1988

Performance investigation of a document retrieval system on a voice-data integrated token ring local area network

Sunil Sharadchandra Gaitonde
Iowa State University

Order Number 8825394

Performance investigation of a document retrieval system on a voice-data integrated token ring local area network

Gaitonde, Sunil Sharadchandra, Ph.D.

Iowa State University, 1988

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Performance investigation of a document retrieval system on a voice-data integrated token ring local area network

by

Sunil Sharadchandra Gaitonde

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Electrical Engineering and Computer Engineering

Major: Computer Engineering

Approved:

Members of the Committee:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Signature was redacted for privacy.

Iowa State University
Ames, Iowa
1988

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Over the past decade or two local area networks have received considerable attention. Local area networks (LANs) generally have four distinctive characteristics [54].

1. A geographical diameter of less than a few kilometers.
2. A total data rate exceeding 1 Mbps/sec.
3. Ownership by a single organization.
4. Small transmission error rate (10^{-8} to 10^{-11}).

The interest in LANs is primarily motivated by two reasons. The first is to connect together a collection of computers, terminals, and peripherals located in the same or in adjacent buildings, allowing them to communicate with each other and remote hosts. The other reason is to exploit the advantages of functionally distributed computing. Specifically, some of the machines are dedicated to perform specific functions, such as file storage, data base management, thus making implementation simpler and, probably, more efficient.

The motivations for considering integrated voice and data traffic in a shared network environment include: the advent of new voice related applications with the technology now existing to support them, and the desire to plan for and design future integrated networks for reasons of economy and flexibility [20].

Table 1.1: Traffic Characteristics

	Voice	Data	Documents
General nature	on-off type, 60-65 % idle time	interactive or bulk	on-off type no stats available
Bandwidth requirements	$\approx$ 64 kbps	$\approx$ 1-2 Mbps	$\approx$ 2-5 Mbps
Error control	None	Essential	Medium
Packet rate	None during silence, constant otherwise	stochastic	None during idle, constant otherwise
Delay requirements	Constant delay with an upper bound	stochastic, but low delay for interactive traffic	Stochastic, a reasonable delay can be tolerated
Blocking probability	1-2 % can be tolerated	None	None

Extensive research has been done on integrating voice and data (both interactive and bulk). In this dissertation we will focus our attention on transmitting text images (documents) along with voice and data.

This chapter is organized as follows. Section 1.1 compares the three traffic types of interest. Section 1.2 states the problem followed by the reasons to solve it in Section 1.3. In Section 1.4 we will have a look at CBX as a possible solution. Sections 1.5 to 1.7 discuss the standards and the current research. Section 1.7 and 1.8 explain why CSMA/CD and the token bus were ruled out as possible approaches to support voice and data. Section 1.9 describes the type of solution sought and the approaches for obtaining it.

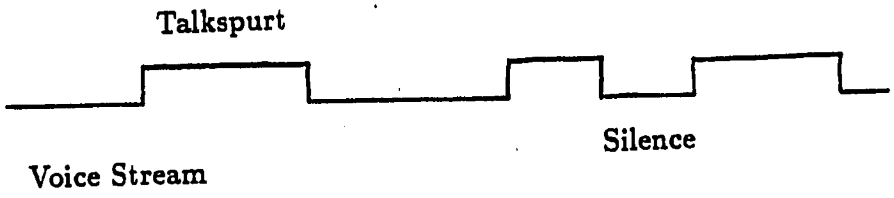


Figure 1.1: Brady's Voice Model

1.1 The Traffic Characteristic

Table 1.1 compares voice, data and documents. The nature of the data traffic is self-explanatory from the table and the documents traffic will be discussed in more detail in Chapter 3. Voice traffic, however, needs further clarification. Voice, for long, was considered as a source of continuous or *stream* traffic. Hence, only the circuit switching method, which provided guaranteed channel bandwidth was considered suitable for supporting voice. Brady's experiments [6], however, indicated that voice is not a continuous traffic source but, in fact, is made of alternating *talkspurts* and *silence periods.* Brady went on to show further that these talkspurts and silence periods are approximately exponentially distributed with means 1.36 sec and 1.802 secs, respectively. Figure 1.1 illustrates this phenomenon and we have used this observation in our simulations and analysis.

1.2 Problem Statement

Design of a protocol for local networks to support

1. Voice transmission
2. Data transmission (Interactive and Bulk)
3. Document image transmission for an interactive library system.

1.3 Why Should it be Solved

The following reasons prompted the study on the above mentioned system.

- Local networks are prolific
- Most local networks operate below capacity
- Office automation demands voice/data integration
- A library system is essential in a research environment

1.4 Comparing CBX with LANs

Voice and data services can be integrated on a LAN or a Computerized Branch Exchange (CBX) [43,35,52]. CBX is a third generation PBX (Private Branch Exchange) which has three basic characteristics: distributed architecture, integrated voice and data and non-blocking configuration [43]. CBX and LAN can be compared as shown in Table 1.2.

Although the CBX appears to be the better choice between the two in most categories, applications which have large data transfer requirements (e.g., file transfer, library systems) interspersed with interactive traffic LANs are most suited and

Table 1.2: LAN vs CBX

	LAN	CBX
Installation	Expensive	Less Expensive
Reliability	Reliable	More Reliable
Data Types	Voice/data/video	Voice/data
Speed	0.1-100 Mbps	9.6-64 Kbps

if LANs are already installed then cost of added services is not a decisive factor, and this makes LANs attractive.

1.5 A Look at Standards

The groups concerned with LAN standardization, the Institute of Electrical and Electronics Engineers (IEEE) Project 802 and the European Computer Manufacturers Association (ECMA) TC24, have adopted a LAN architecture model that describes the relationship of LAN architecture and the Open Systems Interconnection (OSI) Reference Model [10,24,25,26]. Figure 1.2 shows the OSI data-link layer split into two sublayers, the medium-dependent *Medium Access Control* (MAC) sublayer and the medium independent *Logical Link Control* (LLC) sublayer. Local area networks, thus, differ mainly in the medium, the physical layer and the MAC sublayer. Also, the quality of service at the MAC-to-LLC interface differs between different local area networks.

In this section the discussion is focused on the three MAC methods that have been standardized and are widely used: Carrier-Sense Multiple-Access with Collision Detection (CSMA/CD), Token Ring and Token Bus.

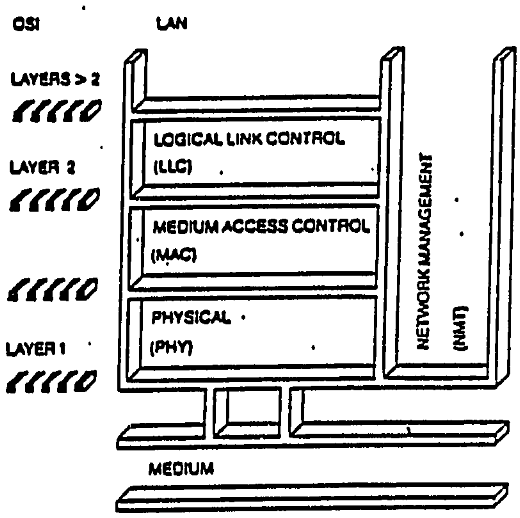


Figure 1.2: LAN architecture reference model

1.5.1 CSMA/CD

Carrier-sense multiple-access with collision detection may be viewed as the offspring of CSMA methods developed for the early broadcast systems, for example, the ALOHA system. While immediate detection of collisions is difficult in radio systems or satellite systems, it can be readily employed in bus systems, resulting in improved performance. The CSMA/CD protocol is described in [24].

1.5.2 Token ring

Token-ring principle was considered for LAN because of its favorable attributes regarding wiring, transmission technology, performance and the potential low cost implementation. Its potential problem, viz., reliability can be overcome by appropriate wiring and suitable access protocols [10,9]. The description of the token-ring protocol is in Chapter 2 and [26].

1.5.3 Token bus

The token bus technique combines the attractive features of a bus topology (e.g., use of broadband transmission) with those of controlled medium access protocol (e.g., good efficiency under high traffic load, speed-distance sensitivity, and fairness of access). The operation of the token-bus protocol is dealt with in [25].

1.5.4 Other local network techniques

In addition to the standard approaches, various other LAN techniques have been proposed and successfully implemented [54,9,39]. Slotted rings, buffer-insertion rings or buses with controlled type access are just a few of them.

1.6 Current Research in Application of Synchronous Protocols

The advent of office automation has necessitated the integration of voice and data onto one local area network. These two traffic types, however, have different characteristics. In particular, voice is stream traffic that has long holding time and requires real-time delivery [23,22], while data traffic can be categorized into two basic types: interactive and bulk [59]. Interactive data traffic is *bursty* in nature and it usually consists of short messages requiring low network delay. Bulk data traffic, on the other hand, consists of long messages and generally requires high throughput. For data, continuity and real-time delivery considerations are not of primary importance [59].

Voice has traditionally been treated as a continuous traffic source. However, it has been shown [7,6] that in a typical conversation, voice traffic exhibits on-off characteristics and the channel capacity in each direction is idle for 60-65 percent of the time. Voice traffic is thus often conveniently modeled as having alternating *talkspurts* and *silences* with synchronous input present during the silence periods and no input during silence periods. Two basic requirements must be met for voice traffic: a small end-to-end delay and a synchronous output at the destination.

Similar to voice requirements, data also may require a small end-to-end delay and synchronous output at the destination. Consider a library system, for example, where a user requests a certain publication from the central library electronic data base. Images to be printed out to a screen are stored in digital format and transferring these images to a terminal is similar to a file transfer operation. Unlike file transfer, however, document retrieval is *time critical* i.e. the end-to-end delay needs to be bounded. So a synchronous protocol with bounded end-to-end delay is a necessity for a library system operation.

A multitude of synchronous protocols have been proposed (mainly for voice/data integration). What follows next is a survey of the main ideas in a few of them.

1.6.1 CSMA/CD

Since the channel access delay is not guaranteed to be bounded in CSMA/CD networks, it is generally difficult to transmit voice in these networks without either modifying its basic operation or developing complex algorithms for handling voice [23].

1.6.1.1 CSMA/CD with channel capture This protocol is developed for a local area network with unidirectional global bus architecture, in which a ready station captures the bus for one packet time only if no other upstream stations transmit along with it [53]. This protocol has a dynamic priority structure in which each station's priority changes according to the state of the system. The protocol ensures fairness.

1.6.1.2 Multirate voice coding on CSMA/CD This interesting idea considered the feasibility of using multirate voice coding to control the traffic intensity on the network [17]. By decreasing the voice coding rate for short periods of time (and thus compromising the voice quality) when the network traffic increases, network delay is reduced. In this scheme the premise is that short term voice quality can be traded for increased throughput on a CSMA/CD network. With some added feedback circuitry this scheme seems to perform better than the standard CSMA/CD.

1.6.1.3 Movable boundary protocol for integrated voice/data This protocol introduces a movable voice-data boundary to framed TDMA/CSMA and eliminates the requirement of system wide synchronized clocks [50]. The movable boundary is a major advantage in any system where fluctuations in voice and data loads are expected because assignment of the idle capacity from one class to the other increases the utilization of the channel.

1.6.1.4 Other techniques Amongst other techniques the one by van de Lagemaat et al. [56] stands out by its complexity in collision resolution. A complex algorithm is executed to prioritize the next transmission after a collision. Another protocol worth mentioning is the L-express [5] protocol which utilizes a simple and efficient virtual token access scheme which provides ordered collision-free transmission. Priorities are assigned to stations with two counters per station. Depending on the values of these counters the state of the station is determined.

1.6.2 Token Ring

Token passing protocols have always seemed attractive for voice-data integration because of their *bounded delay* property.

1.6.2.1 Voice-data integration with distributed control In order to enforce synchronous communication for voice packets, these packets should be transmitted at the same rate they are formed by the encoder. This protocol defines a window width of h seconds such that if the packet is expected to be transmitted at time t_0 then if the token arrives at any instant t, the packet will be transmitted only if $t_0 \leq t \leq t_0 + h$ [22,23]. If the token arrives outside of this interval, the packet is discarded. Thus, the success of this protocol depends on the value of h. Two schemes were proposed [22]: fixed transmission scheme (FTS) and sliding transmission scheme (STS). The former scheme has fixed interval positions with respect to the initial packet transmission, while the latter positions its interval according to the previous transmission. This scheme seems to work better than using standard token ring for voice transmission.

1.6.2.2 Voice data integration with centralized approach Stations are divided into two categories, voice and data [59,9]. The network has a special station called the supervisor. This station looks after voice call management, that is, granting and denying call requests and setting up/tearing down of calls. At regular time instances, the supervisor attempts to seize the circulating token. If the token is free, the supervisor issues a priority token to be used by the voice stations. Each token in talkspurt gets a chance to transmit one packet as the priority token goes around the ring. After all the voice stations have had their turn, the supervisor removes the priority token from the ring and issues a normal token. If the token is busy (when the supervisor attempts to grab it), that is, data transmission is in progress, the supervisor must wait until the data transmission is complete before it can issue the priority token. The IBM token ring is based on this idea.

1.6.2.3 Welnet : A high speed LAN with two unidirectional channels This modification of the standard token ring worth mentioning because of its expensive implementation [37]. Welnet is a high speed local area network with a pair of unidirectional communication channels. While individual stations control their own access to the medium, global scheduling of the medium for various types of traffic is controlled centrally. The central station issues three types of token : C_0, C_1, and C_2. C_0 is the null token which is used at the beginning of every new service cycle. C_1 is the voice token, and only voice stations may transmit upon acquiring it. C_2 is the data token used by data stations only. The null token is used to preempt stations from using C_1 or C_2.

1.6.3 Token bus

Token bus protocol seems to be less popular compared to the token ring, probably because of its large token-passing overhead. This author has not come across a single proposed token bus protocol with voice/data integration. One scheme, however, seems to have the potential for adopting token bus protocol for voice/data integration.

1.6.3.1 Twin-cycle token bus protocol A well known disadvantage of standard token-passing in ring and bus networks is the waste in channel bandwidth often seen in lightly loaded or asymmetric systems. It is possible to make use of the broadcast mechanism in token bus systems to distribute readily the up-to-date information about the state of individual stations to the entire system [11]. This scheme involves the determination of randomly varying set of more active stations. The stations are given a chance to form a second logical ring that characterizes the token bus. The transmission cycle of the system can thus be made to alternate between standard and improved cycles, termed as *passive* and *active* cycles, respectively. The system is shown to be more stable than the standard token bus protocol.

1.7 CSMA/CD is Ruled Out: Why ?

The following reasons make using CSMA/CD difficult.

- CSMA/CD performance degrades rapidly at high utilizations [34].
- Channel access delay is not guaranteed to be bounded in CSMA/CD. This means that real time delivery of voice packets becomes probabilistic, thus making CSMA/CD not suitable for voice transmission.
- CSMA/CD may be used for voice transmission, provided the basic CSMA/CD operation is altered [50,38]. This approach may either be expensive or infea-

sible.

- CSMA/CD may be used for voice transmission by developing a very complex algorithm [50,56] for handling voice. This, again, does not seem to be a feasible idea especially since an alternative, the token passing protocol is available.

1.8 Why is the Token Bus Not Suitable ?

Logically the token bus protocol and the token ring protocol are identical. The implementation and standardization, however, set them apart and, thus, dictate their usefulness in a given situation. Since, *sticking to the standard as closely as possible* is one of the primary goals of this research, these differences are significant and a case can be made against the token bus. The differences between the bus and the ring are as follows.

1. There is no direct way to reserve the priority of the token globally, on a token bus. This means that a station may have high priority data waiting to be transmitted while all other stations that receive tokens before it may transmit lower priority data. A clever selection of timer values may improve the performance of the high priority data but it is still marginal. Token ring priority reservation scheme eliminates this problem, since a token can be hurried through the network to serve high priority data.

2. The token passing time is much higher in a token bus. This deteriorates its performance at high utilizations. A document retrieval system is expected to cause high utilization of the channel bandwidth and hence token ring prevails over token bus.

1.9 The Goal of the Research

The goal of this research is to design a multipriority protocol accommodating voice/data and synchronous data (documents) traffic on a LAN with its performance characteristics at least as good as the existing protocols. Two approaches were adopted for protocol analysis.

1.9.1 Approaches to be adopted for protocol analysis

1.9.1.1 Analytical modeling Analytical modeling is generally less expensive of the two approaches discussed here. Performance analysis with this approach, however, is far more difficult for multipriority protocols than only for data traffic [59,48]. The performance of a LAN is modeled with the following parameters [52].

- The physical structure parameters:
  1. Bandwidth of the channel
  2. Length of the bus or ring
  3. Error rate of the channel
  4. Node buffering capacity
- The protocol structure parameters:
  1. Channel access mechanism
  2. Addressing mechanism
  3. Transmission policy : exhaustive / nonexhaustive
  4. Retransmission strategy
  5. Buffer management policy
  6. Control overhead
- The workload structure parameters:
  1. Arrival rate of packets

2. Packet size
3. Service rate of packets
4. Priority
5. Service requirement of packets

A protocol designer naturally has control over the protocol structure parameters. The physical structure parameters depend on the existing technology and the cost incurred. When analyzing a network analytically, the workload structure parameters are chosen such that they resemble the projected workload as closely as possible, while still maintaining their mathematical tractability [13,31,32,33]. Various mathematically tractable queueing models have been proposed and analyzed extensively [31,32]. The key, therefore, lies in modeling the protocol of interest onto these models with appropriate modifications. This approach is adopted in the ensuing research.

1.9.1.2 Simulation Queueing behavior, described above, can also be studied by means of simulation, which is an expensive undertaking, particularly at high transmission rates. Mathematical tractability, however, is not easy to come by, hence though expensive, simulation provides an attractive alternative to analytical modeling [47,14]. Simulation allows for an arbitrary level of protocol complexity.

Simulation technique is used in this research in order to tune the analytical model closer to the real system. It is also be used to find optimal values of various physical structure and protocol structure parameters.

1.10 Organization

An attempt is made to organize this dissertation to facilitate the reader. Chapter 2 discusses the standard token ring. Chapters 3 and 4 address the design issues related to a document retrieval system with integrated voice and data. Chapter 5 deals with the simulation fundamentals and their use in determining the protocol selection. The analytical models are developed in Chapter 6.

2 THE TOKEN RING ACCESS METHOD

Apart from completeness, the reason for having a chapter for the Token Ring protocol is as follows. The IEEE Standard 802.5 contains all the detail about the functioning of the ring, but for the purpose of this dissertation only a small portion of it is necessary. Thus, this chapter, which is at times copied verbatim from the standard, serves as a summary of the necessary detail from IEEE 802.5. The readers familiar with the protocol may wish to skip this chapter.

2.1 General Description

A token ring consists of a set of stations connected by a transmission medium. Information is transferred sequentially, bit by bit, from one active station to the next. Each station generally regenerates and repeats each bit and serves as the means for attaching one or more devices (terminals and work-stations) to the ring for the purpose of communicating with other devices on the network. A given station (the one that has access to the medium) transfers information onto the ring, where the information circulates from one station to the next. The addressed destination station(s) *copies* the information as it passes. Finally, the station that transmitted the information effectively removes the information from the ring.

