1987

Design and performance analysis of an Integrated Voice/Data (IVD) protocol for a token ring network

Jai Yong Lee
Iowa State University

8716786

Lee, Jai Yong

DESIGN AND PERFORMANCE ANALYSIS OF AN INTEGRATED VOICE/DATA (IVD) PROTOCOL FOR A TOKEN RING NETWORK

Iowa State University PH.D. 1987

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

Design and performance analysis
of an Integrated Voice/Data(IVD)
protocol for a token ring network

by

Jai Yong Lee

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:

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

Iowa State University
Ames, Iowa

1987

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Integrated Services in Local Area Networks (LANs)

Historically, data networks and voice networks have evolved separately because of the uncompatible technology adopted for each network. Voice networks have been based mainly on analog technology, while data networks have been based on digital technology [1]. With the advent and development of digital technology for voice, integration of both voice and data services in a network has become practical. Research work on integrating various services in a network has been spurred by the promise of significant benefits to network users and operators [2-3]. The principal benefits to the user can be expressed in terms of cost savings and convenience. Users can buy integrated service to meet multiple needs and can thus achieve lower cost. Voice annotation of documents, voice message recording and retrieval are among the conveniences that users can obtain. For network owners, more simplified network control and management can be achieved by integration of services in a network.

One can integrate voice and data over either a voice network or over a data network. For a long-haul situation, it appears that voice networks have an edge, as reflected in current active research work on Integrated Services Digital

Network (ISDN) [4-6]. Even for local area networks, this statement may seem true, at least from the point of view of economy, because of cost savings due to use of the existing telephone network. It is not, however, generally true for local area networks. From the performance point of view, voice networks have severe speed constraints, since the speed of a typical Computerized Branch Exchange (CBX) is limited to 64kbps. For such local environments as scientific/engineering labs, factories, and office work-stations, which may require high speed data transfer at speed greater than 1Mbps, a Local Area Network (LAN) is better suited than CBX in serving integrated services [7-9]. In the research described in this thesis, a local . environment requiring both high speed data transfer and voice service will be assumed.

Selection of LAN Protocols

The LAN model and its relationship to the Open Systems Interconnection (OSI) Reference Model of the International Organization for Standardization (ISO) is illustrated in Figure 1.

Since the performance of a LAN depends on the protocol of the MAC layer [7] in Figure 1, a proper MAC layer protocol must be selected to achieve high performance. The Institute of Electrical and Electronics Engineers (IEEE) has

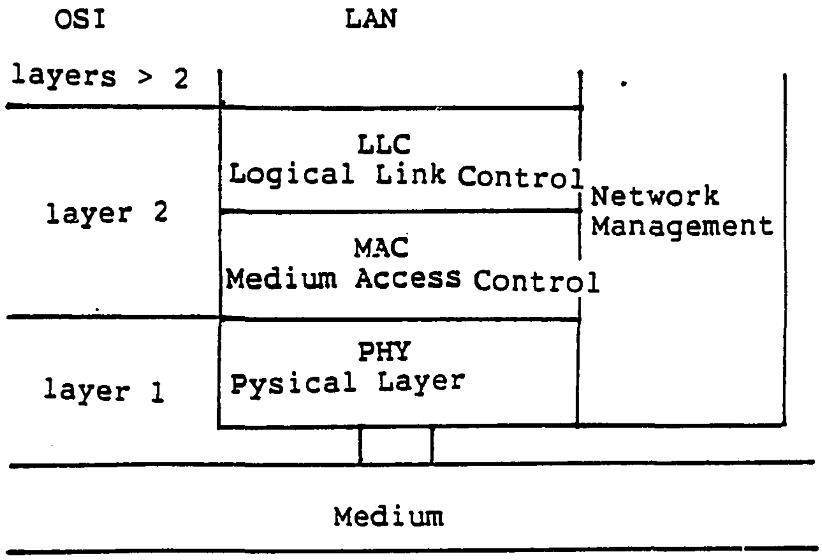


FIGURE 1. Relation of OSI Reference Model to LAN Model [10]

standardized three LAN protocols representing three medium access methods [10-12]. They are:

- IEEE 802.3 (CSMA/CD)
- IEEE 802.4 (Token Bus)
- IEEE 802.5 (Token Ring)

Many researchers have compared the performance of these three protocols under various traffic loads [13-18]. The performance results are summarized as follows:

- Token Ring is the least sensitive to workload.
- CSMA/CD offers the shortest delay under light load, while it is most sensitive under heavy load.
- Token bus shows the longest delay in light traffic

conditions.

Since a fraction of a channel will be used for voice traffic, which may in turn result in a heavy data load, the token ring protocol is selected for the integrated voice/data protocol because of its relative insensitivity to this load.

Token Ring Protocol

Because an integrated voice/data (IVD) protocol based on the token ring protocol will be developed in this thesis, the operation and the priority mechanism of the token ring protocol in IEEE 802.5 [12] will be explained in this section.

General operation

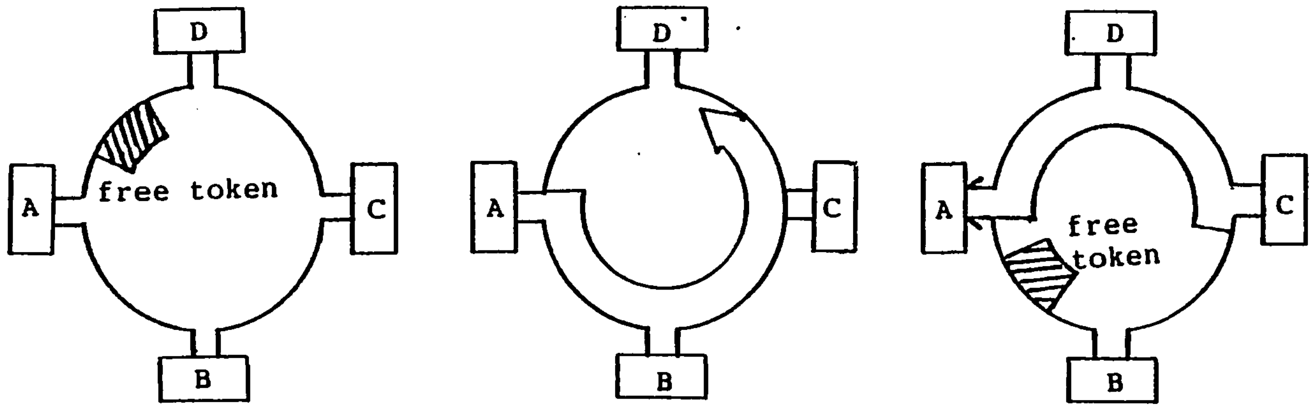
In the token ring protocol, information is transferred sequentially, bit by bit, from one station to the next serially connected station. A station gains the right to transmit when it receives a free token. The station detects the token, modifies it to a start-of-frame sequence and starts to transmit. Each station regenerates and repeats each bit when a bit arrives at a station via the medium. When a packet is addressed to a destination, the destination station copies the information and repeats all bits, passing them to the next station. When the transmitted packet is received by the transmitting station, the packet is removed from the ring by the transmitting station. At that point the start-of-frame sequence, is changed back to a free token and passed to the next station. The operation of the token ring protocol is illustrated in Figure 2.

Priority mechanism

The priority mechanism operates in such a way that fairness is maintained for all stations within a given priority level. This is accomplished by having the same station that raised the service priority level of the ring (stacking station) return the ring back to the original level. For priority operation of the token ring protocol, registers and stacks exist in each data station, as shown in Figure 3. These registers, stacks, and priority and reservation fields of a packet are defined as follows:

- P,R: Priority and Reservation value of current packet on the medium.
- Pm: Priority of a queued Packet Data Unit (PDU).
- Pr,Rr: The value of P and R of the most recently received AC (Access Control) field are stored in registers as Pr and Rr.
- Sx,Sr: These stacks are used to push and pop the ring priority.

The summary of the priority mechanism is given in Table 1. Notice that the first three rows in Table 1 occur when a new token is generated and the last two rows occur only when a



free token arrives station A

station A transmits to station C by modifying free token to busy token

station A changes busy token to free token and continues to remove message from medium

FIGURE 2. Operation of the Token Ring Protocol

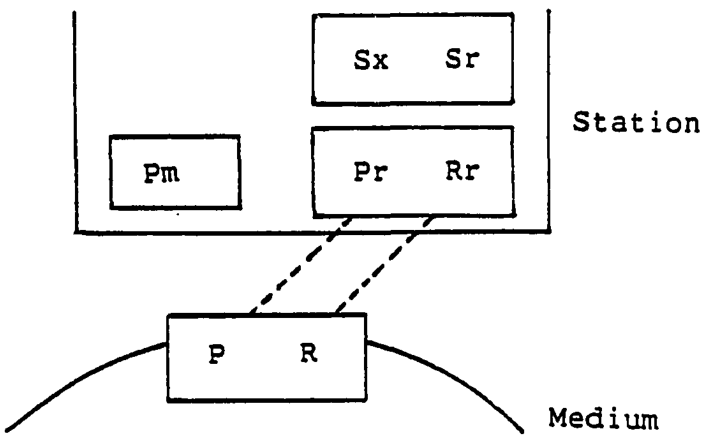


FIGURE 3. Registers and stacks in a station

TABLE 1. Priority mechanism of IEEE 802.5

	Registers & Stack Condition	Output
Token transmission (same priority)	Pr<=Rr&Pm	P=Pr, R=Max(Rr/Pm)
Token transmission (higher priority, push)	(Pr<Rr/Pm) & (Pr>Sx) (if Sx empty, (Pr>Sx) is true)	P=Max(Rr/Pm), R=0, Sx=P, Sr=Pr
Token transmission (higher priority, pop and push)	Pr<Rr/Pm & (Pr=Sx) (if Sx empty, (Pr=Sx) is false)	P=Max(Rr/Pm), R=0, Pop Sx, Sx=P
Free Token received by the stacking station(pop & push)	(Queue_empty V (PDU_queued & Pm<Sx)) & TK(P=Sx) & Rr>Sr	Pop Sx, P=Rr, Sx=P
Free Token received by the stacking station(pop)	(Queue_empty V (PDU_queued & Pm<Sx)) & TK(P=Sx) & Rr<=Sr	Pop Sx, P=Sr, R= Rr, Pop Sr

free token is received by the stacking station. Notice also that the order of assignments in the output column of this table is important for the correct operation.

Problem Statement and Objectives

In dealing with integrated voice/data in a token ring network, two basic problems are encountered. One problem occurs in handling real-time voice in a packet-switching network whose characteristics are dependent on the variable network access delay. The other problem is related to channel-sharing by the two different traffic types. The former problem affects the quality of voice while the latter problem affects data performance. Recently, several protocols have been proposed to support integrated services in a token ring network, as will be discussed in the next chapter [19-27]. In spite of these research efforts, the following problems are still unsolved:

- The problems of LAN in supporting integrated voice/data services haven't been clearly identified.
- Design issues for designing the integrated voice/data protocol haven't been thoroughly studied.
- The quality of voice is poor in medium and heavy data traffic, and high queueing delay of data

packets are caused by voice traffic.

The objective of the present research is to develop an integrated voice/data (IVD) protocol for token ring networks along with a channel allocation strategy that provides high quality of voice over a wide range of data traffic while preserving satisfactory data performance. To achieve this objective, the subgoals of this research have been set as follows:

- Describe LAN problems related to integrating voice/data in a token ring network before designing the IVD-token ring protocol.
- Develop a design approach to achieve high quality of voice and select network parameters necessary for the proposed IVD protocol.
- Develop and specify an IEEE 802.5 compatible IVD-token ring protocol to support practical implementation.
- Develop an accurate model of the proposed protocol for performance analysis.
- Perform careful analysis of the model to predict performance of the proposed protocol.

Thesis Overview

Motivation, problem statements, and objectives of this research are explained in the first chapter. In the second chapter, two voice problems and one data problem in current token ring protocol are identified. The drawbacks and contributions of current research work on integrated voice/data protocols based on token ring networks are described in the third chapter. In Chapter 4, design issues, including a channel allocation strategy, are studied to achieve high voice quality while preserving satisfactory data performance in token ring networks based on IEEE 802.5. In Chapter 5, the proposed IVD-token ring protocol is specified using state transition diagrams, and the operation of the proposed protocol is explained. In Chapter 6, a discrete event simulation model is built using a simulation language (SLAM) [28] for performance evaluation of the proposed protocol. In Chapter 7, the voice and data performance of the proposed protocol are evaluated and analyzed using these discrete simulation methods. Finally, conclusions and proposed future work are discussed in Chapter 8.

CHAPTER 2. PROBLEMS OF A LAN IN SUPPORTING INTEGRATED SERVICES

Since a Local Area Network (LAN) is primarily designed to carry data traffic, problems are raised when integrated data/voice services are handled in LANs. In the following sections, first the characteristics of data and voice are compared because some of the problems in LANs are raised by the differences in the characteristics of each type of traffic. Then the problems of current MAC layer protocols in handling integrated voice/data are studied.

Characteristics of Traffic Types

Characteristics of data

Two types of data traffic currently supported by LANs can be identified: interactive and bulk data. The general nature of each of these types may be described as follows:

- Interactive Data: It is bursty and consists of short messages each of which requires low network delay.
- Bulk Data: It consists of long messages and normally requires high throughput.

Characteristics of voice

While data requires low delay and high throughput, voice traffic has only moderate delay and throughput needs. However, voice traffic requires very low variance in delay. Voice traffic is associated with the following three characteristics.

On-Off characteristics (Talkspurt and Silence) Brady [29,30] describes an experiment with 16 people having conversations by phone and studied On-Off patterns in their conversations. Figure 4 shows a part of the pattern in a conversation of P1 and P2.

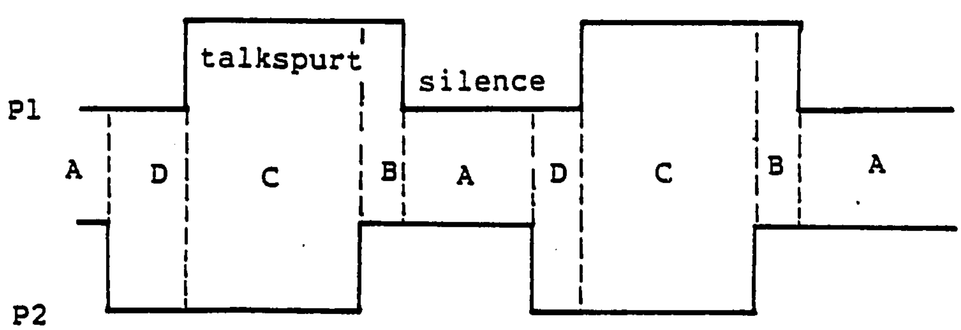


FIGURE 4. On-Off pattern of a conversation

Figure 4 illustrates four distinct states, viz, talkspurts for each person (A and C), mutual talking (B), and mutual silence (D). In his experiments, Brady [30] found that the channel is typically 50-60% idle in a conversation because of the mutual silence. To utilize the channel more efficiently, the idle time for one conversation can be used by another conversation. This feature, called the TASI (Time Assignment Speech Interpolation) advantage, has been used in telephone systems [31] to increase the number of users in the channel. Notice that the TASI advantage can be achieved in the packet network.

<u>Limit on the delay of a talkspurt</u> Another characteristic of voice traffic is that there must be a limit on the delay of a talkspurt for intelligible conversation. This characteristic has been reported by Huggins [32], Gruber [33], and Brady [30]. Huggins showed in his experiments that the intelligibility of voice starts to degrade when the speech segment delay is about equal to the speech segment interval. Gruber and Brady have shown that the limit of delay between intersyllable gaps and listener detected pauses are bounded by 200msec of delay when the minimum length of a talkspurt is about 200msec. Figure 5 shows the effects of delay exceeding 200msec in two talkspurts.

As shown in this figure, 'output' is heard as 'ou - tput' because of the longer delay than limit in the second talkspurt.

<u>Loss of packets allowed</u> In the transmission of data traffic, no loss of any data packet is allowed. However, in

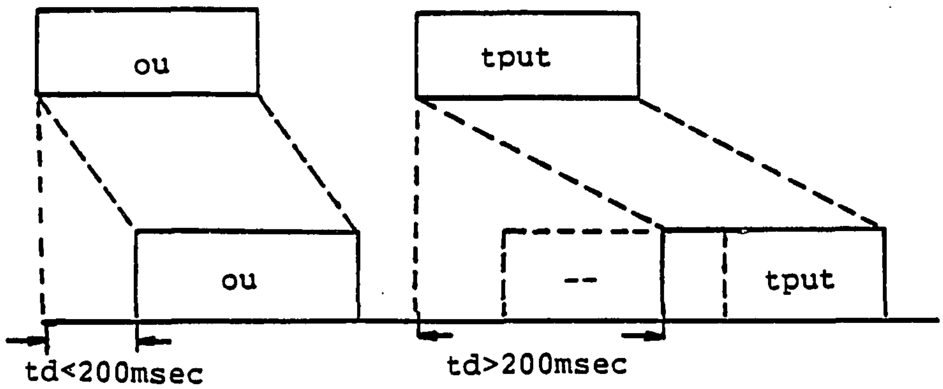


FIGURE 5. Effects of delay over the limit in two talkspurts

voice transmission, a reasonably good quality of voice can be recovered in spite of a 1-2% loss of transmitted voice packets [32],[34].

