently than at some acceptable rate.

Another aspect of interoperability is the ability to communicate across different networks. Connectively between networks is usually made over nodes that are called gateways. (Gateways convert from one protocol to another (TAN 81a: 354). Value-added

gateways are gateways that also do some additional processing (like filtering traffic by security level, encryption/decryption processing, or guard functions); ESC's gateways are all value-added gateways.) An additional means of achieving internetworking is to force a common protocol set among all networks for purposes of homogeneity.

In any case, not all of these safeguards, threats, and characteristics are applicable to this model. The next section shows the relationships of the above concepts to the SLN model developed. It addresses the assumptions made and the physical constraints which define the network's many requirements.

Model's Security Assumptions and Safeguards.

Physical Security. Without physical security, no other security safeguard is effective (WOO 81: 70). The SLN designed in this thesis will have guaranteed physical security. It will be located in a secure building which has active and passive safeguards. All the resources/hardware will be in rooms that will be further secured within the building. Furthermore, all

equipment, as well as the transmission lines, will be sheathed to shield against electromagnetic emanations which would permit eavesdropping. Access controls at each node will insure against the possibility of someone at one node illegally accessing resources at another node.

A More Secure Transmission Medium. There are two major choices for transmission medium for this network, coaxial cable and fiber optics. A comparison of the security characteristics of these two media follows.

If the transmission *medium* chosen were fiber optics instead of coaxial cable, tapping would be more difficult (WOO 81: 70). Also, because the media will be physically secure, another critical security advantage of fiber optics over coaxial cable is found in the realm of electromagnetic radiation. Unlike coaxial cable, electromagnetic impairments are nonexistent in transmissions over fiber optics medium (CLA 81: 23; HOM 80: 980-981; KEN 83). Finally, encryption techniques can be applied with fiber optics just as well as with coaxial cable (WOO 81: 73).

Because of the above mentioned characteristics, fiber optics is a more secure transmission medium and worth any additional expense. Table III-4 (on page 50) summarizes the characteristics of both media.

Encryption: Advantages and Disadvantages. Simmons (SIM 79: 314) and Popek (POP 79: 332-333, 335-336, 338) consider encryption to be the only way to send information over unsecure media and the best way to improve security and message integrity. Wood states that "cryptography is the only cost-effective control" against many threats and is essential for the maintenance of message integrity (DAV 81: 155, WOO 81: 71).

Simmons also argues that encryption helps provide secrecy and integrity. But Simmons warns that it is not perfect and is best used in authentication (SIM 79: 314, 322). Popek and Kline also recommend the use of encryption for authentication (POP 79: 336); but they categorically state that it does not provide protection against inadvertent or intentional modification of data (POP 79: 338). (The use of checksum techniques can provide a modicum of protection in this

area (RUS 83).)

Therefore, encryption is but one control, not a panacea, and is useless without physical protection (WOO 81: 70). But it helps achieve secrecy/confidentiality (i.e. protects data and the source and/or sink from disclosure), it preserves data integrity, and it allows for the introduction of enciphered signals to conceal message length and frequency statistics which are critical for traffic analysis (LAN 83: 87, WOO 81: 71). Wood emphasizes end-to-end rather than less secure and more expensive link-to-link encryption. But the use of both methods simultaneously does add an additional degree of security. Wood also believes that encryption is vital because it can provide message, user, and process authentication and validation assuring integrity of transactions (WOO 81: 74).

Kent states that encryption (and all other security requirements and tasks) can cause unacceptable overhead that adversely impacts upon network performance (KEN 81a: 785; also supported by RUS 83: 55-57); but it is the most effective countermeasure (KEN 83; LAN 83: 87; SEA 83: 54-58). Furthermore, these

adverse effects can in part be offset by high speed communication links (KEN 81a: 785).

Encryption will be the primary means to maintain security within the network. It is a good way to protect against alteration of message contents and message insertion; and it preserves data and transaction integrity (LAN 83; NES 83; POP 79; SIM 79; WOO 81).

Model's Encryption. Stillman's advice on encryption is "rather than attempting to separate multi-level users by monitoring and controlling data accesses, end-to-end encryption attempts to disguise the data at the source, maintain them in unintelligible form all along the communications path, and decrypt them only at the destination" (STI 80: 1473-1474). This advice is followed in the model. All transmissions over the network are encrypted twice. But, agreeing with Stillman (and Rushby and Randell) that security often rests on the secrecy of the key rather than the algorithm, this thesis will not have algorithm selection nor key distribution techniques within its scope.

In this model, there are two levels of encryption which combine link and end-to-end (in this case source host computer-to-final destination host computer) techniques. The inner level is undecipherable to all nodes except the one to which the message was addressed (i.e. a separate key for each pair of source and destination nodes conforming to end-to-end encryption). Furthermore, a distinct and different key is used to encrypt each message. The outer level of encryption is link-to-link and uses another key (which is unique for each channel and is changed periodically) known to all physically connected pairs of nodes which will contain, along with other information, the message destination. The safeguards and protocols associated with proper message handling are discussed in Chapter III.

Miscellaneous Issues. All issues pertaining to key management (i.e. generation, distribution, and control), which were assumed trusted, were beyond the scope of this thesis. Remote key generation and distribution was assumed available through trusted components. Also beyond the scope were the interfaces between the SLN

and any other network. Therefore, security of the communication links into the net from areas outside of the building was assumed adequate. Access was in accordance to the principles delineated by Kent and reiterated by Ames. All three factors presented by Downey for access control (which he defines as clearance/classification, compartmentalization, and need-to-know) were considered (SCH 73: IV-25-26). But all these safeguards were not within the scope of this thesis.

Summary.

The security of the network will be established on four key points. First and foremost, because without it no security is possible, physical security will be assumed. Then, all equipment used will be sheathed as required to protect against electromagnetic emanations. Next, all transmissions will be source host computer-to-final destination computer encrypted with message unique keys as well as encapsulated within link-to-link encryption which uses different keys for each channel which are periodically changed. Finally, Kent's and Downey's security access principles will be assumed

implemented on trusted systems.

The next chapter presents a detailed discussion of the model and how it was designed bearing in mind the security constraints elaborated on in this chapter.

Chapter III: The Model

Overview.

This is a model of a local host-to-host computer network which will be used to support distributed processing and will concurrently support two different levels of security classifications. Security requirements will be considered at each step. Additional requirements which the design should meet are that the resulting model portray a network: 1) that is maintainable, 2) that is fault tolerant, 3) whose arrival and service rates can be varied, and 4) whose traffic, the composition of which can also be varied, can be limited to database transfers (which will be at least 50 percent of the traffic) and "bursty" interactive work primarily associated with distributed processing. "Bursty" traffic is defined as messages of less than 16334 bits. (It was determined that up to 50 -- but not more than 80 -- percent of the bursty traffic would consist of a single screenful of data, this was calculated to be less than 16K bits (HOE 83). The database transfers are messages averaging 100,000 bits. Database transfers will range between 100,000 and

900,000 bits. As specified by ESC/AD, the network will consist of seven nodes; three of the nodes will be communication nodes providing connectivity to different external long haul networks and four of the nodes will be application nodes.

This chapter discusses how and why this particular model was developed. It addresses itself to decisions concerning the topology, the network control, and the protocols. At each step, all pertinent information, especially relevant security considerations, and the options available are presented along with the decisions made. It concludes with a summary of the model.

Topology.

When developing a local network, one of the first decisions involves the choice of backbone topology. (This thesis does not include a discussion of the local access topological design since the research was directed to a host-to-host network. The connection of the hosts, terminals, and peripherals to interface message processors (IMPs) is not within the

scope of this thesis. It is assumed that the nodal hosts are connected to a peripheral local area network or that the peripherals are directly connected to their nodal host.) This decision is affected by such issues as topological simplicity, ease of implementation, message transmission control, fault tolerance and reliability characteristics, and the work the network is expected to perform. In this particular case, the issue of security considerations could be and were relegated to the protocols, but they permeated the selection process of topology, too.

There are three basic topologies applicable to the backbone of a local network to choose from: the star, the ring, and the web (CLA 81: 19-20). These topologies are shown in Figure III-1. It should be noted that the same topologies are often known under different names. These aliases are presented in Table III-1 (page 32) after a discussion of each of the three basic categories.

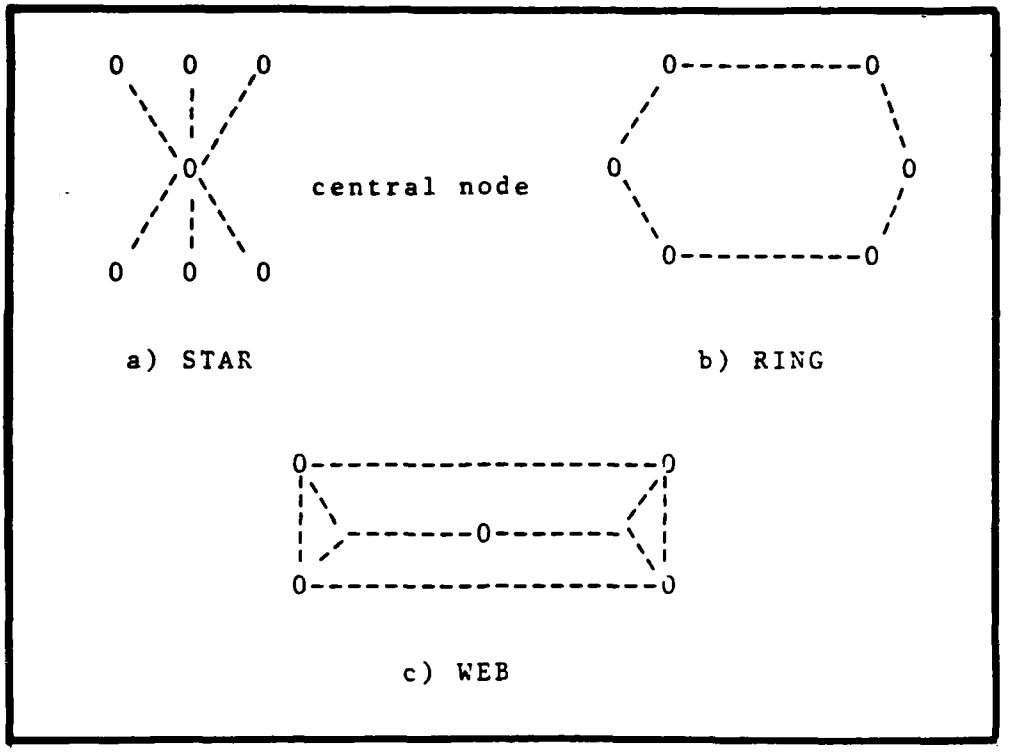


Figure III-1. Topologies: a) Star b) Ring c) Web

Star Network. The star network is a simple structure. Unlike an uncontrolled topology, the star eliminates the need for each node receiving a message to make a routing decision to forward the information by centralizing all message decisions in one node (BAS 81: 366; CLA 81: 19-20; HAB 80: 964-963; PEN 79: 166; STA 80: 83).

While this centralization seems at first to be

an excellent way to maintain security over all traffic; it provides potential availability problems if, for example, the central node fails (CLA 81: 21). A standby redundant control node configuration could overcome this problem. But in any case, the central node could become a bottleneck for traffic (HAB 80: 965) and it presents to the intruder a tempting target at which to disrupt the entire system.

Ring and web topologies attempt to overcome the star network's vulnerability by eliminating the central node without completely sacrificing simplicity (CLA 81: 19-20; TRO 81: 7-11).

Ring Network. In ring topology, we find messages going from node to node along undirectional links until it arrives to its destination. Since each node only has to recognize if the message has arrived at its final destination or else transmit it to the next node in the line, routing decisions are kept to a minimum (WIL 80: 507).

But single loop rings suffer from poor fault tolerance (TRO 81: 53; WOL 81: 149). Fortunately, this problem can be overcome with multiple loops

(PEN 79: 171-172, 228; TRO 81: 53, 73-74; WOL 81: 150).

Web Network. The web is characterized by having all processing elements attached to a common channel which is employed in a broadcast mode (CLA 81: 19-20; PEN 79: 166; TRO 81: 73-74). It is superior in fault tolerance (BAS 81: 366); but suffers from control problems in the areas of synchronization, flow, and error control (HAB 80: 965). Furthermore, for reasons of security, it is not acceptable. Let us next examine the security applicable issues.

In a secure network, a clear audit trail for each transmission is required so that message arrivals can be verified. Each message should only have on destination. With only one destination, security control over the traffic is simplified and it is easier to identify which messages are lost or inserted without authorization (whether or not the cause is from malicious acts or by spurious system errors). Therefore, broadcast modes are not desirable. Because of this and related security complications which arise from broad-

cast modes of operation, the web network is unacceptable.

Table III-1, derived from the works of Bass, Clark, Habara, Penney, Stack, Tropper, and Wolf (BAS 81: 366; CLA 81: 19-22; HAB 80: 964-965; PEN 79: 165-166; STA 80: 83; TRO 81: 7-72, 73-74; WOL 81: 148-150), summarizes the attributes of the topologies discussed.

Table III-1
Comparison of
Controlled Network Topologies with Aliases
Part I

Network Name and Aliases	Advantages	Disadvantages
Star	1) Simplicity of design 2) Localization of damage in case of fault 3) Ease of incremental growth 4) Simplicity of routing 5) Potential centralization of all security tasks	1) Traffic inefficiencies due to central node 2) Central node failure shuts down network 3) From a security perspective central node vulnerability

Table III-1
Comparison of
Controlled Network Topologies with Aliases.
Part II

Network Name and Aliases	Advantages	Disadvantages
Ring Loop	1) Traffic efficiency due to high-way capacity 2) Short average circuit length for intra-ring calls 3) Good fault tolerance with multiple loops 4) Good message audit trail 5) Relatively few routing decisions	1) Design moderately difficult 2) Incremental growth more difficult than for Star
Bus Web Mesh	1) High degree of fault tolerance 2) High degree of flexibility	1) Design very difficult 2) Route processing difficult and further complicated with security controls

Topology Decision. This analysis led to the decision to opt for some form of a ring topology. The advantages of ring networks speak for themselves. Ring networks are relatively simple to implement, relatively easy to modify (i.e. easy to add/delete processing elements/nodes), have relatively low start-up, modification, and maintenance costs (TRO 81: Pp. 8-9, 73), have a high degree of bandwidth efficiency, and, with the advent of multiple-loop ring networks, the fault tolerance problems can be overcome while minimizing security problems (FAR 81: 135; PEN 79: 172, 228; TRO 81: 53-55; WOL 81: 148-150, 158, 162).

After deciding which topology to use, the next issue to be resolved is what network access control scheme to apply. Controlling transmission over a network is an important design issue (CLA 81: 19-20). When can a user gain access to and control over the transmission medium to enter data onto the backbone?

Network Access Control.

There are many different network access control schemes that are applicable to a ring topology. This section presents four of these strategies and

discusses which was chosen to gain access onto the network's transmission medium. The first strategy to be examined is known as contention or random access. This strategy is most often encountered in bus topologies; but it has also been suggested for ring topologies (CLA 81: 21; PEN 79: 166). The next three are considered the "basic" ring access strategies (BUX 81: 1465; CLA 81: 20; TRO 81:8).

Contention. Thre are many contention strategies (TRO 81: 77). In a contention scheme, any node wishing to transmit does so. If two (or more) nodes transmit simultaneously, a collision occurs which will theoretically result in garbled or lost transmissions. Therefore, one contention control strategy (carrier sense multiple access -- CSMA) depends on the node that transmits detecting these collisions and, when it does, waiting a random amount of time before attempting retransmission. Unfortunately, as the number of nodes increases, performance deteriorates.

Also, contention schemes are better suited for "bursty" traffic. This is because contention schemes

lead to a very low limit on the percentage of channel capacity which can be utilized without causing the network to overload (saturate) with retransmission traffic (BUX 81: 1470; CLA 81: 20-21; LIS 83: 30; STU 83: 72-76; TAN 81b: 469; TRO 81: 76, 131-133). This disadvantage of the contention scheme relates to the complexity of the transmit/listen/retransmit if collision detected control technique. Over a ring, the propagation delay is a limiting factor (SALW 83: 184, 190). How long should a node listen for a collision? The unidirectional flow of messages from node to node provides a natural ordering of all nodes that should permit a much lower collision rate (CLA 81: 21). Also, a contention scheme could be implemented between each pair of nodes to limit the propagation to one hop; but then a message that is not destined to an adjacent node has to be retransmitted from every intermediate node that it must cross. The difficulty of implementing any contention scheme is not necessarily warranted if a more feasible network access control scheme exists.

For this model, contention schemes display three major disadvantages. The first critical

disadvantage of contention schemes is that they are meant to handle primarily "bursty" traffic and not the data base transfer transmissions which dominate this network. The next disadvantage is the complexity of a contention scheme -- when a goal is to keep things simple (Chapter I: Methodology, page 4), complexity is a disadvantage. The third undesirable characteristic is that security will be complicated by contention strategies because of "lost" transmissions. Because of these three disadvantages, contention schemes are not deemed appropriate for this model.

Slots. The Pierce loop illustrates the slotted ring access strategy (AGR 78: 674-675; BUX 81: 1466-1467; PEN 79: 167-168; TRO 81: 8-9, 21-22; WOL 81: 149). In this strategy, a (one or more) fixed length time slot, generated and synchronized by a designated supervisory node, continuously circulates around the ring. To inform a node whether or not a slot is in use ("full") or not in use ("empty"), a header is attached to each slot. When a node wishes to transmit a message, it must wait until an empty slot which it can fill reaches it. At that time, the node alters the header to

reflect that it is full and then uses the slot to transmit its message. The filled slot eventually makes its way back to the node that filled it where it is recognized, captured, and, if there is nothing to transmit, marked empty. If there is more traffic to transmit, the slot is reused immediately. It is because of the ability to immediately reuse a slot that a node with a heavy flow of traffic can "hog" a time slot (TRO 81: 70).

The major advantage of this control scheme is that, with more than one slot, simultaneous transmission of messages can occur (TRO 81: 8-9). This strategy was deemed appropriate for this model despite the adverse performance characteristics of "loop hogging".

Tokens. The token ring access strategy is illustrated by the Newhall loop (AGR 78: 675; BUX 81: 1465-1466; PEN 79: 167-169, 176; TRO 81: 9, 11; WOL 81: 148-149). Permission to transmit is passed from node-to-node by a circulating token. When a node receives the token, it may transmit one message. If there is no message to transmit, or after transmitting one, the token is passed to the

next node in the loop. The major advantage of this control scheme is that it allows the transmission of variable length messages (TRO 81: 8-9). Kummerle and Reiser categorically state that token passing is superior over a wider range of parameters than contention schemes (KUM 82) which provides greater potential long-term utilization. This strategy was also deemed appropriate for this model.

Shift Register Insertion Technique. The shift register insertion technique has been applied in the distributed loop computer network (DLCN) and also by the double distributed loop computer network (DDLCN). According to Tropper, the shift register insertion technique has the major advantage of the slot (simultaneous transmission) as well as the variable message length handling ability of token rings (TRO 81: 9). Penney mentions an additional advantage which reflects additional reliability, the shift register insertion technique has completely distributed control of the transmission system (PEN 79: 170). But it does have the disadvantage of additional delays as the message traverses nodes to

its destination (TRO 81: 9). This strategy was also deemed appropriate for this model.

Control Decision. To decide among the three strategies deemed appropriate, an analysis that compared them was required. Fortunately, there are several sources each of which compares simulation results of at least two of the strategies under similar conditions. After reviewing these studies, the shift register insertion technique was selected as the most appropriate because it displayed superior performance (PEN 79: 234-236; TRO 68-72). Table III-2 summarizes the information drawn from the various sources referenced in this section from the standpoint of this model's requirements.

The next step was to analyze the protocols required to meet the model's requirements.

Table III-2
Comparison of Network Control Schemes
Applicable to this Model
Part I

Control Scheme	Example of the Scheme	Advantages	Disadvantages
Contention	CSMA	1) Best for bursty traffic 2) Flexible design	1) Can have low channel capacity utilization 2) Security is complicated 3) Complex implementation
Slot	Pierce Loop	1) Best for packet switching 2) Can transmit messages simulta-neously	1) Can display "loop hogging" (TRO 81: 70)
Token	Newhall Loop	1) Can transmit variable length messages 2) Superior performance to slot 3) No loop hogging	1) Performance inferior to shift register insertion

Table III-2
Comparison of Network Control Schemes
Applicable to this Model
Part II

Control Scheme	Example of the Scheme	Advantages	Disadvantages
Shift Register Insertion	DLCN DDLCN	1) Can transmit variable 2) Can transmit messages simulta-neously 3) Control completely distributed 4) Best overall performance	1) Additional delays upon message 2) Requires additional storage

Protocols.