A station gains the right to transmit its information onto the medium when it detects a free token passing on the medium. The token is a control signal comprised of a unique signalling sequence that circulates on the medium following each information transfer. Any station, upon detection of an appropriate token, may capture the token by modifying it to start-of-frame sequence and appending appropriate control and status fields, address fields, information field, frame-check sequence and end-of-frame sequence. At the completion of its information transfer and after appropriate checking for proper operation, the station initiates a new token, which provides other stations the opportunity to gain access to the ring.

A token holding timer controls the maximum period of time a station shall use (occupy) the medium before passing the token.

Multiple levels of priority are available for independent and dynamic assignment depending upon the relative class of service required for any given message. The allocation of priority shall be by the mutual agreement among the users of the network.

2.2 Token-Generation Strategies

Bux et al. in [9] reported three strategies conceivable in a token ring setup.

2.2.1 Single frame operation

In this strategy the sender issues a free token only after it has received its entire frame back and has erased it. Thus, at any given time there is only one frame on the ring.

2.2.2 Single token operation

In this strategy the sender does not issue a free token until it has received the header of the frame including the busy token. This is effective in cases where small frames are transmitted.

2.2.3 Multiple token operation

In this strategy the sender issues a free token immediately after the end delimiter of a frame is transmitted. This creates the possibility of having multiple tokens on the ring with at most one of them being free.

We opted for the single token operation for the same reason as Bux et al. [9], namely,

- The priority operation requires examining the header of the transmitted frame to check for reservations.

- Token supervision and recovery, though not addressed in this dissertation, are important. They are greatly facilitated by the single token operation. These reasons eliminate multiple token operation from contention.

- Single frame operation is the least efficient in case of small messages. This may not be a factor when the transmitted messages are always greater than the ring latency. In this case, the single token operation degenerates to single frame operation.

Our choice was further limited by the availability of the components, namely, Texas Instruments TMS 380 chip set, which opted for single token operation.

SD = Starting Delimiter (1 octet)
AC = Access Control (1 octet)
ED = Ending Delimiter (1 octet)

Figure 2.1: The Token Format

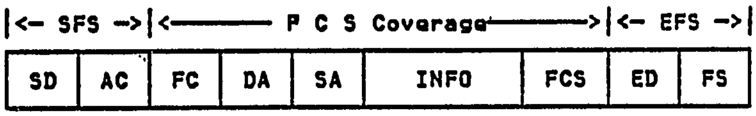
2.3 Formats and Facilities

There are two basic formats used in token rings: tokens and frames. In the following discussion, the figures depict the formats of the fields *in the sequence they are transmitted on the medium*, with left-most bit or symbol transmitted first. (An octet consists of eight bits). The token shall be the means by which the right to transmit (as opposed to the normal process of repeating) is passed from one station to another.

2.3.1 Frame format

2.3.2 Field description

The following is a detailed description of the individual fields in the tokens and frames that are considered relevant in this dissertation.



SFS = Start-of-Frame Sequence
SD = Starting Delimiter (1 octet)
AC = Access Control (1 octet)
FC = Frame Control (1 octet)
DA = Destination Address (2 or 6 octets)
SA = Source Address (2 or 6 octets)
INFO = Information (0 or more octets)*
FCS = Frame-Check Sequence (4 octets)
EFS = End-of-Frame Sequence
ED = Ending Delimiter (1 octet)
FS = Frame Status (1 octet)

Figure 2.2: The Frame Format

2.3.2.1 Access Control (AC)

Priority bits: The priority bits shall indicate the priority of a token and, therefore, which stations are allowed to use the token. In a multiple-priority system, stations use different priorities depending on the priority of the PDU transmitted. The eight levels of priorities increase from the lowest (000) to the highest (111) priority.

PPP = priority bits
T = token bit
M = monitor bit
RRR = reservation bits

Figure 2.3: The Access Control Field

Token bit: The token bit is a 0 in a token and a 1 in a frame. When a station with a PDU to transmit detects a token which has a priority equal to or less than the PDU to be transmitted, it may change the token to a start-of-frame sequence and transmit the PDU.

Reservation bits: The reservation bits allow stations with high priority PDUs to request (in frames or tokens as they are repeated) that the next token to be issued at the requested priority. The exact operation of the reservation scheme will be described later.

2.3.2.2 Destination and source address (DA and SA) fields Each frame shall contain two address fields: the destination (station) and the source (station) address, in that order. Addresses may be either 2 or 6 octets in length; however, all stations of a specific LAN shall have addresses of equal length.

We have omitted description of many other fields and even parts of some of the fields described here. Only the information given here is considered relevant.

2.3.3 Timers

In the current context, the only timer of importance is the token hold timer (THT). Each station will have a timer THT to control the maximum period of time the station may transmit frames after capturing a token. A station may *initiate* transmission of a frame if such transmission can be *completed before the timer expires.*

2.3.4 Priority Registers and Stacks

2.3.4.1 Pr and Sr registers The value of the priority (P) and reservation (R) of the most recently received AC field are stored in registers as Pr and Rr.

2.3.4.2 Sx and Sr stacks If at the time of transmission of a token the value of Rr or Pm (the priority of a queued PDU) is greater than Pr, a token with a priority of the higher of Rr or Pm shall be transmitted. At the same time the station shall store the value of Pr in a stack as Sr and shall store the value of the priority of the token that was transmitted in a stack as Sx.

2.4 Priority Operation

The priority bits (PPP) and the reservation bits (RRR) contained in the access control (AC) field work together in an attempt to match the service priority of the ring to the highest priority PDU that is ready for transmission on the ring. These values are stored in Pr and Rr. The current ring service priority is indicated by the priority bits in the AC field, which is circulated on the ring.

The priority operation operates in such a way that *fairness* is maintained for all stations within a priority level. This is accomplished by having the same station that raised the service priority level of the ring (the *stacking station*) return the ring to the original service priority. Stacks Sx and Sr are used to perform this function.

The priority operation is explained as follows: When a station has a priority (a value greater than zero) PDU (or PDUs) ready for transmission, it requests a priority token. This is done by changing the reservation bits (RRR) as the station repeats the AC field. If the priority level (Pm) of the PDU that is ready for transmission is greater than the RRR bits, the station increases the value of RRR field to the value of Pm. If the value of the RRR bits is equal to or greater than Pm, the reservation bits (RRR) are repeated unchanged.

After a station has claimed the token, the station transmits PDUs that are at or above the present ring service priority level until it has completed transmission of those PDUs or until the transmission of another frame could not be completed before timer THT expires. The priority of all of the PDUs transmitted should be at the present ring service priority value. The station will then generate a new token for transmission on the ring.

If the station does not have additional PDUs to transmit that have a priority (Pm) or does not have a reservation request (as contained in register Pr), the token is transmitted with its priority at the present ring service priority and the reservation bits (RRR) at the greater of Rr or Pm and no further action is taken.

However, if the station has a PDU ready for transmission or a reservation request (Rr), either of which is greater than the present ring service priority, the token is generated with its priority at the greater of Pm or Rr and its reservation bits (RRR) as 0. Since the station has raised the service priority level of the ring, the station becomes a stacking station and, as such, stores the value of the old ring service priority as Sr and the new ring service priority as Sx. (These values will be used later to lower the service priority of the ring when there are no PDUs ready to transmit on the ring whose Pm is equal to or greater than the stacked Sx).

Having become a stacking station, the station claims every token that it receives that has a priority (PPP) equal to its highest stacked transmitted priority (Sx) in order to examine the RRR bits of the AC field for the purpose of raising, maintaining, or lowering the service priority of the ring. The new token is transmitted with its PPP bits equal to the reservation bits (RRR) but no lower than the value of the highest stacked received priority (Sr), which was the original ring service priority level.

If the value of the new ring service priority (PPP equal to Rr) is greater than Sr, the RRR bits are transmitted as 0, the old ring priority contained in Sx is replaced with a new value Sx equal to Rr, and the station continues its role as a stacking station.

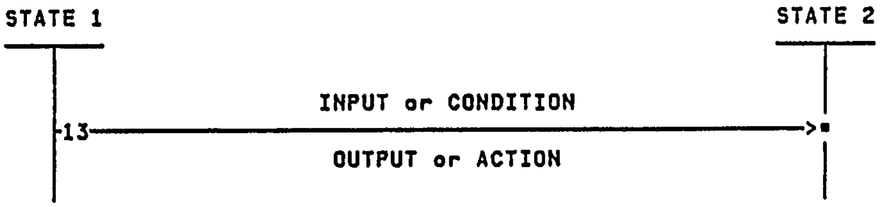
However, if the Rr value is equal to or less than the value of the highest stacked received priority (Sr) the new token is transmitted at a priority value of the Sr, both Sx and Sr are removed (*popped*) from the stack, and if no other values of Sx an Sr are stacked, the station discontinues its role as a stacking station.

2.5 Specification

The Finite State Machine (FSM) method is used for specification of the protocol.

2.5.1 Finite-State Machine Notation

The following notation is used in the FSM diagrams.



A transition begins when the input occurs or the condition specified is met and is complete when the output or the action has occurred. If the state transition is in progress, then no other FSM transition may be initiated. If the exit conditions of a state are satisfied at the time the state is entered, no action is taken in that state and the state is immediately exited.

2.5.2 Abbreviations and Mnemonics

A = Address-Recognized Bit

DA = Destination address

EFS = End-of-Frame sequence

M = Monitor Bit

P = Priority (of the AC)

PDU = Protocol Data Unit

Pm = PDU Priority

Pr = Last Priority Value Received

R = Reservation (of the AC)

RR = Last Reservation Value Received

SA = Source Address

SFS = Start-of-Frame sequence

Sr = Highest Stacked Received Priority

Sx = Highest stacked Transmitted Priority

THT = Timer, Holding Token

TK = Token

Table 2.1: Bit Flipping Loop State Table

REF	INPUT	OUTPUT
02A	PDU_QUEUED & (FR(R<Pm) OR TK(P>Pm>R, P≠Sx)	SET R=Pm
02C	DA=MA (ADDRESS RECOGNIZED)	SET A=1

TX = Transmit

TK(P=x,M=y,R=z) = Token with P=x, M=y, and R=z

FR(P=x,M=y,R=z) = Frame with P=x, M=y, and R=z

2.5.3 Operational Finite-State Machine

The operational finite-state machine is explained as follows:

2.5.3.1 State 0: REPEAT (Repeat State) In the Repeat state, the bits that are received are, in general, repeated on the line to the next station. Certain bits and fields in the repeated bit stream may be modified and certain actions taken without changing state. Transition shall be made to State 1 when there are one or more PDUs queued for transmission and the conditions for transmission are satisfied. Transition shall be made to state 4 for the purpose of modifying stacks.

(01) Usable Token Received If a PDU is queued for transmission and a token is received whose priority (P) is equal to or less than the PDU priority (Pm), the station shall change the token to a start-of-frame sequence (by changing the token bit from 0 to 1) and transmit M and R as 0, initiate the transmission of

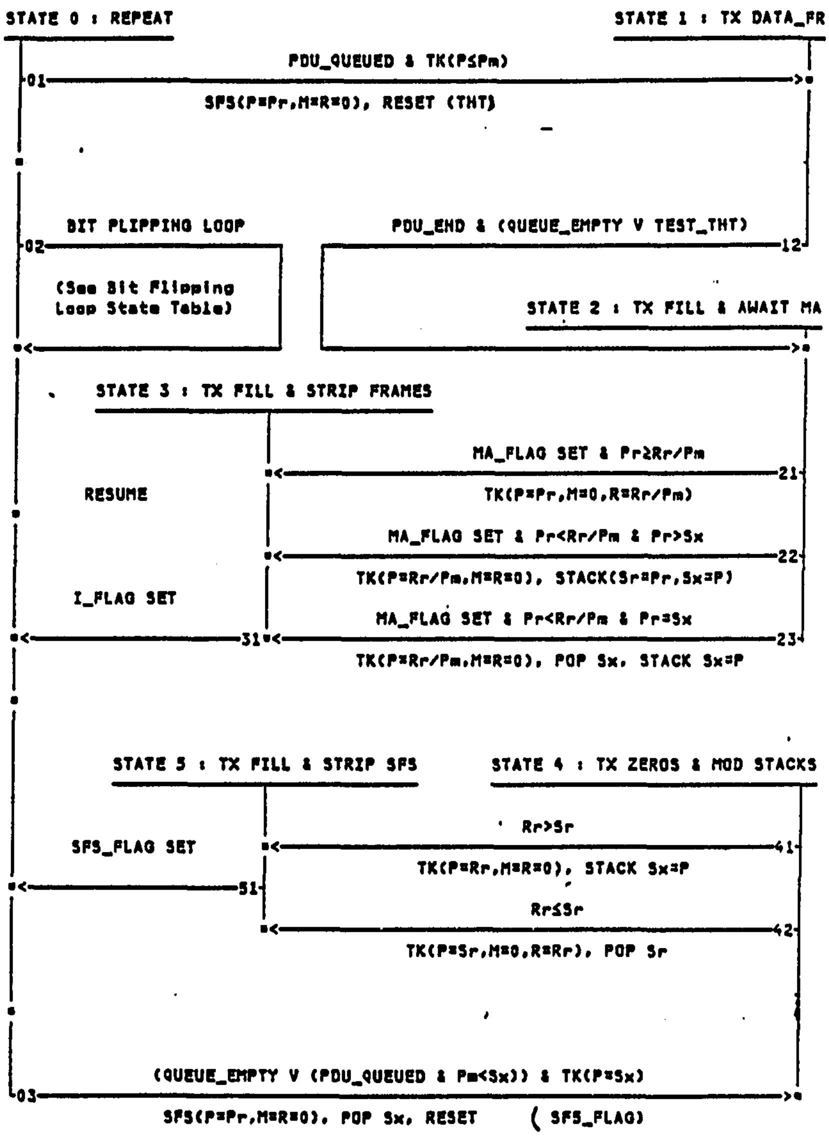


Figure 2.4: Reduced Operational Finite-State Machine Diagram

the enqueued PDU, reset the THT and the MA flag, and make a transition to State 1.

(02) Bit Flipping Loop A number of actions may be taken without changing state. These actions are shown in Table 2.1 and are explained as follows:

(02A) Request Usable Token If there is a PDU queued for transmission with priority Pm, the reservation (R) shall be set to Pm on frames in which the reservation is less than Pm, and on tokens in which the priority is greater than Pm and the reservation is less than Pm and the priority is not equal to the highest stacked transmitted priority.

(02C) Own Address Detected If the station detected its own address in the DA field, the A bit in the FS field shall be transmitted as 1.

(03) Re-stack Operation If there are no frames enqueued with priority equal to or greater than the highest stacked transmitted priority (Sx) and a token is received with priority (P) equal to the highest stacked transmitted priority (Sx), the following actions are taken. The token shall be changed to a start-of-frame sequence by changing the T bit from 0 to 1, popping the Sx from the stack, and making a transition to State 4. If there is no Sx value stacked, the test P=Sx shall be considered to be false.

2.5.3.2 State 1: TX DATA_FR (Transmit Data Frame(s)) While in this state, the station transmits one or more frames. The first and all subsequent PDUs that are transmitted shall have a Pm equal t or greater than the priority of the token that was used. All frames transmitted will have P equal to Pr and M and R equal to 0.

(12) End-of-Frame Transmission If the transmission of the PDU is completed and there are no more PDUs to transmit at this priority or a higher priority, or if transmission of an additional frame could not be completed before THT expires, and end-of-frame sequence (EFS) will be transmitted and a transition shall be made to State 2.

2.5.3.3 State 2: TX FILL & AWAIT MA (Transmit Fill and Await My Address) If a source address equal to the station's address has not been received the station shall transmit fill until the address is received. Upon entering State 2, if the address is already received then transition shall be made directly to State 3 via transitions 21, 22, or 23.

(21) Token Transmission, Same Priority If both the stored value Rr and a queued PDU priority (Pm) are less than or equal to the stored value Pr, a token shall be transmitted with the P equal to Pr, M equal to 0, and R equal to the greater of Rr or Pm, and transition shall be made to State 3.

(22) Token Transmission, Higher Priority, and Pr > Sx (Push Ring Priority) If the Rr or an enqueued PDU priority (Pm) is greater than the Pr, and the highest stacked transmitted priority (Sx) is less than the last priority value received (Pr), a token shall be transmitted with the P equal to the greater of Rr or Pm, and M and R equal to 0. Pr shall be stacked as Sr, P shall be stacked as Sx, and a transition made to state 3. If there is no Sx value stacked, the test Pr > Sx shall be considered true.

(23) Token Transmission, Higher Priority, and Pr = Sx (Pop Ring Priority) If the Rr or an enqueued PDU priority (Pm) is greater than the Pr, and the highest stacked transmitted priority (Sx) is equal to the last priority value received (Pr), a token shall be transmitted with the P equal to the greater of Rr or Pm, and M and R equal to 0. Sx shall be popped from the stack, a new value of P shall be stacked as Sx, and a transition made to state 3. If there is no Sx value stacked, the test Pr = Sx shall be considered false.

2.5.3.4 State 3: TX FILL & STRIP FRAMES (Transmit Fill and Strip Frames) Wait until the last of the transmitted frames arrives and then make a transition to State 0.

(31) Strip Complete Last of the transmitted frames received, make transition to state 0.

2.5.3.5 State 4: TX ZEROS & MOD STACK (Transmit Zeros and Modify Stack) A continuous string of 0's shall be transmitted immediately following the SFS until the internal logic of the station can perform the necessary function to transmit a token.

(41) Reservation Request (Rr) > Highest Stacked Received Priority (Sr) If Rr is greater than the highest stacked priority Sr, a token with its priority (P) set to Rr and its M and R bits set to 0 shall be transmitted, P shall be stacked as Sx, and a transition shall be made to State 5.

(42) Reservation Request (Rr) $\leq$ Highest Stacked Received Priority (Sr) If Rr is equal to or less than the highest stacked priority Sr, then a token with its priority (P) equal to Sr, M equal to 0, and R equal to Rr shall be transmitted, Sr popped from the stack, and a transition shall be made to State 5.

This concludes the description and specification of the IEEE Standard 802.5 for the token ring protocol.

3 COMPONENTS OF A DOCUMENT RETRIEVAL SYSTEM

The advent of optical storage devices has accelerated the use of image archives for storing documents [45]. Even high speed magnetic memories are proving to be competitive for voluminous data storage. The technological development in producing high speed monitors has also increased the feasibility of implementing online document retrieval systems. In a research environment, availability of a document retrieval system is always an asset. A typical document retrieval system to serve a university campus environment is expected to have the following basic characteristics [18]. It should provide

- Economical service to a major campus.
- Quality user interface.
- Access to a wide variety of and a large volume of information.
- For easy upgrading of the system.

Also desirable are the following auxiliary characteristics.

- Privacy of users.
- User ability to update stored information.

The focus of this research is only on achieving the basic characteristics optimally. Specifically, we have concerned ourselves with only transmitting text images on an underlying network (Token Ring) as economically and as fast as possible from a large information base with the ease of upgrading in mind. In the course of the design the following assumptions were made.