LAN Problems

When integrated voice/data are serviced in a LAN, problems related to both voice and data may occur. The voice problems are related to the delay problem which is caused by using a packet network rather than a circuit switching network. The data problems are caused by sharing the same channel with other traffic. Three problems can be identified and studied.

The end-to-end delay limit problem

In the previous section, the end-to-end delay limit of a talkspurt was identified as one of the characteristics of voice traffic. Because a talkspurt consists of voice packets, the end-to-end delay of a talkspurt can be over the limit (200msec) if any voice packet for the talkspurt is delayed more than 200msec or the accumulated delay of voice packets is more than 200msec. When only data traffic is served in a LAN, the end-to-end delay is usually within the range of several tens of msec and it might appear that there is no problem in serving voice traffic. However, there is a possibility of delay exceeding the limit for voice traffic. This possibility is principally caused by the packetization and depacketization of voice packets rather than by channel access delay. To show what kinds of parameters are related to these packetization and depacketization periods, the voice packet generation process will be reviewed. Figure 6 shows the basic structure of a packet voice system.

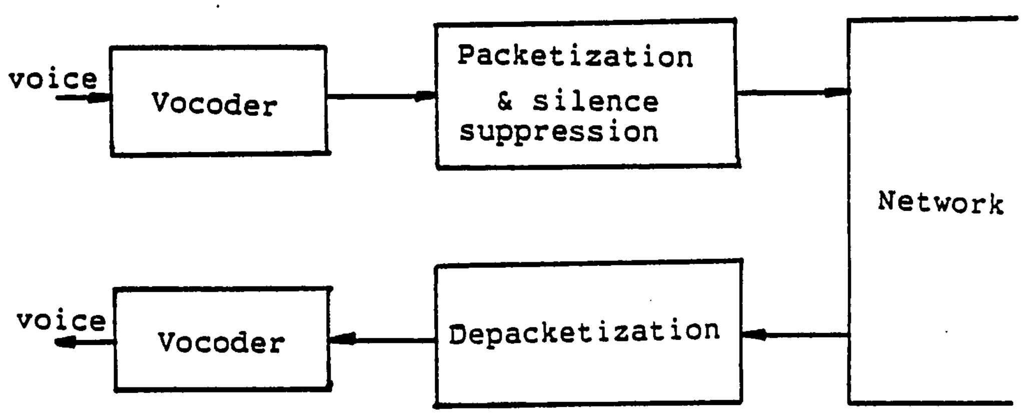


FIGURE 6. A packet voice station

First of all, the analog voice is digitized by a vocoder whose data rate ranges from 64kbps to 2kbps depending on the vocoder technique used. (PCM, ADPCM, LPC, etc. [35]) The next step is the packetization during which packetization delay (Dp) is introduced. Dp may be expressed in terms of packet size (Ps) and vocoder rate (Rv) as follows:

$$Dp = Ps * 1/Rv \tag{2.1}$$

Notice that Dp is only proportional to Ps if the vocoder is selected. To minimize Dp, a small packet size, Ps, can be selected. However, too small a value of Dp requires frequent voice services which in turn increase the data delay. After the packetization process, silence will be suppressed to enhance utilization of the channel [36]. The voice packet then waits for a chance to access the network. The depacketization process is the reverse of the packetization process.

The variable access delay problem

As described earlier, the main problem in carrying voice in a LAN results from the fact that a real time voice signal is carried through a packet switching network rather than a circuit switching network. Because of variable network delay, the network access time of a voice station is not guaranteed. Figure 7 depicts the variable access delay problem of a voice station. In this figure, a voice station generates voice packets after every voice packetization period (Tp). Because the network access delay is variable, the voice station cannot access the network at regular intervals and, as a result, the receiver cannot receive the voice packets at regular intervals and therefore cannot produce the continuous voice signal.

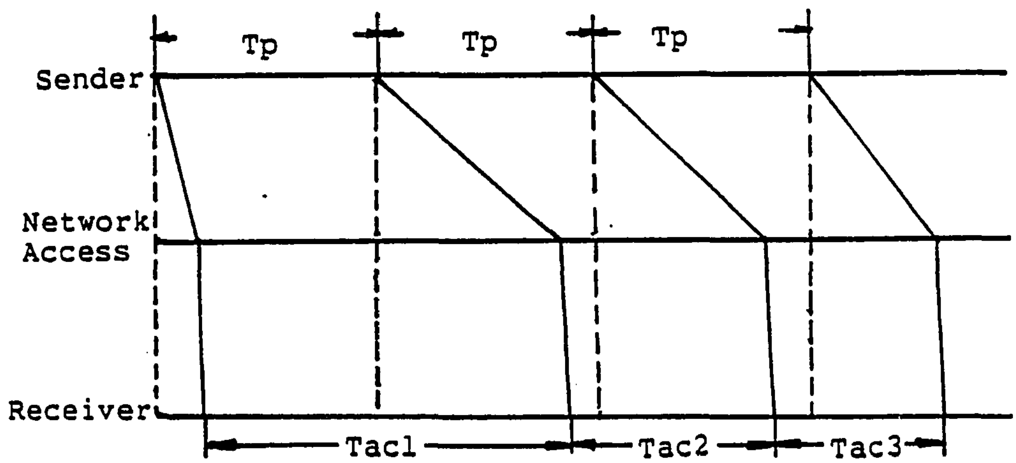


FIGURE 7. A variable network access problem

A closer look reveals two classifications of the variable delay problem: network access problem for the sender and a synchronization problem for the receiver. The network access problem can be stated as follows:

> Current LAN voice protocols do not guarantee the voice stations to access the network just after the arrival of voice packets.

The synchronization problem can be stated as follows:

> The synchronization problem at the receiver is concerned with playing out the packets from the output buffer at regular intervals (equal to Tp) with less than 1-2% packet loss.

Currently, no efficient solution for the network access problem has been developed. Simple and efficient solutions for the synchronization problem have been introduced by a few researchers [37-39].

The data queueing problem

Since a voice packet is generated at every regular interval (Tp), a certain portion of the channel should be allocated to the voice traffic. Data packets may, however, suffer excessive delay if the channel is too frequently allocated to voice traffic to produce a high quality of voice. An example of queueing delay of data in token ring network [20], [40] may be developed by comparing the expected number of data packets in a queue at time t with that at time t + n(Tv + Td) where Tv is the voice cycle interval and Td is the data cycle interval and n = 1,2,... The average incremental data queue size (E[dQ]) after a data and voice cycle may be shown using the Little's theorem [41] to be

$$E[dQ] = \lambda d*Tv + \lambda d*Td - \mu d*Td \qquad (2.2)$$

where λd is the arrival rate and μd is the service rate of the data packet. In (2.2), λdTv and λdTd are the total number of data packets entering the data queue during the voice cycle (Tv) and data cycle (Td) respectively. μdTd is the number of packets serviced during the data cycle. Figure 8 shows the growth of the data queue from E[Q1] to E[Q3] after a few voice/data cycles.

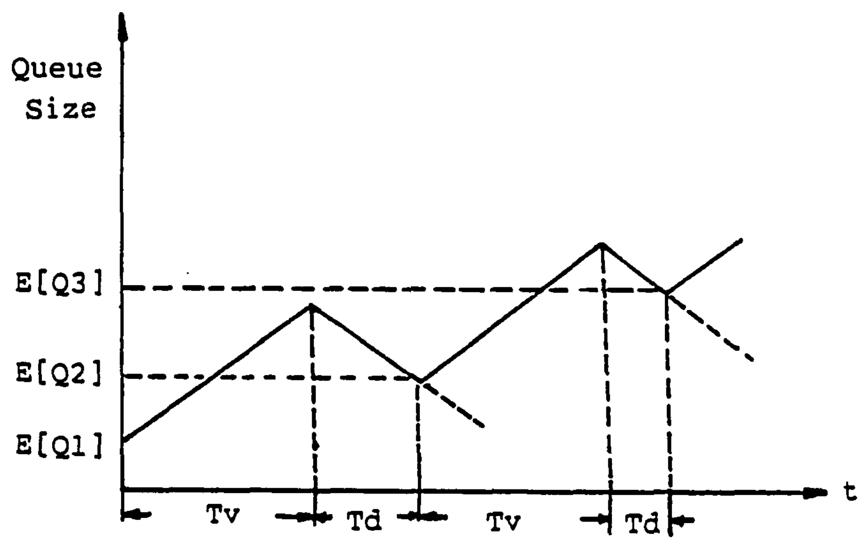


FIGURE 8. Growth of data queue

Notice here that the increase of the data queue size may not be a serious problem for a data type which does not require fast response. However, the data queueing problem is a serious problem for those data which do require fast response.

CHAPTER 3. CURRENT WORK

To handle integrated voice/data traffic in LANs, there have been two different approaches, based on two different MAC layer protocols: CSMA/CD (Carrier Sense Multiple Access/Collision Detection) and token ring. For CSMA/CD protocol, the feasibility of using CSMA/CD for voice traffic has been studied [42] and a few integrated voice/data protocols based on CSMA/CD have been proposed [43-45]. The token ring protocol seems more suitable than CSMA/CD protocol in handling the integrated voice and data for the following reasons:

- Delay is bounded in token ring but not in CSMA/CD.
- Token ring is less sensitive to workload than CSMA/CD particularly under heavy load.

In the following section, current research work on integrated voice/data (IVD) protocols based on token ring protocol will be examined [19-23].

IVD Protocols For Token Ring Network

Since Hafner et al. [27] introduced the loop network to support facilities such as alarms, controls, and voice, several IVD protocols based on token ring protocol have been proposed. These protocols, which basically employ priority mechanisms to give less network access delay to voice packets, can be classified into two different groups according to how the channel is scheduled for each type of traffic (voice/data).

- Centralized Scheduling Method
- Distributed Scheduling Method

Centralized scheduling method

In this method, a central station is responsible for allocating channel for each traffic type. Protocols proposed by Bux et al. [19], Mark [20], and Saydam and Sethi [21] fall into this category. The principal idea of the centralized method can be seen in Bux's protocol. In the following sections, three protocols using centralized scheduling method will be explained.

Synchronous IBM token ring protocol Bux et al. [19] developed a synchronous token ring protocol at IBM to handle voice traffic. Figure 9 illustrates the operation of the protocol proposed by Bux, which works in the following manner:

1. A central Monitor (CM) changes the priority of a token at regular intervals, each of which is equal to the packetization period (Tp).
2. When the token is assigned a high priority (P=1) only voice stations can access the network, and when priority is low (P=0) only data stations can access the network.
3. When the CM receives the token, in order to

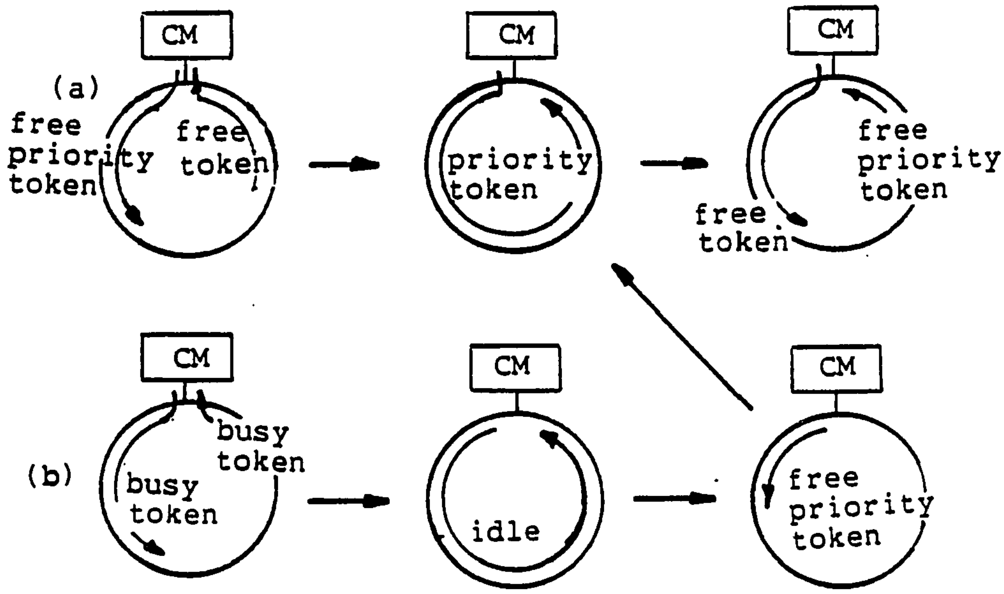


FIGURE 9. Operation of the synchronous token ring protocol (a) free token case (b) busy token case

change its priority from low to high the following cases can occur:

- token free: the priority will be changed immediately as shown in Figure 9(a).
- token busy: the CM waits until the current data station finishes its transmission. Idle characters follow the busy token to prevent other data stations from accessing the channel as shown in Figure 9(b).

Wong and Gopal [26] used the synchronous protocol to examine the effects of the number of voice stations and the data load on the coefficient of variation (CV) for the interdeparture time of voice packets at the destination. No other analysis on the performance of this protocol has been done. In Chapter 6, a protocol of this kind is simulated and shows that a poor quality of voice results from only a medium-sized data load.

Dual ring protocol Mark [20] proposed a dual ring protocol to handle integrated voice/data. In his protocol, a main ring is used for data/voice transmission and a side ring is used for scheduling priorities of the token. The dual ring protocol is almost same as Bux's except that the side ring is used for scheduling priorities and only the data stations transmitting just prior to the voice cycle can access network in the first data cycle immediately after the voice cycle. The main contribution of Mark is an analysis of the data queueing problem using the fluid approximation method [46].

Random priority protocol The random priority protocol was proposed by Saydam and Sethi [21]. In this protocol, a central monitor assigns priorities randomly at the beginning of each token rotation. When the circulating free token has high priority only voice stations may capture it to transmit a message, but when the free token has low priority, both data and voice stations may capture it. The performance of this protocol was evaluated with mean delay and mean queue length of voice and data by using a gated

M/G/1 queueing model. As described by Saydam, the data delay can become a serious problem when the channel is frequently assigned to voice traffic.

Distributed scheduling method

In this method, a channel may transfer either data cycles or voice cycles, and each station is responsible for deciding whether to transmit voice packets or not when the free token arrives to a voice station. Ibe and Gibson [22] developed a distributed scheduling protocol using this transmission mechanism. His protocol works as follows:

1. Every voice station has a time window Δ. A timer for the time window starts when a voice packet arrives at the output buffer.
2. If token arrives within this time window then a voice packet can be sent to the network, otherwise the packet will be discarded.

Figure 10 depicts the role of the time window.

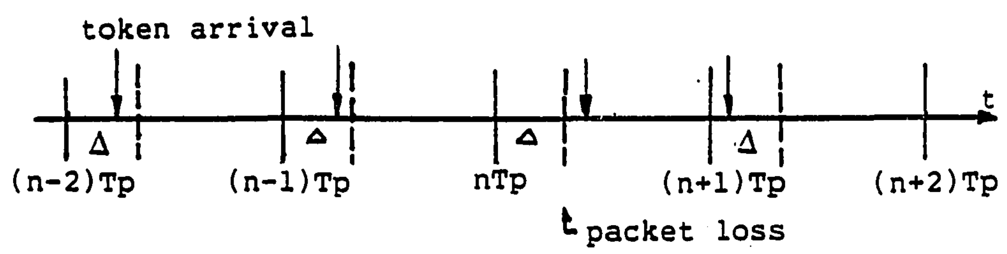


FIGURE 10. Distributed scheduling protocol

While the distributed protocol is more reliable than the centralized protocol because of the distributed function, it has drawbacks in terms of the performance of voice and data transmission. As described in Ibe's paper, this protocol suffers more than 2% of voice packet loss even under light data traffic (λd=0.2) conditions, and reasonable voice quality cannot be obtained. Later, Ibe and Gibson [23] enhanced the performance of the protocol by adding a flow control method for data stations. This flow control method limits the total token cycle time caused by data stations and, as a result, reduces the number of voice packets discarded in each voice station. In the performance simulation of this improved protocol, more than 2% of voice packets are still lost when the data traffic load is over 0.4.

Table 2 summarizes the proposed protocols for token ring network along with their advantages and drawbacks.