Introduction to Protocols. Protocols are the rules and conventions used to control network functions. McQuillan and Cerf state that protocols are logical abstractions of the physical process of communication and they perform three vital tasks: 1) establish standard data elements, 2) establish conventions, and 3) establish standard communication

paths (MCQ 78: 1).

Protocol design is the most critical aspect of the model's development. It is here that the procedures required to meet various design features are set. If the procedures are incorrect, the network will not meet its requirements.

A concensus on protocols has been developed; it is found in the International Standardization Organization's Reference Model for Open Systems Interconnection (ISO OSI). The ISO OSI is presented in an introductory fashion in Tanenbaum's "Network Protocols" and in more detail in his book Computer Networks pages 10-21. From the ISO OSI, protocols have been divided into seven layers. These layers and their interrelationship is illustrated by Figure III-2. (For further information, refer to the bibliography under McQuillan and Tanenbaum.)

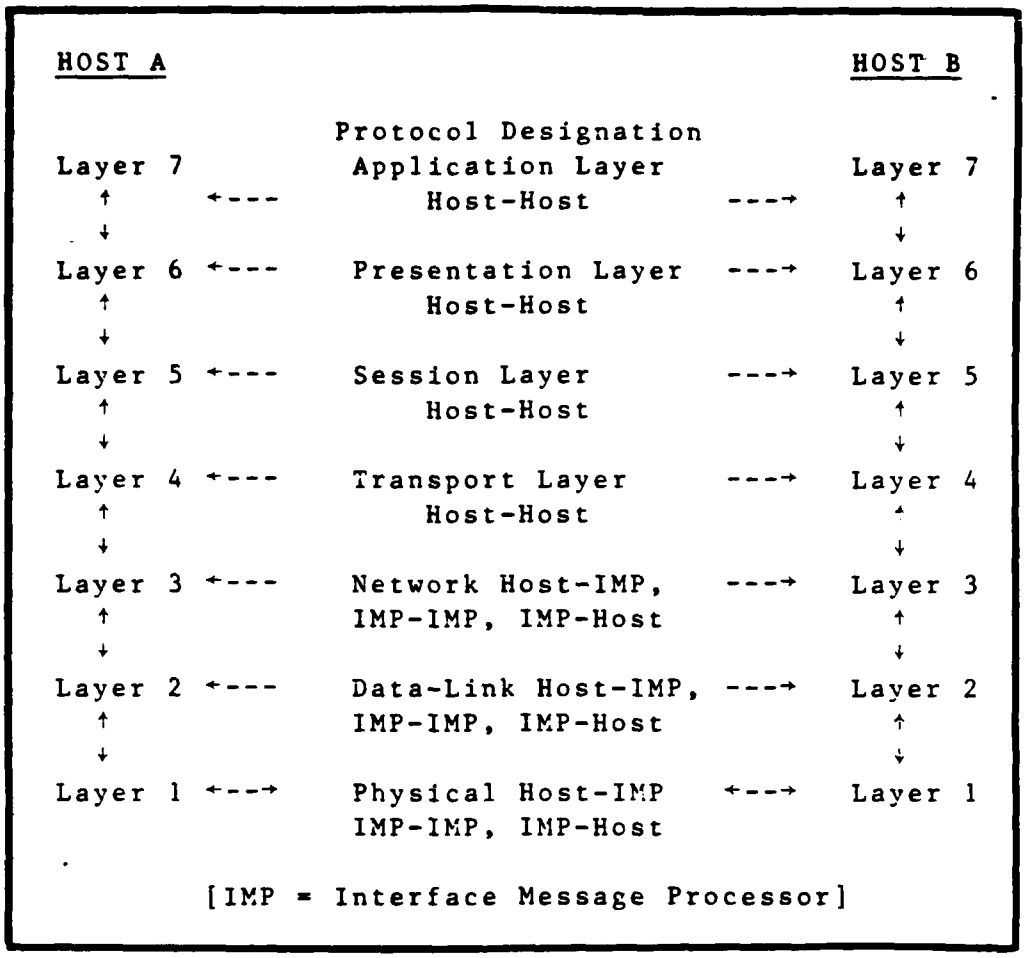


Figure III-2. The Seven-Layer ISO Reference Model. Adapted from Tanenbaum's Computer Networks (TAN 81a: 11, 16).

The protocols and protocol related decisions that this thesis addresses are those that fall within the realm of switching method, flow control, error/fault detection/correction, internetworking, and access/security controls.

The transmission medium is discussed first. Then the switching method. This is followed by the flow control protocol along with the priority scheme which it supports and the manner in which the transmission frequencies are divided to make the priority scheme work while maintaining two security levels. A discussion of the error handling protocols then follows. Finally, a discussion of the security protocols is presented.

The issue transmission medium to be selected is presented here because it impacts upon the switching method for message control and that in turn will affect the transport protocol. (The protocols for the physical, link control, and network and application levels are not within the scope of this thesis. It is assumed that the various standards which have been developed for the lower three levels are followed. The only point concerning this model is that of link level encryption. It is assumed that appropriate equipment is available to perform this task automatically and that this task is handled adequately.)

Switching methods are those techniques that affect how the various users share the transmission medium.

The choices considered are circuit, message, and packet switching (MCQ 78: 12). Each of these methods exhibits different properties which affect transmission efficiencies. Circuit switching establishes an end-to-end dedicated path before any data can be transmitted. Message switching does not establish this circuit in advance; instead the network makes its transmission decision at each node for the next hop. Packet switching, which is best suited for interactive traffic (TAN 81A: 116), acquires and releases the node-to-node link as required. Table III-3 presents a comparison of these three methods.

Table III-3
Comparison of Switching Techniques

Characteristics	Switching Method		
	Circuit	Message	Packet
Dedicated Connection	Yes	No	No
Delays w/ Congestion	No	Yes	Yes
Storage Required	No	Yes	Temporary
Transmission Line Monopolized	Yes	Yes	No
Speed/Code Conversion	No	Yes	Yes
Error Control	No	Yes	Some
Real Time/Interactive Bursty Traffic	No	Maybe	Yes

Flow controls ensure proper functioning of the communication channels with respect to message transmission and reception. The main goal of flow control is to avoid overloading a node (CLA 81: 29; MCQ 78: 24; TAN 81b: 477). Also included in this area is the traffic monitor which enforces flow controls and which 1) supervises queues and the algorithms that permit the entry/exit of messages, 2) inserts dummy traffic that disrupts traffic analysis by an intruder, 3) checks for lost or unauthorized messages, and 4) monitors the loop for transmission link breaks/faults.

An error/fault detection/correction protocol is necessary due to the sensitive nature of the information to be transmitted by the SLN and by the time sensitivy of the same. Detection and retransmission was the obvious solution for two reasons. First, there is no need to implement a costly error correction process when the transmission medium, fiber optics, supports very low error rates making the probability of retransmissions due to bit errors very slight. Second, security is an overriding concern which

is best served by requesting retransmissions as required instead of attempting corrections.

The use of cyclic redundancy code (CRC) checksums was the best means of detection over simpler parity checking mechanisms that would be inappropriate for traffic that must always be correctly interpreted. Furthermore CRC is capable of detecting a greater number of errored bits (MCQ 78: 23). The parity checking is to be implemented at the data link layer. Other parts of the error function are required to handle link breaks (which is handled in the network layer) and message deletions and insertions (which are handled in the transport level).

Internetworking is a major concern in this SLN since three of its nodes (designated as communications or "C" nodes) serve as gateways to external long haul communications networks. As gateways, these "C" nodes perform three functions:

1) network access protocol translation/conversion
2) packet size matching
3) speed matching and synchronization

The most complicated function, that of protocol translation, was simplified when the Department of

Defense (DoD) decided to approach the internetworking issue by declaring a set of internetworking protocols standards for the DoD community's host-to-host data communications networks (DOD 82). The Internet Protocol (IP) developed by the Defense Advanced Research Projects Agency (DARPA) on the ARPANET is the DoD internet standard. Interoperability was further improved by the DoD declaring the Transmission Control Protocol (TCP), to be built above IP, as another standard for its host-to-host data communications networks (DOD 82). The Air Force followed suit by declaring the same standards for all of its networks (USAF 82; USAF 83).

For complete DoD compatibility, other protocol sets to handle terminal (TELNET) and bulk file transfer (FTP) applications are required. (The TELNET and FTP protocols are built above TCP/IP.) Eventually, DoD standards will be established for these functions, too. Dr. Stillman (Technical Advisor, USAF/SIT) strongly supports this approach; she feels that TCP/IP standard protocol sets (and those protocols built upon TCP/IP yet to be declared as standards) will

meet the requirements of at least 95 percent of the DoD's users (STI 83).

Finally, access/security controls are those that perform the necessary and proper checking of a job request. These checks include authentication of the user, verification that the user is authorized to use each requested resource, and a complete mediation check which ensures that the user is indeed on all the pertinent access rosters for all the resources requested and that the desired resources can be used in the requested combination. But the only access control protocols which will be examined and considered pertinent to the model are checks to see that the job is requesting a node which it can access and verification of the legality of the priority requested. Other security controls are assumed properly enforced at the node of origin and re-verified at the node of destination.

Transmission Medium. There are two choices of transmission medium. It could either be coaxial cable or fiber optics. In the first chapter, the security advantages of fiber optics were discussed. In Table

III-4, a comparison of both mediums is presented. Fiber optics are the best choice of transmission medium for this SLN. Fiber optics are strongly recommended as the transmission medium for this network because of its superior electromagnetic emanation, error rate, tapping, and isolation characteristics. It was assumed that this recommendation will be followed.

Table III-4
Comparison of Coaxial Cable and Fiber Optics.

CHARACTERISTIC	Coaxial Cable	Fiber Optics
1) Relative cost outlook		
a) currently inexpensive	Yes	No
b) potentially inexpensive	Yes	Yes
2) Small diameter/weight	No	Yes
3) Supports frequency division	Yes	Yes
4) Supports megabit transmission rates	Yes	Yes
5) Supports extremely high bandwidths (800M bits/sec)	No	Yes
6) Supports point-to-point or broadcast operation	Yes	Yes
7) Supports integrated services	Yes	Yes
8) Supports encryption	Yes	Yes
9) Relatively immune to noise	Yes	Yes
10) No crosstalk	No	Yes
11) Radio Frequency Interference	Yes	No
12) Electromagnetic Interference	Yes	No
13) Electrical isolation problems	Yes	No
14) Very low error rates	No	Yes
15) Tapping more difficult	No	Yes
16) Bidirectional (HAB 80: 960)	Yes	Yes

One way to more efficiently utilize a transmission medium is to apply a multiplexing technology. Multiplexing is a method by which more than one channel of communication are combined into one. The approach selected for this model was frequency division multiplexing.

Frequency division allocates a particular section of bandwidth to each channel all of the time (MCQ 78: 10). With this scheme, potentially only a fraction of the traffic will be intercepted if a tap with incomplete frequency coverage does occur. This limits the traffic that an eavesdropper can listen to and adds a degree of protection against unsophisticated intruders. The increased level of sophistication required for such a comprehensive full-coverage tap can serve as a deterrent to some would be intruders. Further complications can be added to the unsophisticated intruder by changing the frequency assignments at random intervals. For this thesis, the medium will be frequency divided in such a way that each of the message channels will support at least a six megabit per second transfer rate. This is because the size of the data base transfers which the SLN must support. Figure III-3 illustrates how a transmission medium that supports a 60 MBPS transmission rate could be divided to support two security classifications and three message priorities.

Bandwith channel assignments

Channel A:	Flow Control Messages
Channel B:	Security Level 1, Routine
Channel C:	Security Level 1, Overnight
Channel D:	Security Level 1, Immediate
Channel E:	Unused
Channel F:	Security Level 2, Routine
Channel G:	Security Level 2, Overnight
Channel H:	Security Level 2, Immediate
Channel I:	Unused
Channel J:	Unused

NOTES:
1) Each channel (there are ten shown) supports 6 MBPS.
2) In a Coaxial cable medium, each channel would be bracketed with unused bandwidth to decrease crosstalk. This action would result in greater fragmentation of the unused portion of the bandwidth that would be available for growth.
3) If the Bandwidth can support it, there would be more unused channels for future growth of the system.
4) Refer to Priority Scheme section for traffic class definitions (page 58).

Figure III-3. Model's Frequency Division for an 60 MBPS Fiber Optic Medium.

Switching Method. The size of the messages on this network will range from just a few bits (bursty traffic) to 900,000 bits for the data base transfers. To avoid retransmission of large data base transfers because of errors and due to the fact that most of the traffic will be data based transfers, each job request will be limited to a fixed-size transfer block which will consist of a hundred thousand bits for data and 2,400 bits of overhead (100K bits). Because of the size of the data base transfers and as a way to divide these transfers into frames or blocks which will make these long data base transfers more manageable without hogging the transmission lines when a higher priority message must get through, packet switching was chosen. The block size selected equals the size of the average data base transfer (expected to be 100,000 bits) plus the overhead bits for a header and trailer. It should be noted that packet switching will support real time applications as well as data storage, partial error control, fast speed/code conversion, delayed delivery and multiple message addressing (MCQ 78: 12). It is because of this functional flexibility that packet switching was chosen

for the model. The queues in the SLN must be large enough to hold the largest number of blocks that can make up one message.

When a message is longer than the set block size, it is divided into more than one block. These blocks are labeled to maintain proper sequencing when they are reassembled. They are then transmitted in order to the next node. Each block is considered and handled as if it were an integral and complete message. But at the final destination node the blocks are reunited by the transport level protocol to form the original message.

Flow Control. Traffic flow must be controlled to maintain a coherent pattern of transmission which will permit the proper monitoring of traffic in this SLN and to eliminate loss of messages due to insufficient available buffer space (TAN 81b: 477-478). There are several conventions that must be established to implement this control. Also, these conventions will help create a clear audit trail for messages. Some of the conventions are discussed in this chapter under sections on error, fault, and

security controls.

The first convention in this area is that of message acknowledgements. When a message is acknowledged, the sending node can delete it from its buffer space. If it is not acknowledged after some preset delay time, timeout occurs and it is retransmitted. After a predefined number of retransmissions, the problem of message loss due to a potential security breach arises. Control is, in that case, passed over to the security protocols which are covered later in this chapter in the sections on error control and security protocols.

Flow control also prevents one IMP from flooding another. Therefore, to avoid loss of messages due to insufficient buffer space, a convention of message credits is established which explicitly permit transmission from one node to another by informing the transmitting node what the receiver's available buffer space is and allowing transmission only when that space is sufficiently large. This may cause some transmission delay due to the wait that may be required while the receiving

node's buffer space is sufficiently large. But this was considered a necessary cost to maintain proper message audits for security purposes. It seems feasible to add the capability of flushing the receiving node's buffer space with some flow control message or with some control information in the header of a message to that node in the case of high priority messages, but this was not included in this model. It should be noted that implementing this buffer flushing capability could result in unacceptable message loss.

A priority scheme is discussed in these sections on protocols because it affects message handling.

Priority Scheme. There will be three non-preemptive priority classes within each of the security classifications. These classes are, from highest to lowest priority, immediate, routine, and overnight. A round robin technique will be used to address the queue of each of the classifications.

A job request with immediate priority will have first call on the networks resources on a first-come first-served (FIFO) basis within the immediate

class. No request from the lower priority classifications can be upgraded to this classification.

Routine jobs will be routed as soon as possible with a FIFO gueue discipline. They are subject to delays only when an immediate job is present. Jobs may not be routine if the data base transfer required is larger than one half the maximum message size. (The request may be routine, but the response may be such that the priority will be down graded to overnight.)

Overnight jobs have the lowest priority. Messages of this class are released only when jobs of the other classifications are not available for transmission. Only a very small percentage of all the jobs are expected to be classed as overnight.

From the information provided by Mr. Hoelscher (the point of contact for this thesis at HQ ESC), it is expected that immediate jobs will occur even more infrequently than overnight jobs since only a crisis or an emergency will warrant this classification. Routine jobs will be dominate in the SLN's traffic. A few rare

jobs will be overnight and will consist of only large data base transfers; immediate jobs will be negligible in number.

Figures III-4 through III-6 illustrate the network's connectivity and the allowable node resource requests that may originate at a given node. In those figures, the alphabetic character "C" refers to a communication node which only generates job requests and receives answers to these requests. The character "A" refers to an application node which responds to job requests and which may generate requests of its own. There are three communication nodes and four application nodes in this SLN.

Error Control. Dealing with transmission errors is important. Without protocols to handle errors, accurate communication is not possible (KEN 83; MCQ 78; PEN 79; STO 80; TAN 81a; TAN 81b). The reliability of these communications can be greatly improved if there is a high probability that few if any errors go undetected. The protocol primarily responsible with error control and reliable link-to-link transmission resides in the

data link level. It has been already mentioned that a transmission medium with a very low error rate is desireable (Table III-3). To further improve upon the reliability of the communications an error detection mechanism is then necessary.

As Tannenbaum explains, errors can be handled in two ways (TAN 81a: 126). One strategy is to include enough information to the message that allows the receiver to deduce if an error has occurred and have the message transmitted. Another strategy would be to add enough information to not only deduce that an error has occurred, but to also correct it. The second strategy is not very efficient if the transmission medium supports very low error rates. Since the selected transmission medium is fiber optics (which supports very low error rates), the first strategy was selected (MCQ 78: 23; TAN 81a: 129).

The means of detecting the error can be as simple as a parity check. But greater reliability can be achieved by a cyclic redundancy code (CRC) (PEN 79: 227). Therefore, it was assumed that each block that is transmitted within the SLN has a

trailer which provides enough bits of information to implement a CRC scheme at each node. Furthermore, due to the need for error free communication, the CRC can be supplemented with a simple scheme that regards each transmitted block as a rectangular matrix of n by m bits. In this scheme, a separate parity bit is computed for each column and is affixed to the matrix as an additional row which is then transmitted as part of the trailer. In either case, the data link protocol is charged with ensuring reliable link-to-link communications.

(A discussion of either the polynomial that would be employed for the CRC scheme or how to perform the parity scheme is not within the scope of this thesis. But a good general discussion of both techniques can be found in Tanenbaum's text.)

Also within this area is the question of what should be done if after several transmissions an error free communication is not achieved. First, the fault protocol at the transmitting node's network layer (which is waiting for an acknowledgement) is called to determine if the link between the nodes is not

functional. If the determination is a link fault, then transmission is attempted on the alternate loop. If that also is not possible, the node so informs all linked nodes and each node's table of available paths is updated to reflect that no traffic can reach a particular node or set of nodes. Also, if the receiving node continues to receive a message that it has acknowledged and which is still in its buffer, it also calls the fault protocol to determine if there is a link fault. The availability of two loops increases the probability that the nodes will still be linked after one or more link faults. If a message is deemed undeliverable because the addressee cannot be reached, the sender is informed and the message is flushed. (A simulation of the fault-tolerance and redundancy aspects of the SLN is not covered within this thesis. Wolf's work addresses this problem in some detail for a distributed double-loop network.)

If the problem is not a fault, it could be a more subtle problem and both the security and maintenance people at the SLN would be notified and the message would be continuously transmitted until

the maintenance people can attempt to check the problem out or the message is successfully transmitted.

Security Protocols. The main security protocols this thesis is concerned with deal with encryption. The link-to-link encryption (implemented in the data link layer) is assumed automatic and reliably implemented. It is the source host-to-final destination host encryption (implemented in the transport or presentation layer) which provides the necessary additional level of security required for the SLN.

The key used for the link-to-link encryption between each pair of nodes protects the entire packet of information transmitted. Each packet's data is also encrypted with a code used only between a given source and destination node for that security classification and for that particular session. This dual encryption technique forces the intruder to know both codes to get to the information when it is most vulnerable, during transmission. A further enhancement is that these codes change periodically, with each session. In this manner, an intruder will be limited to the session(s) for which

he has all the codes and not all sessions. The remote keying mechanism and the session level protocols that this would entail are not within the scope of this thesis. But the overhead in resources and processing time that security forces upon the network is expected to be relatively high.