3.1 Assumptions

3.1.1 User capacity

About 5000 users use the library system per working day. A working day consists of 16 hours. Each user has 2 sessions per day, i.e., a typical user goes to library twice a day.

3.1.2 The concentrators

There are 10 concentrators available on campus. These concentrators are directly hooked up to the network. Each concentrator serves 15 terminals. Figure 3.1 shows the schematic for a typical concentrator.

3.1.3 User modes

A user session consists of two modes: *Browsing* and *Reading*. In the browsing mode the user simply skims through pages of text and requests for more pages until he finds a page he wants to read. Once such a page is found he can either read it on the screen of his terminal or print a hard copy on a local printer. Obviously, in browsing mode the rate of request is faster than the reading mode. It is assumed

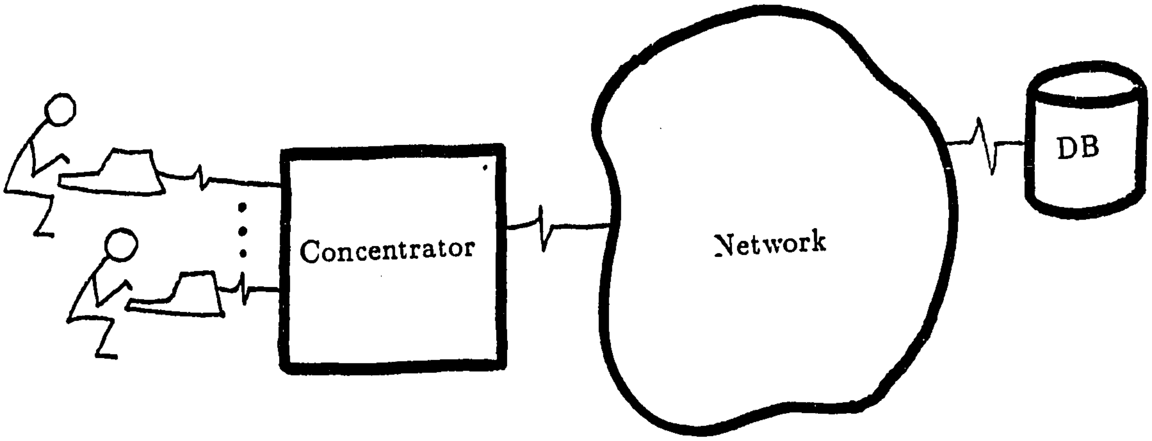
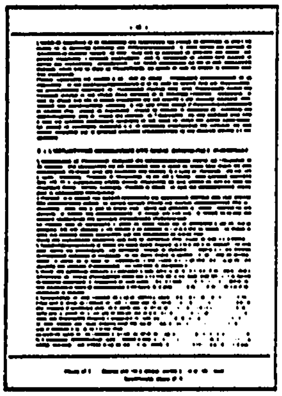


Figure 3.1: A typical concentrator

that the a user is in browsing mode 2/3 of the time and in reading mode 1/3 of the time. A user browses 30 pages per minute and reads 1 page every two minutes, on an average, both modes being exponentially distributed.

3.1.4 The information base

The information base consists of electronically stored text images in digital format. A typical page (such as the ones in magazines and papers) looks like a CCITT test document type 4 (See Figure 3.2), which takes approximately 50 Kbytes of memory [60]. Though this can be compressed further, we have conservatively assumed the memory size to be 50 Kbytes per page.



No. 4

Figure 3.2: CCITT test document type 4

3.1.5 The data search

The information base is assumed to have efficient search algorithms, and a suitable memory architecture. Thus, the information base is able to handle all the requests submitted to it, either by getting the required pages or by negating the request. Our simulations, however, assume that all requests are answered positively.

3.1.6 Terminal requirements

A typical terminal is depicted in Figure 3.3 [41]. Thus, a terminal consists of a codec (coder - decoder), a CRT screen, a computer similar to a personal computer and possibly a printer.

With these explicit assumptions and facts about the document retrieval system, the following criteria are established. The main performance criteria are

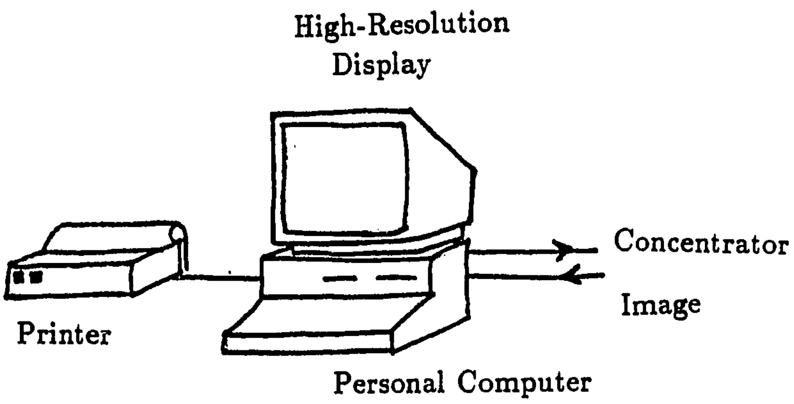


Figure 3.3: A typical terminal

1. Low end-to-end page request delay
2. Low adverse effect on other types of traffic

3.1.7 Low end-to-end page request delay

The *end-to-end page request delay* is defined as the time interval between when a page request is generated at the concentrator, and the time at which the requested page is obtained by the user. A total page request delay of 5 seconds is deemed reasonable. Four factors contribute to this delay [41]:

- Channel speed
- Queueing delay at the concentrator and at the information base
- Codec speed
- Information base access time

3.1.7.1 Channel Speed The channel speed probably contributes the least to the page request delay. For example, a page consisting of 50 Kbytes of data takes only 100 ms to be transmitted on a 4 Mbits network. This factor is not under the protocol designer's control once the network speed is fixed.

3.1.7.2 Queueing delay The queueing delay depends on the traffic on the network and the priority structure. The page request delay could be minimized by assigning the page packets the highest priority. This, however, jeopardizes the performance of other traffic types. Our work is focused on reducing this delay to a minimum, while satisfying the requirements of the other traffic types.

3.1.7.3 Codec speed Since all the pages transmitted on the network are coded, a decoding algorithm is necessary at the user terminal. It is assumed that 1 second is required to execute such a decoding algorithm [41].

3.1.7.4 Information base access time The exact nature of the technology to be used is yet unknown. Two possible alternatives are the optical memory and the high speed magnetic memory. In either case the following assumption holds good. If a page is divided into 5 packets of size 10 Kbytes each, then the first packet takes 100 ms to be queued to be transmitted, each of the remaining packets take 25 ms more than the previous packet to appear in the queue. Therefore, in all, a page takes 200 ms to appear in the queue. A tacit assumption here is that multiple requests can be handled by the information base simultaneously.

3.1.8 Low adverse effect on other traffic types

The real time bounds posed by voice traffic restrict the protocol designer implementing a document retrieval system on a network. In essence, the problem reduces to an optimization problem.

3.1.9 Page request rate calculations

Based on the above assumptions, the page request rate per concentrator may be calculated as follows:

$$\begin{aligned} N_u &= \text{Total number of users} = 5000 \\ S_u &= \text{Number of sessions per user per day} = 2 \\ N_h &= \text{Number of hours per day} = 16 \\ N_c &= \text{Number of concentrators} = 10 \\ T_{sh} &= \text{Number of sessions per minute} \\ T_{sc} &= \text{Number of sessions per concentrator per minute} \end{aligned}$$

Therefore,

$$T_{sh} = \frac{N_u \times S_u}{N_h \times 60} \; (= 10.42) \tag{3.1}$$

$$T_{sc} = \frac{T_{sh}}{N_c} \; (= 1.042) \tag{3.2}$$

Thus, one session generates, on an average, 20.17 pages per minute ($= 30 \times 2/3 + 1/2 \times 1/3$). This calculation is used in our simulation and analysis.

4 PROTOCOL DESIGN

This chapter deals with various constraints faced by a multifarious traffic network and their effect on the design of a suitable protocol. This chapter is divided into four sections.

1. The playout strategy
2. Packet size
3. Priority mechanism
4. Parameters of interest

4.1 The Playout Strategy

In [40], Montgomery discussed the issues regarding the synchronization of packetized voice. Figure 4.1 illustrates the sender's and the receiver's view of the generated voice packets. As is evident from the figure, the voice packets must be played out at the same rate they are generated, in order to have intelligible conversation. However, this task is complicated by the stochastic delay suffered by each packet being different. Thus, the voice packets in a packet network naturally arrive out of synchronization. Therefore, much thought is given to the selection of the playout strategy that would circumvent the above problem. Two basic characteristics of the playout strategies distinguish them from each other. They are,

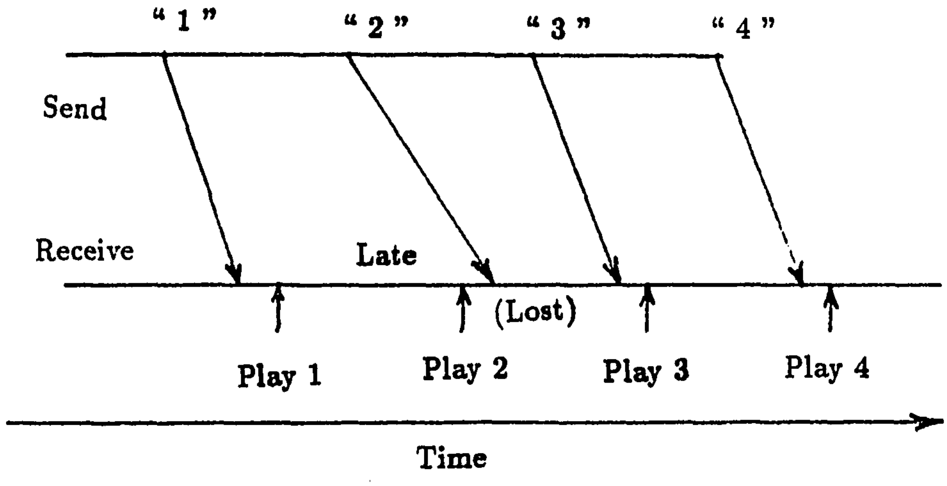


Figure 4.1: Sender and receiver voice stations

1. The first packet delay
2. Late packet policy

4.1.1 First packet policy

Since the stochastic delay suffered by the packets makes it impossible for the receiver to know the exact time at which the packet was generated at the sender, a delay estimation method is adopted by the receiver. The delay experienced by the packet comprises of two components, fixed delay D_F and variable delay D_V. The fixed component consists of the propagation delay, the transmission delay, and the buffering delays. The variable delay is caused by the queueing delay at the sender. In a local area network scenario, the fixed delay, as the name suggests, is deterministic for a given conversation. Also, it is the same for all packets and the packets never arrive out of order. (This is not the case for long haul networks. The route taken by each packet may be different, thus the delay associated with each packet may be different. Therefore, there is a possibility of the packets arriving out of order.) The variable delay, namely the queueing delay, depends on various factors such as the medium access protocol and the priority structure. The delay estimation methods are employed to estimate the time at which the first packet received by the receiver is generated. With this estimation a target playout time is set for the first packet in a *talkspurt*. The ensuing packets, in the same talkspurt are then scheduled to be played regularly after intervals of $(1/\mu)$, where μ is the rate at which the packets are generated. Montgomery [40] enlisted four methods that may be employed for delay estimation. The methods suitable in the context of LANs are,

1. *Blind delay method* : Receiver makes the worst case assumption about the delay encountered by a packet.

2. *Absolute timing method* : Synchronized clocks are maintained at the sender and the receiver and time stamps are used to determine the packet generation time explicitly.

We favor the blind delay method for the following reasons :

1. It is the simplest to implement.

2. The problem of clock synchronization for all potential conversing stations is a huge one, thus eliminating the absolute timing method.

The blind delay method assumes that the first packet suffered the minimal delay, i.e., only the fixed delay (D_F) and it also assumes that the other packets suffered significantly more delay. Therefore, it plays the first packet max $\{ D_V \}$ time units after it arrives and schedules the following packets at regular intervals of $(1/\mu)$ time units. The packets that suffer more delay than anticipated by this method are considered late.

4.1.2 Late packet policy

In [19], Gopal et al. analyzed two playout strategies for voice transmission. Both policies employ the blind delay method and leave it to the receiver to decide what to do with a late packet. The first strategy considers a late packet *lost* and the receiver does not play out anything during the time slot in which that packet is supposed to be played out. (For our purpose all that matters here is that the late packet is thrown out. For completeness, however, it must be mentioned that something else may be played during the same time slot. For example, the previous packet could be played again.) When the receiver uses the second policy, the receiver waits for the late packet, plays it whenever it arrives and reschedules the following packets to be played out at regular intervals of $(1/\mu)$. The first policy is easier to implement but risks losing intelligibility if the losses exceed 2 %. The second policy ensures no packet loss but still has the chance of losing intelligibility since the *gaps* produced by the late packets produce the glitches in a conversation. Also, as we found from our simulation, the percentage of the late packets increases if theses late packets are sent on the network. (This statement will be clearer when we describe our late packet policy.) Another problem arising from the second scheme is that one has to be careful to distinguish between a genuine late packet and a *silence* period.

The case for the first policy is stronger in the context of local area networks and we opted for it with a small modification. In the above schemes the receiver is responsible for the quality of voice, but in our strategy the sender is made responsible. The key argument here is that since the transmission and the propagation delays between the sender and the receiver are fixed and known, the sender knows when a packet is late and refrains from sending it, thus saving some transmission and propagation time. (This statement should shade light our earlier claim about the percentage of the packet loss increasing if the second late packet policy is adopted.) All the properties of Gopal et al.'s first policy are still carried over.

In conclusion, we have adopted a blind delay policy with the late packets not sent by the sender, thus making him partially responsible for the quality of voice.

4.2 Packet Size

In a system with real time constraints packet size becomes an important issue. In general, since a local area network does not use store and forward policy, bigger the packet size smaller is the delay. Apart from the implementation considerations, the following factors impose a bound on the size of the packets.

1. Message length distribution
2. The real time bound

4.2.1 Message length distribution

1. *Data* : We have followed the lead of other well known researchers in assuming an exponential distribution with the mean 1000 bits for all data messages.

Moreover, this distribution is convenient for the comparison of our protocol with other existing protocols.

2. *Voice* : Digitized voice is typically assimilated in the form of packets (instead of messages, as in the case of data). The voice packet length is determined by the sampling rate (in bits/sec) and the packetization period. In our system, the sampling rate is 64 kbits/sec. The packetization period is determined by the real time constraints and that is the topic of discussion in the next section.

3. *Documents* : The message length of documents packets is 50 K bytes after compression of a CCITT document size A4[1]. This is a conservative figure since further compression is possible.

4.2.2 Real time bound

4.2.2.1 Data Data traffic has no real time bound, i.e., there is no constraint that a data message must be delivered within a certain time. However, the delay suffered by a data message must be reasonable. Reasonability being subjective, it is the goal this research to guarantee a data delay obtained from the literature for the same utilization. The data traffic comprises of interactive and bulk and it is intuitive that interactive data should suffer less delay than bulk. This objective is also considered.

4.2.2.2 Voice Since the voice packet end-to-end delay must be less than 200 ms [20] for intelligible conversation, the packetization period cannot be arbitrarily large. In fact, smaller the packetization period, better is the quality of voice. One

[1]See Chapter 3.

must also keep in mind that the end-to-end delay of 200 ms is at the user level and not at the MAC layer level. This consideration further tightens the constraint of 200 ms to a smaller value at the MAC layer level. We have chosen this value to be 51 ms. (30 ms for packetization and 21 ms for waiting time after which the packet is discarded. The value 21 ms comes from the fact that a library packet is 20 ms long and 1 ms time is allowed to improve the probability of transmitting the voice packets without being late.) The packetization period cannot be arbitrarily small because due to the presence of other types of traffic, the availability of the communication channel is limited. This leads to a large number of late packets. If the percentage of late packets exceeds 2 then, again, voice conversations become intelligible [7].

4.2.2.3 Documents As stated in Chapter 3 a bound of 5 seconds is on the end-to-end page request delay is considered tolerable.

4.2.2.4 The interdependency between the real time bounds If a page were to be transmitted entirely as one packet the delay at the MAC level would consist of the following components:

- Queueing, transmission and propagation delay of the page request delay.
- The library request processing delay.
- The reply packet (a page) queueing, transmission and propagation delay.

The first component, though protocol dependent, is negligible. (Since the request packets are small, they can be given high priority without having adverse effect on other traffic.) The second component is technology dependent and is out of the hands of the protocol designer. This leaves the third component, namely, the reply packet delay, as the only component that can be manipulated for optimization. The total transmission time remains the same irrespective of how many packets a page is divided into and the propagation delay is small, hence the idea is to reduce the queueing delay to a minimum.

If a page forms only one packet then that keeps the channel occupied for 100 ms. Since the voice packetization is 30 ms, in rough terms, with 16 voice stations in talkspurt, about 53 packets are lost. Thus, pages must be divided into packets. Our simulations suggested division of the pages into 5 packets. (The division of five packets may not be completely optimal. The reason for this being that, an exhaustive simulation with all possible input parameters is difficult, if not impossible. However, with the values tried and the observed results, it is close to optimal.)

4.3 The Priority Mechanism

The token ring standard keeps provision for eight priority types, type 0 to type 7 [26]. However, the off-the-shelf components that are available may implement a lesser number of priorities. For example, the Texas Instrument chip set TMS 380 has only four priorities available to the user. For the system under consideration, it turns out that these priorities are indeed sufficient.

In a queueing system, specifically with cyclic queues, priorities can be implemented in two ways. The first way is that, at the logical link layer (LLC) or at a higher layer where a priority is first assigned to a protocol data unit (PDU), the PDU is then queued first with respect to its priority and then within its priority class on first-come-first-served basis. In other words, high priority PDUs are always in the front of the queue, even though there may be some low priority PDUs that arrived earlier at the station but are still at the end of the queue. The second way is that the PDUs are queued as they arrive at the station, irrespective of their priority and then at the medium access layer (MAC), using the standard token ring priority reservation scheme, the high priority data is given preference. The standard does not recommend either of the ways. As can be seen, the first way provides the best possible service to the high priority data, whereas the second way is fairer to the low priority data. With the availability of the off-the-shelf components, which implement the second way, we opted for the second choice.

4.4 Parameters of Interest

This section merely enlists the parameters of interest to be optimized.

4.4.1 Mean data delay (Interactive and bulk)

It is the time interval between the arrival of the data message at a station and the arrival of its last bit at its destination.

4.4.2 % voice packet loss

Voice packets out of synchronization are discarded by their source, that is, if voice does not get service within a certain time after it generates a voice packet it is discarded. Thus, the % voice packet loss at a source is defined as the number of packets discarded by the source divided by the number of packets generated by the source.

4.4.3 End-to-end page request delay

It is the time interval between the time at which a request is generated at a terminal and the time at which the requested page is received at the concentrator to which the terminal is attached.