TABLE 2. Summary of the current IVD token ring protocols

Scheduling Method	Name	Techniques used	Performance Parameters	Problems & Comments
Centralized	Bux	CM changes priority periodically	coefficient of variation of interarrival time	no performance analysis on data/voice
	Mark	CM changes priority periodically using side ring	size of data queue	no analysis on voice
	Saydam	CM changes priority randomly	mean delay & mean queue size of voice & data	no analysis on voice & data suffers long delay
Distributed	Ibe	every voice station uses transmission window	loss of voice packets & data delay	high loss of voice packets at data load over 0.2
	Ibe	window for voice station +flow control of data stations	loss of packets & data delay	high loss of voice packets at data load over 0.4

CHAPTER 4. DESIGN OF THE IVD-TOKEN RING PROTOCOL

As described in the previous two chapters, the IVD protocol appears to satisfy the performance requirements of both voice and data. In this chapter, a design approach for designing an IVD-token ring protocol to achieve satisfactory performance of voice and data is described. In this approach, the IEEE 802.5 token ring protocol [12] is assumed as a protocol used by data stations and a single buffer is assumed for voice stations to minimize the end-to-end delay. The protocol is designed with the aid of the simulation model discussed in Chapter 6. Five design issues related to the problems raised in integrating voice and data on the token ring network are studied. The issues are:

- Packet format
- Limit of data packet size (MAX(PSd))
- Allocation of cycles for voice and data
- Synchronization Delay (Ds) at the receiver
- Packetization period (Tp)

Packet Format

Because data stations and voice stations are connected point to point on the ring, the packet format of a voice station must be common with that of a data station. Since IEEE 802.5 is selected as the data protocol, the protocol for voice stations must therefore follow the format of IEEE

802.5. A priority mechanism (in the Access Control field) is used to differentiate the voice packet from data packet and a reserved bit in the FS (Frame Status) field is used for the operation of the voice protocol. Figure 11 shows the packet format of the IVD-token ring protocol along with the modified bit in the FS field.

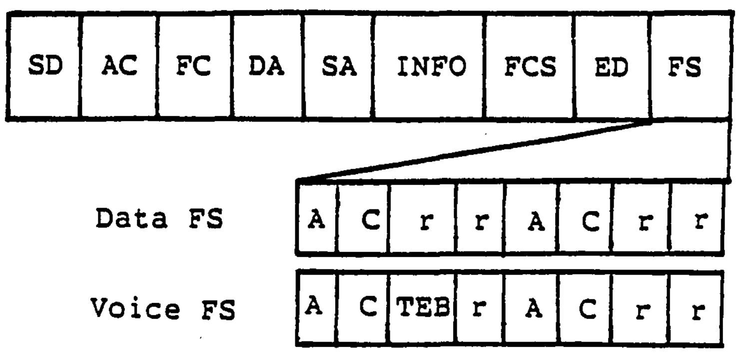


FIGURE 11. Packet format of the IVD-token ring protocol

The purpose of this modified bit (Timer Expired Bit) is to inform the other voice stations they can transmit voice packets. A more detail explanation regarding this bit's function is given in the next chapter.

Limit of Data Packet Size (MAX(PSd))

Chapter 2 describes the two principal problems of voice packets in LANs: limited end-to-end delay and variation of access delay. These problems are depicted again in Figure

12. In Figure 12, the effect of unlimited maximum data packet size is shown. Here, a data cycle longer than the packetization period (Tp) occupies the channel after a voice cycle. The reason for this can be found in the definition of the data cycle (Td):

$$Td = Nd * PSd/Rc \tag{4.1}$$

where Nd is the number of data stations on the ring, PSd is the size of the data packet, and Rc is the channel bit rate. In this equation, overhead such as token passing time, propagation delay, and station delay are ignored. If PSd is not limited, Td may be larger or smaller than Tp. Because a network access opportunity is not guaranteed in each packetization period, the variation of network access delay may be very large. As a result, voice packets may not regularly arrive their destinations and a continuous voice signal may be impossible to obtain at the receiver.

Up to this point, only the quality of voice traffic and its relationship to the limit of the size of a data packet has been considered. It is also important to find the effects of limiting data packet size on data performance. Figure 13 and 14 show the delay and throughput of data at various data traffic levels for different packet sizes: infinite, 4kbits, 640bits. The delay performance of data is shown in Figure 13. As shown in this figure, the delay does not show any difference up to mid-level load for these three

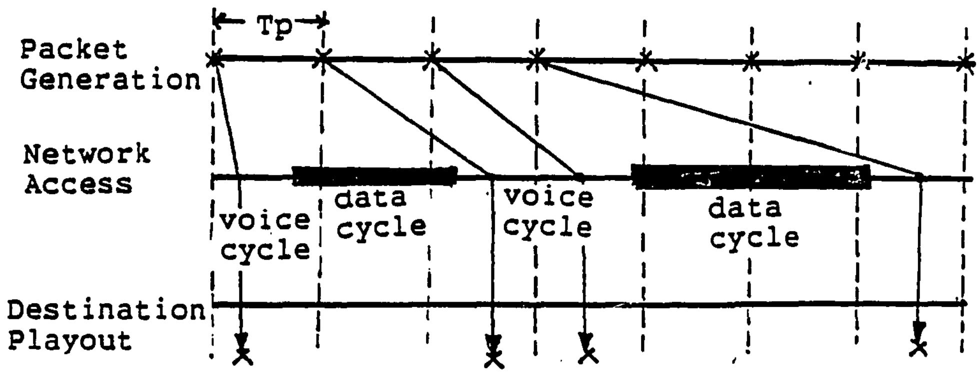


FIGURE 12. Effects of unlimited data packet size

cases. Notice, however, that the performance of delay becomes quite different in heavy data load conditions, where delay is increased significantly when data packet size is 640 bits. As shown in Figure 14, throughput for infinite and 4kbit packet size is nearly equal and there is a slight throughput decrease for small data packet size (640bits) in the heavy traffic region. From these results, the following conclusions can be deduced in deciding the data packet size. There is a need to limit data packet size for good voice performance. However, if too small a data packet size is selected, then the data transmission performance would drop. To achieve desirable performance for both voice and data, a proper data packet size must be selected. In the proposed IVD-token ring protocol, 4kbit is selected as the data packet size.

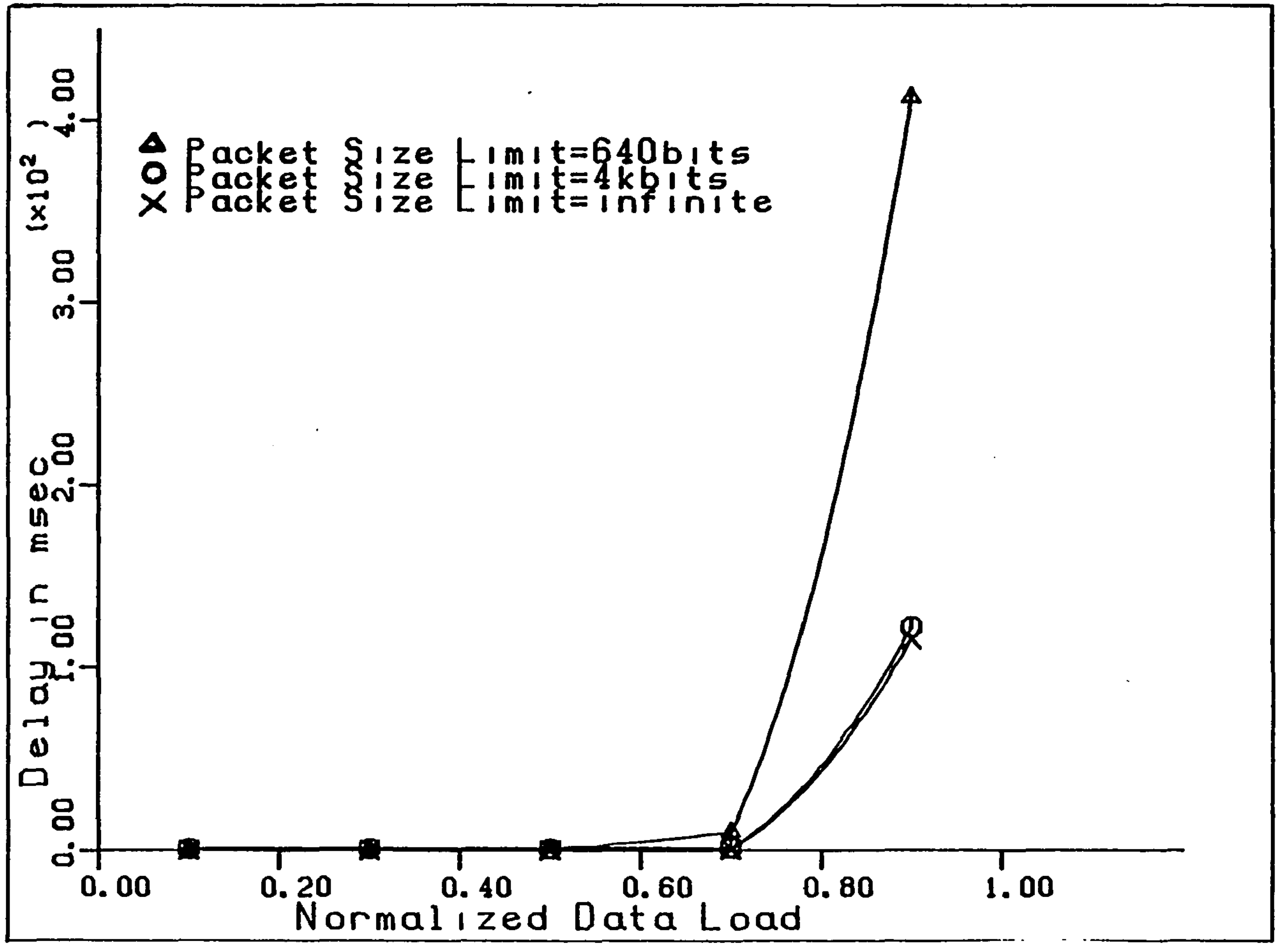


FIGURE 13. Queueing delay of three different data packet sizes

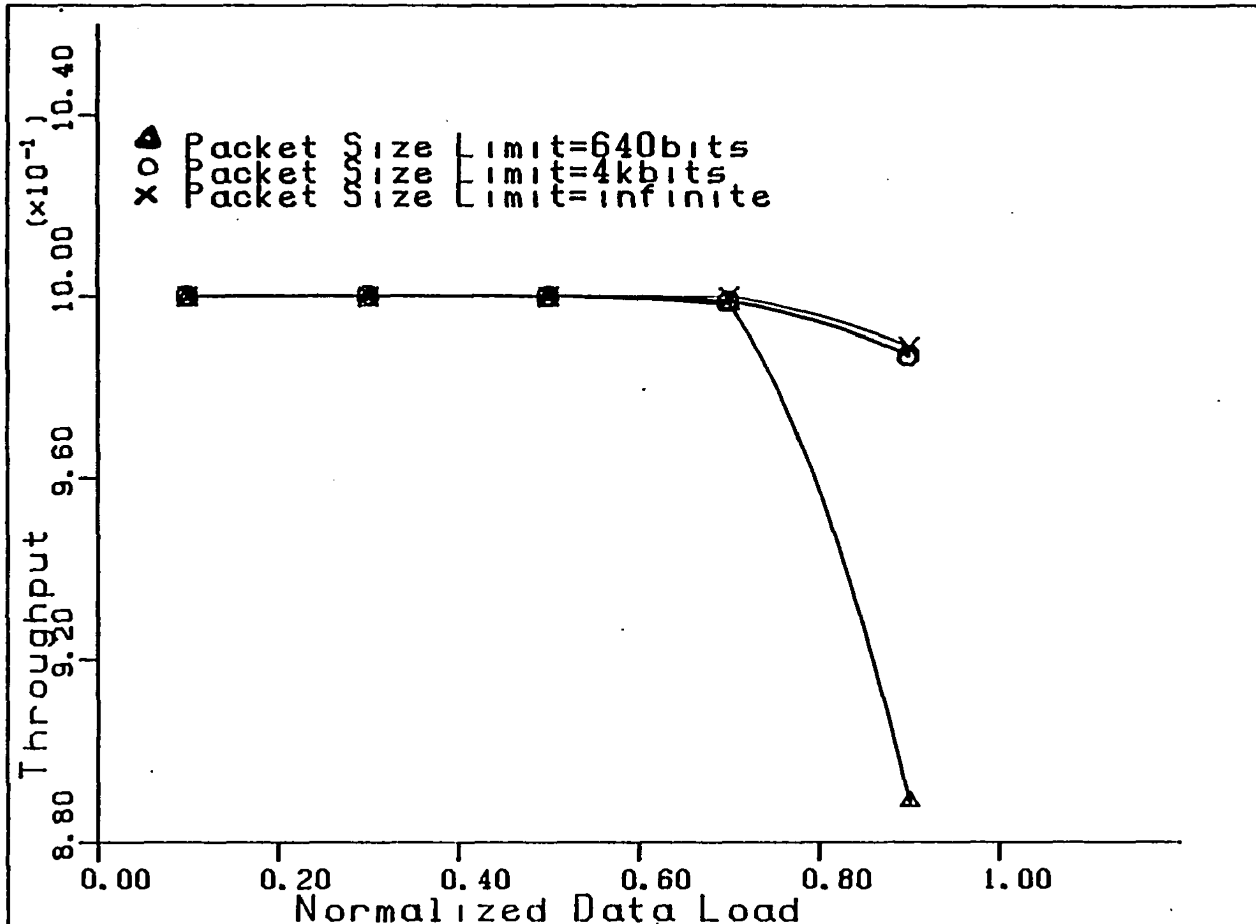


FIGURE 14. Throughput of three different data packet sizes

Allocation of Cycles for Voice and Data

As seen in the previous chapter, the performance of voice and data depends heavily on the cycles allocated to each traffic type. This section starts its discussion by describing the problems of simple allocation strategy: a single voice cycle during a packetization period (Tp) and data cycles for remaining of Tp. Then, cycle allocation strategy to achieve high performance of both voice and data is desribed. Notice that the problem of allocating cycle to each traffic type is the problem of deciding how many voice and data stations to support on the ring network when the packet size of each type of traffic is selected. This can be seen in the following equations which define voice and data cycles (Tv,Td):

$$Tv = Nv(PSv + OH + 64)/Rc \tag{4.2}$$

$$Td = Nd(MAX(PSd) + OH + 64)/Rc \tag{4.3}$$

where OH is the station delay overhead in bits, and a 64 bit packetization overhead is assumed. As seen in these equations, Tv and Td are directly proportional to Nv and Nd because other terms are constant.

Since a voice packet is generated during every voice packetization period (Tp), and voice packets generated must be sent once in every Tp (whether the packet is buffered or not) for the synthesis of the continuous voice signal at the receiver, there must be at least one voice cycle (Tv) in a packetization period.

To meet another requirement of the IVD protocol, that of maintaining the satisfactory performance of data traffic while producing high quality voice output, data cycles (Td) must be allocated in the remainder of the packetization period. The relationship of these cycles can be expressed in (4.4) which includes the token passing overhead (To).

$$Tv + Td + To \le Tp \qquad (4.4)$$

To is defined as:

$$To = 24(Nv + Nd)/Rc \qquad (4.5)$$

Nv and Nd, obtained by substituting (4.2) and (4.3) into (4.4), are shown in Table 3 for assumed values of PSv=640 bits, MAX(PSd)=4kbits, OH=1 bit, Tp=10msec,and Rc=10Mbps, and for Tv/Tp varying from 0.1 to 0.9.

TABLE 3. Nv and Nd for a single voice cycle in Tp

Tv/Tp	Nv	Nd
0.1	15	22
0.2	30	19
0.3	45	17
0.4	60	14
0.5	75	12
0.6	90	9
0.7	105	7
0.8	120	4
0.9	135	2

One problem here is how much fraction of Tp can be allocated to Tv without degrading data performance when data load varies. Allocating one voice cycle during a Tp may cause problems in reconstructing the continuous voice signal because a voice packet arriving immediately after the departure of the token will be lost. This problem is illustrated in Figure 15.

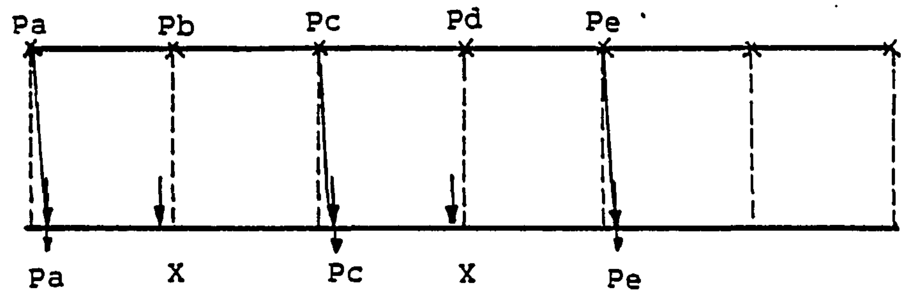


FIGURE 15. Voice packet loss

In Figure 15, the voice packet, Pb, arrives immediately after the departure of a token and the next free token arrives after the arrival of packet Pc. Here, the packets Pb and Pd will be lost.