The fact that nodes communicate with others at particular security levels allows for a design that denies the installation of equipment capable of decoding the traffic that a node is not allowed to access. Thus, each node will have, in addition to the link-to-link encryption/decryption machines for each channel, a pair of encryption/decryption devices for messages that it receives/transmits (one set for each security level). (It may be possible that one remote keying device serve all security levels.) In this model, the maximum number of nodes any single node can communicate with is three and all them fall under the same security classification. Only node C3 communicates in two different security levels and only with one node in each case. (Refer to

Figures III-5 and III-6.)

Another aspect of security is the need to deny the potential enemy reliable traffic analysis. Therefore, there is a need to have fake or dummy messages in the transmission stream. The security protocols will also control the transmission flow of dummy messages.

Dummy Message Control. Whenever there is no message to transmit from a security classification (remember the round robin aspect of these transmissions) and there is available buffer space at the next node, a single block with randomly generated bits is transmitted to the next node and then flushed from the queue immediately. The channel is selected by analyzing a random number which will control what percentage of the time a message should flow in that channel when there is no traffic. The header information for this dummy message will tell the receiving node that this is a trash message so that it is flushed from the buffer immediately. No acknowledgement is required. It is suggested that this dummy traffic travel primarily

down the immediate priority channels since these will normally have the least traffic. The fact that there normally is no traffic on these channels would indicate reaction to some critical problem. Therefore, sending dummy traffic on these channels would deny that certainty to a monitoring enemy.

However, the price of denying traffic monitoring with the use of dummy traffic should be analyzed further. The impact of this traffic could significantly affect throughput of real traffic. Such delays may be considered unacceptable while the security risk of allowing potential traffic monitoring could be considered justified by the responsible authorities.

Summary of the Model.

The next three figures present the dual ring topology of the model and the required traffic connectivity. Figures III-5 and III-6 are specially important because they define the logical link by allowable security classes among the nodes. There are three facts that stand out from those two figures. One is that node C2 does not generate any classification 1 traffic and that node C1 does not generate any

classification 2 traffic. The second is that node A1 is the only recipient of classification 1 traffic and that node A1 cannot process any classification 2 traffic. The third and final fact is that only node C3 communicates in two different security levels and only with one "A" node in each case. Then Figure III-7 presents a summary of how traffic is processed within each of the network's nodes.

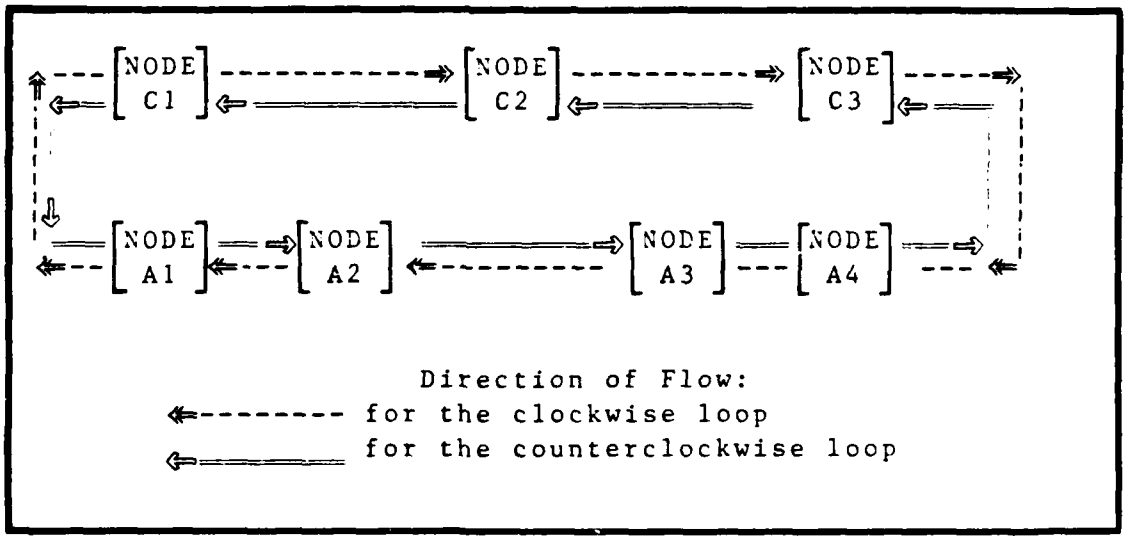


Figure III-4. The Dual Loop Network for this Model.

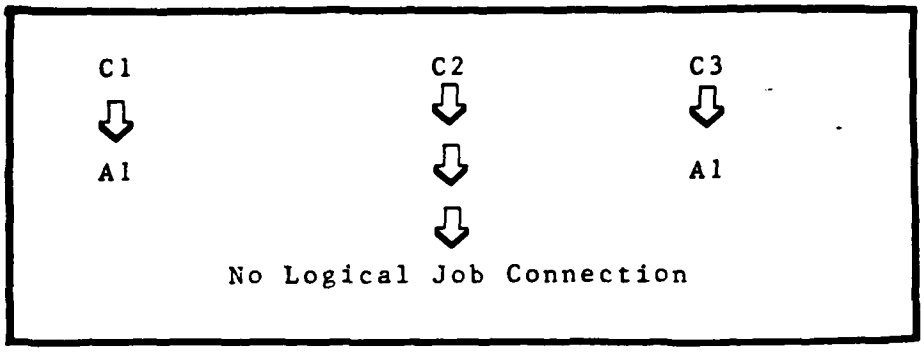


Figure III-5. Allowable Traffic for Security Classification 1.

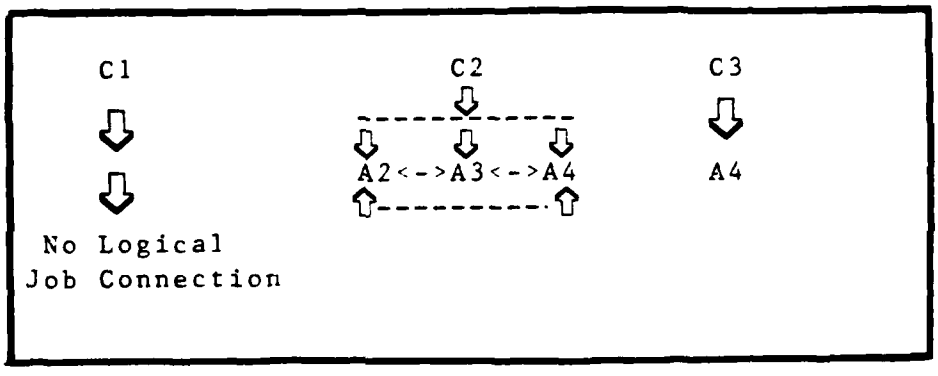
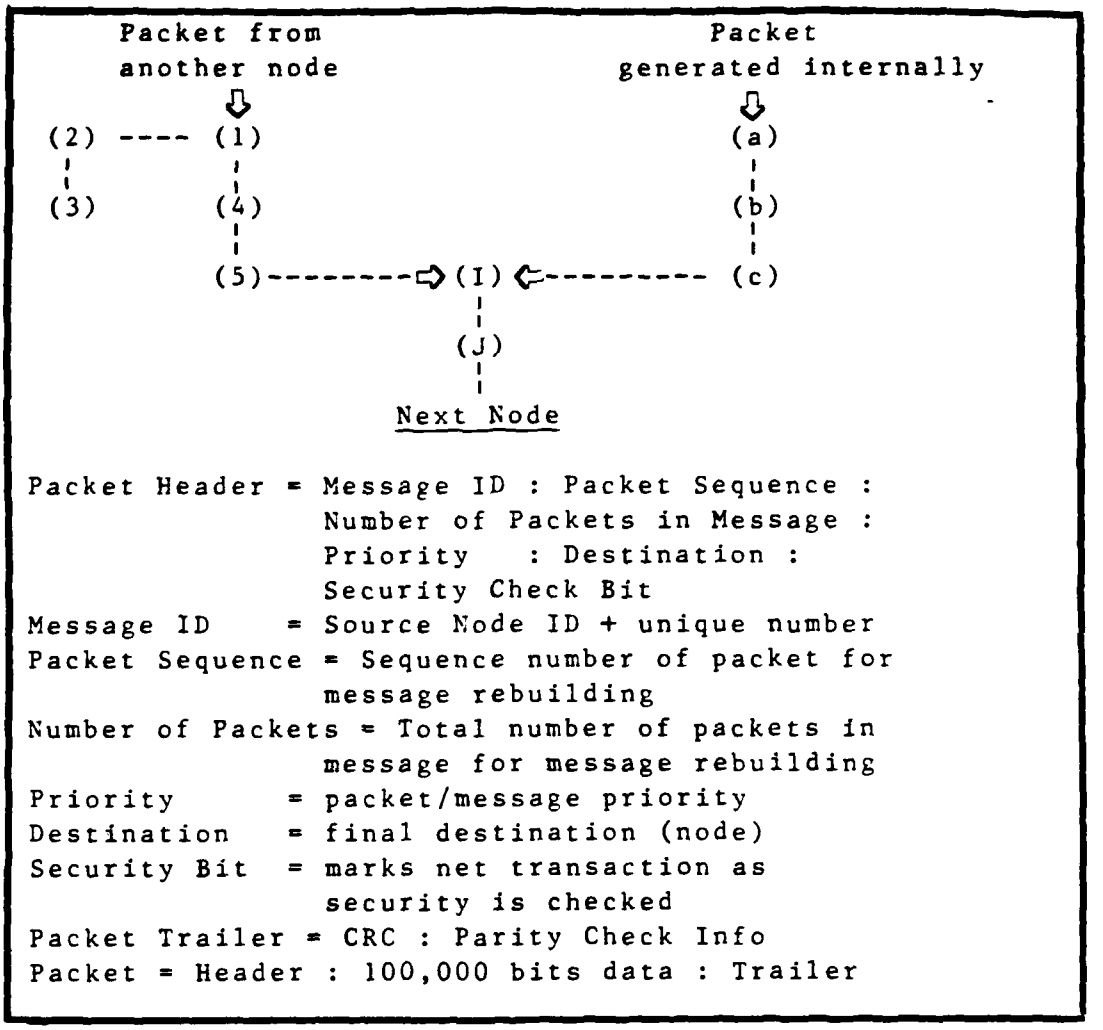


Figure III-6. Allowable Traffic for Security Classification 2.



Packet Header = Message ID : Packet Sequence : Number of Packets in Message : Priority : Destination : Security Check Bit
Message ID = Source Node ID + unique number
Packet Sequence = Sequence number of packet for message rebuilding
Number of Packets = Total number of packets in message for message rebuilding
Priority = packet/message priority
Destination = final destination (node)
Security Bit = marks net transaction as security is checked
Packet Trailer = CRC : Parity Check Info
Packet = Header : 100,000 bits data : Trailer

Figure III-7. Packet Control at SLN Node. Part I

```
(1) :  if flow control packet then
         if acknowledgement then erase acknowledged
           packet from buffer and send credit
           packet to neighbor nodes
         else if credit then
                  update credits for node affected
       go to I
       if retransmission request then
         get requested packet and go to J
       verify checksum and parity correct
       if detected error and
         retransmission counter > a max count
         then notify nodes of problem
              set notification flag
              reset retransmission counter to 0
              go to I
       if detected error then
         request retransmission
         add 1 to retransmission counter
         go to I
       if no error then
         reset retransmission counter to 0
       send acknowledgement packet
       decode HEADER
       go to 2
(2) :  if CRC and parity checks
         and security checked
         and final destination is this node
         and message complete then
         sequence the blocks
         decode the entire message
         go to 3
       else if no error and security checked
                         and for this node then
              strip trailer information
              restore in buffer
              go to I      {* msg not complete *}
            else go to 4   {not for this node *}
```

Figure III-7. Packet Control at SLN Node. Part II

```
(3) :  send on to computer resources (via DMA)
       overwrite buffer space with 0's and 1's of
         the just transferred message
       send credit messages
       go to I
(4) :  recode Header
(5) :  send to proper queue
       within security classification

(a) :  divide message into blocks
       encode message by block
(b) :  compute CRC and parity checks
       attach Trailer to block
       encode Header
(c) :  send to proper queue
       within security classification

(I) :  choose next packet to transmit
       using credit information for that node
       (Round Robin of classification queues,
       FIFO within queue.)
       if no message to transmit in either queue
       then poll queues
       until interrupted by a message arrival
       or until a message can be sent
(J) :  transmit chosen message on correct channel
       if not retransmission then
         decrease credits of node message sent to
       go to I

A head-in required to do band selection is
available at each node due to the different
channels to be selected.
```

Figure III-7. Packet Control at SLN Node.
Part III

From the preceding four figures, it can be seen that the designed SLN has a dual loop ring topology with a store and forward scheme. As transmission medium, the SLN uses fiber optics for point-to-point communications. The frequency division multiplex technique is applied to the medium to provide multiple channels to implement multiple security levels. Packet switching with a block length equal to header and trailer length plus the average data base transfer message length, 100,000 bits, is used to handle variable length messages. Block length is fixed at 100K bits. This, along with the creation of dummy traffic, will hamper traffic analysis. Dummy traffic will provide an additional degree of security. Acknowledgement and credit conventions have been adopted to avoid message losses due to insufficient buffer capacity at the receiving node. There is one queue for each classification. Each queue is long enough to hold the maximum number of blocks which can make up one message. Each queue is ordered according to one of three priority classes. When the entire message arrives at its final destination, it is decoded. Error correction will not be implemented. Instead, correct

data reception will be provided with an error detection scheme. This error detection scheme will be implemented using both CRC and parity techniques. This combination of techniques will yield an extremely low probability of missing any errors. It will also help in the detection of message stream modification when an intruder is not sophisticated enough to properly modify the CRC and parity check fields. Additional memory space is available at each node to provide a work area for decoding the message headers without altering the message in the buffer. But when the entire message is being decoded, the decyphered text is held in the message buffer until it is transfered to the host computer. This transfer is performed, for the model's purposes, instantaneously using direct memory access. Upon completion of the transfer, the area where the decoded message resides in the buffer is overwritten three times with 1's and then three times with 0's to help provide an additional measure of security.

Security is maintained during transmission through a two level encryption process which combines link-to-link as well as session specific source host-

to-final destination host encryption. Actions relating to the session level security aspects are all ignored because they do not fall within the scope of this thesis. How a packet is handled at a node is illustrated in Figure III-7 at the start of this chapter's summary.

With the design of this model complete, the next step was to evaluate it. Jackson's Theorem was applied to the model to enable an analysis of the network's operation in the environment defined above. Chapter IV discusses this analysis and an attempted simulation of the model.

Chapter IV: The Model's Evaluation

Overview.

In this chapter, the analysis of the SLN by applying Jackson's Theorem is presented. Then, the attempted simulation of the network is presented and analyzed. Finally, some conclusions are drawn about the model.

Analysis with Jackson's Theorem.

Simplification of the Model. Jackson's Theorem can only be applied if the model meets specific constraints. A goal of the simplification was to meet those constraints so that analysis using Jackson's was possible. Furthermore, the simplification process had to maintain the main elements of the designed network's traffic pattern to lend credence to the results of the analysis. Therefore, to streamline the model, several steps were taken to highlight the important traffic without seriously affecting the results of any analysis.

The first step resulted in eliminating from consideration the generation of external traffic at all of the "A" nodes. This was done simply because it is expected that no load will be generated which is not the

direct result of requests/traffic received over the "C" nodes (HOE 83).

The next step eliminated the generation of dummy traffic. Then, all consideration of traffic which would result from an explicit acknowledgement function was eliminated. Also, the priority scheme was ignored. These three steps were taken to simplify the traffic load analysis. It was deemed more important to get a gross idea of the model's behavior before expending resources in an effort that could be terminated early on through a simple test.

The fifth and final step was to assume that the packets arrive in order and are fed directly to the host when they arrive at their final destination. This simplifies the processing at each node and can be implemented through protocols. Furthermore, because a very low error rate is expected, all transmissions are assumed error free; therefore, no packages will have to be retransmitted.

The result of the five steps was a simpler version of the network model which did not alter the bulk of the traffic flow and, therefore, did not

grossly affect the analysis. But, the performance results expected from an analysis of a simplified model by applying Jackson's Theorem will most likely be better than those resulting from the application of the same theorem to the complete model. The next major step was to see if the model would fit the Jacksonian constraints.

Applying Jackson's Theorem. An analysis of the network was necessary to see how the model was expected to behave. As stated in the preceding section, the network model was simplified to permit Jacksonian analysis. After determining the general expected behavior of the network under expected constraints, if the results were deemed favorable, follow-on studies could then be used to attain greater confidence in the network's design. If the results of the initial analysis were found to preclude the success of the design, then redirection was possible without having wasted efforts in a detailed and microscopic analysis. Figure IV-1 is an accurate illustration of the simplified version of the network analyzed by using Jackson's Theorem.

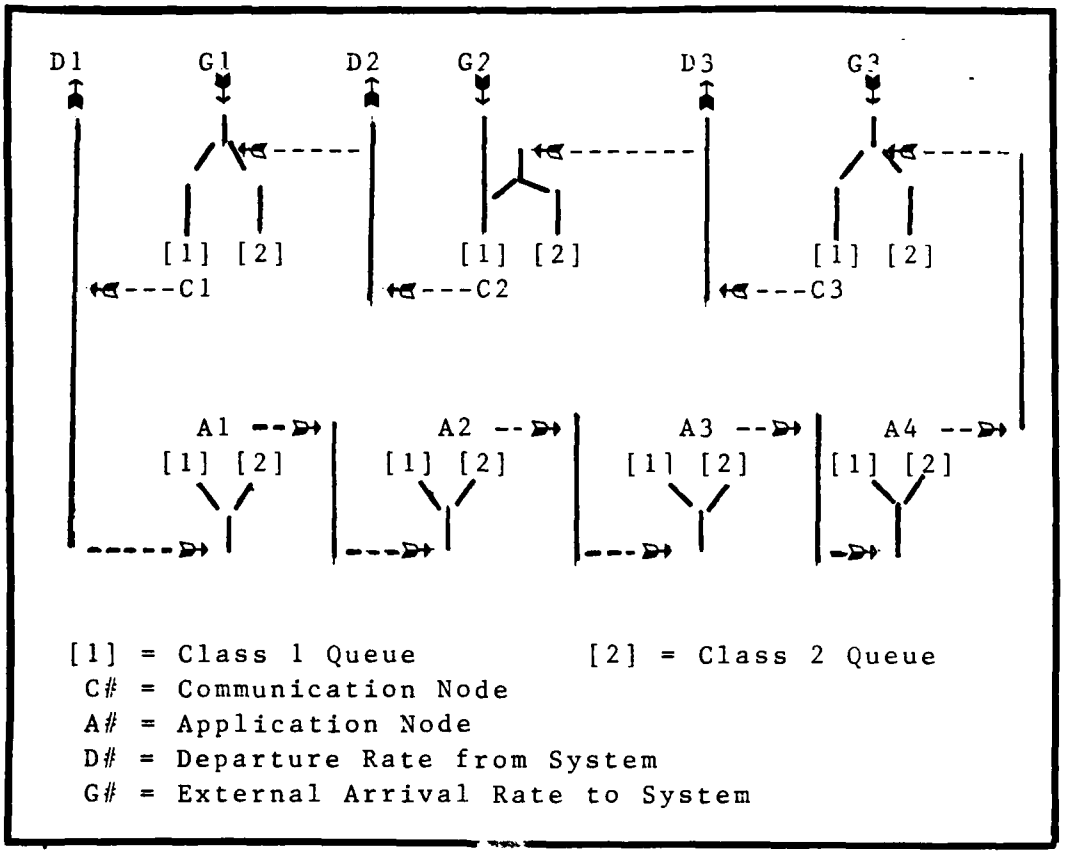


Figure IV-1. The Network.

Due to the traffic that the network supports, each node is actually composed of four components (refer to Figure IV-2). One component processes classification 1 traffic that is addressed to that node. Another component handles classification 1 traffic that is enroute to another node. A third component processes classification 2 traffic for that node. The fourth component handles classification 2 traffic that is addressed to another node.