5 THE SIMULATION MODEL

The aim of this chapter is two-fold:

1. To introduce the fundamental concepts associated with simulation, and

2. To demonstrate the use of these concepts in the context of this research.

Sections 1 to 7 of this chapter deal with various aspects of the important stages in simulation, followed by Sections 8 and 9 which describe the modeled protocols and the corresponding performance of voice, data and video. Sections 10 and 11 highlight the conclusions drawn from these simulations.

5.1 Fundamentals of Simulation

In this section we will concern ourselves with forming definitions for the terms which are often used in this research. A strict adherence to these definitions is attempted throughout this research.

System It is a set of elements, also called components.

Attributes The elements or components have attributes which have either numerical or logical values. The attributes either define the state or the characteristics of the elements.

Environment It is the world outside the system. External relationships connect the elements with the environment.

Inputs and Outputs The system is influenced by the environment through the inputs it receives from the environment. The input is then processed by the system to transform it into the output of the system.

Steady State A system is considered to be in steady state, if the probability of being in the that state does not depend on time.

Transient State If the system is not in steady state, then it is in transient state.

Stable System A system is considered stable if it returns to steady state after an external shock on the system.

5.2 Solution Methods

An analytical solution utilizes properties of the part of mathematics which consists of differential and integral calculus. The solution has a *formula* form, a formula that is true for a range of values for the variables that constitute it.

A popular solution method, called numerical method, usually with the aid of computers, substitutes numbers for independent variables and the parameters of the model and manipulates the equations defining the system. A tacit assumption about the convergence of the solution is made here. Monte Carlo methods and simulation methods are two such numerical methods.

Monte Carlo method in a wide sense could be defined as any technique for the solution of a model using random or pseudorandom numbers [30]. Random numbers are stochastic variables which are uniformly distributed on the interval [0,1] and are stochastically independent. Pseudorandom numbers, on the other hand, are generated by applying deterministic algebraic formulas which results in numbers that for practical purposes are considered to behave as random numbers, i.e., to be also uniformly distributed and mutually independent. Pseudorandom numbers have a cycle-length associated with them. The cycle length is the number of pseudorandom numbers generated before the same sequence of numbers is obtained again [30]. In this dissertation we have resorted to the pseudorandom numbers.

Simulation, in a narrow sense, is defined as experimenting with a model ,usually abstract, over time. Stochastic simulation involves experimenting with an abstract model over time by sampling the values of the stochastic variables from their distributions. Because of the extensive use of random numbers in this type of simulation, it is often called Monte Carlo simulation.

5.3 Modeling and Simulation

The need for a model, usually, is the result of identification of a problem or system which requires a solution [49]. Thus, a model is a representation of a system [4]. The representation , however, depends on the type of the model. With a model is normally associated a collection of variables that contain all the information necessary to describe the system at any point in time. A system consists of entities which are the components of the system that require explicit representation. Associated with the entities are attributes. Entities are involved in activities. Activities are durations of time which have definite lengths. These lengths may be constant, random with a defined probability density function or functions of past or present states. Delay is also a duration of time, but with indefinite or unknown length. The instant at which the delay duration begins, its end is even probabilistically not known. Models can be categorized into various types such as iconic model, symbolic model, discrete model, etc. The types are based on the style of modeling and/or the purpose of modeling.

5.3.1 Simulation vs mathematical models

5.3.1.1 Mathematical Models Mathematical models are characterized by one or a series of equations relating the parameters of interest that measure system performance to the variables which affect the system behavior. These equations are supplemented with the system constraints and the range of operation. The system variables which cannot be controlled may be called decision variables. An example of this is the channel bit rate. The variables are further classified as independent variables, which do not depend on any other variables and dependent variables which depend on the independent variables and other dependent variables.

5.3.1.2 Simulation The distinction between mathematical and simulation models is not clear. Examples normally prove to be useful in drawing a line between the two. Without examples, however, one could state the difference between the two as follows. Mathematical models try to explain the aggregate behavior of the system, while the simulation deals with the individual behavior of the modeled entities and draws inferences about the system behavior from it. Simulation models do have a mathematical side to them. For example, customer arrival rate at a server may be assumed to be Poisson. But, while a mathematical model would give a formula for, say, the average waiting time of a customer, simulation would measure the delay of each customer and predict the average waiting time. Various hybrid techniques combining the mathematical and simulation models have also been proposed [42].

5.4 Discrete-event Simulation

The simulation discussed later in this chapter is called discrete event simulation. *Discrete event simulation* is defined as the simulation of a system model which changes states in discrete points in time. The system state at any point in time is called a snapshot. Discrete event simulation is divided in three basic categories.

1. **Event Scheduling** : The simulation proceeds by arranging all events in chronological order (not all at once, but as the simulation is carried out) and advancing from snapshot to snapshot.

2. **Process interaction** : Allows modeler to represent the system as a set of processes. (For example, Network simulation in SLAM II)

3. **Activity scanning** : The modeler defines the conditions necessary to start and end activity and delay in the system. Simulation time is advanced in equal increments, and if the conditions are appropriate, an activity will be started or terminated.

We used discrete event simulation with event scheduling extensively in this research. The main reason for this is the versatility it offers regarding the actions involved with each event occurrence. The other option, namely the network model, demands inherent simplicity in the simulated models. This requirement is stringent in the case complex protocols such as token ring and token bus. A mixed model, both network and event scheduling combined, could have been used. But this approach was thwarted by the language limitations at the time of simulation. (For example, the maximum number of stations were limited to 50.)

5.5 Modeling Input Processes

The discussion so far was limited to modeling the system without paying much attention to how one must model the input processes to provide the best inference about the system. The ensuing discussion focuses on that aspect of simulation. There are several important computer model characteristics that have a direct bearing on the model input process [27]:

1. The computer model may require many inputs. For example, Arrival processes, Service times.

2. The computer model output may exhibit discontinuities with respect to some of its inputs.

3. The computer model inputs may not behave independently of one another.

4. The computer model predictions may be nonlinear, multivariate, timedependent functions of the inputs.

5. The relative importance of the individual inputs may be a function of time.

5.5.1 Why use probabilistic modeling of inputs?

One could simulate by fixing all inputs except for one and vary values of the one that is not fixed. Such *one-at-a-time* approaches do not provide an efficient way of performing a screening analysis, provide only conditional information, and may be prohibitively expensive. It is highly unlikely that discontinuities in the output would be detected with a *one-at-a-time* approach.

5.5.2 Advantages of the Monte Carlo method

1. If properly done, Monte Carlo modeling can be designed to avoid the pitfalls mentioned above.

2. The Monte Carlo approach varies all inputs simultaneously, thoroughly explores the input space, and can be made very efficient.

3. If the probability distributions assigned to the inputs are meaningful the statistical estimates of output quantiles, means and variances can be made.

5.5.3 How to determine modeling distributions?

1. Just a range of values [a,b] for the input without any pdf.

2. A pdf over a range [a,b] either known or based on expert opinion.

3. Empirical data without a fitted pdf. (Use an empirical distribution function).

4. Empirical data to which a pdf is fit using maximum likelihood procedures and tested with goodness-of-fit technique.

5. Prior and posterior distributions using Bayesian techniques.

In this research we have resorted to methods 1 and 2.

5.6 Credibility Assessment of Simulation Results

Three basic types of errors may be committed in conducting a simulation study [3].

1. **Type I error** : Type I error is committed when based on the study results a hypothesis is rejected when in fact it is true. (Model builder's risk).

2. **Type II error** : Type II error is committed when based on the study results a hypothesis is accepted when in fact it is not true. (Model user's risk).

3. **Type III error** : Type III error is committed when the formulated problem does not completely contain the actual problem. (i.e., solving the wrong problem).

The standard technique of estimating the confidence intervals is used to reflect the type I and II errors. Type III error is minimized, in this research, by using the finite state models to design, develop and debug the simulation software. This approach made changes in the simulated protocols easy and reliable. In addition, previously published results were used as check points for verification in the early stages of the simulation software.

5.7 Variance Estimation

This aspect of simulation pertains to the output, namely, the performance parameters of interest. The significance of the variance estimation and reduction stems from the following facts.

- Simulation results are produced using random number streams and hence the results change if the generated random number input sequence is changed. So the question is how reliably can we state that the output will indeed repeat itself or will be close enough if the simulation is run again ? This indicates that a mere value, such as the mean, of a performance parameter is not sufficient.

- The statement above brings about the next question. How to express the required reliability ? Stating informally, an interval (a,b) is called a $(1-\delta)\times 100\%$ confidence interval of a performance parameter μ when, if the simulations runs were to be repeated independently of each other, the estimated value of μ, $\hat{\mu}$, would fall in the interval (a,b), $(1-\delta)\times 100\%$ of these runs.

In conclusion, a need for the confidence interval is felt. Estimation of the confidence interval involves the estimation of the variance of the parameter of interest as obtained from the simulation, and implementation of various variance reduction techniques to minimize it. In this research, three variance estimation methods were considered [44].

5.7.1 Brute force (Replication)

Several simulation runs, independent of each other, are conducted. If $\hat{\mu}$ is the estimator of the parameter μ, then for n runs, n independent $\hat{\mu}_i$'s are obtained and $\hat{\mu} = \sum_{i=1}^{n} \hat{\mu}_i/n$. A variance estimator for μ is similarly obtained. As the name suggests this method is indeed a *brute force* method, since for obtaining low variances n must be large. This increases the simulation run time and may make it prohibitively high, especially on slow machines.

5.7.2 Subintervals

To overcome the time consuming nature of the brute force method, one simulation run may be divided into n subintervals to compute $\hat{\mu}$ for each subinterval and assuming them to be independent of each other. This reduces the run time by the order n. However, finding a suitable interval size to justify the independence assumption is a difficult task and there is no theoretical basis to facilitate it.

5.7.3 Regeneration

The regeneration theoretically solves the problem of selecting suitable intervals in a simulation run to make the $\hat{\mu}_i$'s independent. This method is applicable when the output process (from which the parameters are estimated) is a stochastic process, or can be approximated to one. This process is called *regenerative* if it always returns to some condition (may be a state), called regenerative condition, such that the future behavior of the system from then on is independent of its behavior in the past and is always governed by the same probability law [12]. Returning to a regenerative state gives rise to n cycles. $\hat{\mu}_i$ is measured in each of these cycles and the variance is estimated as before. The method is theoretically well supported, that is, the $\hat{\mu}_i$'s are indeed independent, but it faces some practical difficulties. First, the number of cycles 'n' is not known at the beginning of the simulation, so the simulation time for a fixed 'n' is unknown. Second, in a computer networking situation, at high utilization, these cycles become fewer and fewer, causing a long simulation run. Third, for some systems exact regeneration is impossible, thus ruling this method out. We have, however, opted in favor of an approximate version of this method due to its sound theory.

5.7.4 The regenerative method : Theoretical background

In the above discussion the notion of the independence of the observations was frequently mentioned. The question is, why is it so important ? The explanation is that the confidence intervals are obtained by applying the *Central Limit Theorem* (CLT), which states that if $X_1, X_2, \ldots, X_n$ is a random sample from the distribution with a mean μ and variance $\sigma^2 < \infty$, then the limiting distribution of

$$Z_n = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma}$$

is a standard normal, Z_n converges in distribution to $Z \sim N(0,1)$ as $n \to \infty$. Thus for satisfying the condition of being a random sample the observations for variance estimation must be independent. (More on the relationship between the confidence interval and the CLT is in the section on confidence interval.)

Without using the regeneration theory each sample in a simulation run cannot be considered independent. For example, in a simple queueing scenario with one server and one queue (G/G/1), waiting time of the first customer is always zero, thus biasing the mean waiting time for short simulation runs to zero. Once the queue builds up, if the k^{th} customer waits for very long then, the chances are, so will the $(k+1)^{th}$ customer. This means that the k^{th} and $(k+1)^{th}$ observations are correlated, thus making independence assumption inapplicable. The regenerative method ensures the independence of the observations for confidence interval estimation. In the ensuing discussion we will focus only on discrete time regenerative processes, which are adequate for discrete event simulation.

Definition 5.7.1 (Regenerative process) *A sequence $\{\vec{X}_n, n \geq 1\}$ of random vectors in k dimensions is a regenerative process if there is an increasing sequence $1 \leq \beta_1 \leq \beta_2 \ldots$ of discrete times, called regeneration epochs, such that at each of these epochs the process starts afresh probabilistically according to the same probabilistic structure governing at epoch β_1 [12].*

That is between any two consecutive regeneration epochs β_j and β_{j+1}, say, the portion $\{\vec{X}_n, \beta_j \leq n \leq \beta_{j+1}\}$ of the process is an independent and identically distributed replicate of the portion between any other two consecutive regeneration epochs. The portion $\{\vec{X}_n, \beta_j \leq n \leq \beta_{j+1}\}$ of the process is called the j^{th} cycle. The portion between epoch 1 and β_1 is allowed to have a different distribution. Let $\alpha_j = \beta_{j+1} - \beta_j$ for $j \geq 1$. It is the sojourn time of the system in the j^{th} cycle. $\{\alpha_j, j \geq 1\}$ is a set of independent random variables with identical distributions. Assume that $\mathrm{E}\{\alpha_1\} < \infty$, which holds for most queueing situations of practical interest. With these definitions the regenerative processes hold the following properties, which are stated without proof.

1. Under very mild conditions a regenerative process has a steady state distribution. (The process needs to be aperiodic, a condition usually satisfied and $\mathrm{E}\{\alpha_1\} < \infty$.)

2. There exists a random k-vector $\vec{X}$ such that the distribution of $\vec{X}_n$ converges to the distribution of $\vec{X}$ as $n \to \infty$, that is, $\lim_{n\to\infty} P\{\vec{X}_n \leq \vec{x}\} = P\{\vec{X} \leq$

$\vec{x}\}$.

Let f be a measurable function in k dimensions [15] having real values, and let the simulation be designed to estimate $r \equiv E\{f(\vec{x})\}$.

Our goal now is to devise a methodology to obtain a confidence interval for r. Let

$$Y_j = \sum_{i=\beta_j}^{\beta_{j+1}} f(\vec{X}_i)$$

Then,

1. The sequence $\{(Y_j, \alpha_j), j > 1\}$ consists of independent and identically distributed random vectors.
2. If E$\{|f(\vec{X})|\} < \infty$ then $r = E\{f(\vec{X})\} = \frac{E\{Y_1\}}{E\{\alpha_1\}}$

With the above four properties the problem has now reduced to [12]:

Given the independent and identically distributed observations $\{(Y_j, \alpha_j), j \geq 1\}$, estimate $r = \dfrac{E\{Y_1\}}{E\{\alpha_1\}}$.

5.7.5 Estimating the confidence interval

With the above basic theory, the following derivation gives a method for estimating the confidence intervals. Our goal is to estimate the confidence interval of r, when the simulation is run for n cycles. Let $U_j = Y_j - r\alpha_j$. Thus, U_j's are i.i.d. and $E\{U_j\} = E\{Y_j\} - rE\{\alpha_j\} = 0$.

$\bar{Y}, \bar{\alpha}$, and $\bar{U}$ are the sample means.

$$\bar{Y} = \frac{1}{n}\sum_{j=1}^{n} Y_j$$

$$\bar{\alpha} = \frac{1}{n}\sum_{j=1}^{n} \alpha_j$$
$$\bar{U} = \frac{1}{n}\sum_{j=1}^{n} U_j$$

and $\bar{U} = \bar{Y} - r\bar{\alpha}$. If $\sigma^2 = E\{U_j^2\}$ and $0 < \sigma^2 < \infty$ then using the CLT,

$$\lim_{n\to\infty} P\left\{\frac{n^{1/2}\bar{U}}{\sigma} \leq x\right\} = \Phi(x) \qquad (5.1)$$

for each $x \in \Re$, where Φ is the standard normal distribution function. Hence,

$$\lim_{n\to\infty} P\left\{\frac{n^{1/2}[\bar{Y} - r\bar{\alpha}]}{\sigma} \leq x\right\} = \Phi(x) \qquad (5.2)$$

and therefore,

$$\lim_{n\to\infty} P\left\{\frac{n^{1/2}[\hat{r} - r]}{\sigma/\bar{\alpha}} \leq x\right\} = \Phi(x) \qquad (5.3)$$

where $\hat{r} = \bar{Y}/\bar{\alpha}$. A confidence interval still cannot be produced from Equation 5.3, since σ is unknown. We have,

$$\sigma^2 = Var\{Y_1\} - 2rCov(Y_1, \alpha_1) + r^2 Var\{\alpha_1\}$$

Let s_{11}, s_{22} and s_{12} be, respectively, the sample variances of Y_j's, α_j's, and the covariance of (Y_j, α_j)'s, that is,

$$s_{11} = \frac{1}{n-1}\sum_{j=1}^{n} (Y_j - \bar{Y})^2$$
$$s_{22} = \frac{1}{n-1}\sum_{j=1}^{n} (\alpha_j - \bar{\alpha})^2$$
$$s_{12} = \frac{1}{n-1}\sum_{j=1}^{n} (Y_j - \bar{Y})(\alpha_j - \bar{\alpha})$$

Now let,

$$s^2 = s_{11} - 2\hat{r}s_{12} + \hat{r}^2 s_{22}$$

It can be shown that $s^2 \to \sigma^2$ with probability 1 as $n \to \infty$. Therefore,

$$\lim_{n\to\infty} P\left\{\frac{n^{1/2}[\hat{r} - r]}{s/\bar{\alpha}} \leq x\right\} = \Phi(x) \qquad (5.4)$$

Let $Z_\delta^* = \Phi^{-1}(1 - \delta/2)$, then from Equation 5.4 we have,

$$\lim_{n\to\infty} P\left\{-Z_\delta^* \leq \frac{n^{1/2}[\hat{r} - r]}{s/\bar{\alpha}} \leq Z_\delta^*\right\} \simeq 1 - \delta$$

for large n. This may be rewritten as,

$$P\left\{\hat{r} - \frac{Z_\delta^* s}{\bar{\alpha}n^{1/2}} \leq r \leq \hat{r} + \frac{Z_\delta^* s}{\bar{\alpha}n^{1/2}}\right\} \simeq 1 - \delta \qquad (5.5)$$

Hence the $100(1 - \delta)\%$ confidence interval for r is,

$$\hat{I} = \left[\hat{r} - \frac{Z_\delta^* s}{\bar{\alpha}n^{1/2}}, \hat{r} + \frac{Z_\delta^* s}{\bar{\alpha}n^{1/2}}\right] \qquad (5.6)$$

5.7.6 Discrete-event simulation and the regenerative approach

In discrete-event simulation, the events are generated recursively. That is, the generation of one event leads to another event which, in turn, generates a future event. In a simple queueing case, where there is one queue and one server a vector $\vec{X}(t)$ given as,

$$\vec{X}(t) = \begin{pmatrix} Q(t) \\ S(t) \\ A(t) \end{pmatrix}$$

represents the system state at any given time t, adequately. $Q(t)$ is the number of customers in the queue, $S(t)$ is the time upto the next service completion event and $A(t)$ is the time upto the next arrival event. (Note that for Markov arrival and service processes, $\vec{X}(t) = Q(t)$ would suffice.) It should be noted that to be a regenerative epoch $Q(t)$ need not be equal to 0, since any state that the system returns to, such that the future behavior of the system is independent of its behavior before that state, is a regeneration state (epoch). Thus, a simulation programmer may define a state as regeneration state and may collect samples accordingly.