A typical example of the problem of allocating one voice cycle within a Tp interval occurs when voice stations are newly connected or disconnected. Figure 16 shows this problem in a voice station. Notice that, in this figure, several voice stations are newly connected and disconnected after the packet Pi.

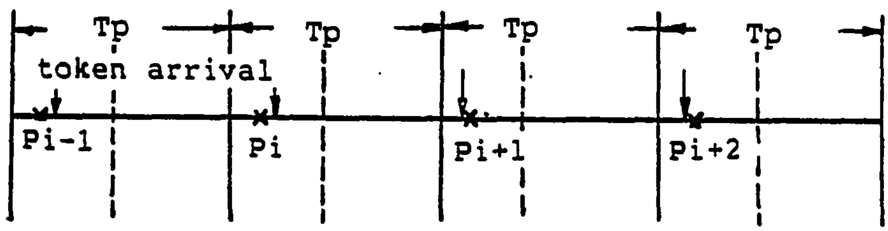


FIGURE 16. Effects of change in number of active voice stations

Because of the change in the active number of voice stations, the token may arrive earlier than it did in the previous cycle. If this happens, voice packets may arrive later than the arrival of the token and these packets will be lost. Notice that, once this occurs, there is a high probability of loss of consecutive packets if the data load does not change drastically.

As mentioned above, the principal reason for losing packets is the variation of the token cycle time. That is, voice packets are generated in a voice station at each voice cycle time, but the token may arrive immediately after or before packet generation because of the variation of this cycle time, and a packet which is generated later than token arrival will be lost. Simulation results shown in Chapter 7 show the high percentage of lost packets in a single voice cycle for a packetization period.

Up to now, problems of allocating cycles for each different traffic have been described. To resolve these problems the following strategy is adopted:

1. Sender is solely responsible for quality of voice. The sender must gurantee no more than 2% of the voice packets are lost and the receiver just plays out the arrived voice packets.
2. To achieve high performance for the interactive data service, those data packets are allowed to be sent during any cycle.
3. When voice stations are idle data stations are allowed to use the ring as a data cycle.

As mentioned in the previous section, the principal reason of lost packets is the variation of token cycle time. To solve this problem at the sender level, the following facts are used. Simulation results show that most of the lost packets have a slight time difference between the token arrival and packet arrival regardless of the data load variation when the Tv and data packet size is bounded. This is because the variation of Tv is majorly affected by the variation of active voice station number rather than the effects of data traffic. (Max(PSd)/Rc is as small as 0.4 microsec.) This brings the idea of employing a second voice cycle after a normal voice cycle to handle the packets arrived later than the token arrival. The second voice cycle is first assumed as large as 0.5Tv. If this assumption works, then the recovery of voice packets at the receiver is a simple process: just plays out at the same interval as that of sender after considering the maximum variation of the interarrival time. This mechanism is explained in the next section. The voice performance using this approach is evaluated in Chapter 7.

The maximum number of voice and data stations following the above strategy can be obtained by modifying (4.4):

$$Tv + p(Tv + Tol) + Td + To <= Tp \qquad (4.6)$$

where p is the probability of losing a chance to transmit in the first voice cycle and Tol is the token passing overhead for second voice cycle. Using the equations for Tv, Td, and To in (4.2), (4.3), and (4.4), Nv and Nd can be obtained for various values of p. Table 4 shows the supported number of voice and data stations for various assumed p values when Rc=10Mbps, Tp=10msec, PSv=640bit, MAX(PSd)=4000bit, station overhead=1bit, and propagation delay=5microsec.

Up to now, only the voice performance has been considered in allocationg cycles. Next problem is how much of the Tp can be allocated for Tv without degrading data performance. For the selection of this maximum boundary of Tv, the delay performance of data packets are examined by varying data load and Tv/Tp using the simulation model. The result is shown in Figure 17 and can be used in selecting

TABLE 4. Nv and Nd in the proposed protocol

Tv/Tp	p	Tvl (msec)	Tvl/Td	Nv	Nd	Tv/Tp	p	Tvl (msec)	Tvl/Td	Nv	Nd
.10	.05	.05	.01	13	22	.60	.05	.30	.07	82	9
.10	.10	.09	.01	13	21	.60	.10	.60	.15	82	8
.10	.15	.14	.02	13	21	.60	.15	.90	.22	82	7
.10	.20	.19	.02	13	21	.60	.20	1.20	.30	82	6
.10	.25	.24	.03	13	21	.60	.25	1.49	.37	82	6
.10	.30	.28	.03	13	21	.60	.30	1.79	.45	82	5
.10	.35	.33	.04	13	21	.60	.35	2.09	.52	82	4
.10	.40	.38	.04	13	21	.60	.40	2.39	.60	82	3
.10	.45	.43	.05	13	21	.60	.45	2.69	.67	82	3
.10	.50	.47	.05	13	20	.60	.50	2.99	.75	82	2
.20	.05	.10	.01	27	19	.70	.05	.35	.12	96	6
.20	.10	.20	.02	27	19	.70	.10	.70	.23	96	5
.20	.15	.30	.04	27	18	.70	.15	1.05	.35	96	4
.20	.20	.39	.05	27	18	.70	.20	1.40	.47	96	3
.20	.25	.49	.06	27	18	.70	.25	1.75	.58	96	3
.20	.30	.59	.07	27	17	.70	.30	2.10	.70	96	2
.20	.35	.69	.09	27	17	.70	.35	2.45	.82	96	1
.20	.40	.79	.10	27	17	.70	.40	2.80	.93	96	-
.20	.45	.89	.11	27	17	.70	.45	3.15	1.05	96	-
.20	.50	.98	.12	27	17	.70	.50	3.50	1.17	96	-
.30	.05	.15	.02	41	16	.80	.05	.40	.20	109	4
.30	.10	.30	.04	41	16	.80	.10	.79	.40	109	3
.30	.15	.45	.06	41	16	.80	.15	1.19	.60	109	2
.30	.20	.60	.09	41	15	.80	.20	1.59	.79	109	1
.30	.25	.75	.11	41	15	.80	.25	1.99	.99	109	-
.30	.30	.90	.13	41	14	.80	.30	2.38	1.19	109	-
.30	.35	1.05	.15	41	14	.80	.35	2.78	1.39	109	-
.30	.40	1.20	.17	41	14	.80	.40	3.18	1.59	109	-
.30	.45	1.35	.19	41	13	.80	.45	3.58	1.79	109	-
.30	.50	1.49	.21	41	13	.80	.50	3.97	1.99	109	-
.40	.05	.20	.03	54	14	.90	.05	.45	.45	123	-
.40	.10	.39	.07	54	13	.90	.10	.90	.90	123	-
.40	.15	.59	.10	54	13	.90	.15	1.35	1.35	123	-
.40	.20	.79	.13	54	12	.90	.20	1.79	1.79	123	-
.40	.25	.98	.16	54	12	.90	.25	2.24	2.24	123	-
.40	.30	1.18	.20	54	11	.90	.30	2.69	2.69	123	-
.40	.35	1.38	.23	54	11	.90	.35	3.14	3.14	123	-
.40	.40	1.57	.26	54	10	.90	.40	3.59	3.59	123	-
.40	.45	1.77	.30	54	10	.90	.45	4.04	4.04	123	-
.40	.50	1.97	.33	54	10	.90	.50	4.48	4.48	123	-
.50	.05	.25	.05	68	11						
.50	.10	.50	.10	68	11						
.50	.15	.74	.15	68	10						

TABLE 4 (continued)

Tv/Tp	p	Tvl (msec)	Tvl/Td	Nv	Nd	Tv/Tp	p	Tvl (msec)	Tvl/Td	Nv	Nd
.50	.20	.99	.20	68	9						
.50	.25	1.24	.25	68	9						
.50	.30	1.49	.30	68	8						
.50	.35	1.74	.35	68	8						
.50	.40	1.98	.40	68	7						
.50	.45	2.23	.45	68	6						
.50	.50	2.48	.50	68	6						

the maximum boundary of Tv when the network load is predictable.

From this figure, about 30% of Tp can be allocated to voice cycles (20% for first voice cycle and 10% for second voice cycle) without degrading data performance if data load is 0.5. Figure 17 along with the Table 4 can be used in deciding the maximum number of voice and data stations to achive a high performance for both voice and data. In the performance evaluation of this protocol, data load is assumed 0.5 and p is assumed 0.5. (From Table 4 and Figure 17, maximum boundary of Tv is 30% of Tp with Nv=27, and Nd=17.) Because of the simulation language limit 20 conversational voice stations and 20 data stations are assumed to be on the ring.

To enhance the data performance further, interacive data packets are allowed to be sent in any cycle and a data cycle will be scheduled when all the voice stations are idle or in silence state. The detail of this implementation will be discussed in the implementation issues of Chapter 5.

Synchronization Delay (Ds) at the Receiver

To construct a continuous voice signal at the receiver, voice packets arriving at the receiver must be played out and depacketized at the same rate as they were generated at the sender. Montgomery [37] summarized the four possible

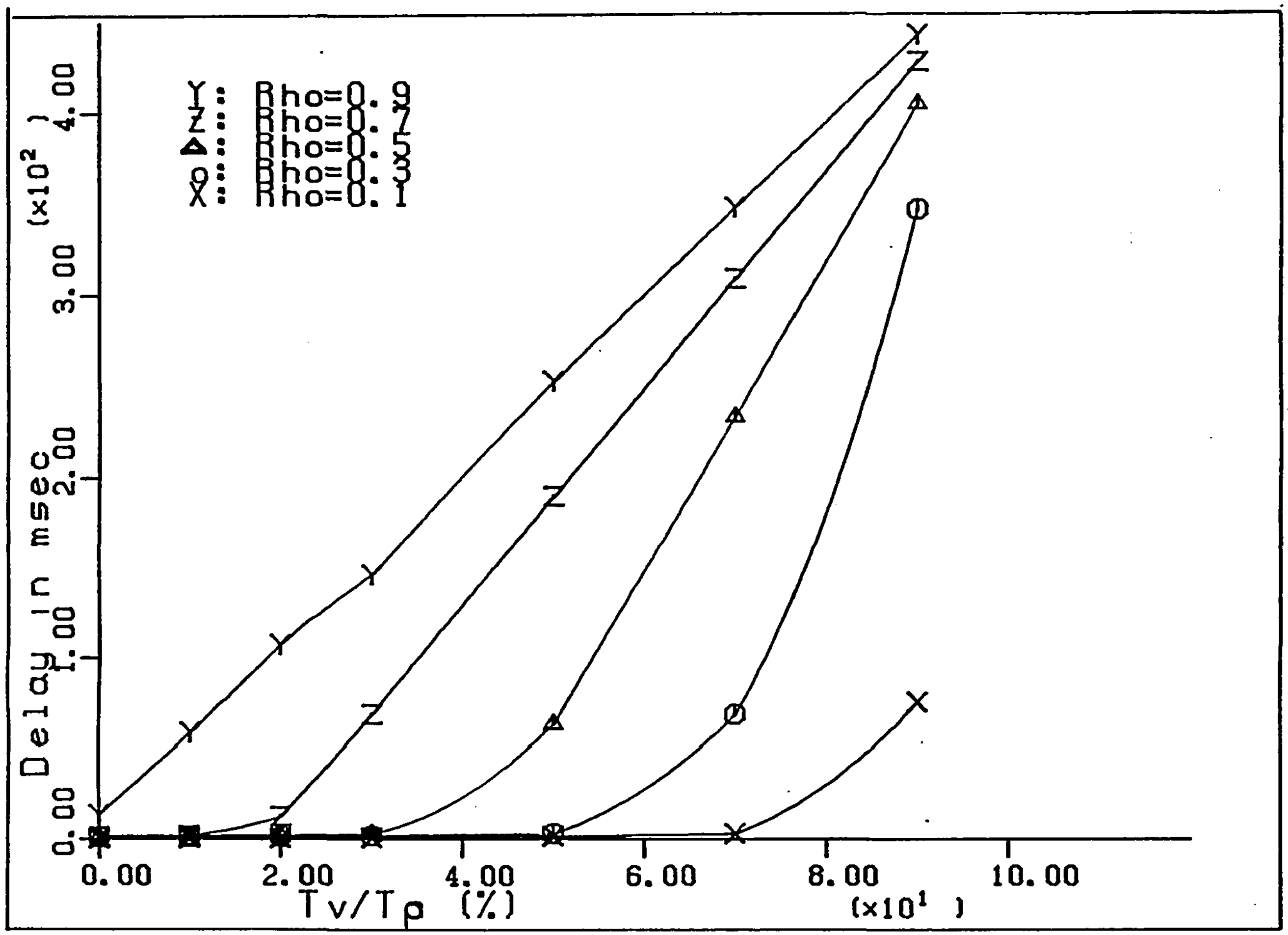


FIGURE 17. Tv/Tp effects on data performance

solutions for the synchronization of packets at the receiver in a packet network. Among these four techniques, the blind method using buffers [38-39] is used in this research because of its simplicity.

Since the late arrival packet problem has been handled at the sender (by allowing a second voice cycle for those packets just missing their chances to transmit), the only matter to be considered at the receiver is the variable network delay (network access delay + network transmission delay) which each packet experiences. Since each packet experiences a different network delay, a maximum variance of network delay, called the synchronization delay (Ds), must be allowed at the receiver before the packets are played out at regular intervals for the depacketization. In selecting Ds, a value less than the maximum variance access delay must be avoided because a large fraction of packets arrive late for playout, resulting in a high loss of packets at the receiver. If Ds is selected too large, some voice packets may suffer longer delay than the end-to-end delay limit. Ds must be properly selected to produce a high quality of vocie at the receiver. In the following, the method for selection of Ds will be explained.

The variance of the network access delay is due to two components, Dad and Dav, as illustrated in Figure 18. In this figure, Dad is the variance of delay caused by a

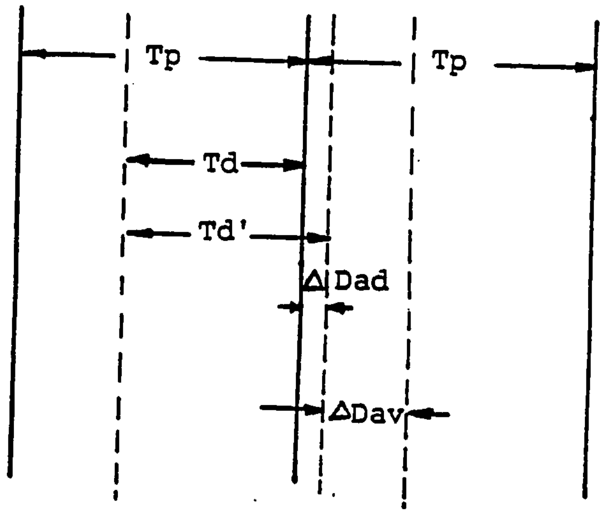


FIGURE 18. Variance of network access delay

data station. The data station which is transmitting a packet cannot stop immediately when the voice cycle is requested, so the actual data cycle will be Td', and the variation caused by this data station can be expressed as follows:

$$\Delta Dad = Td' - Td \qquad (4.7)$$

Since the maximum size of a data packet is known, the maximum variation of Dad is:

$$MAX(\Delta Dad) = MAX(PSd)/Rc \qquad (4.8)$$

Another variation of network access delay is caused by the location of voice stations and its maximum value is:

$$MAX(\Delta Dav) = (Nv - 1)*PSv/Rc \qquad (4.9)$$

The worst-case variance of network access delay is the sum of (4.8) and (4.9).

$$MAX(PSd)/Rc + (Nv - 1)*PSv/Rc < Ds \qquad (4.10)$$

When the following network values are substitutied in the above equation, Ds can be obtained for the assumed case.

MAX(PSd) = 4Kbit

PSv = 640 bit

Nv = 40 stations

Rc = 10Mbps

The worst-case Ds must be greater than 2.96msec.

Packetization Period (Tp)

Figure 19 shows all elements contributing to end-to-end delay experienced by a voice packet when it is transmitted through a voice packet network.