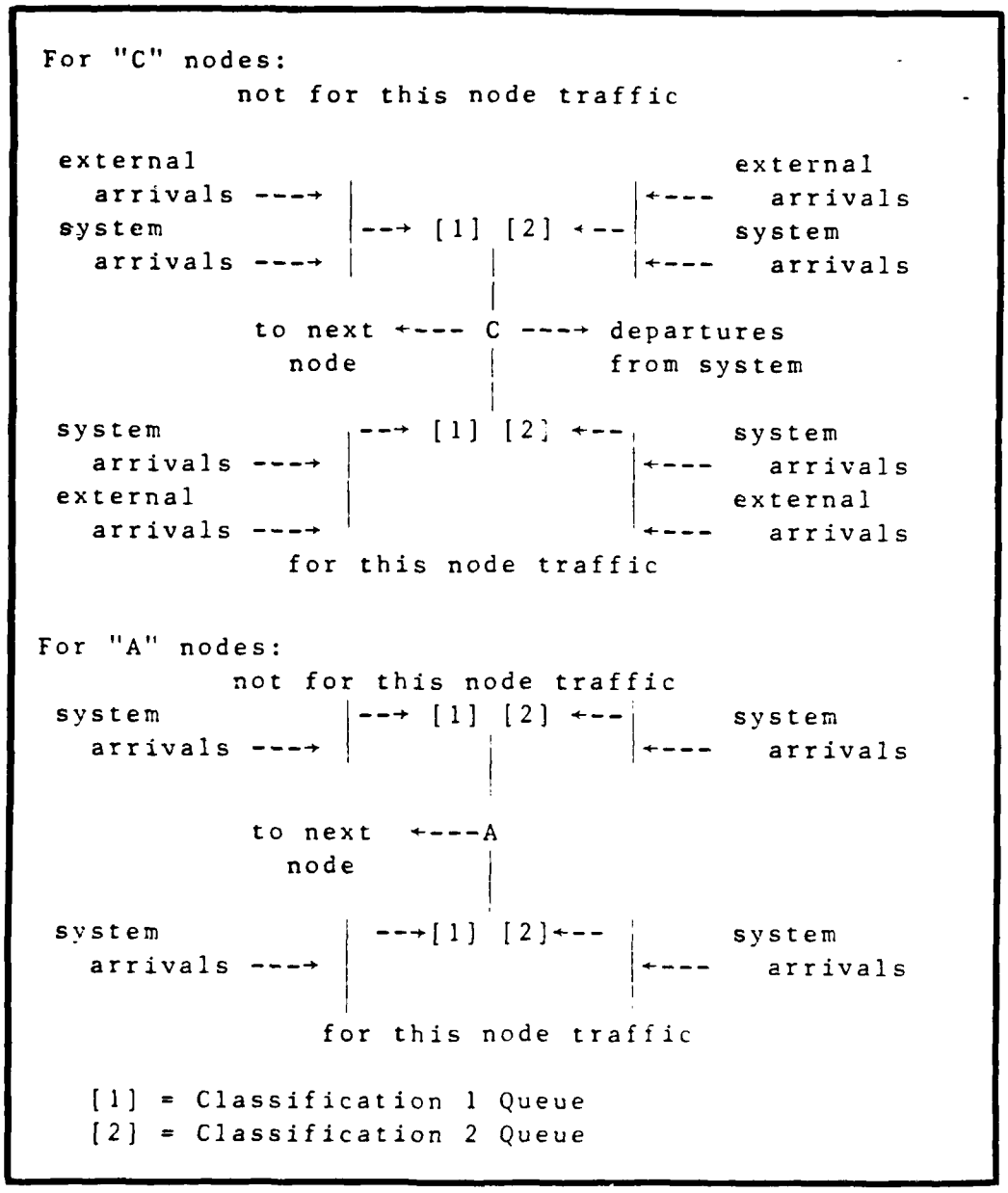
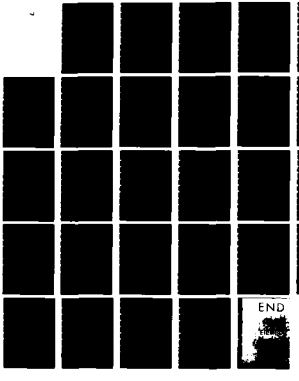
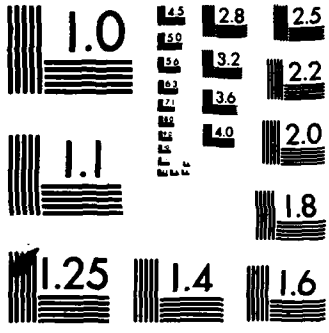


Figure IV-2. Nodal Components.

AD-A138 079 DESIGN OF A SECURE LOCAL NETWORK(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING R G CUADROS DEC 83 AFIT/GCS/EE/83D-6 2/2

UNCLASSIFIED F/G 17/2 NL





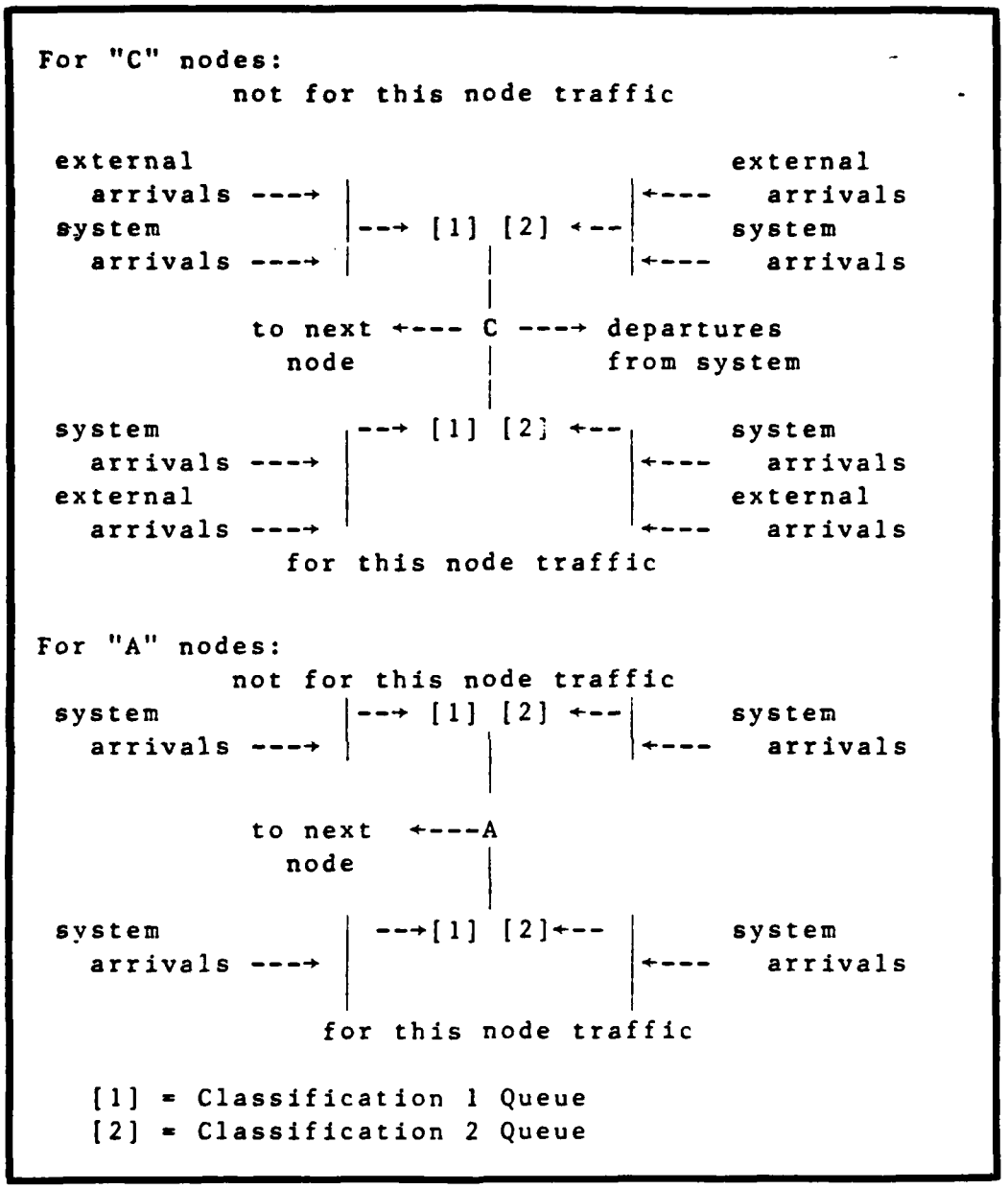


Figure IV-2. Nodal Components.

The reason for this breakdown is that traffic is not uniformly distributed by classification nor is it uniformly distributed by destination. Furthermore, traffic that is not destined for a given node is processed differently than traffic that is destined for that node. This latter traffic has a longer service time. Even though the processing time at the IMP for all traffic is roughly equivalent, additional time is required for "this node" traffic due to the response which is assumed generated for all traffic from the host computer connected to that node. This difference in service rate affects performance for "this node" traffic. Therefore, the network is actually composed of seven nodes each with four servers.

For traffic that is not addressed to a node, a fixed, deterministic, processing time was used to reflect the constant time required for packet handling. For traffic that is addressed to a node, each server uses an exponentially distributed processing time to which a fixed, deterministic time is added. But, to apply Jackson's Theorem, some assumptions had to be made.

Jackson's Theorem stated that the joint distribu-

tion for all nodes factored into the product of each of the marginal distributions is given as the solution to the M/M/m system (KLE 75: 150). This theorem applies to open networks of queues with Poisson arrivals, FCFS queues, exponential service times, and no saturated queues (KLE 75: 149, SAU 81: 80-81). Furthermore, thanks to Burke's Theorem, a network of multiple-server nodes connected in a feedforward fashion still preserve the node-by-node decomposition that makes Jackson's Theorem so useful (KLE 75: 149). For this evaluation all of the conditions were met or could be assumed as met for analytical purposes when the service times for all traffic was idealized to exponential service rates. The deterministic service rate was added to the mean of the expected service rate to yield a new exponential service rate. This shifted the mean service rate but did not totally ignore their deterministic component.

Having met the necessary conditions for Jackson's Theorem, Table IV-1 was developed presenting the arrival rates in terms of the external arrival rates to the system and the necessary performance parameters were computed (Table IV-2).

Table IV-1
Mean Arrival Rates for the Simulation
Using Jackson's Theorem.

Node	Lamda (in terms of external arrival rates)
A1[1]t	.5 G1 + G2 + .5 G3
A1[1]n	0
A1[2]t	0
A1[2]n	.5 (G1 + G3)
A2[1]t	0
A2[1]n	.5 G1 + G2 + .5 G3
A2[2]t	1/6 (G1 + G3)
A2[2]n	1/3 (G1 + G3)
A3[1]t	0
A3[1]n	.5 G1 + G2 + .5 G3
A3[2]t	1/6 (G1 + G3)
A3[2]n	1/3 (G1 + G3)
A4[1]t	0
A4[1]n	.5 G1 + G2 + .5 G3
A4[2]t	1/6 (G1 + G3)
A4[2]n	1/3 (G1 + G3)
C1[1]t	.5 G1
C1[1]n	.5 G1 + G2 + .5 G3
C1[2]t	.5 G1
C1[2]n	.5 (G1 + G2)
C2[1]t	G2
C2[1]n	.5 G1 + G2 + .5 G3
C2[2]t	0
C2[2]n	.5 (G1 + G3)
C3[1]t	.5 G3
C3[1]n	.5 G1 + G2 + .5 G3
C3[2]t	.5 G3
C3[2]n	.5 (G1 + G3)

C# = Communication Node A# = Application Node
[1] = Class 1 traffic [2] = Class 2 traffic
n = traffic not for this node
t = traffic for this node
G# = External Arrival Rate to System
(there are three gateways to the system)

Results. It was, of course, known that these results were idealistic since each node really was a single-server and processing times could be deterministic depending on the type of traffic being processed. But the careful selection of the parameters helped provide confidence in the results of the analysis.

The computations made for Table IV-2 were based on one packet per message, external arrival rate of 0.0001 messages per millisecond (i.e., G1 = G2 = G3 = 0.0001), a service rate of 0.001 millisecond per packet for "not-this-node", and a service rate of 0.006 milliseconds per message for "this node" traffic. This arrival rate is considerably faster than the expected and forseeable average traffic load for the network of 100,000 bits of raw data per second over one "C" node and 50,000 bits of raw data per second for each of the other two "C" nodes (HOE 83). This faster rate was chosen to provide greater confidence in the results of an analysis performed on an idealistic representation of the model. The service rates are those expected with the equipment that is planned for the actual network's implementation (HOE 83).

Table IV-2. Performance Parameters
Computed Using Jackson's Theorem

Node	Lamda	Utilization	Queue Length
A1[1]t	.0002	.033	.034
A1[1]n	0		
A1[2]t	0		
A1[2]n	.0001	.1	.1
A2[1]t	0		
A2[1]n	.0002	.2	.25
A2[2]t	.000033	.0055	.0055
A2[2]n	.000067	.067	.07
A3[1]t	0		
A3[1]n	.0002	.2	.25
A3[2]t	.000033	.0055	.0055
A3[2]n	.000067	.067	.07
A4[1]t	0		
A4[1]n	.0002	.2	.25
A4[2]t	.000033	.0055	.0055
A4[2]n	.000067	.067	.07
C1[1]t	.00005	.0083	.0084
C1[1]n	.0002	.2	.25
C1[2]t	.00005	.0083	.0084
C1[2]n	.0001	.1	.11
C2[2]t	.0001	.1	.11
C2[2]n	.0002	.2	.25
C2[2]t	0		
C2[2]n	.0001	.1	.11
C3[1]t	.00005	.0083	.0084
C3[1]n	.0002	.2	.25
C3[2]t	.00005	.0083	.0084
C3[2]n	.0001	.1	.11

C# = Communication Node A# = Application Node
[1] = Class 1 traffic [2] = Class 2 traffic
n = traffic not for this node
t = traffic for this node

From the computational results, it can be inferred that the designed fullblown SLN model should provide adequate performance and process effectively the bulk data traffic that characterizes the expected traffic load. As Table IV-2 shows, the system is very capable of handling traffic at one packet per message with an arrival rate of 0.0001 messages (packets) per millisecond and a service rate of one message (packet) per millisecond. Even if each message was made up of more than one packet, the utilization rate (arrival rate divided by service rate) would still be less than one. As stated earlier, the chosen arrival rate used is an extreme case load that is ten to twenty times greater than what could be considered within the realm of possibility. Yet, at every point, the utilization rate is considerably less than one. Therefore, the network should be stable and capable of handling a heavier traffic load.

The Simulation and Throughput Performance.

The simulation should show how throughput is affected by different mixes. Factors that influence throughput are the error rate and the

resulting retransmission, maximum message size, block size, medium speed, arrival rates, and service rates at the nodes. Arrival and service rates and message length are the only variables addressed by the thesis; the other variables are left for further study.

Guidance provided by the thesis sponsors limited the range of some of these variables (HOE 82; HOE 83). All traffic entering the system would be uniformly distributed over the three communication nodes. (The distribution of the classification of this traffic was previously addressed in Figures III-5 and III-6.) Short bursty transmissions and data base transfers would be the only type of traffic. The data base transfers would range from 50 to 80 percent of all messages. Data base transfer traffic is expected to average about 100,000 bits in length with a range from 100,000 to 900,000 bits. Three priority classes were generated for the model. At least 50 percent of the traffic would be routine and traffic for the highest priority could be considered rare to non-existent except in a crisis.

To focus on the network, it was assumed for this thesis that each individual host would have its own priority scheme and would handle the messages as it deemed appropriate. But handling the priority scheme was beyond the scope of the analysis performed. Table IV-3 shows the areas actually addressed by the simulation.

Table IV-3. Variables Used in the Analysis of the Network's Throughput Performance.

1) Arrival rate 2) Service rate 3) Message length (range: 1 to 10 packets)

Some areas are left unexamined by the simulation. Such areas as the impact of link faults, buffer size, and error rates on the SLN's throughput, are left for follow-on projects. This simulation concentrates on the three areas identified in the preceding table.

But how are these areas studied?

Examining Throughput Performance. The simulation program implementing the model had to have flexible entries for the features listed in Table IV-3

to be examined. Runs were performed changing only one of those three parameters between executions. To help in the evaluation, the maximum number of packets held in each node's buffer for each run was to be kept, as well as the number of messages and packets processed at each node. This would permit analysis on how variations affected results.

Since the processing of the SLN's traffic consumes time and the traffic could not be generated in real time, the program had to simulate the passing of time. Events are therefore created and processed to simulate this passage of time. The program implements an event driven simulation.

The Design Process. Software engineering techniques were applied. First, the requirements had to be explicitly defined and the functions that were to be performed defined and refined until a structure chart of modules is fully developed. Most of the initial work was spent on the generation of what is illustrated in Figure III-7. It was critical to know or decide how messages were to be processed at each node so that the network analysis could be

determined. General traffic flow requirements were defined in Figures III-4, III-5, and III-6.

After developing the functions that were to be performed at each node (which resulted in Figure III-7), a chart presenting the functions to be performed was drawn. Initially, the functions to be implemented included retransmissions and flow control. Then, the number and diversity of these functions was limited by the problems that arose with the language being used to implement the simulation and by the mathematical tools available to perform the analysis. After the decision was made to restrict and simplify the model, the next step was to see how the functions necessary to simulate the SLN could be grouped or developed. This resulted in Figure IV-3. The technique of stepwise refinement was used to get the simulation down to a level that could lead to code. From the very start, a data dictionary (Appendix C) was maintained and every effort was made to use names that were meanginful. The names of constants, variables, procedures, and functions were made self-explanatory whenever possible within the constraints placed on their

length by the compiler and by the programmer's additional constraint of avoiding multiple lines for simple data manipulations. Furthermore, the programmer avoided nesting of "if" statements to ease debugging. This latter constraint could be changed later if code optimization were desireable.

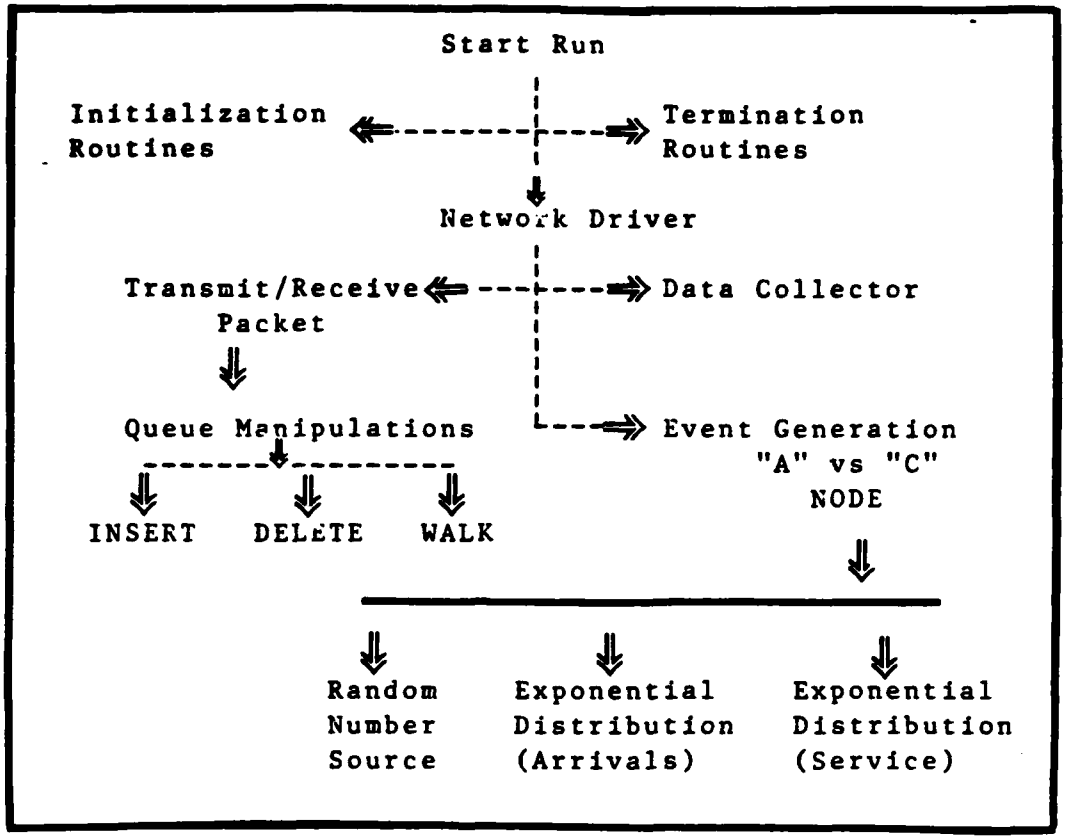


Figure IV-3. Functions Performed by the Simulation Program.

It was obvious at the start that there would be variable parameters in each run. A parameter initialization module had to be the first module which had to interact with the user who would input parameters. Of special importance was the start time for statistics collection since the simulation would

have to run some undetermined amount of time to reach steady state prior to data collection. This time was to be arbitrarily set and hopefully a reasonable delay time would become apparent through trial-and-error. But before any initialization module was designed, the first step taken was to translate the traffic load into an event generating algorithm that represented it.