5.7.7 Approximation techniques

The critical requirements in the above methodology are [12],

- that the process repeatedly returns to some fixed state,
- that the expected time between returns is finite, and
- that the process starts afresh probabilistically each time it enters that state.

Thus, the regenerative method, though appears to be attractive does not lend itself well in situations where the expected time between returns to the regeneration state is not finite or is very large. In most practical situations with continuous distributions (such as exponential service and inter-arrival times) the expected time between returns is not finite. The approximate regeneration scheme circumvents this problem by approximating the original process to a regenerative process. Mathematically, $\beta_k(\epsilon)$ denotes the k^{th} time the process enters the trapping interval $(\vec{X}_0(t) - \epsilon, \vec{X}_0(t) + \epsilon)$, where $\epsilon > 0$ is fixed and $\vec{X}_0(t)$ is the original regeneration state. The times $\beta_k(\epsilon)$ are approximated to β_k in the hope that for small ϵ the system would behave like the original regenerative process. The analysis from this point on is same as before. For the sake of completeness, it is imperative to mention that *Partial State-Space Discretization* is another technique for approximating regenerative processes [12]. In this research we have resorted to the approximation regeneration scheme.

5.7.8 Defining the state vector

The systems simulated in this research employ discrete event simulation technique. Due to their complexity, to define these systems completely at any point in time large state vectors are needed. For example, at any point in time, the state vector must have components depicting,

1. The token priority and its position in the ring.

2. The number of packets in each station queue and their order. (Since some stations have packets with different priorities).

3. The remaining time for the arrival of the next message at each station, and its priority.

4. The remaining service time of the message that is being served currently.

This information, if to be conveyed completely, could require state vectors of enormous dimensions. In this research, we have reduced the dimension of the state vector, RSTATE(t), to five, by focusing only on the components of most importance. The state vector represents,

1. The number of document packets in the library queue.
2. The number of voice packets, all queues lumped together.
3. The number of interactive data packets in all queues.
4. The number of bulk data packets in all queues.
5. The token priority.

Obviously, a large amount of information is lost in the process. We contend that, since voice, bulk and interactive data generate light traffic, they have less influence in defining the structure of the state vector, for being given separate representation to each station. Th library traffic, being relatively influential, is given separate representation (First component of RSTATE(t)). The remaining service time and the next arrival events are not represented. This fact, obviously causes errors, but our results indicate that it is tolerable. With this definition of RSTATE(t) in mind, ϵ is also a five-dimensional vector, which represents the error window for determining the regeneration state.

5.7.9 Variance reduction techniques (VRT)

It is evident from Equation 5.6 that higher the value of s for a given n, larger is the width of the confidence interval. The Variance Reduction Techniques (VRT) are employed for reducing the value of estimated variance by replacing the *straight on* or *crude* sampling by more sophisticated sampling. In this research no VRT techniques were incorporated in the simulations because there are no direct methods that can be employed in token ring networks easily. It is, however, an area for further research [30,44].

5.7.10 Numerical Validation of the use of reduced state vector

Due to lack of any plausible theoretical explanation to justify the choice of the sate vector, which is chosen intuitively, in this section we offer a numerical comparison of the time saved and the obtained confidence interval. The busy cycles for the video traffic and the other traffic (data and voice) were separated. A busy cycle was always started by the library station whenever the defined regeneration state was reached. For document packets, at every fifth packet a new busy cycle was started, if all other queues were found empty. (The number five indicates the number of packets per page.)

The input parameters for the simulations conducted to produce the results listed in Table 5.1 and Table 5.2 are given in the next section, and hence are not repeated here. The important points to note is how the simulation run time is saved by using our approximate version of the regenerative method to obtain the same level of confidence interval, or conversely, with the same simulation run time, the difference in confidence interval levels. Voice packet loss confidence interval was obtained by considering each voice conversation total packet loss as an independent observation. This methodology is used for the rest of the chapter.

5.8 Protocol Simulation

The protocols chosen for simulation, using the above techniques, are based mainly on intuition and past experience. Furthermore, the available protocols from the literature played a significant role in the choice of protocols to be simulated and tested.

Table 5.1: Four Simulations and Confidence Intervals

	Brute Force		Regeneration	
	Mean Wait.	Time	Mean Wait.	Time
Bulk Data (ms)	12.57	4	12.19	4
95 % Conf. Int.	± 1.11		0.90	
Int. Data (ms)	8.57	4	8.19	4
95 % Conf. Int.	± 0.86		± 0.62	
Library Req. (s)	0.8584	4	0.8549	4
95 % Conf. Int.	± .40		± 0.084	

Table 5.2: Simulation: Time Comparison

	Brute Force		Regeneration	
	Mean Wait.	Time	Mean Wait.	Time
Bulk Data (ms)	12.36	16	12.13	8
95 % Conf. Int.	± 0.68		0.75	
Int. Data (ms)	8.42	16	8.17	8
95 % Conf. Int.	± 0.45		± 0.51	
Library Req. (s)	0.7069	16	0.7089	8
95 % Conf. Int.	± .1546		± 0.0615	

As mentioned in [28] most protocols utilize the idea of prioritizing the traffic type, namely, voice, data and the documents. Blind prioritization as will be evident, does not necessarily optimize the performance of even the highest priority traffic. In fact, it may even worsen the performance of the high priority traffic. The following simulation studies were conducted to study the effect of prioritization and to draw conclusions about its suitability in our application.

5.8.1 The input parameters

- The input variables consist of voice, data and document traffic, the characteristics of which are described in Chapters 3 and 4.
- The channel utilization by the data traffic, contributed to by 57 stations, varies from 0.1 to 0.3. It is constant for each simulation run. The data comprises of 40% interactive and 60% bulk.
- Sixteen simultaneous voice conversations are in effect on the network, simulated using Brady's [7] findings. (The number sixteen is coincidentally significant, since Brady's experiments were conducted with 16 telephone conversations. We chose the number sixteen because of the language limitations.)
- Ten concentrators request documents at the rate 0.33 pages per second per concentrator.

5.8.2 The constraints

- Channel bit rate = $4 Mbits/sec$
- Implementation feasibility.

5.8.3 Controllable parameters

- The priority structure (Maximum priorities = 4).
- Various timer values. The timers include the ones specified in the IEEE 802.5 standard, such as the token hold timer, and the ones implemented in modification of the protocol.

5.8.4 Parameters to be optimized

- End-to-end data delay, for interactive and bulk data.
- End-to-end page request delay for documents.
- Percentage of lost voice packets.

5.9 Simulated Protocols

5.9.1 Same Priority Protocol (SPP)

As the first step, all traffic types were assigned the same priority and the standard token ring protocol was simulated. The results, though disastrous, were not surprising. Intuitively, since the concept of real time bounds is not addressed by this protocol, it is expected to fail. The simulation, however, was still carried out to put the prioritization into perspective by comparison. Table 5.3 shows the results.

5.9.2 Four Priority Protocol (FPP)

This protocol uses the standard token ring protocol, using four out of eight possible priorities for the four traffic types. The documents, bulk data, interactive

Table 5.3: Same Priority Protocol

Data Util.	Mean Int. Delay (ms) (95% Conf. int)	Mean Bulk Delay (ms) (95% Conf. int)	End-to-end Page Request Delay (sec) (95% Conf. int)	% Lost Voice Packets
0.1	9.119±0.525	9.119±0.525	0.305±0.030	13.19±1.03
0.2	10.597±0.780	10.597±0.780	0.3569±0.077	24.74±1.74
0.3	14.571±0.719	14.571±0.719	0.4815±0.064	33.68±1.80

Table 5.4: Four Priority Protocol

Data Util.	Mean Int. Delay (ms) (95% Conf. int)	Mean Bulk Delay (ms) (95% Conf. int)	End-to-end Page Request Delay (sec) (95% Conf. int)	% Lost Voice Packets
0.1	8.048±0.526	9.270±0.696	0.5388±0.071	4.81±0.33
0.2	8.635±1.070	11.62±1.480	0.8912±0.146	4.09±0.12
0.3	9.962±0.269	16.02±0.333	2.236±0.778	4.23±0.64

data and voice use the priorities 0, 1, 2 and, 3, respectively. Intuitively, it may seem that this protocol should guarantee satisfactory voice performance. It turns out, as listed in Table 5.4, that this is not so. Voice packet loss exceeds 2%, thus, causing intelligible conversations. The conclusion section of this chapter explains this anomaly.

Table 5.5: Centralized Priority Protocol

Data Util.	Mean Int. Delay (ms) (95% Conf. int)	Mean Bulk Delay (ms) (95% Conf. int)	End-to-end Page Request Delay (sec) (95% Conf. int)	% Lost Voice Packets
0.1	7.760±0.595	8.715±0.709	0.4933±0.047	9.14±0.62
0.2	9.527±0.771	11.053±0.893	0.7725±0.0627	11.82±0.94
0.3	11.367±1.030	14.246±1.292	1.179±0.218	15.88±1.77

5.9.3 Centralized Priority Protocol (CPP)

This scheme is implemented by the International Business Machines (IBM). A matter of note here is that with IBM's implementation [9] only two priorities are possible. In our implementation the idea is extrapolated to materialize four priorities. The key idea in this scheme is that only one station is allowed to change the token priority. This station, called the Synchronous Bandwidth Manager (SBM)(The library station is the designated SBM), controls the number of calls and video sessions below a predetermined value to maintain quality service on the ring. The SBM interrupts asynchronous operation at a station if it sees the reserve bits set to a higher priority. Once all the high priority stations get a chance to transmit the priority is lowered again. To ensure fairness, the asynchronous station that was interrupted, remembers it, and gets the chance to transmit first. This operation becomes complex as the priorities increase and as our results (Table 5.5) verify this protocol does not perform well.

Table 5.6: Three Priority Protocol

Data Util.	Mean Int. Delay (ms) (95% Conf. int)	Mean Bulk Delay (ms) (95% Conf. int)	End-to-end Page Request Delay (sec) (95% Conf. int)	% Lost Voice Packets
0.1	7.216±0.342	9.283±0.415	0.5528±0.063	5.32±0.31
0.2	7.964±0.371	12.067±0.560	0.8231±0.099	4.90±0.38
0.3	9.028±0.213	15.663±0.427	1.985±0.384	4.83±0.18

5.9.4 Three Priority Protocol

In the light of the performance by the FPP, intuitively, this protocol is not expected to do better. Even so, this protocol serves as a stepping stone for our proposed modification. The assigned priorities for our proposed modification. The assigned priorities are 0, 1 and, 2 for documents, bulk data, interactive data, respectively. Voice is now assigned priority 2. As expected, more voice packets are lost than FPP (See Table 5.6).

5.9.5 Modified Three Priority Protocol

In the absence of the documents transmitted over the network, the voice traffic gets the best possible service, that is, no voice packets are lost. Thus, it is evident that the token ring priority scheme favors high priority traffic considerably, if it is given a chance to reserve the priority and a high priority token is issued quickly. Hence, most of the lost packets are lost during or after the transmission of a large packet, namely, a document packet. The idea used in this protocol, that circumvents this problem, is as follows. The priority of a voice packet is stepped up from 2 to

Table 5.7: Modified Three Priority Protocol

Data Util.	Mean Int. Delay (ms) (95% Conf. int)	Mean Bulk Delay (ms) (95% Conf. int)	End-to-end Page Request Delay (sec) (95% Conf. int)	% Lost Voice Packets
0.1	7.450±0.504	9.558±0.723	0.5928±0.099	1.57±0.38
0.2	8.310±0.494	12.14±0.739	1.001±0.158	1.15±0.16
0.3	9.408±0.489	15.585±0.932	2.146±0.467	1.16±0.15

3 only if waits for more than a certain time (less than the document transmission time). Thus, long awaiting voice packets have higher priority over the other voice packets. This key step improves the performance of the protocol drastically (See Table 5.7). In the a later section, we have discussed why this protocol performs better.

5.9.6 The effect of asymmetric nodes on synchronous traffic

Although, the literature in queueing theory is biased towards balanced traffic due to its simplicity and upto a certain extent, reasonability, from time to time, asymmetric traffics are studied and are commented upon. The emphasis so far, in the author's view, has been on non-real-time traffic, that is on how data delays depend on asymmetry. Bux pointed out in [8], with a heuristic explanation, that the balanced traffic delays in case of exhaustive service are indeed the worst case data delays, thus giving an upper bound on system performance. Ferguson and Aminetzah in [16] analytically computed, in the exact form the delays at the nodes to observe the effect of their distance from an asymmetric node. This effect is lost in Bux's approximation and it turns out that it is significant. Coming back to the initial problem of real time traffic, the voice stations in our simulation are set up such that there are a few adjacent (that is no data stations in the middle) downstream from the asymmetric node, namely, the library. (The term downstream refers to the relative positioning of the voice stations with respect to the direction of token passing.) In this section simulation outputs are given to observe the pattern of voice packets lost with respect to their distance from the library. Figures 5.1, 5.2 show the loss pattern in the CPP and, FPP, TPP and MTPP protocols.

Though no linear relationship can be obtained between the distance of the stations from the library and the number of packets lost, it is obvious from the figures that there is a general tendency to lose more packets as the distance increases. The MTPP protocol is proposed for smoothening the effect of distance by using selective high priority assignment.

5.10 Discussion of Results

The following conclusions are drawn from the performance of the protocols studied.

1. As expected intuitively, SPP performs badly as far as data and voice are concerned. SPP has the best performance for document end-to-end delay. Overall, the performance of this protocol is not acceptable. The performance of SPP, thus, merely underscores the importance of prioritization.

2. As noted by Kleinrock in [32] "You do not get something for nothing." In other words, preferential treatment given to one class of traffic is at the expense of

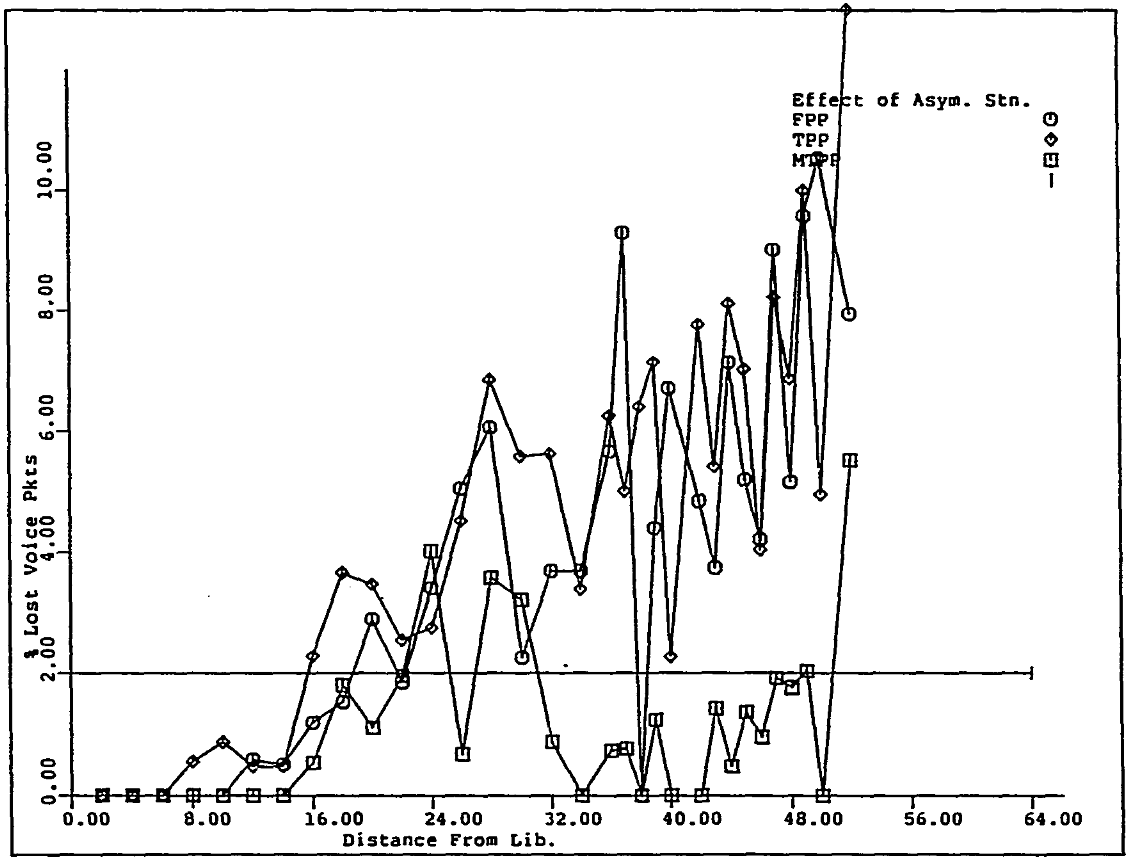


Figure 5.1: Effect of distance on CPP

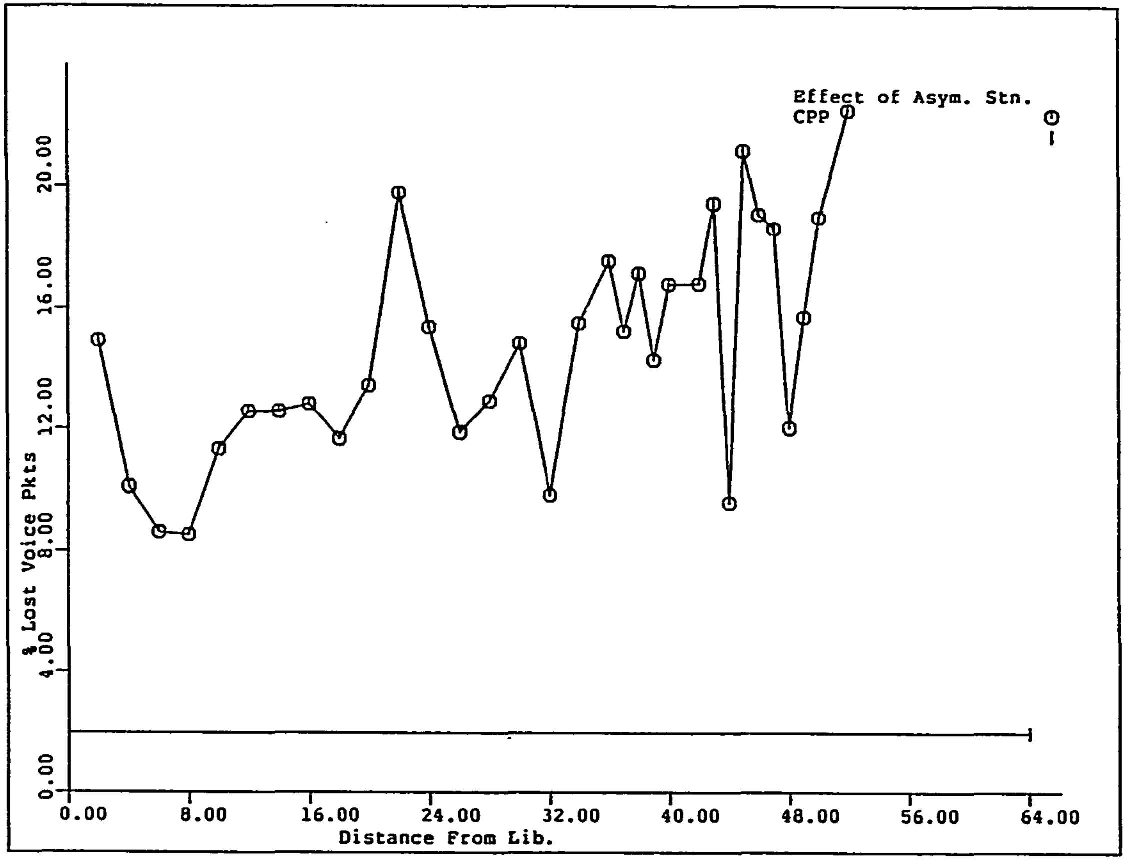


Figure 5.2: Effect of distance on FPP, TPP, MTPP

other traffic. Though the system under consideration is not conservative (because work is destroyed within the system when voice packets are discarded by the source), it may be approximated to one because the amount of work destroyed is negligible. Thus, when a traffic type is given preference (or high priority) care should be taken to minimize its effect on other traffic types. Illustrating this by example, the documents suffer a delay in seconds, say, 1.2 sec. An additional delay of a few milliseconds, say, 50-100, will not be noticeable at the user level. (But, the same additional delay will cause problems for data.) Now, if this additional daly for the document traffic can somehow enhance the performance of, say, voice traffic, then it is a good trade-off. This idea is used here when protocols are compared.