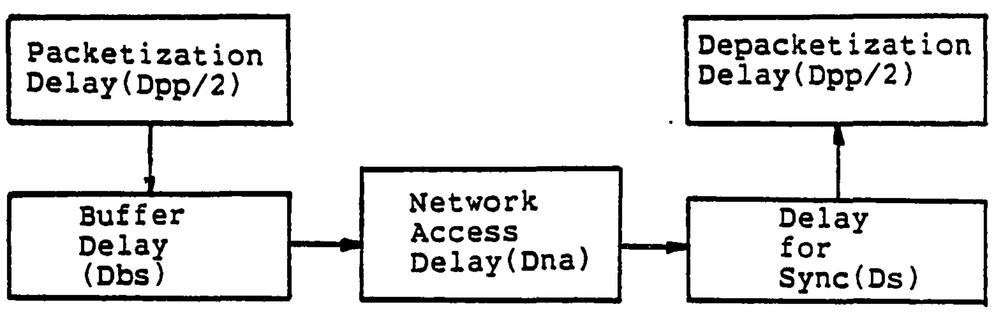


FIGURE 19. Elements of end-to-end delay of a voice packet

As explained in the previous chapter, the end-to-end delay of a voice packet must be less than 200msec to preserve the interactive nature of conversation. Because these delays are related to the packetization period, the selection of Tp is decided by the condition that the total end-to-end delay must be less than 200msec.

The packetization and depacketization delay (Dpp)

In a voice station, the voice signal must be sampled and packetized to be sent to the network and the packet must be depacketized at the receiver. The delay for packetization and depacketization (Dpp) can be expressed as:

$$Dpp = 2 * Tp \qquad (4.11)$$

Sender buffer delay (Dbs)

Because of the network access delay, at least one buffer (Nbs) is needed at the sender. If the token cycle time is longer than the packetization period , more buffers are needed to hold voice packets. If there is only one buffer at the sender, the voice packet will experience only the network access delay. However, if there are multiple buffers the delay at the sender will be increased. Generally, the delay caused by the sender buffers is:

$$Dbs = (Nbs - 1) * Tp \qquad (4.12)$$

where Nbs is the number of buffers in the sender (Nbs >= 1).

Network access delay (Dna)

The network access delay of a voice packet also affects the allocation strategy of a channel. In the proposed protocol, a second voice cycle is allowed for those packets which just missed a chance for transmission. Because of this strategy, only one output buffer is needed in the sender. In this case, the worst network access delay (Dna) is Tp. So, the worst network access delay of the prepared voice packet is:

$$Dna = Tp \tag{4.13}$$

Since the total end-to-end delay must be limited to 200 msec:

$$Dpp + Dbs + Dna + Ds <= 200msec \tag{4.14}$$

By (4.10), (4.11), (4.12), and (4.13):

$$Tp <= (200msec - Ds)/(Nbs + 1.5) \tag{4.15}$$

When Nbs=1 and Ds=3msec,

$$Tp <= 78.8msec \tag{4.16}$$

In the proposed IVD-token ring protocol, Tp is selected as 10msec to limit the available number of voice/data stations on the ring. This is mainly because the total simulation time is proportional to the number of stations on the ring as will be explained in the next chapter.

CHAPTER 5. IVD-TOKEN RING PROTOCOL

In this chapter, implementation issues, specification, and operation of the proposed protocol will be explained. The IVD-token ring protocol described here is based on the design approach described in the previous chapter. The network parameters selected using this design approach are given in Table 5.

TABLE 5. System parameters for IVD-token ring protocol

Tv/Td	Ds	MAX(PSd)	Ts	Rv	PSv	Nbs	Nbr	Nv	Nd
1/4	3msec	4K	10msec	64kbps	640bit	1	1	40	20

Packet Format and Facilities

The packet format of IEEE 802.5 [12] is used in the proposed IVD-token ring protocol, with only one reserved bit in the FS field modified to permit indication of the start of a voice cycle. Since the proposed protocol is based on the IEEE 802.5 data protocol, the functions of flags/timers in data stations are those described in the standard. For voice stations, a voice timer and two flags are added in addition to the timers and flags which already exist in data stations. The functions of these timers and flags in voice stations are:

- VTMR (Voice Timer): The purpose of this timer is to provide voice stations with the information that the packetization timer has expired and to enable the voice station to reserve the channel when voice packets are ready for transmission in their output buffers. Every voice station has such a timer but only one timer in the network is active at a time and the timers in other stations are idle and will be active when the active timer fails. The initial value of this timer is the packetization period (Tp). When this timer is expired, TMRF flag will be set to 1 and a new Tp will be loaded.
- TMRF (Timer Flag): TMRF will be set to 1 when the VTMR is expired. This flag is used to modify the TEB (Timer Expired Bit) bit in the FS field when a packet passes by a station. After the TEB bit is modified, the TMRF will be reset to 0.
- ICF (Init voice Cycle Flag): This flag is set when a free priority token (in the first voice cycle) is passed to the next station. This flag is used for voice stations which just miss packet transmissions because of late arrival packets. When ICF=1 and the output buffer of a voice station is not empty, the channel is reserved for the second voice cycle (R=1).

Since the functions of various fields are explained in detail in the IEEE 802.5 standard and since functions of most fields are unchanged in the proposed protocol, only those fields which are modified will be described here. They are:

- P (Priority bit): Among the four levels of priorities, three levels of priorities are used to serve voice packets and two different kinds of data packets. When the ring priority is highest (P=2), voice packets and data packets of interactive type are allowed to transmit. In the second highest priority (P=1), voice packets having just missed transmission and data packets of the interactive type are transmitted through the network. When the priority is lowest (P=0), only data packets (of any type) are sent through the network.
- TEB (Timer Expired Bit): This bit is used to inform voice stations that the VTMR has expired. By reading this bit any voice station ready to transmit can reserve the channel for voice cycles. Notice that, even when TEB=1, the channel may not be reserved for voice cycles if all voice station buffers are empty. This bit is set to 0 when the first voice cycle starts (by the station which pushes the ring priority).

- R (Reserve field): This field is used to reserve the channel for voice cycles (either first or second). When at least one voice packet is queued and TEB is 1, the channel will be reserved (R=2) for the first voice cycle. When the priority of the channel is 2, ICF=1, and voice packets are queued, the channel will be reserved for the second voice cycle (R=1).

Operation of the IVD Token Ring Protocol

The operation of the proposed IVD-token ring protocol is described as follows:

- The channel usage is divided up into first voice cycles, second voice cycles, and data cycles. Data packets of interactive type can access the network during any cycle. In the second voice cycle, the voice packets which have just missed being transmitted in a first voice cycle are sent. Only data packets (of any type) can be sent during data cycles.
- When VTMR expires (TMRF=1), the TEB shall be set to 1 if the token is busy (i.e., a data station is transmitting a packet) or a voice cycle should be immediately established if the free token is passed to the station whose TMRF=1 and whose output buffer

is not empty.

- When this TEB is set to 1 (i.e., VTMR has expired) the data cycle is interrupted and a voice cycle is initiated if voice packets have been placed in their output buffers. If all the buffers are empty when TEB=1, then the data cycle will be uninterrupted. Figure 20 shows a cycle change when TEB=1.

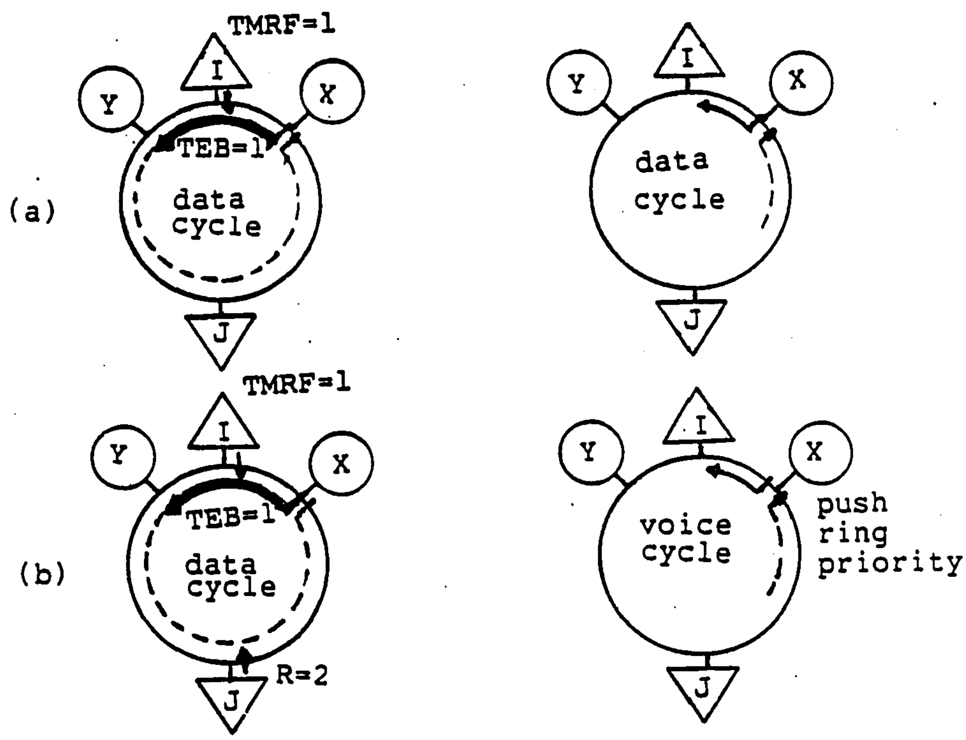


FIGURE 20. Cycle change when TEB=1 (a) Uninterrupted data cycle (b) Interrupted data cycle

Figure 20(a) shows the uninterrupted cycle when TEB=1. In this figure VTMR in station I expires during the transmission from data station X to Y. Since all the voice station buffers are empty at the time when TEB=1, the channel is not reserved for voice and the data cycle continues without any interrupts. When a voice station J whose output buffer is not empty finds TEB=1, the channel will be reserved (R=2) ring priority pushed by transmitting data station X and the cycle is changed to a first voice cycle as shown in Figure 20(b).

- During a first voice cycle, all voice stations and data stations with buffered packets of interactive type will have chances to transmit. When the free token passes by voice stations in the first voice cycle, each ICF shall be set to 1.
- If a voice packet arrives an output buffer when ICF=1, the station reserves the channel for a second voice cycle for packets arriving after token departure. When the first voice cycle is finished, the same station (X in the figure) that pushed ring priority will change ring priority. The cycle change from a first voice cycle is shown in Figure 21. Figure 21(a) shows the change from the first

voice cycle to the second voice cycle when the channel is reserved by voice stations with buffered voice packets arriving later than token departure in a first voice cycle. Notice that voice cycle will be changed immediately to a data cycle when the channel is not reserved for the second voice cycle, as shown in Figure 21(b).

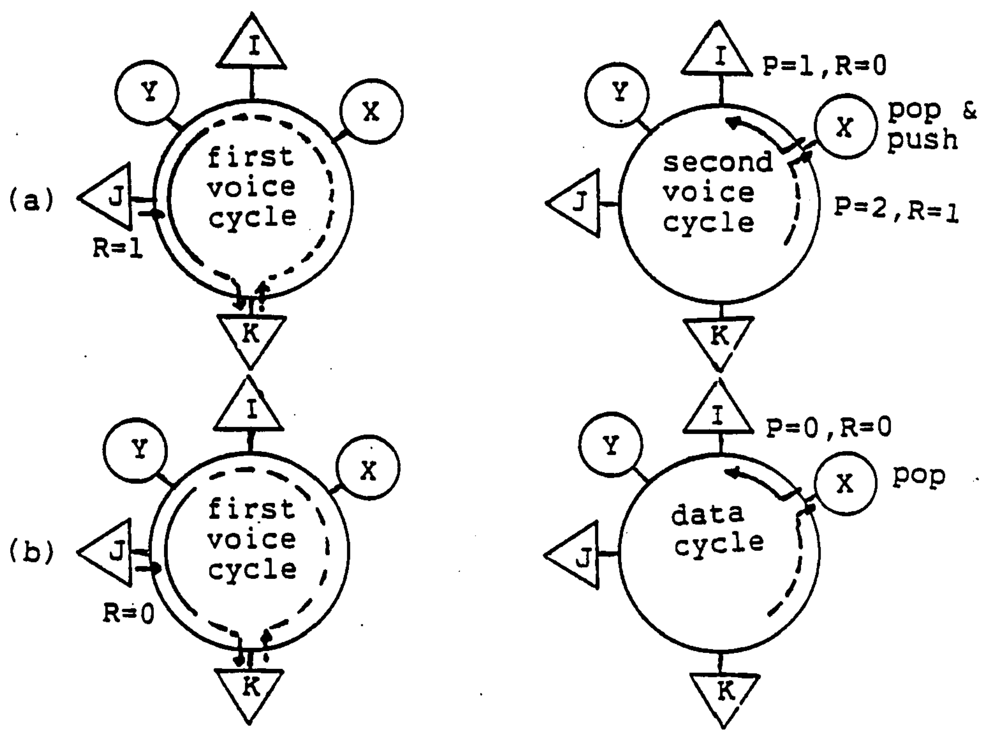


FIGURE 21. Cycle changes from a first voice cycle (a) when R=1 (b) when R=0

- When voice cycles are completed, the station which originally pushed ring priority is responsible for recovering the original ring priority.

The operation of the proposed IVD-token ring protocol is summarized in Figure 22.

Example

An example of the operation of the IVD-token ring protocol is shown in Figure 23, which illustrates the following situation:

- 4 data stations at odd addresses and 4 voice stations at even addresses are on the ring.
- VTMR is active only at voice station 2.
- Initially, the free token is passed to data station 1.
- During a data cycle (station 1 transmitting to station 5), VTMR expires.
- In the first voice cycle, station 4 misses its chance and a packet at station 4 is generated when the token is being passed to the station 6.

Notice that a data cycle may continue without any interrupttion when all voice station buffers are empty with TEB=1. Also, notice that there will be no second voice cycle if there are no voice packets which arrive later than token departure. The possible cycles of the proposed IVD-token ring protocol are shown in Figure 24.