The event generation function was a straight forward implementation thanks to the detailed information made available on the expected traffic load (refer to Chapter III, especially the sections entitled: Overview, Switching Method, Priority Scheme, and Summary of the Model). The only hitch in the entire algorithm development process was the lack of random number generators in the chosen language, PASCAL. Books by Hillier and Sauer (HIL 73; SAU 81) eventually helped by providing formulas for exponential distributions. But the cleanest solution was the one finally implemented, to use CBASIC II (Compiler Systems, Inc., version 2.0, July 1981) to generate, initially, a two thousand entry file of uniformly distributed random

numbers which could then be accessed by the simulation program whenever it required a uniformly distributed number. (After much trial-and-error, the best cycling that was achieved for a uniformly distributed pseudo-random number generator was every 574 times, this was deemed, after consultation with the thesis advisor, borderline acceptable. Reading from a file of uniformly distributed random numbers was easier to follow for purposes of programming and debugging.)

Next, after developing the event generating algorithm, handling of the created event record via a linked-list queue was tackled. The queue manipulation function was much more difficult. Translating Figures III-4, III-5, and III-6 and Figures IV-1 and IV-2 into code was just the beginning. Event insertions and deletions, walking the queue, moving events about in the queue to simulate the flow of a packet around the network to its destination and the integration of calls to modules to generate new events as well as the insertion of code to trap required data for follow-on analysis was not trivial. Fortunately, the decision not to include flow and error control traffic

simplified the implementation. The final program design is reflected by the structure chart in Appendix B.

The Differences. As Figure IV-1 illustrates, several SLN functions discussed in Chapter III were not implemented in the simulation. There are six important differences which resulted from the model's simplification. The rationale for this simplification is discussed in detail at the beginning of this chapter. Briefly, the simplifications were required to permit analytical validation of the model with Jackson's Theorem.

The first difference is the lack of external traffic generation at the "A" nodes. The next difference is the lack of dummy traffic generation. The third difference is the lack of an explicit acknowledgement function. The fourth difference is that packets are assumed to arrive in order and to be fed directly to the host when they arrive at their final destination. Next, the priority scheme is ignored. Finally, the sixth major difference is that all transmissions are assumed error free.

The Problems. As has already been remarked, the

simulation was an additional attempt to further validate the network model that was designed. Unfortunately, the simulation was never completed. Several problems hindered the successful execution of the simulation. The most critical problem was the language chosen for the simulation.

Language and Machine Decisions. The SLN model developed over the preceding two chapters was a severely constrained by the chosen simulation environment. The simulation was to be performed on a microcomputer to see what could be accomplished on a small system. As far as could be determined, no network simulation had yet been performed on a microcomputer. Performing the simulation on a microcomputer would present constraints on the simulated model due to available memory and computing power. The choice of language would also affect the implementation due to routines available and ease of use. A machine and a language had to be chosen. The process is presented below.

The machine desired was a microcomputer with a proven processor chip. Other desired characteristics were a large main memory and as much easily accessible

secondary storage as possible. Finally, the machine had to be available for use.

Because of availability, an Intertec Data Systems "Superbrain" Z80A microcomputer with dual 5.25 inch single-sided soft-sectored floppy disk drives (each with 162K useable storage capacity) with 64K RAM was used. When that machine shorted out, it was replaced with a microcomputer of the same make, but with double-sided floppy disk drives. The upgrade in disk storage capacity was a definite asset during the development of the thesis because of the additional 332K of secondary storage.

Because of software availability, the language choices were limited to some form of Basic, C, or Pascal. Due to the unstructured nature, non-overlay features, and language construct limitations of the Basic softwares available, Basic was not chosen. Both C and Pascal did not suffer these handicaps. They are structured languages and they both support overlays. After talks with some members of the faculty and using a timely article in ACM Computing Surveys by Alan R. Feuer, Pascal was chosen since it was structured, its

dynamic storage for link lists was deemed highly appropriate for event-driven simulations, and the available compiler was apparently well-documented and supports overlays (critical in a RAM constrained environment), and this researcher was familiar with the language through courses recently completed.

Once Pascal and the machine were chosen, the next phase was to see how code the model and evaluate the network's performance.

The Language. The Pascal language supports both overlays and recursive calls, has a good diagnostic package to aid in debugging, is structured, and the author had some programming experience in the language. But the software did not provide any number generator routines and does not provide the programmer with a simple and direct capability for direct bit manipulation. In retrospect, for this restricted memory environment, the bit manipulating capability of C was a more important characteristic which should have led to it being chosen instead. Besides, C also provided several number generator routines. But the restrictive memory in itself was not the problem since

overlays could in part offset it by not having the entire program in main memory.

Unfortunately, the most blatant problem during the development of this thesis was the language chosen. This problem manifested itself in primarily two ways. In the first place, overlays were never possible. In second place, the debugging package was not fully useable.

Without overlays, the number of functions that could be simulated was reduced. This caused considerable simplification of the model which in itself was not as discomfitting as the reason why overlays were not performed. After working with Pascal for a while, it became apparent that the documentation package was not as good as advertised and therefore, expected.

The other major problem was that to use the debugger, the program size was drastically limited. That may have been solved with overlays, but as mentioned above, the documentation was not that easily or well understood. In fact, no one was found to provide any aid in this area. Thus, overlays were not

performed and the debugger was not available to help during the debugging phase. But even if the debugger had been available for use, its usefulness was severely handicapped by the fact that it could not handle real numbers. This severe handicap was not discovered until the software development was well into the coding phase. All in all, it may be best to have C as the language for any follow-up work on a microcomputer.

The last related problem was that when the simulation program was finally compiled clean, it did not execute as expected. This was never resolved prior to the thesis effort being terminated. But it was the development of a means to handle random numbers that caused the single most frustrating period during the generation of this thesis.

The Random Number Generator. The development of the uniform random generator was more difficult than expected. Several sources presented good examples for mini and other large computers, but none presented one for a microcomputer.

Finally, the theory presented by Sauer and

Hillier was used to program a number generator. But when it was tested, cycling occurred so quickly that its value was questionable, though considered acceptable. Finally, after some study and trial-and-error, the solution adopted was to generate a uniform number file using C-BASIC II which was then read as necessary by the Pascal program. This was quickly tested and proved a clean implementation prior to its inclusion in the network simulation program.

Conclusions.

Application of Jackson's Theorem validated the designed network. Even though the results of this analysis are idealistic, the careful simplification and streamlining of the model and the judicious selection of arrival and service rates provide a high degree of confidence in the design's ability to meet its traffic goals.

As for the simulation program (Appendix A), it would be interesting to see the model validated in this manner. Definitely, it would behoove whomever desired this SLN to have it simulated with as realistic a set of constraints as possible before the immense cost of

actually developing the network were made. A SLN is not an inexpensive system since heavy software costs are involved to develop protocols and interfaces which are not in existance today.

Chapter V: Conclusions and Recommendations

Overview.

As shown in the preceding chapter, the simplified version of the designed model should be able to handle the projected work load. Based on that analysis, it is expected that the more complex model (summarized in the last section of Chapter III) would also meet the work load requirements. In any case, the model was designed to: 1) effectively process bulk data traffic, 2) provide a high level of security, and 3) permit multiple concurrent transmissions of different classifications. In this last chapter, areas for further study are presented and some conclusions are drawn from the experience of completing this thesis.

Areas for Further Study.

There are at least five areas left for further study. The five areas discussed below were not fully developed within the scope of this thesis, but they all deserve additional research and examination.

In the first place, an attempt to generalize the network model for applications more interactive/bursty

in nature could result in different design elements. This researcher believes that the major differences between the design of this SLN and one with more bursty traffic would be in the area of topology (a web might be more appropriate) and network access control (possibly contention instead of shift register insertion).

But, within the framework of this design and ESC's specific constraints, the addition of dummy traffic, of new arrivals from the "A" nodes, of flow control traffic, of error/reliability traffic (retransmissions), and of priority traffic to a simulation for the purpose of examining throughput would be of major interest. Of course, this would entail successfully developing the simulation attempted for this thesis work. In any case, the traffic that is potentially the most damaging to throughput is the dummy load. It could cause unacceptable delays which would require the re-examination by higher authorities of its need for security.

A third area would be research into the interoperability and interface issues of a SLN and other secure and/or non-secure networks. An analysis of

TCP/IP and the projected national level long haul communications networks like the Defense Data Network would be within the scope of such work.

Another area that deserves more study is that of fault tolerance and fault limitation/isolation in both physical (hardware) design and in the design of protocols. But probably the most intriguing area would be in the fifth area, the expansion of the security aspects of this thesis.

The encryption of this model revolves about the secure/trusted generation and distribution of keys and their management. This area has been addressed by many without, to this researcher's knowledge as of August 1983, an accepted way of doing so. (Accepted by this country's national level security agencies.) Any follow-on work in this area could bring great dividends to this nation's security.

Conclusions.

The interplay of topology, network access, switching method, and flow and error control protocols was challenging, extremely enlightening, and definitely

interesting. The addition of security constraints does cloud the issue of performance, but flexible designs with inherently good performance characteritics seem to be best suited for security, too. The design process is definitely influenced by security issues, especially those which deal with the need to limit the electromagnetic emanations of the hardware and the need to guard against traffic analysis. But, the key to achieving security seems to exist primarily within the realm of software access controls implemented in the network's protocol structure (even if these protocols are implemented through micro-code).

Bibliography

AGR 78 Agrawala, A.K., J.R. Agre, and K.D. Gordon. "The Slotted Ring vs. the Token-Controlled Ring: A Comparative Evaluation," IEEE COMPSAC: 1978 Pp. 674-678.

AME 83a Ames, Stanley R. Jr., Morrie Gasser, and Roger R. Schell. "Security Kernel Design and Implementation: An Introduction," IEEE Computer, Vol 16, No 7: 14-22 (July 1983).

AME 83b Ames, Stanley R. Jr. and Peter G. Neumann. "Computer Security Technology: Guest Editor's Introduction," IEEE Computer, Vol 16, No 7: 14-22 (July 1983).

BAL 81 Ball, J. Eugene, Jerome Feldman, James R. Low, Richard Rashid, and Paul Rovner. "RIG, Rochester's Intelligent Gateway: System Overview," in Tutorial: Local Computer Networks. Pp. 316-323.

BAS 81 Bass, Charlie, Joseph S. Kennedy, and John M. Davidson. "Local Network Gives New Flexibility to Distributed Processing," in Tutorial Local: Computer Networks, Thurber (Ed.) Pp. 358-366.

BLAC 83 Blackmarr, Brian B. "Local-Area Nets," Computerworld OA, Vol 15, No 48A: 40-44 (1 Dec 1983).

BLA 82 Blair, Gordon S. and Doug Sheperd. "A Performance Comparison of the Ethernet and the Cambridge Digital Communication Ring," Computer Networks, Vol 6, No 2: 105-113 (May 1982).

BOG 80 Boggs, D.R., et. al. "Pup: An Internet Architecture," IEEE Transactions on Communications, COM-28: 612-624 (April 1980).

BOO 81 Booth, Grayce M. The Distributed System Environment: Some Practical Approaches. New York: McGraw-Hill Book Co., 1981.

BUX 81 Bux, Werner. "Local-Area Subnetworks: A Performance Comparison," IEEE Transactions on Communications, COM 29, No 10: 1465-1473 (October 1981).

CLA 81 Clark, David D., Kenneth T. Pogran, and David P. Reed. "An Introduction to Local Area Networks," in Tutorial: Local Computer Networks, Thurber (Ed.) Pp. 16-35.

COO 83 Cooper, Edward B. "Broadband Network Design: Issues and Answers," Computer Design, Pp. 209-216 (March 1983).

COR 81 Cornell, Roger G. and David J. Stelte. "Progress Towards Digital Subscriber Line Services and Signaling," IEEE Transactions on Communications, COM-29 No 11: 1589-1594 (November 1981).

COV 80 Coviello, Gino J., Irwin Lebow, Raymond L. Pickholts, and Donald L. Schilling. "Preface: Military Communications -- An Overview of the Special Issue," IEEE Transactions on Communications, COM-28, No 9: 1441-1444 (September 1980).

DAV 81 Davida, George, Charles K. Wilk, Charles S. Wood, Robert Morris, and Melvin Klein. "Panel Session on Cryptography," IEEE Security and Privacy, 1981 Pp. 151-161 New York: 1981.

DEN 79 Denning, Dorothy E. and Peter J. Denning. "Data Security," Computing Surveys, Vol 11 No 3: 227-249 (September 1979).

DEN 82 Denning, Dorothy E. Cryptography and Data Security. Reading, MA: Addison-Wesley, 1982.

DID 82 Didic, Milena and Bernard Wolfinger. "Simulation of a Local Computer Network Architecture Applying a Unified Modeling System," Computer Networks, Vol 6, No 2: 75-91 (May 1982).

DOD 77 DOD 5520 22-M, Industrial Security Manual for Safeguarding Classified Information. Defense Intelligence Agency: October 1977.

DOD 83 DRS 2600-3779-83, DODIIS Network Security for Information Exchange (DNSIX). Defense Intelligence Agency: August 1983.

DOD 82 "DOD Policy on Standardization of Host-to Host Protocols for Data Communications Networks," USDRE Memo (23 Mar 1982).

DON 79 Donaldson, Hamish. Designing a Distributed Processing System. New York: John Wiley and Sons, 1979.

FAR 81 Farber, David J. "A Ring Network," in Tutorial: Local Computer Networks, Thurber (Ed.) Pp. 134-136.

FEU 82 Feuer, Aland R. and Narain H. Gehani. "A Comparison of the Programming Languages C and PASCAL," ACM Computing Surveys, Vol 14, No 1: 73-92 (March 1982).

FIN 81 Findlay, William and David A. Watt. Pascal: An Introduction to Methodical Programming. (2nd ed.) Rockville, MD: Computer Science Press, Inc., 1981.

FOS 83 Foster, Susan. "Networking Micros: Sharing Resources," Computer Decisions, Pp. 100-213 (April 1983).

FRA 83 Fraim, Lester J. "SCOMP: A Solution to the Multilevel Security Problem," IEEE Computer, Vol 16, No 7: 26-34 (July 1983).

FRE 80 Freeman, H.A. and K.J. Thurber. "Updated Bibliography on Local Computer Networks," Computer Architecture News, Vol 8: 20-28 (April 1980).

GIB 81 Gibson, Ronald W., Charles A. Harris, S.P. Border, and A.T. Matsumoto. "Technology Survey of Local Area Networks," a Boeing Computer Services Report, December 1981.

GLA 83 Glazer, Sarah. "Committees, Vendors Chase Elusive Networking Standards," Mini-Micro Systems, Pp. 101-110 (July 1983).

GOL 83 Goldberger, Alex and Stephen Y. Lau. "Understand Datacomm Protocols by Examining Their Structure," EDN, Pp. 109-118 (3 March 1983).

HAB 80 Habara, Kohei and Takao Aratani. "Toward Local Network Digitalization: The View from Japan," IEEE Transactions on Communications, COM-28, No 7: 956-966 (July 1980).

HIL 73 Hillier, Frederick S. and Gerald J. Lieberman. Introduction to Operations Research. Pp 444-450. San Francisco, CA: Holden-Day, Inc., 1973.

HOE 82 Hoelscher, Wilbur L. Correspondence with HQ ESC/AD. (Hoelscher, point of contact for this thesis) March 1982 - November 1982.

HOE 83 -----. Conversations with HQ ESC/AD. (Hoelscher, point of contact for this thesis) January 1983 - August 1983.

HOM 80 Homayoun, Fred. "Loop Evolution: Its Dynamics and Driving Forces," IEEE Transactions on Communications, COM-28, No 7: 976-982 (July 1980).

IKE 80 Ikeda, Katsuo, Yoshinoko Ebihara, Michiro Ishizaka, Takao Fujima, Tomoo Nakamura, and Kazuhiko Nakayama. "Computer Network Coupled by 100 MBPS Optical Fiber Ring Bus -- System Planning and Ring Bus Subsystem Description --," IEEE COMPCON Fall 1980 Pp. 159-165 New York: 1980.

JON 83 Jones, J. Richard. "Consider Fiber Optics for Local-Network Designs," EDN, Pp. 105-107 (3 March 1983).

KAT 78 Katzan, Henry J. An Introduction to Distributed Data Processing. New York: Petrocelli Books, Inc., 1978.

KEN 83 Kent, Stephen T. Conversations with Dr. Kent to Discuss Security and Networking Issues, 10 March 1983, 22-23 June 1983, 2-3 and 8 August 1983.

KEN 76 -----. Encryption-Based Protection Protocols for Interactive User-Computer Communication ADA026911 Technical Report 162 Massachusetts Institute of Technology Cambridge, MA: Laboratory for Computer Science, 1976.

KEN 81a -----. "Security Requirements and Protocols for a Broadcast Scenario," IEEE Transactions on Communications, COM-29, No 6: 778-786 (June 1981).

KEN 81b -----. "Security in Computer Networks," in Protocols and Techniques for Data Communication Networks, Kuo (Ed.) Pp. 369-432.

KLE 75 Kleinrock, Leonard. Queueing Systems: Volume I: Theory. New York: John Wiley and Sons, 1975.

KOF 81 Koffman, Elliot B. Problem Solving and Structured Programming in Pascal. MA: Addison-Wesley Publishing Co., Inc., 1981.

KON 81 Konheim, Alan G. "Guest Editor's Prologue," IEEE Transactions on Communications, COM-29, No 6: 761 (June 1981).

KOT 83a Kotelly, George. "Local-Area Networks," EDN, Pp. 109-122 and 127-150 (17 February 1983).

KOT 83b -----. "Personal Computer Networks," EDN, Pp. 83-100 (3 March 1983).

KUM 82 Kummerle, K. and M. Reiser. "Local Area Communications Networks -- An Overview," Journal of Telecommunication Networks, Pp. 349-370 (Winter 1982).

KUO 81 Kuo, Franklin F. (Editor) Protocols and Techniques for Data Communication Networks. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1981.

LAN 83 Landwehr, Carl E. "The Best Available Technologies for Computer Security," IEEE Computer, Vol 16, No 7: 86-100 (July 1983).

LAP 83 La Padula, L. Conversations with Mr. La Padula to discuss Network Security Issues, 12 April 1983.

LEM 79 Lempel, Abraham. "Cryptology in Transition," Computing Surveys, 11, No 4: 285-303 (December 1979).

LEN 81 Lennon, Richard E., Stephen M. Matyas, and Carl H. Meyer. "Cryptographic Authentication of Time-Invariant Quantities," IEEE Transactions on Communications, COM-29, No 6: 773-777 (June 1981).

LEW 81 Lewin, Leonard (Editor). Telecommunications in the United States: Trends and Policies. Dedham, MA: Artech House, Inc., 1981.

LIS 83 Lissack, Tsvi, Basil Maglaris, and Ivan T. Frisch. "Digital Switching in Local Area Networks," IEEE Communications, Vol 21, No 3: 26-37 (May 1983).

LIV 82 Livingston, William D. "Local Area Network Improves Real Time Intelligence Systems," Defense Electronics, Pp. 68-80 (Dec 1982).

LOR 80 Lorin, Harold. Aspects of Distributed Computer Systems. New York: John Wiley and Sons, 1980.

MAR 77 Martin, James. Future Developments in Telecommunications. (2nd edition) Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1977.

MCQ 78 McQuillan, John M. and Vinton G. Cerf. (Editors) Tutorial: A Practical View of Computer Communications Protocols. New York: IEEE Computer Society Press, 1978.

NAC 82 NACSIM 5100A, Compromising Emanations Laboratory Test Requirements: Electromagnetics. National Security Agency: June 1982.

NES 83 Nessett, Dan M. "A Systematic Methodology for Analyzing Security Threats to Interprocess Communication in a Distributed System," IEEE Transactions on Communications, COM-31 No 9: 1055-1063 (Sept 1983).

PAR 83 Parker, Richard and Sydney F. Shapiro. "Untangling Local Area Networks," Computer Design, Pp. 159-172 (Mar 1983).

PAU 80 Paulish, Daniel J. "A Fail-Soft Distributed Processing System," IEEE COMPCON Fall 1980 Pp. 179-184 New York: 1980).

PEN 79 Penney, B.K. and A.A. Baghdadi. "Survey of Computer Communications Loop Networks: Part 1 and Part 2," Computer Communications, Vol 2, No 4-5: 165-180, 224-241 (Aug-Oct 1979).