3. CPP performs better than SPP but the voice performance is still not satisfactory. In all fairness to the centralized protocol, it must be stated that its performance could probably be enhanced with clever modifications. We, however, opted for a distributed protocol due to its better reliability.

4. FPP and TPP perform better than CPP and have close performance parameters, but both exceed 2% voice packet loss.

5. The Modified Three Priority Protocol (MTPP) performs the best on all accounts, at the expense of some extra delay for document retrieval. The reason for better performance of MTPP may be explained as follows. When a document is being transmitted the channel is occupied for a long time (20 ms). Voice packets generated at the beginning of this time have to wait for a long time. As the voice packets wait they approach their discarding time. Now,

once the document transmission stops, in FPP and TPP all voice stations with high priority vie for the token irrespective of whether their packets are late. This may delay the transmission of the late packets farther downstream from the library. By using MTPP, only those packets that are late are given a chance to transmit first. This saves a lot of packets from being lost.

5.11 Summary

In conclusion, the design of a voice-data integrated token ring network is now complete. We have opted for the Modified Three Priority Protocol. Though the performance of the protocol may change with the change of parameters, the concept can be extended to any combination of input parameters. The distributed nature of the protocol and its implementation feasibility makes it reliable and attractive.

As a byproduct of the simulation studies we have also demonstrated the use of regenerative techniques and a technique to reduce the simulation time. As a follow-up on this work efforts will be devoted to choosing values of ϵ for the best results and variance reduction techniques.

6 ANALYTICAL MODELING

6.1 Prologue

In most practical queueing situations, the systems may be categorized as, in queueing theory parlance, G/G/1 systems. In other words, most queueing systems have a general (G) customer arrival distribution, a general (G) service time distribution and m servers. However, closed form expressions for the parameters of interest (For example, waiting time) are still unknown [1]. Typical token ring and token bus systems fall into the category of systems with cyclic service, where a server serves multiple queues in a predetermined fashion. In the strictest sense, in real networks, this gives rise to a G/G/1 system. The problem of obtaining analytical closed form expressions for the parameters of interest is complicated further by the incorporation of prioritized data, i.e., by not having a strict first in first out (FIFO) order of service at every queue. Thus, due to lack of mathematical tractability, approximations are made to make the problem tractable. A general tendency is to somehow make the problem a Markovian one and simplifying it. Most approximations, however, tend to solve a local problem, meaning thereby that the solutions do not hold under all types of traffic patterns, service times and other input parameters. For example, an approximation that works fine in a symmetrical traffic pattern may produce disastrous results for asymmetric traffic case. Similarly, the stage at which an approximation is made is also vital, since that is what determines if the modeler is obtaining the exact solution to an approximate model or pursuing the approximate solution to an exact model.

In [36], Leon-Garcia noted three general trends in performance modeling. The first trend is that the *closed form* solutions are being replaced by a set of linear equations, that is, the problem is considered done when a set of linear equations are obtained, which when solved numerically, would yield the desired parameters. The emphasis in this approach is on reducing the complexity of computation, while solving the equations, as much as possible [16]. A second trend is the use of *decomposition* approach. In this approach, a *test queue* is isolated from the system, and the input and output processes are approximated to express the effect of other entities in the system. Kuehn in [33] and Karvelas and Leon-Garcia in [29] are examples of this and are discussed later in this chapter. The third trend pointed out by Leon-Garcia is about the continuation of the use of simulation models to validate analytical models. A change in this trend is expected in the direction of actual network measurements. Many networks are in operation now and a glut of measurements is expected to appear to support or contradict modeling approximations.

In the following discussion we will first take a look at the difficulties encountered in modeling our protocol, followed by our analysis for modeling voice as a batch process and a comparison with similar work by Karvelas and Leon-Garcia [29].

6.2 Definitions

Before we plunge into the queueing model, some definitions are due. These definitions are stated in the context of token passing systems and are used consistently.

6.2.1 Exhaustive service discipline

This is a service discipline in which a node empties all the messages it has into the network before it passes the token to the next node.

6.2.2 k-Nonexhaustive service discipline

In this service discipline the node that has the token transmits only k or less (if it has less than k messages) messages before passing the token to the next node. A special case is 1-Nonexhaustive service discipline.

6.2.3 Gated service discipline

In this service discipline the node that has the token transmits only those messages that arrived before the free token was acquired.

6.2.4 Waiting time

It is the time elapsed between the arrival of the message at a node and the transmission of its first bit. (Here, the term message is used as the message at LLC layer, this message may be just a part (packet) of a higher level user message.)

6.2.5 Transmission time

It is the time elapsed between the arrival of the message at a node and the arrival of its last bit at its destination. Notice that for a given source-destination pair the transmission time differs from the waiting time only by a constant.

6.3 The Modeling Difficulties

Various difficulties arise when a modeling endeavor in undertaken in a system such as the one described in previous chapters. The initial goal of this research was to model the entire system with the proposed protocol. This goal, though was considered unrealistic, was expected to provide insight into subsystem modeling, that is, modeling parts of the system to give numerical solutions to some parameters of interest. We were able to attain this goal partially.

First we will look at the difficulties encountered in modeling the protocol in its entirety.

1. Though the data traffic may be justifiably assumed to have Poisson arrival rate, the assumption does not hold for voice and the document traffic. With the lack of this assumption and therefore the desirable *memoryless property*, the system analysis becomes complex, if not impossible.

2. Introduction of timers is a must in all systems with real time constraints. The queueing theory is not yet sufficiently developed to analyze multiple queues with, possibly, interrupt driven service time.

3. The system under consideration is not conservative [32], that is, work is destroyed in the system when voice packets are lost. Thus, most developments in

queueing theory, which mainly apply to conservative systems are not directly applicable. An approximation could be made that since the loss is negligible the system is almost conservative.

4. The priority mechanism complexity of the standard token ring is overwhelming for the available mathematical tools. Priority models are developed for queues with restricted queueing mechanisms, however, none of them apply to a system with complex reservation scheme such as the token ring.

The reasons mentioned above have led to the lack of numerical analytical solutions. Thus, our goals have been reduced to analyzing a subset of the entire system to shed some light on some performance parameters.

6.4 The Analysis Goals, Notation and Assumptions

6.4.1 Goals

Since at low data utilizations, 1-nonexhaustive and exhaustive system behaviors are close, as the first step, we will analyze a system with token ring configuration and 1-nonexhaustive policy. Our focus is on being able to predict the mean data delay, characterizing the voice traffic, modeling it as a batch process and capturing its effect on the data traffic. Also, it is desired to estimate the mean voice delay. Another goal was to be able to predict the pattern of lost packets by computing the voice delay distribution. Some numerical difficulties arose and hence we report only the method to compute the tail-end probabilities without any results.

6.4.2 Notation

The following notation was used in this analysis[1]:

- The arrival process at a station j, A_j, has the following Cumulative Density Function (CDF), $P\{T_{A_j} < t\} = A_j(t)$.
- H_j is the service time at station j with CDF, $P\{T_{H_j} < t\} = H_j(t)$.
- The batch size of messages arriving at a station j have the following distribution, $P\{B_j = k\} = q_k, k = 0, 1, 2 \dots$
- Mean arrival rate at station $j = \frac{1}{E\{A_j\}}$
- U_j is the switch-over time from station j to station $j+1$ and $P\{T_{U_j} < t\} = U_j(t)$.
- $\rho_j = \lambda_j h_j$, where ρ is the utilization at the j^{th} node, $h_j = E\{H_j\}$ and $h_j^{(2)} = E\{H_j^2\}$.
- $\rho_0 = \sum \rho_j$, for all j's that are data stations.
- T_C is the random cycle time (token cycle time), $c = E\{T_C\}$.
- The ring latency, $c_0 = \sum_{j=1}^{N} E\{U_j\}$, where N is the total number of stations.
- Total number of data stations $= D$.
- Total number of voice stations $= V$, note that $D + V = N$.
- T_P is the voice packetization period.

[1]Note that T_k indicates the time associated with the process k.

- T_a and T_s are the talkspurt and silence period means, respectively. As per Brady's model both are exponentially distributed.
- The Laplace-Stieltjes transform of a CDF $F(t)$ is expressed as $F^*(s)$ and $F^*(s) = \int_0^\infty e^{-st} dF(t)$

6.4.3 Characterization of the voice process

As in Karvelas and Leon-Garcia [29] and Heffes and Lucantoni [21], we have also modeled the voice packet arrival process as the one consisting of geometrically distributed number of packets arriving with the constant inter-arrival time equal to T_p plus an exponentially distributed silence period. Mathematically,

$$A_j(t) = P\{T_{A_j} < t\} \qquad \text{for all voice stations} \tag{6.1}$$

$$= \left[\left(1 - \frac{T_p}{T_a}\right) + \frac{T_p}{T_a}\left(1 - e^{-\left(\frac{t-T_p}{T_s}\right)}\right)\right] U(t - T_p) \tag{6.2}$$

where $U(t)$ denotes the step function. The Laplace-Stieltjes transform (LST) of $A_j(t)$ is $A_j^*(s)$,

$$A_j^*(s) = \left[1 - \frac{T_p}{T_a} + \frac{T_p}{T_a(sT_s + 1)}\right] e^{-sT_p} \tag{6.3}$$

Our approach is to approximate this process, which itself is a model for real voice, into a batch arrival process to make the computation of the performance parameters mathematically tractable.

6.5 General Analysis at the Data Stations

Kuehn's analysis in [33] is the basis for the ensuing analysis for computation of the waiting time at the data stations.

6.5.1 Conditional cycles

The concept of cycle times inherently furnishes a method to consider the performance parameters at a specific queue, while expressing the effect of other queues through the cycle time. We define the cycle time T_{C_j} at a queue j as the time interval between two successive scan instances [33]. The scan instances are defined depending upon the context, but generally pertain to the server departure instances at a queue. A j customer is either served in T_{C_j} or not. We distinguish between two types of cycles, type 2 and type 1, depending upon whether a j customer is served in T_{C_j} or not, respectively. Thus, cycle time $T_{C_{j1}}$ indicates a cycle in which a customer from j was not served and $T_{C_{j2}}$ indicates a cycle in which a customer from j was served. The corresponding CDFs are $C_{j1}(t)$ and $C_{j2}(t)$.

Let α_{j1i} and α_{j2i} be the probabilities that an i customer ($i \neq j$) is served in a cycle $T_{C_{j1}}$ and $T_{C_{j2}}$, respectively. Then,

$$C^*_{j1}(s) = \prod_{i=1}^{N} U^*_i(s) \prod_{i \neq j} \left(\alpha_{j1i} H^*_i(s) + \left[1 - \alpha_{j1i}\right]\right) \tag{6.4}$$

$$C^*_{j2}(s) = \prod_{i=1}^{N} U^*_i(s) \prod_{i \neq j} \left(\alpha_{j2i} H^*_i(s) + \left[1 - \alpha_{j2i}\right]\right) H^*_j(s) \tag{6.5}$$

By the law of total probability, the unconditional cycle LST can be expressed as,

$$C^*_j(s) = (1 - \alpha_j) C^*_{j1}(s) + \alpha_j C^*_{j2}(s) \tag{6.6}$$

(Note : α_j will be defined later.)

We have, $c_{j1} = E\{T_{Cj1}\}$, $c_{j2} = E\{T_{Cj2}\}$ and $c_j = E\{T_{Cj}\}$ then,

$$c_{j1} = c_0 + \sum_{i \neq j} \alpha_{j1i} h_i \tag{6.7}$$

$$c_{j2} = c_0 + \sum_{i \neq j} \alpha_{j2i} h_i + h_j \tag{6.8}$$

$$c_j = (1 - \alpha_j) c_{j1} + \alpha_j c_{j2} \tag{6.9}$$

Also assume,

$$\alpha_{j1i} = \lambda_i c_{j1} \tag{6.10}$$

$$\alpha_{j2i} = \lambda_i c_{j2} \tag{6.11}$$

$$\alpha_j = \lambda_i c_j \tag{6.12}$$

From the above equations, we have

$$c_{j1} = \frac{c_0}{1 - \rho_0 + \rho_j} \tag{6.13}$$

$$c_{j2} = \frac{c_0 + h_j}{1 - \rho_0 + \rho_j} \tag{6.14}$$

For later analysis we also need the variances of T_{Cj1}, T_{Cj2} and T_{Cj}. We have,

$$Var\{T_{Cj1}\} = \sum_{i=1}^{N} Var\{T_{U_i}\} + \sum_{i \neq j} \left(\alpha_{j1i} h_i^{(2)} - \alpha_{j1i}^2 h_i^2 \right) \tag{6.15}$$

$$Var\{T_{Cj2}\} = \sum_{i=1}^{N} Var\{T_{U_i}\} + \sum_{i \neq j} \left(\alpha_{j2i} h_i^{(2)} - \alpha_{j2i}^2 h_i^2 \right) + Var\{T_{H_j}\} \tag{6.16}$$

$$Var\{T_{C_j}\} = \left(1 - \alpha_j\right) \left[Var\{T_{C_{j1}}\} + c_{j1}^2 \right] + \alpha_j \left[Var\{T_{C_{j2}}\} + c_{j2}^2 \right] - c_j^2 \tag{6.17}$$

6.5.2 Queueing analysis of the $M^{[X]}/G/1$ system

The arrival process in such a system consists of customers arriving in batches. Obviously, any system in which incoming data is packetized can be modeled as a batch arrival, $M^{[X]}$, system.

Let $Q(x)$ be the probability generating function of the batch size variable B_j at queue j, i.e.,

$$P\{B_j = k\} = q_{kj} \qquad k = 0, 1, 2, \ldots \tag{6.18}$$

and,

$$Q(x) = \sum_{k=0}^{\infty} q_{kj} x^k \tag{6.19}$$

At the data stations, since we are interested only in the mean waiting time per message, we will use the formula derived by Kuehn [33] here. Therefore,

$$w_j = \left(\frac{c_{j1}^{(2)}}{2c_{j1}} + \frac{\lambda_j c_{j2}^{(2)}}{2(1-\lambda_j c_{j2})} \right) + \frac{c_{j2}}{2(1-\lambda_j c_{j2})} \left[\frac{E\{B_j^2\}}{E\{B_j\}} - 1 \right] \tag{6.20}$$

where $\lambda_j = \lambda_{B_j} \cdot E\{B_j\}$.

As a special case, consider a queue at which exponentially distributed messages arrive. We need to find $E\{B_j\}$ and $E\{B_j^2\}$. Let the mean message length be equal to β_j and the constant packet size h_j.

Then,

$$E\{B_j\} = \frac{1}{\left[1 - e^{-h_j/\beta_j}\right]} \tag{6.21}$$

$$E\{B_j^2\} = \frac{1 + e^{-h_j/\beta_j}}{\left[1 - e^{-h_j/\beta_j}\right]^2} \tag{6.22}$$

$$\lambda_j = \lambda_{B_j} \cdot E\{B_j\} \tag{6.23}$$

6.6 General Analysis at Voice Stations

The analysis at voice stations is two faceted. First, on top of Brady's measurements which concluded that the voice traffic on a network is essentially a sequence of alternating, approximately exponentially distributed, talkspurts and silence periods, another layer of modeling needs be added to make the voice traffic performance parameters mathematically tractable. We will look at how this can be achieved, in this section. Second, the performance parameters of interest at the voice stations are not the same as at the data stations. At the data stations, it was sufficient to estimate the mean data delay to get an idea about the performance of the data on the network. Simply stating, higher the delay, lower is the performance. At voice stations, however, the mean delay does not convey the performance of voice effectively. Reasons for this anomaly are obvious. One, it does not matter how early a voice packet arrived at its destination, it is not played until its scheduled time is reached. Two, a late packet degrades the quality of voice conversation, so *just getting to the destination* is not sufficient and that is precisely what is conveyed by the mean delay. There are two general ways one can compute a more meaningful performance statistic in the case of voice stations.

1. One can still compute the mean delay, but also compute the variance of the delay and thus, shed some more light on how many packets are late by using the Central Limit Theorem or Chebychev's inequality or any other suitable method to satisfy a particular level of accuracy.

2. The tail-end probabilities of the voice packets may be computed from the distribution function of the voice delay, provided such a function could be

obtained. An inference regarding the late packets may be derived from the tail-end probabilities. In a network that implements the blind delay strategy, a late packet is lost. Thus, from the tail-end probabilities the lost packets may be estimated. Our attempt included using this approach. However, numerical difficulties were encountered in Laplace inversion, thus making numerical results impossible. We have computed the mean voice delay and have shown a method to compute the tail-end probabilities.

As stated earlier, in this section we will limit ourselves to modeling voice, which is assumed to be a sequence of alternating exponentially distributed talkspurts and silence periods. The approach is to model the talkspurts as batch processes, as in Karvelas and Leon-Garcia [29]. We have modeled the voice stations as batch arrival processes, using Karvelas's approach, with constant and geometrically distributed batch sizes. Our approach differs from Karvelas's in the batch size distribution.