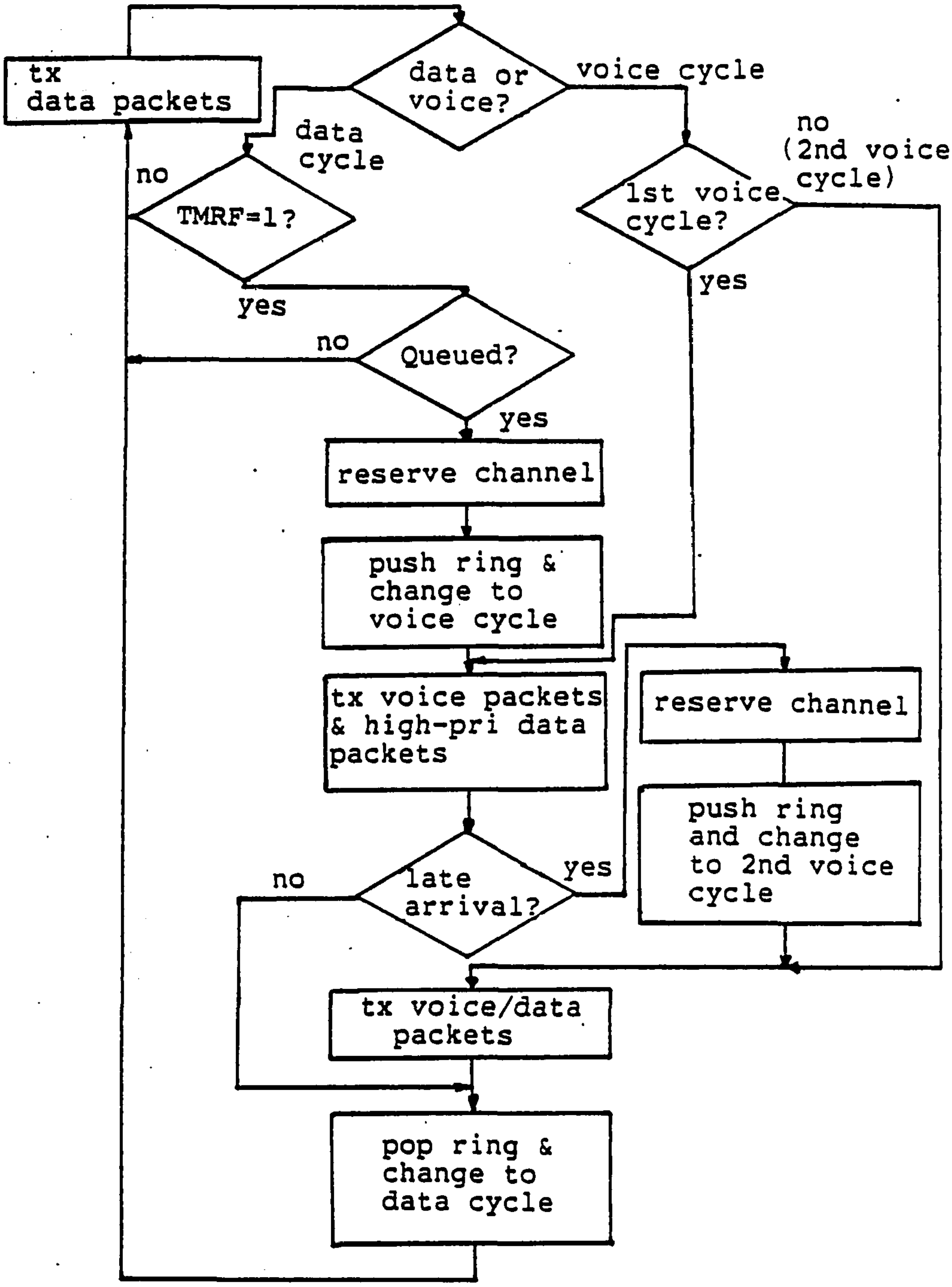


FIGURE 22. The operation of the IVD-token ring protocol

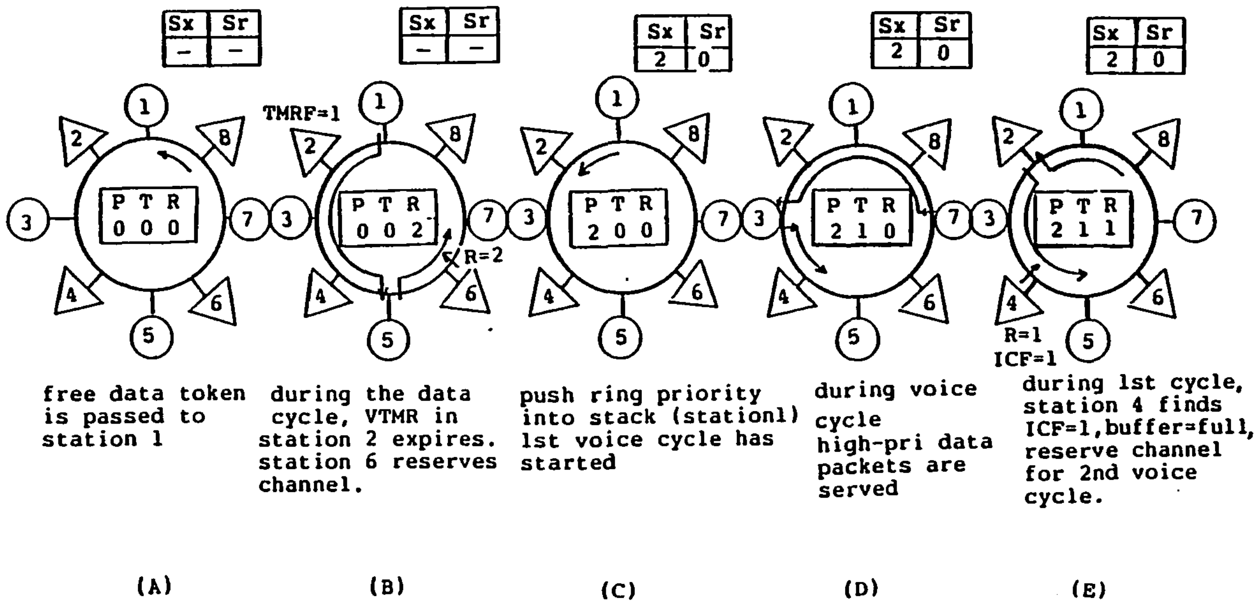
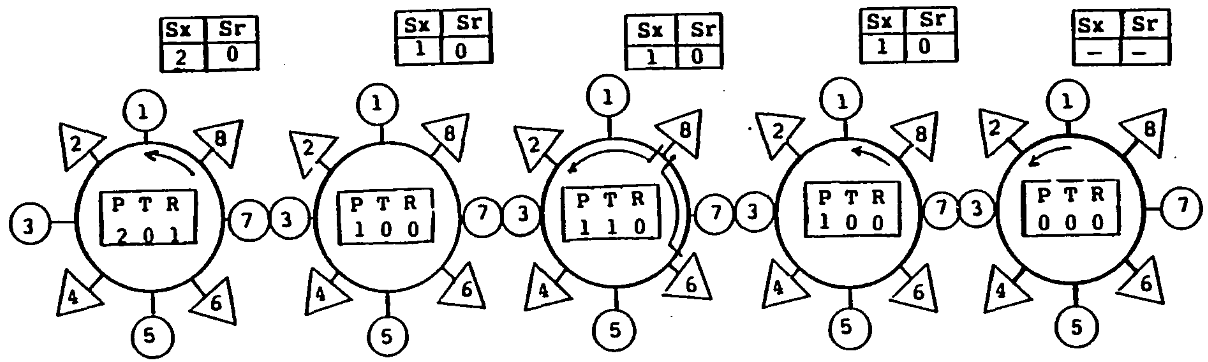


FIGURE 23a. Example of the IVD-token ring operation



(F)	(G)	(H)	(I)	(J)
at the end of 1st voice cycle token returns to stacking station with P=2,R=1.	pop Sx and push for 2nd voice cycle	2nd voice cycle	token returns to stacking station	stacking station pops ring and change to data cycle.

FIGURE 23b. Example of the IVD-token ring operation

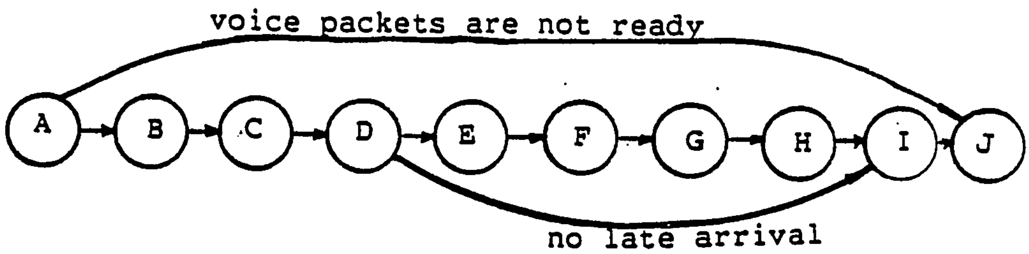


FIGURE 24. Cycles of the example

Specification

The proposed IVD-token ring protocol will be specified by state transition diagrams in which error states are not handled, only the states in normal operation being considered. Since the same packet format is used for voice and data stations, and IEEE 802.5 protocol is used for data stations, the state transition diagram for voice and data protocol are almost alike, the exception being the bit flipping state. The state transition diagram for the proposed IVD-token ring protocol is shown in Figure 25. Table 6 shows the bit flipping state for voice and data protocol. In Table 6(a), notice that the reservation of the channel for data stations is disabled and the data stations send their prioritized packets in a passive way. That is, the transition 02A in Table 6(b) does not exit in Table 6(a).

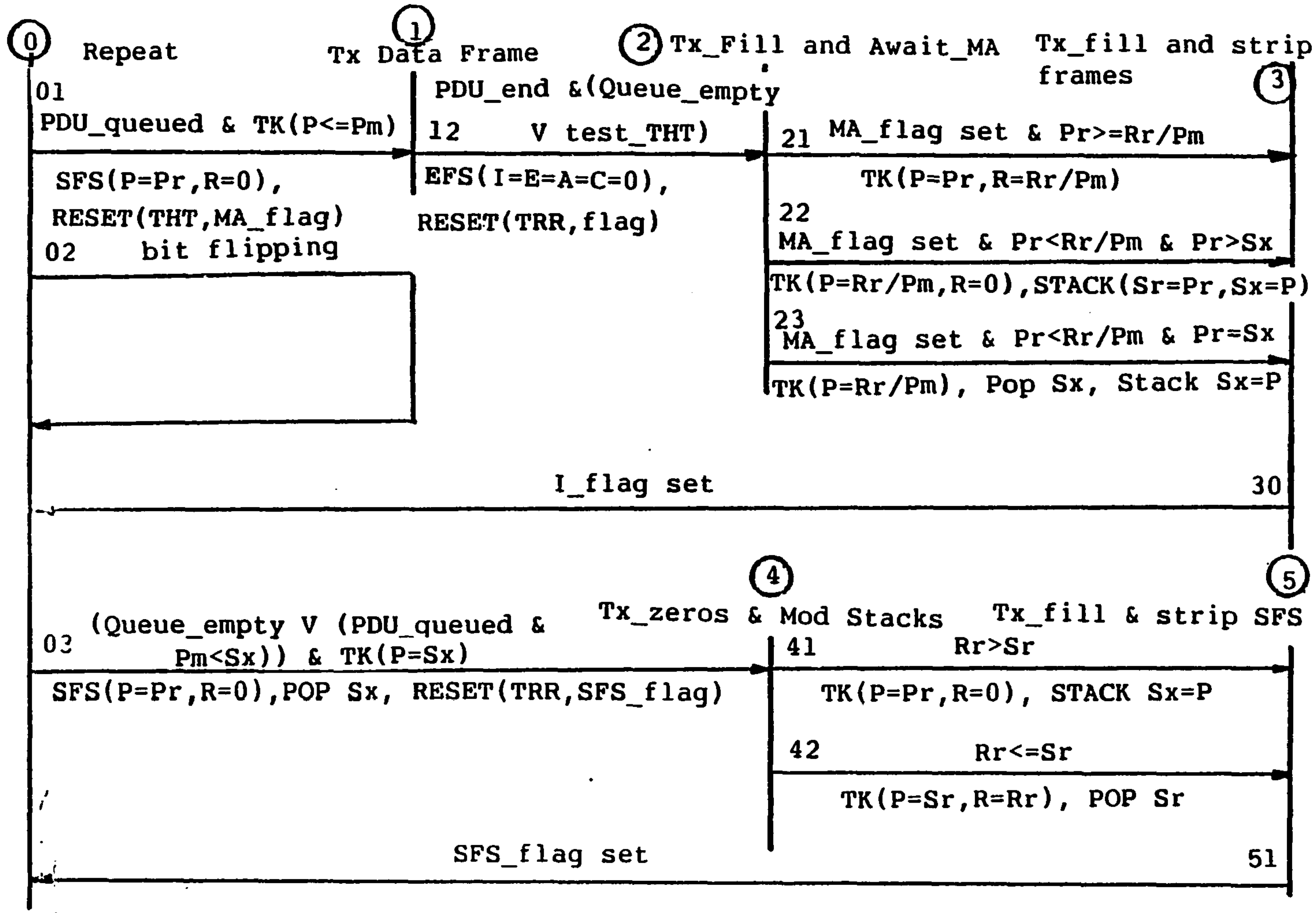


FIGURE 25. State transition of the IVD-token ring protocol

TABLE 6. Bit flipping state (a) data protocol (b) voice protocol

(a)

REF	INPUT	OUTPUT
02A	FR_WITH_ERROR	SET E=1
02B	DA=MA	SET A=1
02C	FR_COPIED	SET C=1

(b)

REF	INPUT	OUTPUT
02A	PDU_QUEUED & T=1&(FR(R<Pm) V (P>Pm>R, P=/Sx))	SET R=Pm
02B	FR_WITH_ERROR	SET E=1
02C	DA=MA	SET A=1
02D	FR_COPIED	SET C=1
02E	VTMR expired	SET TEB=1

The state transitions are explained as follows:

- (01):Usable token is received. The station starts to transmit a packet. Transition is made to state 1.
- (02): The frame or token is not destined to this station. This station only acts as a passive station by flipping bits according to the input conditions.
- (12): Transmission is completed or the token hold timer expires. An end-of-frame (EFS) shall be transmitted. A transition shall be made to state 2.

- (21): The frame sent by this station has returned (MA=1) with Pr >= Rr/Pm. A token shall be transmitted with P equal to Pr, and R equal to the greater of Rr or Pm, and transition shall be made to state 3.
- (22): The frame sent by this station has returned (MA=1) with Pr < Rr/Pm & Pr > Sx. A token shall be transmitted with P equal to the greater of Rr or Pm, and R equal to 0. Pr shall be stacked as Sr, P shall be stacked as Sx, and a transition shall be made to state 3.
- (23): The frame sent by this station has returned (MA=1) with Pr < Rr/Pm & Pr=Sx. A token shall be transmitted with the P equal to the greater of Rr or Pm, R equal to 0. Sx shall be popped as Sx and a transition shall be made to state 3.
- (31): Whether the frame sent has been received completely or not shall be tested. When the frame is removed by this station completely, a transition is made to state 0.
- (03): The ring priority is pushed by this station (P=Sx) and the queue is either empty or a packet is queued with Pm < Sx. The token shall be changed to a start-of-frame sequence by changing the T bit from 0 to 1 and popping Sx from the stack. A

transition shall be made to state 4.

- (41): The received packet has Rr greater than Sr. A token with its priority set to Rr and its R bit to 0 shall be transmitted and P shall be stacked as Sx, and a transition made to state 5.
- (42): The received packet has Rr less than Sr. A token with P=Sr, R=Rr shall be transmitted, Sr popped from the stack, and a transition made to state 5.
- (51): Upon receipt of SFS, a transition shall be made to state 0.

CHAPTER 6. PROTOCOL MODELING

For evaluation and analysis of the performance of the IVD-token ring protocol, it will be modeled and simulated using SLAM [28]. In the first section, the general aspects of SLAM are described along with its applicability for modeling the IVD-token ring protocol. In the second section, model building and event subroutines are explained along with the assumptions made and, finally, a method for the verification of the operation of the protocol model is described.

SLAM

SLAM, a Simulation Language for Alternative Modeling, was developed by Pritsker. The main advantage of SLAM over other simulation languages such as GASP IV and Q-Gert is that it allows users to develop models from process-interaction, next-event, or activity-scanning perspective by combining network, discrete event, and continuous modeling capabilities. Among these capabilities, the discrete event modeling is used in the simulation described here. In descrete event modeling of SLAM, a set of standard subprograms is provided by SLAM for use by the modeler to perform common discrete event functions such as event scheduling, file manipulation, statistics collection, and random sample generation. The executive control program of

SLAM controls the simulation by advancing time and initiating calls to the appropriate event subroutines at the proper points in simulated time. Hence, the modeler is completely relieved of the task of chronological sequencing of events.

The Discrete Event Framework of SLAM

A discrete event simulation program consists of three components: the main program, input statements, and a set of event subroutines.

- Main program: A user-written main program calls the subroutine SLAM which provides the executive control for discrete event simulation.
- Input statements: To run a discrete event simulation program input statements are required to specify such as quantities as run length, initial conditions, output options, and file ranking.
- A set of event subroutines: Event subroutines consist of event routines that correspond to user models. Each event subroutine is programed in FORTRAN and combined with SLAM subprograms for performing the commonly encountered simulation fuctions.

Figure 26 shows the SLAM next-event logic for simulating discrete event models.

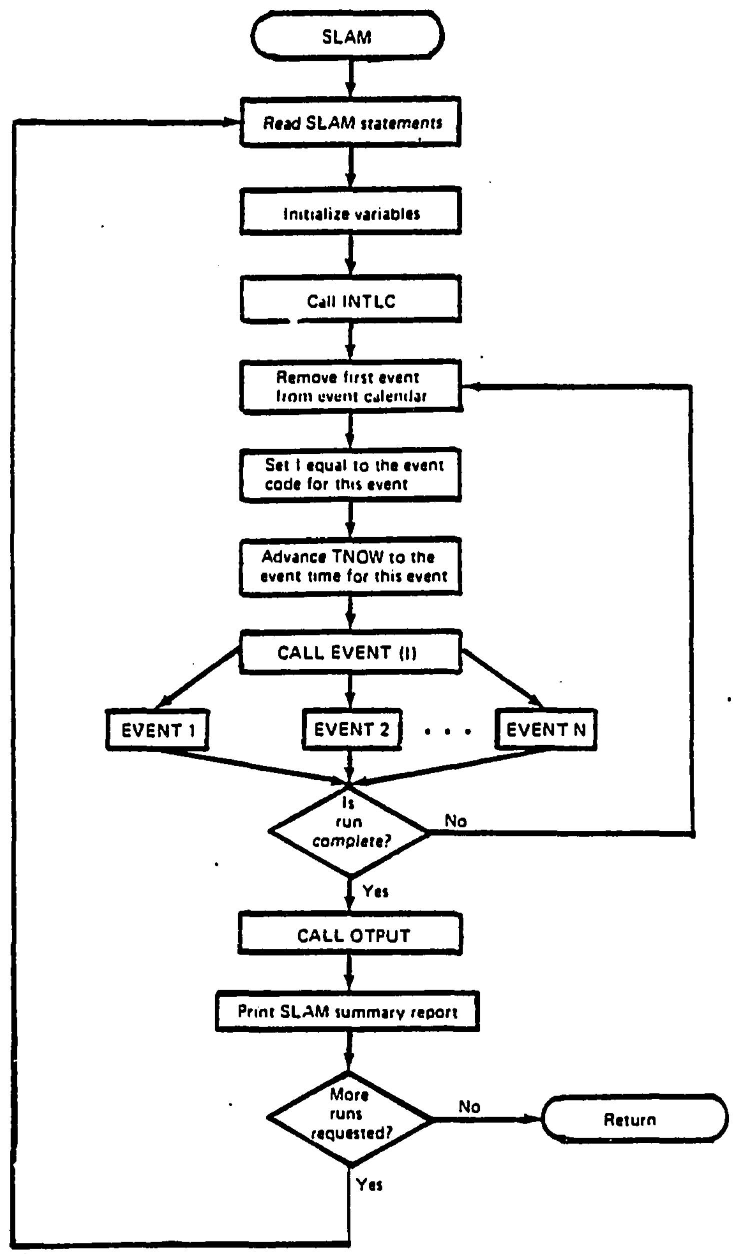


FIGURE 26. The SLAM next-event logic [28]

Modeling in SLAM

The structure of the discrete event program for the IVD-token ring protocol is described in this section along with the description of SLAM terms and variables used in modeling the protocol. The SLAM subprograms used in modeling are explained.