POP 79 Popek, Gerald J. and Charles S. Kline. "Encryption and Secure Computer Networks," Computing Surveys, 11, No 4: 331-356 (Dec 1979).

POS 80 Postel, J.B. "Internetwork Protocol Approaches," IEEE Transactions on Communications, COM-28: 604-611 (April 1980).

RIL 82 Riley, Wallace B. "Local-Area Networks Move Beyond the Planning Stage," Systems and Software, Pp. 50-71 (Nov 1982).

ROS 82 Rosner, Roy D. "Packet Switching," Signal, Pp. 110-122 (May/Jun 1982).

RUS 83 Rushby, John and Brian Randell. "A Distributed Secure System," IEEE Computer, Vol 16, No 7: 55-67 (July 1983).

SAA 83 Saal, Harry. "Local Area Networks: An Update on Microcomputers in the Office," Byte, Pp. 60-79 (May 1983).

SALW 83 Salwen, Howard C. "In Praise of Ring Architecture for Local Area Networks," Computer Design, Pp. 183-192 (Mar 1983).

SALT 79 Saltzer, J.H. and Kenneth T. Pogran. "A Star-Shaped Ring Network with High Maintainability," Local Area Communications Symposium, Pp. 179-190.

SALT 81 -----, D.D. Clark, and Kenneth T. Pogran. "Why a Ring?," Proceedings of the Seventh Data Communications Symposium, Pp. 211-217 (Oct 1981).

SAU 81 Sauer, Charles H. and K. Mani Chandy. Computer Systems Performance Modeling. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1981. Pp. 194-282.

SCH 73 Schell, Roger R., Downey, Peter J. and Popek, Gerald J. Preliminary Notes on the Design of Secure Military Computer Systems. Chapter IV: "Secure Military Computing Systems" by Downey the most pertinent of the material. Air Force Systems Command: January 1973.

SCHW 77 Schwartz, Mischa. Computer-Communication Network Design and Analysis. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1977.

SEA 83 Seaman, John. "Data Communications: Beware of the Wireless 'Datanapper'," Computer Decisions, Pp. 54-58 (July 1983).

SHU 81 Shu, Lin and Daniel J. Costello, Jr. "Coding for Reliable Data Transmission and Storage," in Protocols and Techniques for Data Communication Networks, Kuo (Ed.) Pp. 240-318.

SIM 79 Simmons, Gustavus J. "Symmetric and Asymetric Encryption," Computing Surveys Vol 11, No 4: 305-330 (December 1979).

SMI 81 Smid, Miles E. "Integrating the Data Encryption Standard into Computer Networks," IEEE Transactions on Communications, COM-29, No 6: 672-772 (June 1981).

STA 80 Stack, Thomas R. and Kathleen A. Dillincourt. "Protocols for Local Area Computers," IEEE Trends and Applications: Computer Network Protocols. New York: 1980.

STI 83 Stillman, Rona B. Conversation with Dr. Stillman on Internetworking Protocols for the Department of Defense (9 Aug 1983).

STI 80 -----, and Casper R. Defiore. "Computer Security and Network Protocols: Technical Issues in Military Data Communications Networks," IEEE Transactions on Communications, COM-28, No 9: 1472-1477 (September 1980).

STO 80 Stover, Harris A. "Network Timing/ Synchronization for Defense Communications," IEEE Transactions on Communications, COM-28 No 8: 1234-1244 (August 1980).

STU 83 Stuck, Bart W. "Calculating the Maximum Mean Data Rate in Local Area Networks," IEEE Computer, Vol 16, No 5: 72-76 (May 1983).

SUN 81 Sunshine, Carl A. (Editor) Computer Protocol Modeling. Debham, MA: Artech House, Inc., 1981.

TAN 81a Tanenbaum, Andrew S. Computer Networks. Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 1981.

TAN 81b -----. "Network Protocols," Computing Surveys. Vol 13, No 4: 453-489 (Dec 1981).

TEJ 83 Teja, Edward R. "Powerful Local-area-network Controllers Make Networking More Accessible Than Ever," EDN, Pp. 61-66 (3 Mar 1983).

TENE 81 Tenenbaum, Aaron M. and Moshe J. Augenstein. Data Structures Using Pascal. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1981.

TEN 81 Teng, Albert Y. and Ming T. Liu. "The Transmission Grammar Model for Protocol Construction," IEEE Security and Privacy, 1981 Pp. 110-120 New York: 1981.

THO 71 Thomas, Roland E. and Daniel W. Buehler. Signals and Systems: An Introduction to Electrical Engineering, Vol II. (USAFA 71-0287114) US Air Force Academy, CO: USAFA, 1971.

THU 82 Thurber, Kenneth J. Guest Speaker, IEEE Meeting at AFIT/EN (3 Feb 82).

THU 81 -----, and Harvey A. Freeman. (Editors) Tutorial: Local Computer Networks. New York: IEEE Computer Society Press, 1981.

TRO 81 Tropper, Carl. Local Computer Network Technologies. New York: Academic Press, 1981.

USAF 82 USAF/XOKC/ACDS/ACDT. "Local Area Network (LAN) Protocols," Message 232130Z Nov 82.

USAF 83 USAF/XOK/ACD. "Policy on Protocols for Packet-Oriented Local Area Networks," Message 191245Z Apr 83.

WEI 80 Weitzman, Cay. Distributed Micro/Minicomputer Systems: Structure, Implementation, and Application. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1980.

WIL 80 Wilson, Thomas C. and Charles B. Silio, Jr. "Distributed Control of Ring Networks Using a 'Play Through' Protocol," IEEE COMPCON Fall 1980 Pp. 507-576 New York: 1980.

WIT 83 Witt, Michael. "An Introduction to Layered Protocols," BYTE, Vol 8, No 9: 385-398 (September 1983).

WOO 81 Wood, Charles Cresson. "Future Applications of Cryptography," IEEE Security and Privacy, 1981 Pp. 70-74 New York: 1981.

WOL 81 Wolf, Jacob J. and Ming T. Liu. "A Distributed Double-Loop Computer Network (DDLCN)," in Tutorial: Local Computer Networks, Thurber (Ed.) Pp. 148-163.

```
Pascal/MT+ Release 5.5
Copyright (c) 1981 MT MicroSYSTEM, Inc.
Compilation of:  B:WORKG
Stmt    Nest    Source Statement
   1       0
   2       0   {$K1}
   3       0   {$K2}
   4       0   {$K4}
   5       0   {$K7}
   6       0   {$K13}
   7       0   {$K14}
   8       0   {$K15}
   9       0   PROGRAM SLN_SIM (INPUT, OUTPUT):
  10       0   { CONFIG_CONTROL = '04 JULY 1983:  VERSION 2G' }
  11       0   {         IMPLEMENTATION OF A                   }
  12       0   { SECURE LOCAL AREA NETWORK (A SLN)              }
  13       0   { THIS SIMULATION MODEL WAS DEVELOPED TO MEET    }
  14       0   { THESIS REQUIREMENTS FOR THE GCS PROGRAM AT     }
  15       0   { THE AIR FORCE INSTITUTE OF TECHNOLOGY          }
  16       0   { ELECTRICAL ENGINEERING DEPT (AFIT/EN)          }
  17       0   { THIS PROGRAM WAS USED TO VERIFY THE RESULTS    }
  18       0   { DERIVED USING JACKSON'S THEOREM IN THE THESIS}
  19       0
  20       0   {AUTHOR:  RICARDO G. CUADROS, CAPT USAF          }
  21       0   {ADVISOR: WALTER D. SEWARD, MAJOR USAF, PhD      }
  22       0   { PROGRAM DATES:  12 FEB 1982 - 24 JULY 1983     }
  23       0   { ENVIRONMENT:                                   }
  24       0   {         INTERTEC DATA SYSTEMS SUPERBRAIN QD    }
  25       0   {         CP/M 2.2 OPERATING SYSTEM              }
  26       0   {         DIGITAL RESEARCH PASCAL MT+ VER 5.5    }
  27       0   { GENERAL DESCRIPTION:                           }
  28       0   { GENERATE AN EVENT QUEUE SORTED BY TIME         }
  29       0   {  AND INCLUDING NODE AND CLASSIFICATION DATA    }
  30       0   { PROCESS THE EVENT QUEUE TO SIMULATE            }
  31       0   {  TRAFFIC FLOW                                  }
  32       0   { COLLECT TRAFFIC DATA                           }
  33       0   { TRAFFIC FLOW: COUNTER-CLOCKWISE          }
  34       0   {          v <- 3 - 2 - 1 -< ^             }
  35       0   {          -> 4 - 5 - 6 - 7 ->             }
  36       0   { NODES 1, 2, 3 ARE COMMUNICATION NODES    }
  37       0   { NODES 4, 5, 6, 7 ARE APPLICATION NODES   }
  38       0
  39       0   { LIST OF PROCEDURES AND FUNCTIONS    ## }
  40       0   { PROCEDURE INITIAL;                   01 }
  41       0     { PURPOSE:  TO INITIALIZE VARIABLES,    }
  42       0     { ASSIGN FILES, AND TO CONTROL FIRST    }
  43       0     { THREE EVENTS                          }
  44       0
```

```
Stmt   Nest   Source Statement

  45      0   { PROCEDURE GENEVENT (SRC NODE: INTEGER): 02 }
  46      0     { PURPOSE: GIVEN THE NODE, CREATE THE    }
  47      0     {          NEXT EVENT                    }
  48      0     {                                        }
  49      0   { PROCEDURE COMMNODE;                   03 }
  50      0     { PURPOSE: CONTROLS COMM NODE INFO FOR GENEVENT}
  51      0     {                                        }
  52      0     {                                        }
  53      0
  54      0   { PROCEDURE COMMNODE;                   03 }
  55      0     { PURPOSE: GIVEN TH TIME, INSERTS AN EVENT IN  }
  56      0     {          THE PROPER PLACE OF THE EVENT QUEUE }
  57      0     {                                        }
  58      0
  59      0   { PROCEDURE DELEVENT:                    05 }
  60      0     { PURPOSE: DELETES AN EVENT FROM THE HEAD OF   }
  61      0     {          THE EVENT QUEUE               }
  62      0     {                                        }
  63      0
  64      0   { PROCEDURE MOVEVENT;                    06 }
  65      0     { PURPOSE: MOVES EVENTS ABOUT THE MODELED NET; }
  66      0     {          HAS ALGORITHMS FOR COUNTERCLOCKWISE }
  67      0     {          TRAFFIC FLOW; AND SERVES AS TRAFFIC }
  68      0     {          CONTROLLER                    }
  69      0     {                                        }
  70      0
  71      0   { PROCEDURE QWALK;                       07 }
  72      0     { PURPOSE: TO HELP COLLECT QUEUE INFO FOR RUN  }
  73      0     {                                        }
  74      0
  75      0   { PROCEDURE WRAPUP;                      08 }
  76      0     { PURPOSE: RUN TERMINATION CONTROL FOR A NORMAL}
  77      0     {          CLOSE OF FILES AFTER RUN      }
  78      0     {                                        }
  79      0
  80      0   { PROCEDURE UFILREAD;                    09 }
  81      0     { PURPOSE: TO READ FROM THE UNIFORM NUMBER FILE}
  82      0     {                                        }
  83      0
  84      0   { FUNCTION SRC : REAL;                   10 }
  85      0     { PURPOSE: TO PROVIDE ARRIVAL TIME INFORMATION }
  86      0     {                                        }
  87      0
  88      0   { FUNCTION SVC : REAL;                   11 }
  89      0     { PURPOSE: TO PROVIDE SERVICE TIME INFORMATION }
  90      0     {                                        }
  91      0
```

```
Stmt   Nest   Source Statement
  92      0   CONST  { GLOBAL CONSTANTS }
  93      1   CONFIG CONTROL = '04 JULY 1983: VERSION 2G';
  94      1   ARRIVAL_RATE= 0.001; { IN MSG PER MILLISEC FOR   }
  95      1   SERVICE_RATE= 0.003; { ARRIVAL AND SERVICE RATES }
  96      1   COMPLETE    = 'C'    { ALL PKTS FOR THIS MSG RCVD}
  97      1   PARTIAL     = 'P'    { NOT COMPLETE              }
  98      1   LEN1        = 0.500; {LEN# :                     }
  99      1   LEN2        = 0.750;   { GIVES PROBABILITY MSG   }
 100      1   LEN3        = 0.875;   { IS <- #PKTS LONG        }
 101      1   LEN4        = 0.9375;  { (0 REPRESENTS 10 PKTS)  }
 102      1   LEN5        = 0.96875; { THESE VALUES CHOSEN     }
 103      1   LEN6        = 0.984375; { TO MEET REQUIREMENT    }
 104      1   LEN7        = 0.9921875; { THAT MSG BE LEN 1 50% }
 105      1   LEN8        = 0.99609375; { OF THE TIME.         }
 106      1   LEN9        = 0.9990234375;
 107      1   LEN0        = 1.0000000000;
 108      1   EOF_UNIF    = 999.999; {EOF OF UNIFORM_DAT FILE}
 109      1   FIXED_PROCESS_TIME = 0.015;
 110      1
 111      1   TYPE  EVENTPTR = ^EVNTREC;
 112      1         EVNTREC  = RECORD
 113      1       E_TIME   : REAL;    {EVENT TIME; SORT KEY   }
 114      1       AT_NODE  : INTEGER; {CURRENT POS: 10-30, 1-7}
 115      1       TO_NODE  : INTEGER; {INBOUND DEST NODE 4-7}
 116      1       EX_NODE  : INTEGER; {OUTBOUND NODAL SINK 1-3}
 117      1       CLASS    : INTEGER; {CLASS: 1 OR 2 }
 118      1       C_OR_P   : CHAR; {COMPLETE (C) OR PARTIAL (P)}
 119      1       E_NEXT   : EVENTPTR; { NEXT EVENT }
 120      1       END;
 121      1
 122      1 VAR     DFILE         :   TEXT;
 123      1         UFILE         :   TEXT;
 124      1         { WORK ELEMENTS FOR MSGS }
 125      1         WRK_E_TIME    :   REAL;
 126      1         WRK_AT_NODE   :   INTEGER;
 127      1         WRK_TO_NODE   :   INTEGER;
 128      1         WRK_EX_NODE   :   INTEGER;
 129      1         WRK_CLASS     :   INTEGER;
 130      1         WRK_C_OR_P    :   CHAR;
 131      1         WRK_E_NEXT    :   EVENTPTR;
 132      1         { POINTERS }
 133      1         ATPTR, END_PTR : EVENTPTR;
 134      1         HDPTR, TEMP_PTR: EVENTPTR;
 135      1         { TIMES }
 136      1         ELAPS_TM     :    REAL;
 137      1         START_TIME   :    REAL;
 138      1         STOP_TIME    :    REAL;
 139      1         TIME_NOW     :    REAL;
```

```
Stmt   Nest   Source Statement
 140      1   { COUNTERS: INDEX CORRESPONDS TO 'RELATIVE' NODE }
 141      1           CLASS1_CNT      :  REAL;
 142      1           CLASS2_CNT      :  REAL;
 143      1           C_STRTSTP       :  ARRAY [1..7] OF REAL;
 144      1           HI VALUES       :  ARRAY [1..7] OF REAL;
 145      1           MAX-IN BUFFER   :  ARRAY [1..7] OF REAL;
 146      1           MSGS            :  ARRAY [1..7] OF REAL;
 147      1           PCKTS           :  ARRAY [1..7] OF REAL;
 148      1           P_STRTSTP       :  ARRAY [1..7] OF REAL;
 149      1           SMSGS           :  ARRAY [1..7] OF REAL;
 150      1           SPCKTS          :  ARRAY [1..7] OF REAL;
 151      1           { MISC VARIABLES }
 152      1           ERROR LEVEL :  INTEGER;
 153      1           EVENT_Q_LEN :  INTEGER;
 154      1           IO_STATUS   :  INTEGER;
 155      1           LCNT        :  INTEGER;
 156      1           MAX PCKTS   :  INTEGER;
 157      1           MODULE_NAME :  ARRAY [1..12] OF CHAR;
 158      1           PCKT_NUM    :  INTEGER;
 159      1           PCKTS_IN_MSG:  INTEGER;
 160      1           RDT         :  ARRAY [1..20] OF CHAR;
 161      1           SRC_NODE    :  INTEGER;
 162      1           TEMP VAL    :  INTEGER;
 163      1           U_VALUE     :  REAL;
 164      1
 165      1   { * * * PROCEDURES AND FUNCTIONS * * * * * }
 166      1   PROCEDURE INITIAL;
 167      1   VAR LCNT : INTEGER;
 168      2    BEGIN
 169      2     MODULE_NAME := 'INITIAL     ';
 170      2     WRITELN('ENTER REMARKS FOR THIS RUN - 20 CHAR');
 171      2     LCNT := 1;
 172      2     WHILE LCNT <= 19 DO   BEGIN
 173      3       WRITE('_');
 174      3       LCNT :=LCNT + 1
 175      3      END; {  END  WHILE  }
 176      2     WRITELN('*');
 177      2     FOR LCNT := 1 TO 20 DO BEGIN
 178      3         READ(RDT[LCNT])
 179      3         END;
 180      2     READLN;
 181      2     WRITELN('ENTER MAX NUM OF PCKTS PER MSG - INT'):
 182      2     READLN(MAX_PCKTS);
 183      2     IF MAX_PCKTS > 10 THEN MAX PCKTS := 10;
 184      2     WRITELN('ENTER TIME TO STOP RUN - REAL - SEC');
 185      2     READLN(STOP TIME);
 186      2     WRITELN('ENTER DATA COLLECT START TIME
                          - REAL - SEC');
 187      2     READLN(START_TIME);
```

```
Stmt   Nest   Source Statement
 188      2     FOR LCNT := 1 TO 7 DO BEGIN {'0' OUT COUNTERS}
 189      3         PCKTS[LCNT]          := 0.0;
 190      3         HI_VALUES[LCNT]      := 0.0;
 191      3         MSGS[LCNT]           := 0.0;
 192      3         MAX_IN_BUFFER[LCNT]  := 0.0;
 193      3         SMSGS[LCNT]          := 0.0;
 194      3         SPCKTS[LCNT]         := 0.0;
 195      3         C_STRTSTP[LCNT]      := 0.0;
 196      3         P_STRTSTP[LCNT]      := 0.0
 197      3    END;
 198      2    EVENT_Q_LEN := 0;
 199      2    ERROR_LEVEL := 0; {STATUS OK; '9' MARKS PROBLEM }
 200      2    CLASS1_CNT := 0.0;
 201      2    CLASS2_CNT := 0.0;
 202      2    { INITIALIZE QUEUE AND QUEUE POINTERS }
 203      2    NEW(HDPTR);
 204      2    WITH HDPTR^ DO BEGIN
 205      3       E_TIME     := 0.0;
 206      3       AT_NODE    := 0;
 207      3       TO_NODE    := 0;
 208      3       EX_NODE    := 0;
 209      3       CLASS      := 0;
 210      3       C_OR_P     := '0';
 211      3       E_NEXT     := NIL
 212      3    END;
 213      2    ATPTR := HDPTR;
 214      2    END_PTR := HDPTR;
 215      2    TEMP_PTR := HDPTR;
 216      2    WRK_E_TIME     := 0.0;
 217      2    WRK_AT_NODE    := 0;
 218      2    WRK_TO_NODE    := 0;
 219      2    WRK_EX_NODE    := 0;
 220      2    WRK_CLASS      := 0;
 221      2    WRK_C_OR_P     := '0';
 222      2    WRK_E_NEXT     := NIL;
 223      2    ASSIGN(DFILE,'A:RUNDATA.OUT');
 224      2    REWRITE(DFILE);
 225      2    ASSIGN(UFILE,'A:UNIFORM.DAT');
 226      2    RESET(UFILE);
 227      2    WRITELN(DFILE,CONFIG_CONTROL,' REMARKS = ',RDT);
 228      2    WRITELN(DFILE,'START ',START_TIME,' ;STOP ',
                                      STOP_TIME);
 229      2    WRITELN(DFILE,' ARRIVAL ',ARRIVAL_RATE,
                           ;SERVICE ',SERVICE_RATE);
 230      2    WRITELN(DFILE,' MAX PKTS ',MAX_PCKTS);
 231      2    WRITELN(DFILE,'INITIAL    ,ERROR_LEVEL);
 232      2    { GENERATE 1ST 3 ARRIVALS - 1/C NODE }
 233      2    WRITELN(' GENERATING THE FIRST THREE EVENTS ');
 234      2    TIME_NOW  := 0.0;
```

```
Stmt   Nest   Source Statement
 235      2     FOR LCNT := 1 TO 3 DO BEGIN
 236      3      GENEVENT(LCNT)
 237      3      END; { NOW SET TIME TO 1ST ARRIVAL }
 238      2     TIME NOW :- HDPTR^.E TIME
 239      2    END;
 240      1
 241      1   PROCEDURE GENEVENT(VAR SRC_NODE: INTEGER);
 242      1   VAR GLCNT: INTEGER;
 243      2   BEGIN { ALGO IMPLEMENTS FIG. II-5 & 6 OF THESIS }
 244      2   MODULE_NAME := 'GENEVENT    ';
 245      2   WRITELN('IN ',MODULE_NAME,'FOR SRC_NODE= ',
                                          SRC_NODE);
 246      2   WRITELN(DFILE,MODULE_NAME,ERROR_LEVEL,' ',
                                          SRC_NODE);
 247      2   TEMP_VAL := SRC_NODE;
 248      2   IF SRC_NODE < 10 THEN SRC_NODE := SRC_NODE * 10
 249      2      ELSE ERROR_LEVEL := 9;
 250      2   IF ERROR_LEVEL <> 9
 251      2      THEN BEGIN
 252      3      UFILREAD;
 253      3      WRK AT_NODE := SRC_NODE;
 254      3      IF SRC NODE < 40 THEN WRK EX_NODE := TEMP_VAL;
 255      3      IF SRC_NODE < 40 THEN COMMNODE
 256      3         ELSE { SRC NODE > 30 }
 257      3         WRK_E_TIME := TIME NOW + SVC;
                          { RESPONSE AT APPL }
 258      3      UFILREAD;
 259      3      IF U-VALUE <= LEN9 THEN PCKTS IN MSG:=9;
 260      3      IF U-VALUE <= LEN8 THEN PCKTS IN MSG:=8;
 261      3      IF U-VALUE <= LEN7 THEN PCKTS IN MSG:=7;
 262      3      IF U-VALUE <= LEN6 THEN PCKTS IN MSG:=6;
 263      3      IF U-VALUE <= LEN5 THEN PCKTS IN MSG:=5;
 264      3      IF U-VALUE <= LEN4 THEN PCKTS IN MSG:=4;
 265      3      IF U-VALUE <= LEN3 THEN PCKTS IN MSG:=3;
 266      3      IF U-VALUE <= LEN2 THEN PCKTS IN MSG:=2;
 267      3      IF U-VALUE <= LEN1 THEN PCKTS IN MSG:=1
 268      3         ELSE PCKTS_IN_MSG := 10;
 269      3      IF PCKTS IN_MSG > MAX PCKTS THEN
 270      3         PCKTS IN_MSG := MAX PCKTS;
 271      3      WRK_C_OR_P := PARTIAL;
 272      3      FOR GLCNT := 1 TO PCKTS IN MSG DO BEGIN
 273      4          IF GLCNT = PCKTS_IN-MSG
                       THEN WRK_C_OR P := COMPLETE;
 274      4          INSRT(WRK_E TIME)
 275      4          END { FOR }
 276      4      END; { IF ERROR_LEEL <> 9 }
 277      2   WRITELIN('BYE ',MODULE_NAME);
 278      2   SRC_NODE := TEMP_VL
             { SETS SRC_NODE TO ORIGINAL CALLING PARAM }
 279      2   END; {GENEVENT}
 280      1
```

```
Stmt   Nest   Source Statement
 281      1   PROCEDURE COMMNODE;
 282      1   BEGIN
 283      2     MODULE NAME := 'COMMNODE      ';
 284      2     WRITELN(DFILE,MODULE NAME,ERROR LEVEL,' ',
                                                SRC_NODE);
 285      2     WRK_E_TIME := TIME_NOW + SRC;
 286      2     WRK_CLASS := 1;
 287      2     IF (SRC NODE <> 20) AND (U_VALUE < 0.50)
 288      2        THEN WRK_CLASS := 2;
 289      2     IF WRK_CLASS = 1 THEN BEGIN
 290      3        WRK_TO_NODE := 4;
 291      3        CLASS1 CNT := CLASS1_CNT + 1.0
 292      3        END;
 293      3     IF WRK_CLASS = 2
 294      2        THEN BEGIN
 295      3        CLASS2 CNT := CLASS2_CNT + 1.0;
 296      3        WRK_TO_NODE := 7
 297      3        END;
 298      2     IF ((WRK CLASS = 2) AND (U_VALUE < 0.66666667))
 299      2        THEN WRK_TO_NODE := 6;
 300      2     IF ((WRK_CLASS = 2) AND (U_VALUE < 0.33333333))
 301      2        THEN WRK_TO_NODE := 5
 302      2   END;  { COMM NODE }
 303      1
 304      1   PROCEDURE INSRT (VAR TTIME; REAL);
 305      1   BEGIN { LINK-LIST IN ASC ORDER BY E_TIME }
 306      2   MODULE_NAME := 'INSRT        ';
 307      2   WRITELN(DFILE,MODULE_NAME,ERROR_LEVEL,' ',TTIME);
 308      2   WRITELN(MODULE_NAME,ERROR_LEVEL,' ',TTIME);
 309      2   EVENT_Q_LEN := EVENT_Q LEN + 1;
 310      2   WITH HDPTR^ DO BEGIN
                    { KEEP TRACK OF MAX PCKTS IN BUFFER }
 311      3   IF ((AT NODE > 0) AND (AT_NODE < 10)) THEN
 312      3      BEGIN
 313      4       HI_VALUES[AT_NODE] := HI_VALUES[AT NODE] + 1.0;
 314      4       IF HI_VALUES[AT_NODE] < MAX_IN_BUFFER[AT_NODE]
                    THEN
 315      4          MAX IN_BUFFER[AT_NODE] := HI_VALUES[AT_NODE]
 316      4      END
 317      4   END; { WITH }
 318      2   IF (HDPTR^.E_TIME = 0.0) THEN
 319      2    BEGIN { LIST EMPTY }
 320      3     WITH HDPTR^ DO BEGIN
 321      4      E_TIME     := WRK_E_TIME;
 322      4      AT_NODE    := WRK_AT_NODE;
 323      4      TO NODE    := WRK_TO NODE;
 324      4      EX_NODE    := WRK_EX_NODE;
 325      4      CLASS      := WRK_CLASS;
 326      4      C_OR P     := WRK_C_OR_P;
 327      4      E_NEXT     := NIL
 328      4    END
```

```
Stmt   Nest   Source Statement
 329      4    END
 330      3   ELSE
 331      2    IF TTIME < HDPTR^.E_TIME THEN
 332      2     BEGIN { INSERT AT HEAD OF LIST }
 333      3      NEW(TEMP PTR);
 334      3      WITH TEMP_PTR^ DO BEGIN
 335      4      E TIME     :=  WRK_E_TIME;
 336      4      AT NODE    :=  WRK_AT_NODE;
 337      4      TO_NODE    :=  WRK_TO_NODE;
 338      4      EX_NODE    :=  WRK_EX_NODE;
 339      4      CLASS      :=  WRK_CLASS;
 340      4      C_OR_P     :=  WRK_C_OR_P;
 341      4      E_NEXT     :=  HDPTR
 342      4      END;
 343      3     HDPTR := TEMP_PTR
 344      3    END
 345      3    ELSE BEGIN { INSERT AFTER START OF THE LIST }
 346      3     ATPTR := HDPTR;
 347      3     WHILE TTIME >= ATPTR^.E_NEXT^.E_TIME DO
 348      3      ATPTR := ATPTR^.E_NEXT; { END WHILE }
 349      3     NEW(TEMP_PTR);
 350      3     WITH TEMP_PTR^ DO BEGIN
 351      4     E TIME      :=  WRK_E_TIME;
 352      4     AT_NODE     :=  WRK_AT_NODE;
 353      4     TO NODE     :=  WRK_TO_NODE;
 354      4     EX_NODE     :=  WRK_EX NODE;
 355      4     CLASS       :=  WRK CLASS;
 356      4     C OR_P      :=  WRK_C_OR_P;
 357      4     E_NEXT      :=  ATPTR^.E_NEXT
 358      4     END;
 359      3    IF TTIME >= END_PTR^.E_TIME
                 THEN END_PTR := TEMP PTR;
 360      3    ATPTR^.E NEXT := TEMP_PTR
 361      3    END
 362      3   END;  {INSRT}
 363      1
 364      1   PROCEDURE DELEVENT;
 365      1   BEGIN
 366      2   {SHOULD ONLY BE DELETING FROM THE HEAD OF THE LIST}
 367      2   MODULE_NAME := 'DELEVENT    ';
 368      2   WRITELN(DFILE,MODULE_NAME,ERROR_LEVEL);
 369      2   IF ((HDPTR^.AT_NODE > 0) AND (HDPTR^.AT NODE < 10))
 370      2      THEN HI_VALUES[HDPTR^.AT_NODE] - 1.0;
 371      2   IF HDPTR^.E_NEXT = NIL THEN BEGIN
 372      3      HDPTR^.AT_NODE := 0;
 373      3      HDPTR^.AT_TIME := 0.0
 374      3      END
```

```
Stmt   Nest   Source Statement
 375      3   ELSE BEGIN
 376      3        ATPTR := HDPTR^.E NEXT;
 377      3        DISPOSE(HDPTR);
 378      3        HDPTR := ATPTR
 379      3        END;
 380      2   EVENT Q_LEN := EVENT Q_LEN - 1
 381      1   END;  {DELEVENT}
 382      1
 383      1   PROCEDURE MOVEVENT;
 384      1   VAR LCNT : INTEGER;
 385      2   BEGIN
 386      2   { CHECK FOR ARRIVAL AT COMM TO GENERATE NEW ONE }
 387      2     MODULE_NAME := 'MOVEMENT    ';
 388      2     WRITELN(DFILE,MODULE_NAME,ERROR_LEVEL,' ',
                                          HDPTR^.AT_NODE);
 389      2     WRITELN(MODULE_NAME,ERROR_LEVEL,' ',
                                          HDPTR^.AT_NODE);
 390      2   LCNT := 0;
 391      2   CASE HDPTR^.AT_NODE OF
 392      2        10 : LCNT := 1;
 393      3        20 : LCNT := 2;
 394      3        30 : LCNT := 3
 395      3        END;
 396      2   WRITELN(MODULE_NAME,ERROR_LEVEL,' ',LCNT);
 397      2   IF LCNT <> 0 THEN GENEVENT(LCNT);
 398      2
 399      2   IF ((TIME_NOW < STOP_TIME) AND
                  (TIME_NOW >= START_TIME))
 400      2      THEN BEGIN
 401      3      TEMP VAL := HDPTR^.AT_NODE;
 402      3      IF TEMP VAL >= 10
 403      3         THEN BEGIN
 404      4         TEMP_VAL := (TEMP_VAL DIV 10);
 405      4         PCKTS[TEM_VAL] := PCKTS[TEM_VAL] + 1.0;
 406      4         IF (HDPTR^.C_OR_P = COMPLETE) THEN
 407      4            MSGS[TEM_VAL] := MSGS[TEM_VAL] + 1.0
 408      4         END
 409      4     END;
 410      2
 411      2   WITH HDPTR^ DO BEGIN
 412      3   { MOVE TO NEXT NODE }
 413      3   IF ((AT_NODE = 7) OR (AT_NODE = 70))
 414      3      THEN AT_NODE := 1
 415      3   ELSE
 416      3      IF ((AT_NODE > 0) AND (AT_NODE < 7 ))
 417      3         THEN AT_NODE := AT_NODE + 1;
 418      3   IF (AT_NODE > 9) AND (AT_NODE < 70 ))
 419      3      THEN AT_NODE := ((AT_NODE + 10) DIV 10)
 420      3   END; { WITH }
 421      2
```