A voice process with alternating exponential talkspurts and silences can be thought of as a renewal process. This voice process is then approximated to a batch process with exponential inter-arrival time and a batch size expectation of $E\{B_j\}$, at any station j.

$$E\{B_j\} = \frac{T_a^2 + 2T_aT_s + 2T_aT_s^2/T_p}{2(T_a+T_s)^2} \tag{6.24}$$

$$\lambda_{B_j} = \frac{T_a}{E\{B_j\}T_p(T_a+T_s)} \tag{6.25}$$

6.7 Tail-end Probability Computation Method

Since $W_j^*(s)$ gives the LST of the waiting time per customer at a voice station, the Laplace inverse of $W_j^*(s)$ gives the probability density function of the waiting time at time $t = t_1$ using existing Laplace inversion methods. However, if such a density does not exist then this inversion is, mathematically, meaningless. Here, we will derive a known formula to compute the waiting time distribution directly, i.e., without having to do the integration after obtaining the density function. It is known that, for a nonnegative function $h(T)$,

$$E\{h(T)\} = h(0) + \int_0^\infty h'(u)(1 - F(u))du$$

where $F(t)$ is the CDF of T.

Let $h(t) = e^{-st}$. Therefore, $h'(t) = -se^{-st}$. Hence,

$$E\{e^{-sT}\} = 1 - \int_0^\infty se^{-su}(1 - F(u))du$$

which means $T^*(s) = 1 - sG^*(s)$, where $G(u) = 1 - F(u)$.

$$G^*(s) = \frac{1 - T^*(s)}{s}$$

Now, if T represents the waiting time then, $T^*(s) = W^*(s)$ or alternately $T = W$.

Thus, $G(t) = 1 - W(t)$, will precisely gives the tail probabilities. ($G(t)$ is obtained by inverting $G^*(s)$ using known numerical methods.)

From Kuehn [33], the following formula is obtained for the LST of the waiting time at the j^{th} voice station.

$$\begin{aligned} W_j^*(s) &= \text{LST of the waiting time per packet} \\ &= \frac{1 - \lambda_j c_{j2}}{\lambda_j c_{j1}} \frac{1 - C_{j1}^*(s)}{C_{j2}^*(s) - x} \frac{1}{R_j(x)} \end{aligned} \qquad (6.26)$$

where $x = f(s)$ is the solution of $s = \lambda_{B_j}(1 - Q_j(x))$, where $Q_j(x)$ is the probability generating function of the batch size distribution at voice station j.

6.8 Numerically Speaking

Some simulation experiments were conducted for 20 queues (10 data stations and 10 voice stations). Following the notation in Section 6.4,

- D = 10, V = 10, and N = 20.
- The arrival process is Poisson at all data stations such that it gives rise to a designed total utilization of ρ_0.
- The data packets arrive in geometrically distributed batch sizes with $E\{B_j\}$ = 4.
- $c_0 = 20\mu$sec and the switch-over time U_j is constant and equal for all stations.
- $T_p = 30$ms, $T_s = 1802$ms and $T_a = 1360$ms.

The numerical analysis here is based entirely on Karvelas's approximation of batch voice process. Table 6.1 lists the voice parameters, mean voice delays as derived by Karvelas, using both geometrical and constant batch sizes for voice. Figure 6.1 depicts the mean data delays for Karvelas's and our model. Figure 6.2 shows the voice delays provided by the empirical formula (18), in [29]. As can be seen, the mean data delay is reasonably accurately predicted. However, the mean voice delay leaves much to be desired. Both the geometrical and constant batch size approximations fail miserably and though the empirical formula does a little better, it has a high percentage error. The reason for such high error for the constant and exponential batch size approximations is obvious. Voice packets do not arrive in batches, especially large sized batches. The chances of a voice packet waiting for another voice packet in the same queue are relatively low, particularly when the voice packetization period is high. Thus, the batch size approach in its present form fails. As for the empirical formula, we have no information on how it was arrived at and, thus, it may be valid only for a certain range of parameters, with our parameters being out of range.

6.9 A Modification to the Voice Batch Process

6.9.1 Theory

In the batch approximation method above, the large batches seem to estimate high mean voice delays. We have, from here on, used the *Queueing Network Analyzer* (QNA), approach described in [1,58,51] to arrive at a batch size distribution.

The main idea is to express the dependence among successive inter-arrival times in the aggregate voice packet arrival process by a batch process. In other words, equating a Poisson point process (the one with exponential inter-arrival time and a known batch size distribution) to the aggregate voice process generated by the voice stations. The voice process interference with data is well expressed by Karvelas's assumption and we continue to use it. The modification we propose is a combination of the one proposed by Sriram and Whitt in [51] and Albin [1], with a few changes. First, a couple of definitions are due.

Definition 6.9.1 *Let* $\{X_k, k \geq 1\}$ *represent the sequence of inter-arrival times of an arrival process. Let* $\{X_k, k \geq 1\}$ *be stationary. Let* $S_k = X_1 + \cdots + X_k$ *denote the sum of consecutive inter-arrival times. Then the Index of Dispersion of intervals*

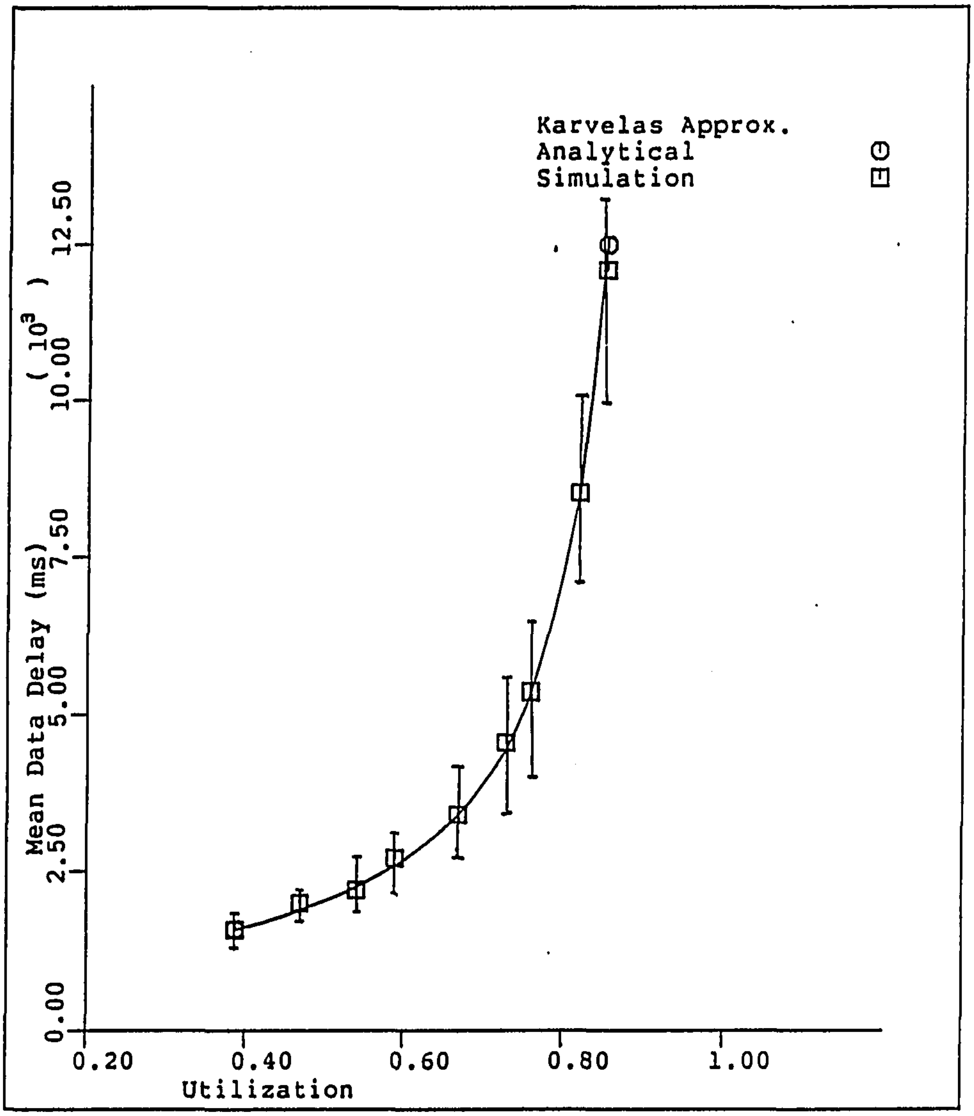


Figure 6.1: Mean Data Delay

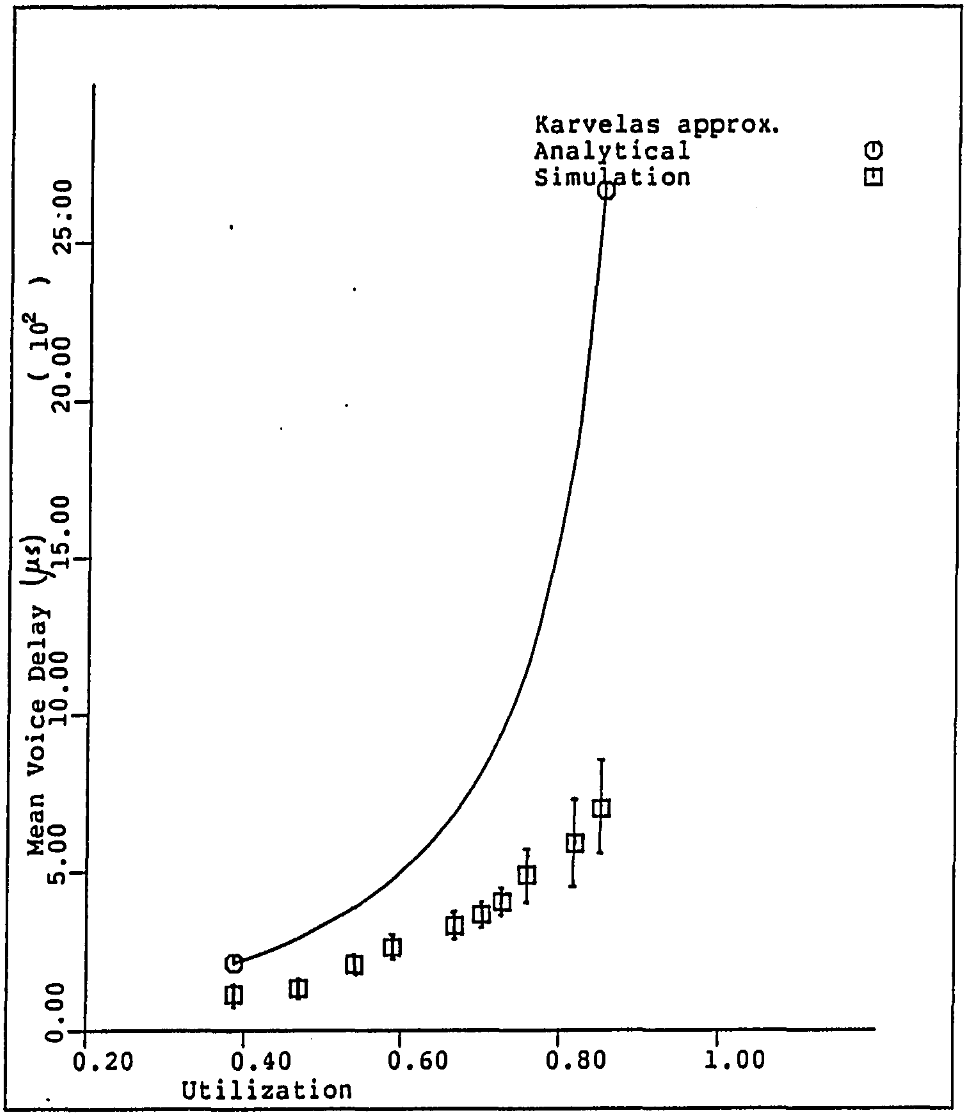


Figure 6.2: Mean Voice Delay : Karvelas Model (Empirical)

Table 6.1: Voice Delay : Karvelas's Batch Assumption

Utilization		Mean Voice Delay (ms)		
Data	Voice	Const. Batch	Geo. Batch	Simulation (95 % Conf. Int.)
.387	.078	6.55	13.013	.114 $\pm$ .011
.469	.078	7.71	15.324	.133 $\pm$ .013
.541	.078	9.135	18.151	.209 $\pm$ .019
.589	.078	10.413	20.695	.263 $\pm$ .026
.668	.078	13.531	26.902	.330 $\pm$ .028
.702	.078	15.534	30.892	.366 $\pm$ .031
.728	.078	17.519	34.844	.405 $\pm$ .058
.759	.078	20.671	41.120	.489 $\pm$ .056
.819	.078	31.743	63.158	.589 $\pm$ .093
.852	.078	45.039	89.594	.699 $\pm$ .095

(IDI) is the sequence $\{c_k^2\}$ is defined by

$$c_k^2 = \frac{kVar(S_k)}{[E(S_k)]^2}$$

More on this may be found in [51].

Definition 6.9.2 *Let $N(t)$ denote the counting process associated with the above arrival process. (It simply counts the arrival instances in time t). The Index of Dispersion for Counts (IDC) is then the function*

$$I(t) = \frac{Var[N(t)]}{E[N(t)]}, \ t > 0$$

Again, for some interesting properties of $I(t)$, see [51].

Now, let c_{kn}^2 and $I(t;n)$ be the two indexes of dispersion associated with the superposition process of n independent voice sources (See [51]). Then,

$$c_{1n}^2 = 1 - \frac{2}{n+1} + \left(\frac{1-p}{\frac{T_p}{T_s} + 1 - p}\right)^{n+1} \left(\frac{2}{1-p} - \frac{2n}{n+1}\right) \tag{6.27}$$

where $p = 1 - \lceil\frac{T_a}{T_p}\rceil$. (Note that, $\lceil X \rceil$ is the smallest integer $\geq X$).

A lucid derivation of this formula is given in [51]. In [57], Whitt suggested two basic methods to approximate point processes by renewal processes. Albin, in [1], suggested a hybrid procedure to combine these two basic methods to generate a better approximation. The first basic method, called the stationary interval method, equates the stationary distribution of an interval of the superposition process (consisting of various independent subprocesses) to the approximating renewal process. With the second basic method, called the asymptotic method, the moments of the interval between the renewals in the approximating process is obtained by matching the moments of the renewal counting process over a large time interval with the corresponding moments of the point process over a large time interval [57]. It turns out that neither of these two methods work very well when used alone. Albin [1] combined the two methods, using a convex function to improve the approximation. (Her mode of validation was the extensive simulations that were run on a Cray-1.) Her formula for the squared coefficient of variation, c_a^2, of the inter-arrival time distribution in the approximating renewal process for the aggregate packet arrival process entering the multiplexer queues is given below.

$$c_a^2 = wc_{AM}^2 + (1-w)c_{SI}^2 \tag{6.28}$$

Where,

$$\begin{aligned} c_{AM}^2 &= \text{coefficient of variation using the asymptotic method} \\ c_{SI}^2 &= \text{coefficient of variation using the stationary method} \\ w(\rho_0, \nu) \equiv w &= \left[1 + 4(1-\rho_0)^2(\nu - 1)\right]^{-1} \end{aligned}$$

and ν = the effective number of component processes that are being superposed. (= the number of voice stations, in our case.)

The formula for $w(\rho_0, \nu)$ is not completely arbitrary, but in fact must follow certain properties [1].

1. $0 \leq w(\rho_0, \nu) \leq 1$.

2. As $\rho_0 \to 1, w(\rho_0, \nu) \to 1$.

3. As $\rho_0 \to 0, w(\rho_0, \nu) \to 0$.

4. As $\nu \to \infty, w(\rho_0, \nu) \to 0$.

Without getting into the detail of these properties, it suffices to say that, the formula for $w(\rho_0, \nu)$ satisfies them. Albin also discovered that if c^2_{SI} is replaced by 1 (which is the coefficient of variation of exponential distribution) a good approximation is still obtained. This makes the formula for c^2_a linear at the expense of marginal increase in the error.

6.9.2 The proposed modification

We propose two modifications to Albin's method. First, our approach is to express the superposed voice process (with V components) as a batch process at one station. The rate of arrival is maintained at the same value as in Karvelas and Leon-Garcia [29]. However, the batch size distribution is changed as follows.

If the arrival rate at a station is λ, with a batch size distribution having a mean μ and variance τ^2, then, if $Z(t)$ is the corresponding counting process, we have (see [55]),

$$E\{Z(t)\} = \lambda\mu t \qquad Var\{Z(t)\} = \lambda\left(\tau^2 + \mu^2\right)t \tag{6.29}$$

$$\begin{aligned} c^2_B &= \text{Coefficient of variation of the batch counting process} \\ &= \frac{\lambda\left(\tau^2 + \mu^2\right)t}{\lambda\mu t} \\ &= \frac{E\{B^2_j\}}{E\{B_j\}} \end{aligned}$$

Thus, by equating this to c^2_a, the coefficient of variation of the superposed process, we have

$$\frac{E\{B^2_j\}}{E\{B_j\}} = c^2_a \tag{6.30}$$

The other modification involves changing the constant 1 suggested by Albin to 1.2. The reasoning is as follows. Albin's approximation is directed towards multiplexers where first-come-first-served discipline is followed more closely since there is no switching overhead like the one created by token passing. This may give rise to a variance for the counting process larger than the exponential variance. After running several simulations 1.2 was found to be a suitable value.

6.10 Results and Conclusion

Figure 6.3 indicates that our approximation does work better for a wide range of utilizations. The key to this lies in the fact that though the voice process may be modeled as a batch process, the batch size distribution should have a lower mean than the one suggested by Karvelas. Our approach ensures this and produces better results. The following conclusions are drawn form this exercise.