Program structure

Figure 27 illustrates the organization of the SLAM discrete event program. In this figure, IVD.F is the collection of event subroutines modeled for the IVD-token ring protocol. The events in IVD.F are controlled and executed in sequences by the SLAM processor, which is called in MAIN.F. IS.DAT consists of input statements for statistics and start/end simulation times. The traffic information for data stations and the time between arrivals is read from the TRATE file. After MAIN and IVD are compiled and linked with the SLAM library file (LSLAM1) and IS.DAT, the executable file RSLAM is produced. RSLAM generates two output files: SOUT and LOUT. SOUT,the SLAM summary report, displays the statistics collected by SLAM. LOUT has the information on the performance of voice transmission (number of lost packets for each station).

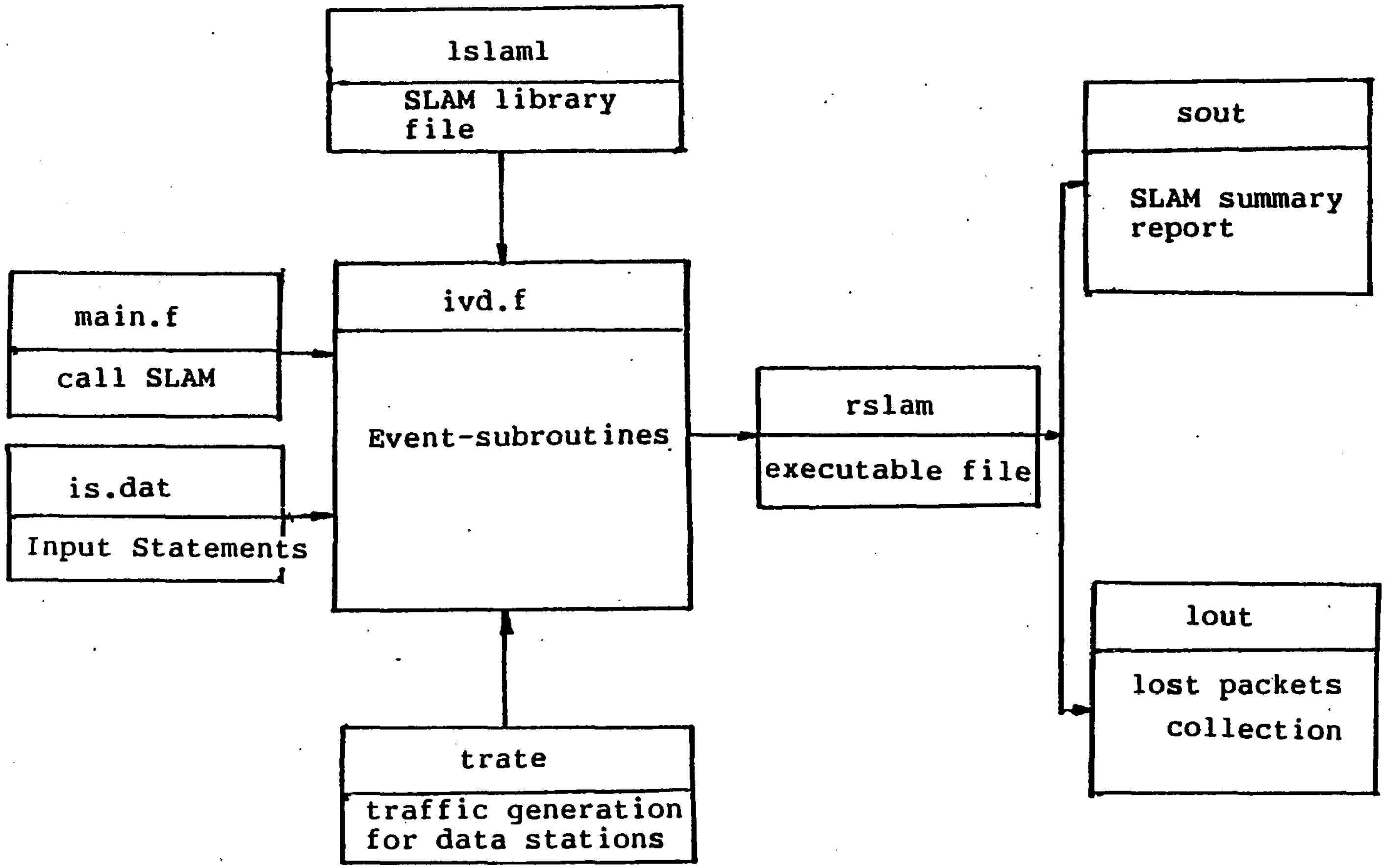


FIGURE 27. Discrete event program structure for the proposed protocol

Terms and variables of SLAM

- Entity: The entity is a unit of traffic that flows through the system. An entity corresponds to a packet.
- Attributes (ATRIB(I)): Attributes are numerical values carried along with an entity to specify its characteristics. When a packet is generated in a station, each field of the packet is represented by a corresponding element of the attribute array. The information on the statistics are also carried in this attribute array.
- File: A file provides the mechanism for storing the atttrbutes of an entity in a prescribed ranking with respect to other entities in the file. A file number is also used to denote the station address and the queue number.
- XX(I): This is a global variable I.

Each field of a packet on the medium is represented by each element of the XX array. Some of the information on the statistics are carried in the XX array. Table 7 shows the assignments of XX variables.

Subprograms of SLAM

To assist the analyst in writing event subroutines, SLAM provides a set of FORTRAN subprograms for programming all commonly encountered functions such as event scheduling,

TABLE 7. Assignment of XX variable

XX array	Assignment	XX array	Assignment
XX(1)	token	XX(2)	message length
XX(3)	priority	XX(4)	source address
XX(5)	destination addr	XX(6)	reserve
XX(7)	TEB	XX(8)	active voice
XX(9)	MA	XX(11)	cycle time
XX(12)	Packet arvl time	XX(13)	Packet service time
XX(14)	packet delay	XX(15)	token arvl time
XX(16)	access delay	XX(17)	TRT
XX(20)	voice cycle strt	XX(21)	voice cycle time
XX(22)	v. cycle strt MA	XX(23)	VTMR expired time
XX(30)	high priority packet sent		

statistics collection, and file manipulaions.

File manipulation   Figure 28 shows how a packet is enqueued, copied to the medium, and removed from a queue.

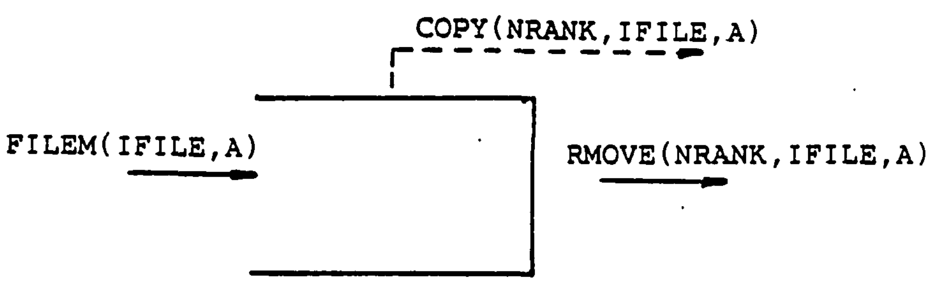


FIGURE 28. File manipulations in SLAM

When a packet is generated in a station, the packet, whose attributes are contained in array A, is enqueued to the queue IFILE, in a first-in/first-out manner using FILEM(IFILE,A). When a free token arrives to a station

IFILE, the first-in packet is copied to the medium along with its attributes by the subprogram COPY(1,IFILE,A). When a packet returns back to the source station along with its attributes in array A, the packet served will be removed from the queue by RMOVE(1,IFILE,A).

Scheduling events The SLAM processor completely relieves the user of the responsibility for chronologically ordering the events on an event calendar. The user simply schedules events to occur by calling subroutines SCHDL(KEVNT,DTIME,A), and SLAM causes each event to be processed at the appropriate time in simulation. KEVNT denotes the event code of the event being schduled and DTIME denotes the number of time units after the current time TNOW. Attributes associated with an event are specified by passing the buffer array A as an element of SCHDL.

Statistics collection SLAM provides a SLAM summary report which displays statistical results for the simulation. This report automatically prints the file statistics at the end of a simulation. Because the file number represents the queue number of a station, the average queueing delay for each station is automatically obtained. To obtain access delay of a packet, the STAT input statement and the COLCT subprogram are used. In Figure 29, an example for collecting statistics on the access delay of a station is shown.

```
Input Statement:                      .
                                      .
                       STAT,1, Access Delay 1
Event Subroutine;                     .
                                      .
                       ACDLY = XX(15) - XX(12)
                       CALL COLCT(ACDLY,1)
                                      .
                                      .
                                      .
SLAM Summary Report:
```

	Mean value	Standard Deviation	Coeff. of Variation	Min	Max	Num.
Access Delay 1	xx	xxx	xxx	xx	xx	xx
		.				
		.				

FIGURE 29. An example of statistics collection

In this example, XX(15),XX(12) represent token arrival time and packet arrival time respectively.

Modeling the IVD Token Ring Protocol

In this section, assumptions made in modeling the protocol are first explained, then the states of the system are identified for modeling the protocol in discrete events. Each event subroutine will be explained in detail and followed by verification of the model.

Assumptions

Since the IVD-token ring protocol corresponds to the token ring operation in normal environment, the following assumptions are made in the operation of the protocol:

- The channel is reliable.
- No maintenance functions for addition/deletion of stations, and failure of stations are needed.

Along with these assumptions for the protocol operation, additional assumptions on the locations of voice/data stations are made to permit accurate performance evaluation of the protocol. These assumptions are:

- Voice and data stations are evenly distributed on the network as shown in Figure 30.

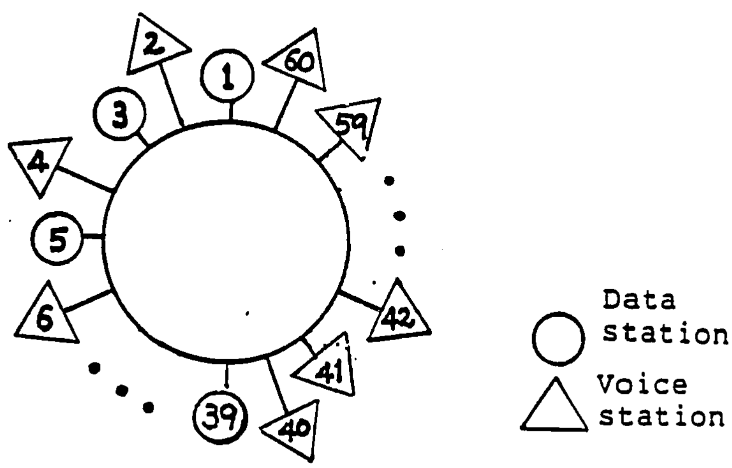


FIGURE 30. Location of voice/data stations

Among the 40 voice stations and 20 data stations selected in the design of this protocol, 20 voice stations and 20 data stations are placed alternately, and 20 additional voice stations are grouped together after the last data station (number 39).

- The total length of the cable is assumed to be 1.2Km, and the propagation delay is distributed equally among stations. The propagation delay between stations is 0.1 microsec.
- A 1-bit delay is assumed for the station delay.

In addition to these system assumptions, the following assumptions are made for voice/data station traffic. For data stations:

- Packets are generated independently in data stations.
- As shown in the measurement of data traffic [47], [48], in some applications, the arrival process is reasonably assumed to be a Poisson process with the packet size exponentially distributed with a mean value of 1000 bits/packet.
- Each data station has two queues: one high-priority queue with message size less than 4kbits, the other for low-priority packets. In the low-priority queue, a message is segmented if the message size is over the maximum data packet size (4kbits).

For voice stations:

- To reflect the real situation that a call connection can occur at any time, voice calls are not restricted to fit within a voice cycle, but are assumed to be uniformly distributed between 0 and the packetization time.

- A voice station can detect the presence of silence and no packets are generated during a silence period. Although not specified in the protocol, silence information is assumed to be implicitly contained in the first packet of a talkspurt to permit reconstruction of the voice signal.
- Following Brady's statistics [30], the talkspurt length is assumed to be exponentially distributed with mean of 1360 msec and the silence length is assumed to be exponentially distributed with mean of 1802 msec.
- As shown in Brady's experiment [30], the percentage of time for double-talking in a conversation is usually negligible (typically 5-6% of a total conversation time) and it is assumed that there is no double-talking during a conversation.

Figure 31 shows a sequence of talkspurts and silences for a caller A communicating with a callee B. As shown in this figure, the talkspurt length of a called station is assumed to be less than or equal to the silence length of a calling station. In this figure, although the callee B generates the second talkspurt extending beyond the silence length of the caller A, the talkspurt length of the callee will be assumed equal to the silence length of the caller, ignoring the double-talk interval.

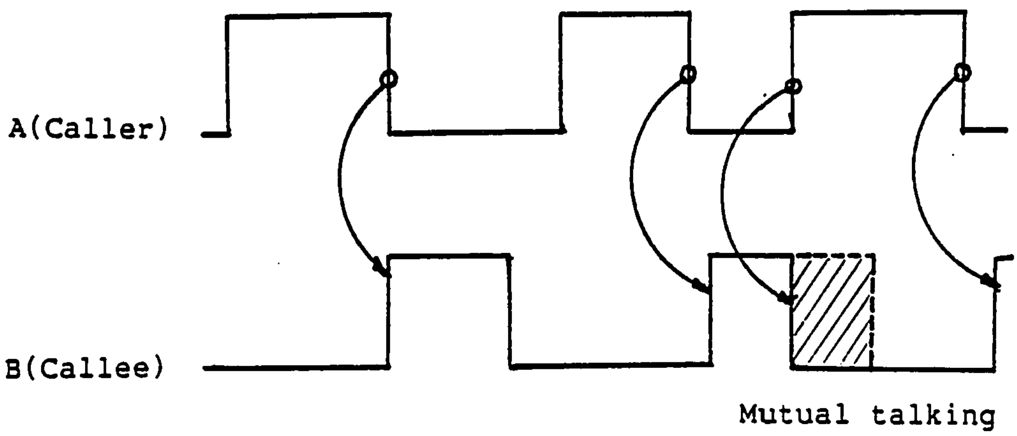


FIGURE 31. Talkspurts and silences in a conversation

Event modeling

If we consider a packet generated in a voice/data station and transmitted through stations and medium, each state of the system is determined by a combination of the following events:

- packet generation at ith data station.
- packet generation at jth voice station.
- packet arrival at the ith data station from the medium.
- packet arrival at the jth voice station from the medium.
- token arrival to the kth station.
- packet departure from the kth station.

- packetization timer value.

Here i=1,3,5...39, j=2,4,6...40,41,42,...60, and k=1,2,...60.

As seen in the states of the system of the IVD-token ring protocol, the location of the packet or token plays an important role in determining the states of the system. In addition to this, the location of stations affects voice performance significantly. To evaluate performance accurately, the location of stations must be properly reflected in the protocol modeling. In the following section, the effect of station location on performance of voice traffic will be explained.

Location of stations   Since some of the bits in a packet on the medium may be modified during transmission, and the future-states of the system may depend on these modified bits, the location of each station must be considered in modeling. The importance of station location in modeling may be explained by the example shown in Figure 32. Let's assume there are two voice stations and two data stations as shown in Figure 32(a) and station A transmits a packet to station D at time t0 and receives back its packet at t3 as shown in Figure 32(b). As shown in this figure, voice station C generates a voice packet at t1 (<t2) and the VTMR in station B expires at t2. Following the protocol, since the VTMR in B expires after t0, and station B finds

TEB=1 when its voice packet is ready, station C reserves the ring, and setting it up to be changed to support a voice cycle after t3. In this example, if the location of the voice stations are not considered in event modeling, the channel will not be reserved for voice after t3 because VTMR expires after the packet arrival at C.