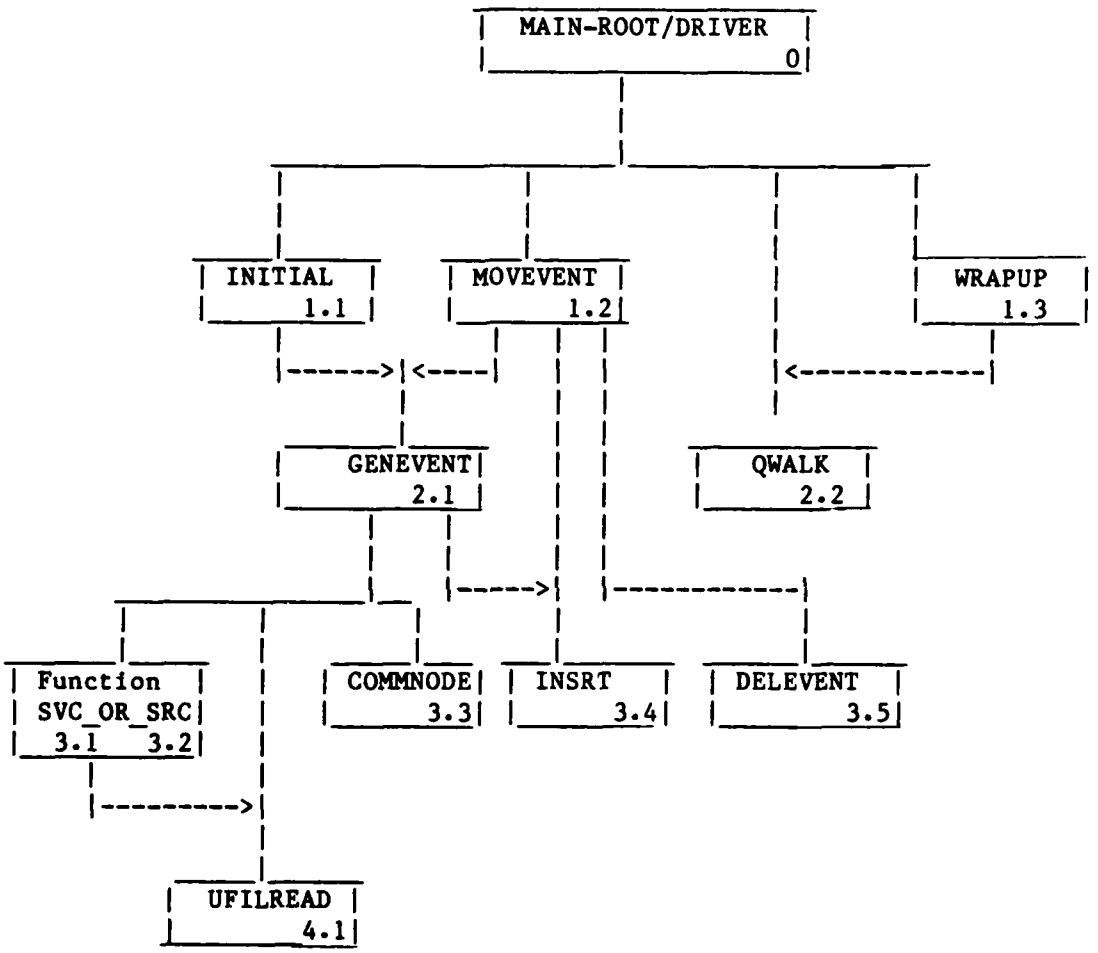
```
Stmt   Nest   Source Statement
 422      2    IF HDPTR^.AT_NODE <> HDPTR^.TO_NODE
 423      2      THEN { THAT ENTRY AND CREATE A NEW ONE }
 424      2      BEGIN
 425      3        WRK_E TIME     := HDPTR^.E TIME +
                                     FIXED_PROCESS_TIME;
 426      3        WRK_AT_NODE    :=  HDPTR^.AT_NODE;
 427      3        WRK_TO NODE    :=  HDPTR^.TO_NODE;
 428      3        WRK_EX_NODE    :=  HDPTR^.EX_NODE;
 429      3        WRK CLASS      :=  HDPTR^.CLASS;
 430      3        WRK_C_OR_P     :=  HDPTR^.C_OR_P;
 431      3        INSRT(WRK_E_TIME)
 432      3      END { <> }
 433      3    ELSE
 434      2      IF HDPTR^.AT_NODE = HDPTR^.TO NODE
 435      2         THEN { ARRIVED TO APPLICATION SINK }
 436      2         BEGIN
 437      3         IF HDPTR^.C_OR_P = COMPLETE THEN
 438      3            BEGIN
 439      3              WRK_E_TIME    :=  HDPTR^.E TIME;
 440      4              WRK_AT_NODE   :=  HDPTR^.AT_NODE;
 441      4              WRK_TO_NODE   :=  HDPTR^.EX_NODE;
 442      4              WRK_EX_NODE   :=  HDPTR^.EX_NODE;
 443      3              WRK_CLASS     :=  HDPTR^.CLASS;
 444      4              GENEVENT(WRK_AT NODE)
 445      4            END { COMPLETE }
 446      4         END; { = APPLICATION NODE ARRIVAL }
 447      2
 448      2    IF ((TIME_NOW < STOP_TIME) AND
                   (TIME_NOW >= START_TIME))
 449      2       THEN BEGIN
 450      3       IF ((HDPTR^.AT_NODE = HDPTR^.EX_NODE) OR
 451      3          (HDPTR^.AT NODE = HDPTR^.TO_NODE))
 452      3          THEN BEGIN
 453      4          SPCKTS[HDPTR^.AT_NODE] :=
                    SPCKTS[HDPTR^.AT_NODE] + 1.0;
 454      4          IF HDPTR^.C_OR_P = COMPLETE THEN
 455      4             SMSGS[HDPTR^.AT_NODE] :=
                       SMSGS[HDPTR^.AT_NODE] + 1.0
 456      4          END
 457      4        END;
 458      2
 459      2    IF ((HDPTR^.AT_NODE = HDPTR^.EX NODE) OR
 460      2       ((HDPTR^.AT NODE = HDPTR^.TO_NODE) OR
 461      2       ((HDPTR^.AT_NODE <> HDPTR^.TO_NODE))
 463      2      THEN DELEVENT
 463      2    ELSE ERROR_LEVEL := 9;
 464      2
 465      2    TIME_NOW := HDPTR^.E TIME
 466      2    END; { MOVEVENT}
 467      1
```

```
Stmt   Nest   Source Statement
 468      1   PROCEDURE QWALK;
 469      1   VAR LCNT : INTEGER;
 470      2   BEGIN
 471      2     MODULE NAME := 'QWALK       ';
 472      2     WRITELN(MODULE_NAME,ERROR
 473      2     WRITELN(DFILE,MODULE_NAME,ERROR_LEVEL);
 474      2     ATPTR := HDPTR;
 475      2     LCNT := 0;
 476      2     WHILE ATPTR^.E_NEXT <> NIL DO
 477      2      BEGIN
 478      3       LCNT := LCNT + 1;
 479      3       WITH ATPTR^ DO;
 480      3       BEGIN
 481      4       IF ((AT_NODE > 0 ) AND (AT_NODE < 10)) THEN
 482      4          IF (C_OR_P = COM LETE) THEN
 483      4                C_STRTSTP[AT NODE] :=
                          C STRTPSTP[AT NODE] + 1.0
 484      4          ELSE  P STRTPSTP[AT_NODE] :=
                          P_STRTPSTP[AT_NODE] + 1.0
 485      4       END { WITH }
 486      4      END; { WHILE <> NIL }
 487      2     WRITELN(DFILE,'LCNT = ',LCNT,' Q_LEN = ',
                                      EVENT_Q_LEN);
 488      2     FOR LCNT := 1 TO 7 DO BEGIN
 489      3         HI_VALUES[LCNT] := P-STRTSTP[LCNT];
 490      3         MAX_IN_BUFFER[LCNT] : HI VALUES[LCNT]
 491      3     END { FOR }
 492      3     END;   { QWALK }
 493      1
 494      1     PROCEDURE WRAPUP;
 495      1     VAR LCNT : INTEGER;
 496      2     BEGIN
 497      2     { WRITE OUT TO DFILE THE SIM DATA DESIRED }
 498      2     QWALK;
 499      2     ELAPS TM := TIME_NOW - START_TIME;
 500      2     WRITELN(DFILE,'ERROR_LEVEL = ',ERROR LEVEL);
 501      2     WRITELN(DFILE,'DATA COLLECTED FOR ',ELAPS_TM,
                              SEC; TIME NOW = ',TIME_NOW);
 502      2     FOR LCNT := 1 TO 7 DO BEGIN
 503      3      WRITELN('IN WRAPUP AT NODE # ',LCNT);
 504      3      WRITELN(DFILE,'AT NODE # ',LCNT);
 505      3      WRITELN(DFILE,'STOP STATUS: MSGS = ',
                       C_STRTSTP[LCNT]);
 506      3      WRITELN(DFILE,'             PCKTS = ',
                       P_STRTSTP[LCNT]);
 507      3      WRITELN(DFILE,'MSGS  GENERATED = ',MSGS[LCNT];
 508      3      WRITELN(DFILE,'PCKTS GENERATED = ',PCKTS[LCNT];
 509      3      WRITELN(DFILE,'BUFFER USED     = ',
                       MAX_IN BUFFER[LCNT])
 510      3      END;
```

```
Stmt   Nest   Source Statement
 511      2   WRITELN(DFILE,'EVENT QUEUE LEN AT STOP TIME = ',
                          EVENT_Q_LEN);
 512      2   CLOSE (UFILE,IO_STATUS);
 513      2   IF IO_STATUS = 255
                 THEN WRITELN('ERROR IN UFILE CLOSURE')
 514      2      ELSE WRITELN('UFILE CLOSED');
 515      2   CLOSE (DFILE,IO STATUS);
 516      2   IF IO_STATUS = 255
                 THEN WRITELN('ERROR IN DFILE CLOSURE')
 517      2      ELSE WRITELN('DFILE CLOSED')
 518      2   END;  {WRAPUP}
 519      1
 520      2   PROCEDURE UFILREAD;
 521      1   BEGIN
 522      2     MODULE_NAME := 'UFILREAD     ';
 523      2     WRITELN('* * *ENTERING ',MODULE_NAME);
 524      2     READ(UFILE,U_VALUE);
 525      2     IF U_VALUE = EOF_UNIF THEN BEGIN
 526      3        RESET (UFILE);
 527      3        READ(UFILE,U_VALUE)
 528      3        END; { IF }
 529      2     WRITELN('* * * * *EXITING ',MODULE_NAME);
 530      2     WRITELN(DFILE,MODULE_NAME,ERROR_LEVEL,
                              ' U_VALUE := ',U VALUE)
 531      2   END;
 532      1
 533      1   FUNCTION SRC : REAL;
 534      1   VAR INT_RESULT: REAL; { SRC/COMM NODE ARRIVALS }
 535      2   BEGIN { RETS VALUE FROM EXPONENTIAL DIST. }
 536      2   UFILREAD;
 537      2   INT_RESULT := -((ARRIVAL_RATE)*(LN(1.0 - U_VALUE)));
 538      2   IF INT_RESULT <= 0.0 THEN BEGIN
 539      3          WRITELN('****ERROR IN SOURCE ***');
 540      3          ERROR_LEVEL := 9
 541      3      END
 542      3   WRITELN(DFILE,'SRC READ ',INT_RESULT,' ' ,ERROR LEVEL)
 544      2   END;  { END OF SRC }
 545      1
 546      1   FUNCTION SVC : REAL;
 547      1   VAR INT_RESULT: REAL; { SERVICE RATE W/SKEW-TIME}
 548      2   BEGIN { RETS VALUE FROM EXPONENTIAL DIST. }
 549      2   UFILREAD;
 550      2   INT RESULT := -((SERVICE RATE)*(LN(1.0 - U_VALUE)));
 551      2   IF INT_RESULT <= 0.0 THEN BEGIN
 552      3          WRITELN('*** ERROR IN SERVICE ***');
 553      3          ERROR_LEVEL := 9
 554      3      END
 555      3      ELSE SVC := INT_RESULT + FIXED PROCESS_TIME;
 556      2   WRITELN(DFILE,'SVC READ ',INT_RESULT,' ',ERROR_LEVEL)
 557      2   END;  { END OF SVC }
```

```
Stmt   Nest   Source Statement
 558      1
 559      1   { * * * * * * * * * * * * * * * * * * * * * * * * * }
 560      1   BEGIN { M  A I N - D R I V E R }
 561      1   INITIAL;
 562      1   WRITELN('ERROR LEVEL := ',ERROR_LEVEL,
                                    ' AFTER INITIAL');
 563      1   WRITELN(DFILE,'ERROR LEVEL := ',ERROR_LEVEL,
                                    ' AFTER INITIAL');
 564      1   WRITELN('* * * * * ** * * * * * * *  MAIN1');
 565      1   IF ERROR_LEVEL = 9 THEN TIME NOW := 9.60E+15;
 566      1
 567      1   WHILE (TIME NOW < START_TIME) DO
 568      1     WHILE (TIME_NOW < HDPTR^.E_TIME) DO
 569      1     BEGIN
 570      2      MOVEVENT;
 571      2      IF ERROR LEVEL  9 THEN TIME NOW := 9.60E+15
 572      2     END; { TIME_NOW < HDPTR^.E-TIME }
 573      1   { END WHILE TIME_NOW < START_TIME }
 574      1
 575      1   WRITELN(' IN MAIN AFTER SET-UP; ERROR LEVEL = '
                                              ,ERROR_LEVEL);
 576      1   WRITELN(DFILE,'IN MAIN AFTER SET-UP;ERROR LEVEL = ',
                    ERROR_LEVEL);
 577      1   WRITELN('* * * * * * * * * * * * * * *  MAIN2');
 578      1
 579      1   IF TIME_NOW <> 9.60E+15 THEN BEGIN
 580      2   QWALK;
 581      2   WRITELN(DFILE,'START TIME STATUS: ');
 582      2   FOR LCNT := 1 TO 7 DO BEGIN
 583      3    WRITELN(DFILE,'AT NODE # ',LCNT);
 584      3    WRITELN(DFILE,' MSGS: ',C-STRTSTP[LCNT],
                           ' PCKTS: ',P_STRTSTP[LCNT])
 585      3   END { FOR LOOP }
 586      3   END; { TIME_NOW <> 9.60E+15 }
 587      1
 588      1   WRITELN('IN MAIN READY TO START UP ',ERROR LEVEL);
 589      1   WRITELN(DFILE,'IN MAIN READY TO START UP ',
                                          ERROR_LEVEL);
 590      1   WRITELN('* * * * * * * * * * * * * * *  MAIN3');
 591      1
 592      1   WHILE (TIME_NOW < STOP_TIME) DO
 593      1      WHILE (TIME NOW < HDPTR^.E_TIME) DO MOVEVENT;
 594      1   { END WHILE TIME_NOW < STOP_TIME }
 595      1
 596      1   WRAPUP;
 597      1   WRITELN(' DONE ')
 598      1   END .   { END OF THE PROGRAM }
 598      0   --------------------------
 598      0   Normal End of Input Reached
```