- Data may be modeled as a batch process with excellent accuracy.
- The effect of a co-existing voice process (as in a voice-data integrated environment) on the data traffic is well represented by using the batch process approximation by Karvelas, for voice traffic.
- The same batch process approximation, however, does not produce good results for the mean voice delay. Hence, another approximation is needed to accurately predict the mean voice delay.
- The QNA approach, using the index of dispersion of the counting process (IDC) associated with the voice process provides a suitable alternative to

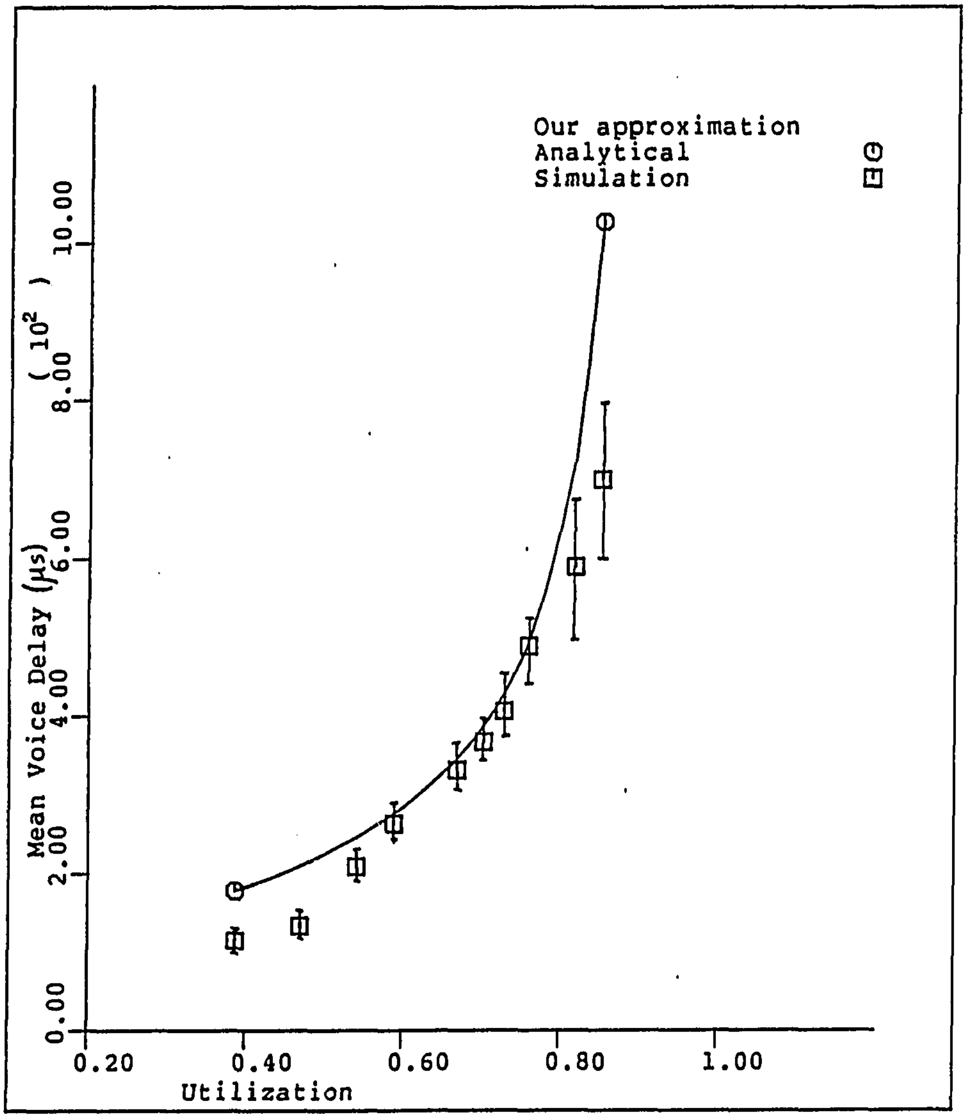


Figure 6.3: Mean Voice Delay : Our Model

Karvelas's approximation. However, modifications are needed to extend the method to token passing rings. Suggesting the modifications is the key contribution in this chapter.

- The advantage of using the batch process lies in the fact that a known distribution function of the waiting time exists and can be inverted if the input parameters are within a certain range to obtain the tail-end probabilities of the voice packet waiting time, which in turn could give an indication regarding the number of packets lost. Though, our approach does provide the methodology to obtain the tail probabilities, the numerical constraints were overwhelming for the Laplace inversion of the waiting time distribution LST.

6.11 Future Work

The work presented in this chapter is merely a small step towards solving a huge problem. Though we do not foresee being able to solve analytically, a protocol like the one proposed in its entirety, the following steps may be useful in getting closer.

- In another experiment we tried our model for 100 queues (67 data and 32 voice stations). The model did not work as well, since the delays were always underestimated for both data and voice. The error probably lies mainly in Kuehn's nonexhaustive model [33], which does not work very well for more than ten queues. Hence, a possible area for further work would be to first modify his model and apply the above technique for voice delay estimation.

- For twenty queues model, the effect of an asymmetric station, contributing a large chunk to the total utilization, is also of interest.

- For the studies conducted in this chapter, simulations were conducted extensively. However, since they were conducted on slow machines, where the turnaround time was high, the accuracy suffered. For future studies we recommend use of faster machines.

7 THE CONCLUSION

There are three main themes presented in this dissertation. First, design issues concerning the integration of voice and data in presence of heavy document retrieval traffic are dealt with and a possible design is proposed. Second, several aspects of simulation are categorized and a statement is made regarding how to extract the maximum possible information out of a simulation run. Third, the modeling difficulties encountered in analytical modeling are addressed and relevant subsystem analysis is carried out. This chapter is divided in three sections dealing with each of the three themes focusing on the work done and the work to be done.

7.1 The Design

7.1.1 The work done

Our approach to the design of the proposed protocol was essentially *ad hoc.* The implementation of the priority protocol was based on intuition and extensive simulations to rule out the alternatives. This research deals with integrating voice and data on a token ring when long document packets are also being transmitted. The research on asymmetric traffic has received relatively little attention in the literature, as compared with balanced traffic. Our contribution, thus, lies in designing a protocol suitable for asymmetric rings with real time constraints. Additionally, we feel, that the merit of the design is enhanced by the fact that it utilizes the existing standard and off-the-shelf components, an important, but often ignored, fact.

7.1.2 Future directions

7.1.2.1 Implementation Before a full scale implementation is carried out, an experimental laboratory is being set up to validate the claims made in this dissertation and also to grasp the protocol implementation difficulties at the hardware and software level.

7.1.2.2 A possible addition A possible addition of moving television images may be appropriate if a token ring network of bandwidth more than the current 4 Mbits/sec, can be implemented. Hardware limitations, however, make it difficult for non-fiber token rings to operate at higher bandwidths. So a low speed fiber-optic token ring network may be a possible course.

7.2 The Simulation

Simulation is, probably, the easiest way to try out different ideas during the design process and was used extensively in this research. It is often emphasized in the literature that simulations are expensive. To a normal user with a *free* available computer system to use, this may seem like an irrelevant fact. The stated expensiveness of the simulations does not pertain to the *real* money, but refers to the turn around time involved in getting meaningful results from the simulation programs. On the slow systems that are available to some users, efficient simulations are a must. As an example, it sometimes took us a day for simulating ten seconds of network operation. Thus, some of the research was devoted to reducing the simulation time.

7.2.1 The work done

In this research, we focused on reducing the simulation time, given our constraints, without sacrificing the reliability of the simulation results. We incorporated the regeneration technique to estimate the variance for calculating the confidence intervals. The existing methods for G/G/1 queues with single server and single queue were extended to single cyclic server with multiple queues. The validity of the approximations was tested by comparing our method with the brute force method.

7.2.2 Future directions

This is probably the most promising area in performance analysis for future research. With protocols with high complexities, simulations are becoming indispensible and if one cannot afford high speed computers, attention must be paid to reducing the simulation time to a minimum. Our approach, since improving the simulation time was not our main goal, was essentially *ad hoc*. A great deal of theoretical work needs be done to get an insight into how to design regenerative statistics collection efficiently. Furthermore, techniques should be developed to implement variance reduction in multiqueue situation, an area we did not focus on.

7.3 The Analytical Model

Forming an exact analytical model for the designed protocol is a formidable task and the current developments in queueing theory propose no solution to this class of systems, namely, the G/G/1 queues [1]. We focused only on certain related subsystems for analysis as a stepping stone for future analysis.

7.3.1 The work done

A system with cyclic queues with non-exhaustive service discipline was analyzed. The incoming traffic consisted of a mixture of data and voice. Explicit expressions for mean data delay, mean voice delay and the voice waiting time tail probabilities were obtained.

7.3.2 Future directions

Optimistically, the goal, probably, should be to analyze the entire system. We feel, from our experience, that, unless there are some landmark discoveries in queueing theory, this goal is unattainable. However, more information about the behavior of the queues in a system, such as the standard token ring, may be obtained from extensive simulations. Similar work on the behavior of single server G/G/1 queues was carried out at the Bell Laboratories [58]. However, undertaking such a project makes it necessary to have high speed computing facilities.

8 ACKNOWLEDGEMENT

I have, at times impatiently, waited for this moment which I had been describing as *the end*, only to realize now that this is just *the beginning*. Anyway, at this time I take the opportunity to thank those who made this wait bearable.

First, I would like to thank Prof. A. V. Pohm, my major professor, for suggesting this problem to me in its abstract form and Prof. Doug Jacobson, my co-major professor, for helping me define it further and put it in a tangible form. Prof. J. O. Kopplin provided the financial backing and I am thankful for it.

Second, I am indebted to the following three who helped me either directly or indirectly in making my education at ISU fruitful: Prof. Jim Davis, for giving me the opportunity to work on the Gateway Project, an experience I wouldn't trade for any amount of coursework; Prof. K. B. Athreya, for answering my innumerable questions on queueing theory; Prof. T. F. Piatkowski, for teaching me the importance of abstraction and for introducing me to the fascinating world of Computer Networking.

Third, I thank Prof. T. A. Smay, Prof. A. E. Oldehoeft and Prof. C. S. Comstock for serving on my committee.

To share the research blues, one needs friends and I have a few to thank. I would like to thank Mansoor Sarwar for suppressing his urge to kill my ever-running processes on the AT & T 3B15 system. In the fear of leaving someone out I would like to thank my officemates over the years, my classmates and the guys who worked with me on various projects, collectively.

My parents have had an everlasting influence on me in my quest for a career, knowledge and all other *nice* things in life. I am dedicating this dissertation to them.

I would be remiss if I did not mention my wife, Julie, for her patience and perseverance and for her stabilizing presence.

Finally, this dissertation is one of the first printed using LaTeX, a task that would have been quite unpleasant without the timely help of George A. Christensen from the Computation Center.

9 BIBLIOGRAPHY

[1] Albin, S. L. "Approximating a point process by a renewal process, II: Superposition arrival processes to queues." *Operations Research*, 32, No. 5 (1984): 1133-1162.

[2] Ammar, M. H. and Wong, J. W. "Response time optimization in teletext systems." *IEEE INFOCOM*, (1985) : 266-272.

[3] Balci, O. "Credibility assessment of simulation results." *Winter Simulation Conference Proceedings*, Washington D. C., 1986, 38-43.

[4] Banks, J. and Carson J. "Introduction to discrete-event simulation." *Winter Simulation Conference Proceedings*, Washington D. C., 1986, 17-23.

[5] Borgonovo, F., Cadorin, E., Fratta, L., and Pezze, M. "Voice and data performance measurements in L-express net." *ACM SIGCOMM SYMPOSIUM*, (1986):120-125.

[6] Brady, P. T. "A Technique for investigating on-off patterns of speech." *B. S. T. J.*, 44 (1963): 1-22.

[7] Brady, P. T., "Effects on transmission delay on conversational behavior on echo free telephone circuits ". *B. S. T. J.*, 50 (1971): 115-134.

[8] Bux, W. "Local area subnetworks: A performance comparison." *IEEE transactions on communications*, COM-29, No. 10 (October 1981) : 1465-1473.

[9] Bux, W., Felix, C. H., Kuemmerle, K., Keller, H. J., and Mueller, H. R. "Architecture and design of a reliable token ring network." *IEEE Journal on Selected Areas in Communications*, SAC-1, No. 5 (Nov 1983) : 756-765.

[10] Bux, W. "Performance issues in local area networks." *IBM Systems Journal*, 23, No. 4 (1984): 351-374.

[11] Chauhan, V. and Sethi, A. S. "Performance studies of token based local area networks." *10th Conference on Local Computer Networks*, Minneapolis, 1985, 100-107.

[12] Crane, M. A. " An introduction to the regenerative method for simulation analysis." Lecture Notes in Control and Information Sciences, Spinger-Verlag, Berlin, 1977.

[13] Daigle, J. N. and Langford, J. D. "Queueing analysis of a packet voice communication system." *IEEE INFOCOM*, (1985) : 18-26.

[14] DeTreville, D. "A simulation based comparison of voice transmission on CSMA/CD networks and on Token Buses." *AT & T Bell Laboratories Technical Journal*, 63, No. 1 (January 1984): 33-55.

[15] Feller, W. An introduction to probability therory and its applications. Vol. II. John Wiley, New York, N.Y., 1971.

[16] Ferguson, M. J. and Aminetzah, Y. J. " Exact results for Nonsymmetric token ring systems." *IEEE Transactions on Communications*, COM-33, No. 3 (March 1985): 223-231.

[17] Frost, V. S., Friedman, E. M., and Minden, G. J. "Multirate voice coding for load control on CSMA/CD local computer networks." *10th Conference on Local Computer Networks*, Minneapolis, 10 (1985):10-19.

[18] Gifford, D. K., Lucassen, J. M., and Berlin, S. T. "The application of digital braodcast communication to large scale information systems." *MIT Report*, MIT/LCS/TM-299, April 1986.

[19] Gopal, P. M., Wong J. W. and Majithia, J. C. "Analysis of playout strategies for voice transmission using packet switching techniques." Performance Evaluation, 4 (1984): 11-18.

[20] Gruber, J. G. "Delay related issues in integrated voice and data networks." *IEEE Transactions on Communications*, COM-29, No. 6 (June 1981): 786-800.

[21] Heffes, H. and Lucantoni, D. M. " A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. " IEEE Journal on Selected Areas in Communications, SAC-4, No. 6 (September 1986): 856-867.

[22] Ibe, O. C. and Gibson, D. T. "Synchronous communication protocols for local networks." *10th Conference on Local Computer Networks*, Minneapolis, 10 (1985):4-9.

[23] Ibe, O. C. and Gibson, D. T. "Protocols for integrated voice and data local area networks." *IEEE Communications Magazine*, 24, No. 7 (July 1986) : 30-36.

[24] *IEEE Standard 802.3.* CSMA/CD Access Method and Physical Layer Specifications. IEEE Computer Society, Silver Springs, MD, Dec. 1984.

[25] *IEEE Standard 802.4.* Token Passing Bus Access Method and Physical Layer Specifications. IEEE Computer Society, Silver Springs, MD, Dec. 1984.

[26] *IEEE Standard 802.5.* Token Ring Access Method and Physical Layer Specifications. IEEE Computer Society, Silver Springs, MD, Dec. 1984.

[27] Iman, R. L. "Modeling input processes." *Winter Simulation Conference Proceedings*, Washington D. C., 1986, 34-37.

[28] Jacobson, D. W. " Progress Report : Local Area Network Protocols for Integrated Voice, Data and Video Services." Computer Systems Research Group, Department of Electrical Engineering and Computer Engineering, Iowa State University, November 30, 1987.

[29] Karvelas, D. and Leon-Garcia, A. " Performance Analysis of integrated Voice/Data Token Passing Rings ." *IEEE Journal on Selected Areas in Communications*, SAC - 4, No. 6 (September 1986): 823-832.

[30] Kleijnen, J. P. C. *Statistical techniques in simulation. Part I.* Marcel Dekker, Inc. ,New York, 1974.

[31] Kleinrock, L. *Queueing Systems.* Vol. I: Theory. John Wiley, New York, N.Y. 1975.

[32] Kleinrock, L. *Queueing systems.* Vol. II : Computer Applications. John Wiley, New York, N.Y. 1975.

[33] Kuehn, P. J. "Multiqueue systems with nonexhaustive cyclic service." *B. S. T. J.*, 58 (March 1979): 671-698.

[34] Lam, S. S. "A carrier sense multiple access protocol for local networks." *Computer Networks* 4 (1980): 21-32.

[35] Lee, J. Y. *Research Proposal.* Computer Systems Research Group, Department of Electrical Engineering and Computer Engineering, Iowa State University, 1985.

[36] Leon-Garcia, A. " Guest Editorial ." *IEEE Journal on Selected Areas in Communications*, SAC - 4, No. 6 (September 1986): 773-774.

[37] Mark, J. W. "Priority scheduling for integrated voice/data services in local area networks." *CCNG report E-111.* University of Waterloo, Waterloo, Ontario, Canada, March 1983.

[38] Maxemchuk, N. F. "A variation on CSMA/CD that yields movable TDM slots in integrated voice/data local networks." *B. S. T. J.*, 61 (1982): 1527-1550.

[39] Mittal, K. and O'Reilly, P. "Comparative evaluation of different access schemes for local area ring networks." *10th Conference on Local Computer Networks*, Minneapolis, 10 (1985):114-122.

[40] Montgomery, W. A. " Techniques for packet voice synchronization." *IEEE Journal on Selected Areas in Communications*, SAC-1, No. 6 (December 1983): 1022-1028.

[41] Ogawa, R., Takashima, Y., Omachi, T., and Usubuchi, T. "Experimental system document page image filing and communication." *IEEE Globecom*, 1 (1985): 368-362.

[42] O'Reilly, P. J. P. and Hammand, J. L. "An efficient simulation technique fro performance studies of CSMA/CD local networks." *IEEE Journal on Selected Areas in Communications*, SAC-2, No. 1 (January 1984): 238-249.

[43] Pfister, G. M. and O'Brien, B. V. "Comparing the CBX to the local area network– and the winner is?" *Data Communications*, 11 (July 1982) : 103-113.

[44] Pritsker, A. A. B. *Introduction to Simulation and SLAM II.* Second Edition, Halsted Press, Stroudsburg, Pennsylvania, 1984.

[45] Ramakrishnan, K. K. and Howe, W. "Performance of an extended local area network for image applications." *IEEE Phoenix Conference*, (1986):314 - 320.

[46] Rego, V. and Hughes, H. "A comparison of two token passing bus protocols." *ACM SIGCOMM Symposium*, 1986, 58-66.

[47] Sachs, S. R., Kan. K., and Silvester, J. A. "Performance analysis of a token bus protocol and comparison with other LAN protocols." *10th Conference on Local Computer Networks*, Minneapolis, 10 (1985):46-51.

[48] Saydam, T. and Sethi, A. S. "Performance evaluation of voice-data token ring LANs with random priorities." *IEEE INFOCOM* (1985) : 326-332.

[49] Schmidt, J. M. "Introduction to systems analysis, modeling and simulation." *Winter Simulation Conference Proceedings*, Washington D. C., 1986, 5-16.

[50] Sharrock, S. M., Maly, K. J., Ghanta, S., and Du, H. C. "A framed, movable boundary protocol for integrated voice/data in a LAN." *ACM SIGCOMM SYMPOSIUM*, 1986, 111-119.

[51] Sriram, K. and Whitt, W. "Characterizing superposition arrival processes in packet multiplexers for voice and data." *IEEE Journal on Selected Areas in Communications*, SAC-4, No. 6 (September 1986): 833-846.

[52] Stallings, W. *Local Networks - An introduction.* Macmillan Publishing Company, New York, 1984.

[53] Soueid, E. and Pendse, S. "CSMA/CD with channel capture." *10th Conference on Local Computer Networks*, Minneapolis, 10 (1985):54-60.

[54] Tannenbaum, A. S. *Computer Networks.* Prentice Hall Inc., Englewood Cliffs, N.J., 1981.

[55] Taylor, H. M. *An introduction to stachastic modeling.* Academic Press Inc., Orlando, Florida, 1984.

[56] van de Legemaat, J., Daeman, J. M. A., and Niemegeers, L. G. "Voice transmission in a priority CSMA/CD LAN: An efficient protocol using hybrid switching." *ACM SIGCOMM SYMPOSIUM*, 1986, 102-110.

[57] Whitt, W. "Approximating a point process by a renewal process, I: Two basic methods." *Operations Research*, 30, No. 1 (1982): 125-147.

[58] Whitt, W. "The queueing network analyzer." *B. S. T. J.*, 62 (1983): 2779-2815.

[59] Wong, J. W. and Gopal, P. M. "Analysis of a token ring protocol for voice transmission." *Computer Networks*, 8, No. 4 (August 1984) : 339-346.

[60] Yamazaki, Y. "Recent advances in facsimile communications." *IEEE Globecom*, 1 (1985): 118-125.