Appendix B: Structure Chart



```
TRAFFIC FLOW:  COUNTER-CLOCKWISE
               V < - 3 - 2 - 1 --< ^
               - > 4 - 5 - 6 - 7 ->|
               NODES 1, 2, 3 ARE COMMUNICATION NODES
               NODES 4, 5, 6, 7 ARE APPLICATION NODES

PROCEDURES AND FUNCTIONS                                    #.#
1.   PROCEDURE INITIAL:                                     1.1
     PURPOSE:  TO INITIALIZE VARIABLES, ASSIGN
     FILES, AND TO CONTROL 1ST 3 EVENTS
2.   PROCEDURE GENEVENT(SRC NODE: INTEGER);                2.1
     PURPOSE: GIVEN THE NODE, CREATE THE NEXT EVENT
3.   PROCEDURE COMMNODE:                                   3.3
     PURPOSE: CONTROLS COMM NODE INFO FOR GENEVENT
4.   PROCEDURE INSRT((TIME: REAL);                         3.4
     PURPOSE: GIVEN TH TIME, INSERTS AN EVENT IN
              THE PROPER PLACE OF THE EVENT QUEUE
5.   PROCEDURE DELEVENT:                                   5.5
     PURPOSE:  DELETES AN EVENT FROM THE HEAD OF
              THE EVENT QUEUE
6.   PROCEDURE MOVEVENT:                                   1.2
     PURPOSE: MOVES EVENTS ABOUT THE MODELLED NET;
              HAS ALGORITHMS FOR COUNTERCLOCKWISE
              TRAFFIC FLOW; AND SERVES AS TRAFFIC
              CONTROLLER
7.   PROCEDURE QWALK:                                      2.2
     PURPOSE: TO HELP COLLECT QUEUE INFO FOR RUN
8.   PROCEDURE WRAPUP;                                     1.3
     PURPOSE: RUN TERMINATION CONTROL FOR A NORMAL
              CLOSE OF FILES AFTER RUN
9.   PROCEDURE UFILREAD;                                   4.1
     PURPOSE: TO READ FROM THE UNIFORM NUMBER FILE
10.  FUNCTION SRC : REAL;                                  3.1
     PURPOSE: TO PROVIDE ARRIVAL TIME INFORMATION
11.  FUNCTION SVC : REAL;                                  3.2
     PURPOSE: TO PROVIDE SERVICE TIME INFORMATION
```

```
CONSTANT
  GLOBAL
   ARRIVAL_RATE= 0.0001;      { IN MSG PER MILLISEC FOR    }
   COMPLETE    = 'C'          { ALL PKTS FOR THIS MSG RCVD }
   CONFIG_CONTROL = LITERAL ALTERED BY MANUALLY TO TRACK
                    PROGRAM VERSION
   EOF_UNIF       = 999.999; { EOF OF UNIFORM_DAT FILE    }
   FIXED_PROCESS_TIME = 0.015;
```

```
 LEN1           = 0.500;       { LEN# :                          }
 LEN2           = 0.750;       {   GIVES PROBABILITY MSG         }
 LEN3           = 0.875;       {   IS <= #PKTS LONG              }
 LEN4           = 0.9375;      {   (0 REPRESENTS 10 PKTS         }
 LEN5           = 0.96875;     {   THESE VALUES CHOSEN           }
 LEN6           = 0.984375;    {   TO MEET REQUIREMENT THAT      }
 LEN7           = 0.9921875;   {   MSG BE LEN 1 50% OF TIME.     }
 LEN8           = 0.99609375;
 LEN9           = 0.9990234375;
 LENO           = 1.0000000000;
 PARTIAL        = 'P';         { NOT COMPLETE                    }
 SERVICE_RATE = 0.003;         { ARRIVAL AND SERVICE RATES       }

TYPE  EVENTPTR = ^EVNTREC;
      EVENTREC = RECORD
       E_TIME       :   REAL;     { EVENT TIME; SORT KEY    }
       AT_NODE      :   INTEGER;  { CURRENT POSITION: 10-30, 1-7}
       TO_NODE      :   INTEGER;  { INBOUND DESTINATION NODE 4-7}
       EX_NODE      :   INTEGER;  { OUTBOUND NODAL SINK 1-3}
       CLASS        :   INTEGER;  { CLASSIFICATION: 1 OR 2 }
       C_OR P       :   CHAR;     { COMPLETE (C) OR PARTIAL (P)}
       E_NEXT       :   EVENTPTR; { NEXT RECORD/EVENT }
     END;

VARIABLES
  COUNTERS:  INDEX CORRESONDS TO 'RELATIVE' NODE
       CLASS1_CNT       :  REAL; {NUM MESSAGES ENTERING THE }
       CLASS2 CNT       :  REAL; { NETWORK FOR A GIVEN CLASS}
             {ARRAYS TO STORE NODAL INFO:}
       C STRTSTP        :  ARRAY [1..7] OF REAL; {COMPLETE MSGS}
       HI_VALUES        :  ARRAY [1..7] OF REAL; {TEMP FOR MAX}
       MAX_IN_BUFFER    :  ARRAY [1..7] OF REAL; {MAX PCKTS}
       MSGS             :  ARRAY [1..7] OF REAL; {TOTAL SEEN}
       PCKTS            :  ARRAY [1..7] OF REAL; {TOTAL SEEN}
       P STRTSTP        :  ARRAY [1..7] OF REAL; {PARTIAL MSGS}
       SMSGS            :  ARRAY [1..7] OF REAL; {MSGS FROM A}
       SPCKTS           :  ARRAY [1..7] OF REAL; {PCKTS FROM A}

  FILES
       DFILE          :    TEXT; {STATISTICS/DEBUF FILE}
       UFILE          :    TEXT; {UNIF-RAND FILE}

  MISC VARIABLES
       ERROR_LEVEL :       INTEGER;  { 0 = OK; 9 = ABORT RUN }
       EVENT_Q_LEN :       INTEGER;  {TO DETERMINE MAX_IN_BUFFER}
       IO_STATUS   :       INTEGER;  { USED IN CLOSE CMD }
       LCNT        :       INTEGER;  { GENERAL PURPOSE COUNTER }
       MAX PCKTS   :       INTEGER;  { LIMITS MSG LEN }
       MODULE_NAME :       ARRAY [1..12] OF CHAR;  { DEBUG RMKS }
       PCKT_NUM    :       INTEGER;  { USED IN MSG GENERATION }
       PCKTS IN_MSG:       INTEGER;  { USED IN MSG GENERATION }
       RDT         :       ARRAY [1..20] OF CHAR;  { RUN REMARKS }
```

```
        SRC_NODE     :      INTEGER;  { USED IN MSG GENERATION }
        TEM VAL      :      INTEGER;  { GENERAL PURPOSE TEMP HOLD }
        U_VALUE      :      REAL;     { RESULT OF READ FROM UFILE }

  POINTERS
        ATPTR, END_PTR:     EVENTPTR;
        HDPTR, TEM_PTR:     EVENTPTR;

  TIMES
        ELAPS_TM     :      REAL; { ELAPSED TIME }
        START_TIME   :      REAL; { START DATA COLLECTION }
        STOP TIME    :      REAL; { STOP DATA COLLECTION }
        TIME_NOW     :      REAL; { CURRENT SIMULATION CLOCK TIME }

WORK ELEMENTS FOR MESSAGES
        WRK_E_TIME   :      REAL;
        WRK_AT NODE  :      INTEGER; { CURRENT POSITION: 10-30, 1-7}
        WRK TO_NODE  :      INTEGER; { INBOUND DESTINATION NODE 4-7}
        WRK_EX_NODE  :      INTEGER; { OUTBOUND NODAL SINK 1-3}
        WRK_CLASS    :      INTEGER; { CLASSIFICATION: 1 OR 2 }
        WRK C_OR_P   :      CHAR;    { COMPLETE (C) OR PARTIAL (P)}
        WRK_E_NEXT   :      EVENTPTR;
```

VITA

Ricardo Gerardo Cuadros was born on 10 March 1951 in Santurce, Puerto Rico. In 1969, he graduated from St. John's High School. He earned his B.S. (with honors) from the U.S. Air Force Academy in 1973. In August 1973, he attended Minuteman Missile Combat Crew training at Vandenberg AFB, California enroute to his first assignment to the 510th Strategic Missile Squadron (351st Strategic Missile Wing), Whiteman AFB, Missouri. While at Whiteman, he earned a Master's in Business Administration (University of Missouri, 1977). He attended Squadron Officer's School (Class 77D) at Maxwell AFB, Alabama and received training in programming at Keesler AFB, Mississippi enroute to a tour at Headquarter's Strategic Air Command (SAC/ADW, 1978-1981). While serving as System's Analyst, JCS War Plans, he was selected to attend the AFIT School of Engineering. He went in June 1981. While at AFIT, he completed Air Command and Staff. In January 1983, he arrived at the Electronic Security Command (ESC). He has served as chief, Systems Engineering Division (ESC/ADTE) and is presently chief, Systems Software Division (ESC/ADTS). While at ESC, he completed his AFIT thesis.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GCS/EE/83D-6	2. GOVT ACCESSION NO. AD-A138079	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DESIGN OF A SECURE LOCAL NETWORK		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ricardo G. Cuadros Capt USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DCS/Computer Resources HQ Electronic Security Command San Antonio, Texas 78243		12. REPORT DATE December, 1983 13. NUMBER OF PAGES 153
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

FREDRIC C. LYNCH, Major, USAF
Director, Public Affairs

Approved for public release: IAW AFR 190-17
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

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

Secure Networks
Network Design
Local Area Network Design
Jackson's Theorem

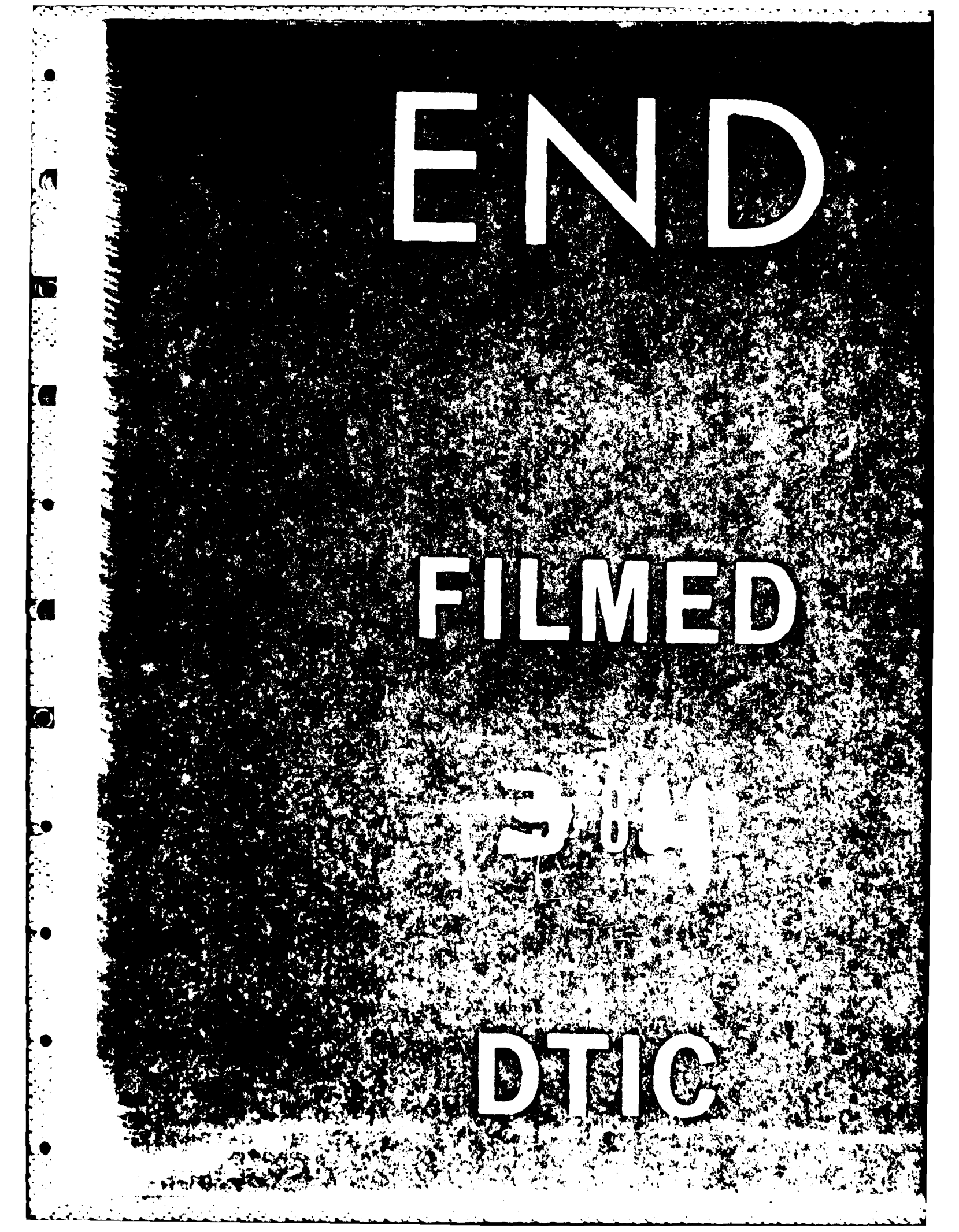
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research sponsored by the USAF's HQ ESC/AD develops a multi-level secure host-to-host computer local area network. The design process is presented. The resulting network uses a ring topology with packetized point-to-point switching over fiber optics communication links. For transmission security, packets are source host-to-destination host encrypted as well as encapsulated with link-to-link encryption. Message

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

transmission is controlled with message acknowledgements and credits within a non-preemptive three priority class queue. A simplified version of the resulting network was validated by applying Jackson's Theorem. Additionally, the simplified view was modeled with a PASCAL simulation program executed on a 64K microcomputer. Unfortunately, the comparison of the simulation against the analytical results that were obtained using Jackson's Theorem was not possible due to problems modeling the network on the microcomputer. Follow-on work in the area of simulation is needed to successfully complete the simulation and compare results.


END
FILMED
DTIC