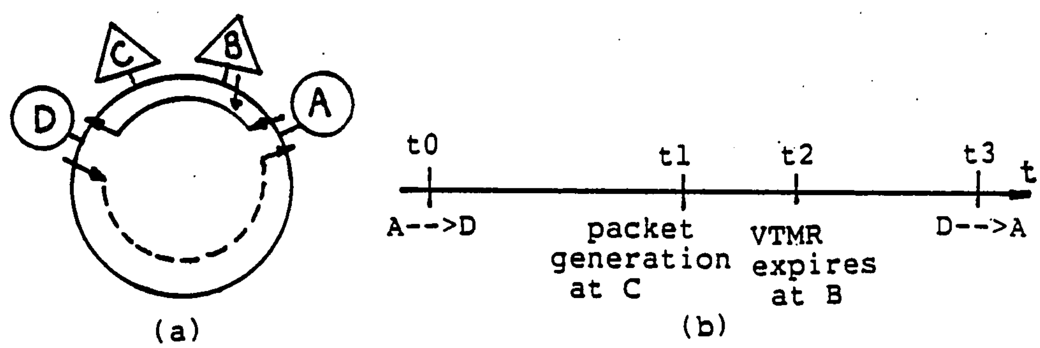


FIGURE 32. Effects of location of stations (a) location of 4 stations (b) timing diagram

Figure 33 compares the situation when the location information is properly included in the modeling to that when it is not.

A similar result can be obtained in considering the reservation of the second voice cycle. To properly reflect the location of stations in discrete event modeling of the proposed protocol, the following information is included in the modeling:

- The stations in an arrival event are classified as

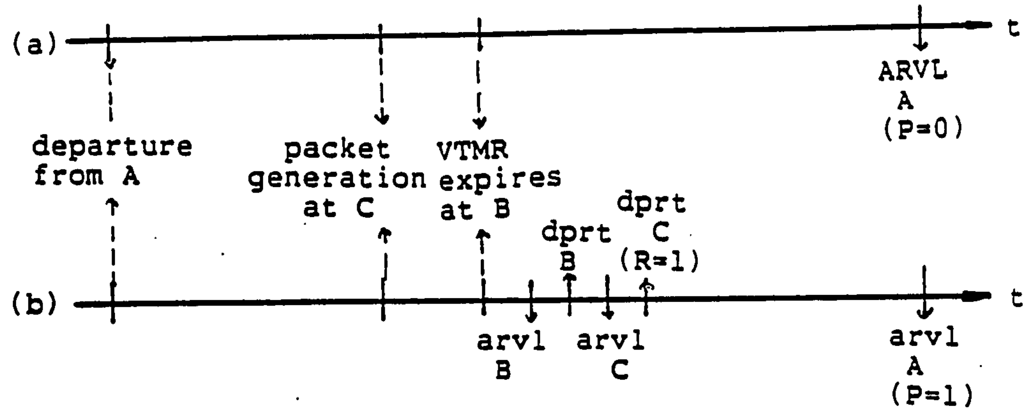


FIGURE 33. Results of the example in Figure 32 (a) when the location is not considered (b) when the location is considered

source stations, intermediate stations, and destination stations. The departure event only increases the location of a station by one.

- Because the timing information is very critical, propagation delay and station delay are attributed to each station.

Event subroutines To reflect the states of the system accurately and to include the location information on stations, the following event subroutines are developed.

Subroutine EVENT(I) In this subroutine, each event subroutine is assigned a positive integer numeric code called the event code. The event code is mapped onto a call to the appropriate event subroutine EVENT(I), where the argument I is the event code.

Event INTLC The principal functions of this routine are to initialize ring parameters, to generate input traffic for data stations, to prepare the output file for the collection of statistics, and to start the ring.

Event PGEN This subroutine is used for data stations for the following functions:

- to generate messages according to the arrival rate.
- to build packets.
- to collect statistics on station traffic.
- to enqueue data packets into the priority queue and the nonpriority queue.

When a packet is built, five elements of the ATRIB array are used for the packetization and collection of statistics information.

ATRIB(1) = packet generation time

ATRIB(2) = data length

ATRIB(3) = priority

ATRIB(4) = source address

ATRIB(5) = destination address

Events PVPG-PVS and SVPG-SVS These subroutines are used for the generation of voice traffic and packetization for voice stations. In a voice station, the talkspurts and silences of a conversation are generated by the PVPG and SVPG subroutines. The voice traffic of the caller is generated by PVPG and that of callee is generated by SVPG. The packetization process using the ATRIB array for voice packets in caller and callee voice stations are described in PVS and SVS respectively. The attributes of a voice packet are same as those of a data packet except that the priority is always assigned as 1 in a voice packet.

Event VTMR This routine sets the TMRF flag every packetization time to allow voice stations to reserve voice cycle if they are ready to transmit.

Event ARVL1 This subroutine is called when a packet arrives at a data station. Different functions are performed depending on the source address or destination address of the packet received. If the station is an intermediate station (XX(4) =/MA & XX(5) =/ MA), then the DPRT event is called after a station delay. Subroutine SRCE1 is called if this station is the source station, while DSTN1 is called if it is a destination station.

- Subroutine SRCE1: If a packet is originated by this station and the packet returns back to this station, this subroutine will be called. First of all, the R bit is checked to see whether the voice cycle is reserved or not when the packet comes back through the network. If the ring is reserved then the current ring priority is pushed down in the stack (SX(MA)) and priority is changed to 1. Meanwhile, the returned packet is removed from the queue via RMOVE subprogram. Finally, DPRT event will be scheduled after a station delay.

- Subroutine DSTN1: When a free token or message is headed for this station, DSTN1 is called. If a message is headed for this station, then DPRT event is scheduled after the station delay. In the case of a free token, the cycle time is measured and the network access delay is measured if the queue is not empty. If a voice cycle (P=1, T=0) is in progress, the stack will be tested. If the station is the stacking station, the stack is popped and a free data token is generated.

<u>Event ARVL2</u> This event occurs when a packet arrives at the voice station. If the station is the source station, the packet is removed from the queue and DPRT event is scheduled after a station delay. If the station is the destination or an intermediate station, then DSTN2 or INTR2 subroutine will be called, respectively.

- Subroutine INTR2: When XX(4)=/MA & XX(5)=/MA, this routine is called. If the current cycle is a voice cycle, then the DPRT is scheduled after a station delay. In the case of a data cycle, VTMR is tested if the timer is active in this station. When VTMR is expired and the output buffer is not empty, the channel is reserved (R=1).

- Subroutine DSTN2: When XX(5)=MA, this event is called. When a message is headed for this station, the message is copied and DPRT is scheduled after the station delay. When a free token is received, a test is made on the priority of the token to identify the cycle. In the case of a data cycle, VTMR is tested and cycle time is measured. For a voice cycle, the stack is tested, and the access delay is measured and lost packets collected if the queue is not empty.

Event DPRT When a packet leaves a station, the packet is always passed to the neighbor station after a propagation delay and a station delay. The location of a station (MA) is incremented by one. Either event ARVL1 or ARVL2 is called depending on the address of the neighbor station (even addresses for voice stations and odd addresses for data stations).

Event OTPUT This event is scheduled when the simulation is finished. In this event, all the necessary statistics for data and voice stations are collected and calculated. Information for each stations includes the average packet size (Avg(PS(I))), throughput (S(I)), average arrival rate (ARVL(I)) in message/sec, average transmission time (Avg(TXT(I))) in sec, and normalized traffic load (Norm(TRFF(I))). These values for data stations are given in (6.1), (6.2), (6.3), (6.4), and (6.5).

$$Avg(PS(I)) = TRFFo(I)/NPG(I) \tag{6.1}$$

$$S(I) = TRFFs(I)/TRFFo(I) \tag{6.2}$$

$$ARVL(I) = NPG(I)/TBAL(I) \tag{6.3}$$

$$Avg(TXT(I)) = TTXT(I)/NPG(I) \tag{6.4}$$

$$Norm(TRFF(I)) = ARVL(I)*Avg(TXT(I)) \tag{6.5}$$

where TRFFo is the number of bits offered to the channel, TRFFs is the number of bits served in the network, and NPG(I) is the number of packets generated at station I. The throughput (S) and the normalized channel traffic load for data stations can be obtained by (6.6) and (6.7).

$$S = \sum TRFFs(I)/TRFFo(I) \tag{6.6}$$

$$Rho = \sum Norm(TRFF(I)) \tag{6.7}$$

For voice stations, in addition to the traffic information, the following statistics are collected for the evaluation of the performance of the voice protocol.

- Number of voice packets served in the first cycle (NSRV1) and in the second voice cycle (NSRV2).
- Number of packets lost (NLOST(I)).

<u>Verification of the model operation</u>

The operation of the protocol model is verified using the MONTR input statement with the TRACE option. The format of this input statement is:

MONTR, option, TFRST, TBTWN, Variables

where TFRST,TBTWN are the time to start the trace and to stop the trace, respectively. To trace whether the cycles are reserved and changed at the right time the MONTR statement is used with the following variables.

- XX(9): The current location of the station a packet/token arrives.
- XX(4) and XX(5): The source and destination address.
- XX(1): Priority of a packet/token.
- XX(3): Token

CHAPTER 7. PERFORMANCE ANALYSIS

In the previous chapters, an IVD-token ring protocol has been developed and modeled. In this chapter the performance of voice and data under the proposed protocol will be analyzed and evaluated. Parameters to be used in the performance evaluation of both voice and data will be defined and performance will be analyzed in terms of these parameters. Assumptions made in the design of the proposed protocol will be verified and two modified protocol will be used to justify the design approaches of the proposed protocol.

Evaluation Parameters

Since the characteristics of voice and data are quite different, different evaluation parameters must be defined. Data performance is evaluated by finding how many bits can be accommodated by a channel (throughput) and how quickly the packets can be transmitted (queueing delay) to the network. The channel throughput is the sum of the served traffic in a channel divided by the sum of the traffic generated by all data stations. The queueing delay (Dq (i)) of a data packet j for a data station i is defined in (7.1).

$$Dq_j(i) = TARVLT_j(i) - PGENT_j(i) \tag{7.1}$$

Here, $TARVLT_j(i)$ is the token arrival time for packet j at station i and $PGENT_j(i)$ is the packet generation time of station i for the packet j. The average queueing delay(Dq) of a channel is:

$$Dq = \Sigma\, Dq(i)/Nd \qquad (7.2)$$

One of the identified characteristics of voice is that packet loss of up to 1-2% of total transmitted packets may be allowed without significantly degrading the performance of voice when reproduced at a receiver. Since the quality of voice becomes poor when packet loss exceeds this value, the proposed protocol should meet a corresponding packet loss limit requirement to support high voice quality. Notice that the performance of voice is determined by the sender in the proposed protocol. This is because the initial delay (Ds) at the receiver is set to a value which can accomodate the worst network access delay. The protocol performance for voice is determined by the voice station which shows the worst performance regardless of good performance in other voice stations. If the percentage of packet loss of station i is PLOST(i), the performance of the proposed protocol is determined by the maximum value of PLOST(i), where i=2,4,6,...38,40,41,42,..60.

Load Variation

Since protocol performance heavily depends on the load offered to a channel, load variation of combined voice and data will be examined before analyzing the proposed protocol. Since channel activity is divided into voice cycles and data cycles, the performance of each traffic type is affected by the fraction of the channel occupied by each.

It is evident that the performance of data transfer will be improved if the load of voice traffic decreases because when less voice traffic is generated more channel capacity is allocated to data traffic. Since it is evident that the performance of data will be enhanced when voice traffic decreases, voice traffic is fixed to a high value to observe the worst-case data performance. In the simulated system, 20 among 27 voice stations are activated in the initialization of network operation (voice load = 0.74). The principal reason to set the voice load to 0.74 rather than 1.0 are the limit on file numbers in SLAM and the long simulation-run time.

One of the important factors which affects voice performance is the relative timing between arrival of a token and a packet. Since the token arrival time at a station varies with the data load in each data cycle, the performance of voice may be affected by variation of this load. The data traffic load is therefore varied from light load to heavy load in the performance evaluation. The normalized data load (Norm(Rho)) given below is used.

$$\text{Norm(Rho)} = (\text{Nd/TBAL} * 10^{**}6)/\text{Rc} \qquad (7.3)$$

where TBAL is the time between message arrival in microsec/message and Rc is the channel bit rate in Mbps. In Table 8, the arrival rate of a message for a station is shown for various normalized channel loads.

TABLE 8. Normalized channel load and arrival rate of a station

Arrival Rate(1/TBAL) for a data station (message/sec)	Normalized Channel Load(Norm(Rho))
50	0.1
100	0.2
150	0.3
200	0.4
250	0.5
300	0.6
350	0.7
400	0.8
450	0.9

Analysis

The simulation model for the proposed IVD token ring protocol was run on an AT&T 3B5 computer. The results of this simulation are used in the following analysis.

Verification of assumption on p

In allocating a channel for a second voice cycle, p, the probability of losing a chance to transmit in a first cycle, was assumed very low with a worst-case value of 0.5. Because the channel fraction occupied by a second voice cycle is normally used for a data cycle, p must be shown to be small enough such as not to increase the size of the data queue drastically. For proof of the assumptions made, the total number of packets served (NPCKT) and number of packets served by second voice cycles (NSVC) were observed for various data traffic levels. p is defined as:

$$p = NPSC/NPKT \tag{7.4}$$

Table 9 shows p for various data loads.

TABLE 9. p for various data loads

Norm(Rho)	0.1	0.3	0.5	0.7	0.8	0.9
NPCKT	9963	9962	9960	9962	9961	9964
NPSC	870	976	904	725	620	701
p	0.087	0.098	0.091	0.073	0.062	0.070

As shown in this table, the maximum value of p is 0.098, consistent with earlier assumptions. Since less than 10% of the channel allocated to data cycles is used by second voice cycles, it is expected that the data transmission performance will not be degraded drastically. Notice that the load due to voice is actually 0.74 according to the Table 4 and 11 more voice stations can be added to the network. The effects of the second voice cycle on data transmission performance will be explained in the section 'data performance'.

Voice performance

As mentioned at the beginning of this chapter, the quality of voice is affected by the percentage of packet loss (PLOST) and PLOST must be less than 1-2% under various data traffic levels for proper reconstruction of a continuous voice signal.

For performance evaluation of the proposed protocol, the following information for various data traffic levels was collected for each voice station:

- NPCKT(i): number of packets generated by a voice station i.
- NPSC(i): number of packets served in a second voice cycle for voice station i.
- NLOST(i): number of packets lost by a station i.
- PLOST(i): percentage of packets lost in a station i.

The statistics collected for each station when data traffic changes from 0.1 to 0.9 are given in the Appendix. Notice that NPSC would be counted as NLOST and, as a result, PLOST would be much higher than the critical limit for the desirable quality of voice if second voice cycles are not allowed. The summary of the data in the Appendix is given in the first row of Table 10 (PLOST) for various data traffic levels, assuming that the performance of voice for a specific data traffic level is determined totally by the voice station which shows the worst performance.

TABLE 10. PLOST when data load varies from 0.1 to 0.9

Norm(Rho)	0.1	0.3	0.5	0.7	0.9
PLOST(%)	0.52	0.34	0.41	0.35	0.0
PLOST0(%)	0.42	0.34	0.41	0.34	0.28
PLOST1(%)	14.4	21.9	25.1	25.1	28.1

Notice that PLOST of the proposed protocol in all data traffic levels falls within 0.52%, which represents a high quality of voice. Notice also that PLOST does not depend on the level of data traffic. The principal reason for this is that voice cycles are scheduled regularly regardless of the level of data traffic.

In the proposed protocol, high-priority data packets are allowed to transmit during voice cycles for fast response. To observe the effects on performance of sending high-priority data packets in voice cycles, the proposed protocol was modified so as not to allow such high-priority packets during voice cycles. PLOST0 in Table 10 shows the percentage of lost packets in the modified protocol. Comparing PLOST and PLOST0 in all data traffic levels, we can find that allowing high-priority packets in voice cycles does not degrade the performance of voice traffic appreciably. Meanwhile, as will be seen in the next section, performance for high-priority packets is improved significantly.

Finally, to observe the definite necessity of the second voice cycle in achieving a high quality of voice, the IVD-token ring protocol was modified so that only the first voice cycle is available for voice packets. The observed percentage of packet loss (PLOST1) in Table 10 of the modified protocol shows PLOST1 in the range of 14-28.1%, resulting in a totally unacceptable voice quality.

Data performance

In the previous section, it has been shown that the proposed protocol provides a high performance for voice traffic. In this section, data transmission performance will be examined in terms of throughput and delay. Before examining this performance for the proposed protocol, the effects of allowing high-priority packets in voice cycles will be examined.

In Figures 34 and 35, the effects of allowing high-priority packets in voice cycles on the data performance are shown. As shown in Figure 34, the delay performance of high-priority data packets is enhanced significantly and is as low as 1.7msec at 0.88 data load when such data packets are allowed in voice cycles. Notice here that the delay of low-priority packets is slightly increased (about 9% when Rho=0.88) only in the heavy traffic region.

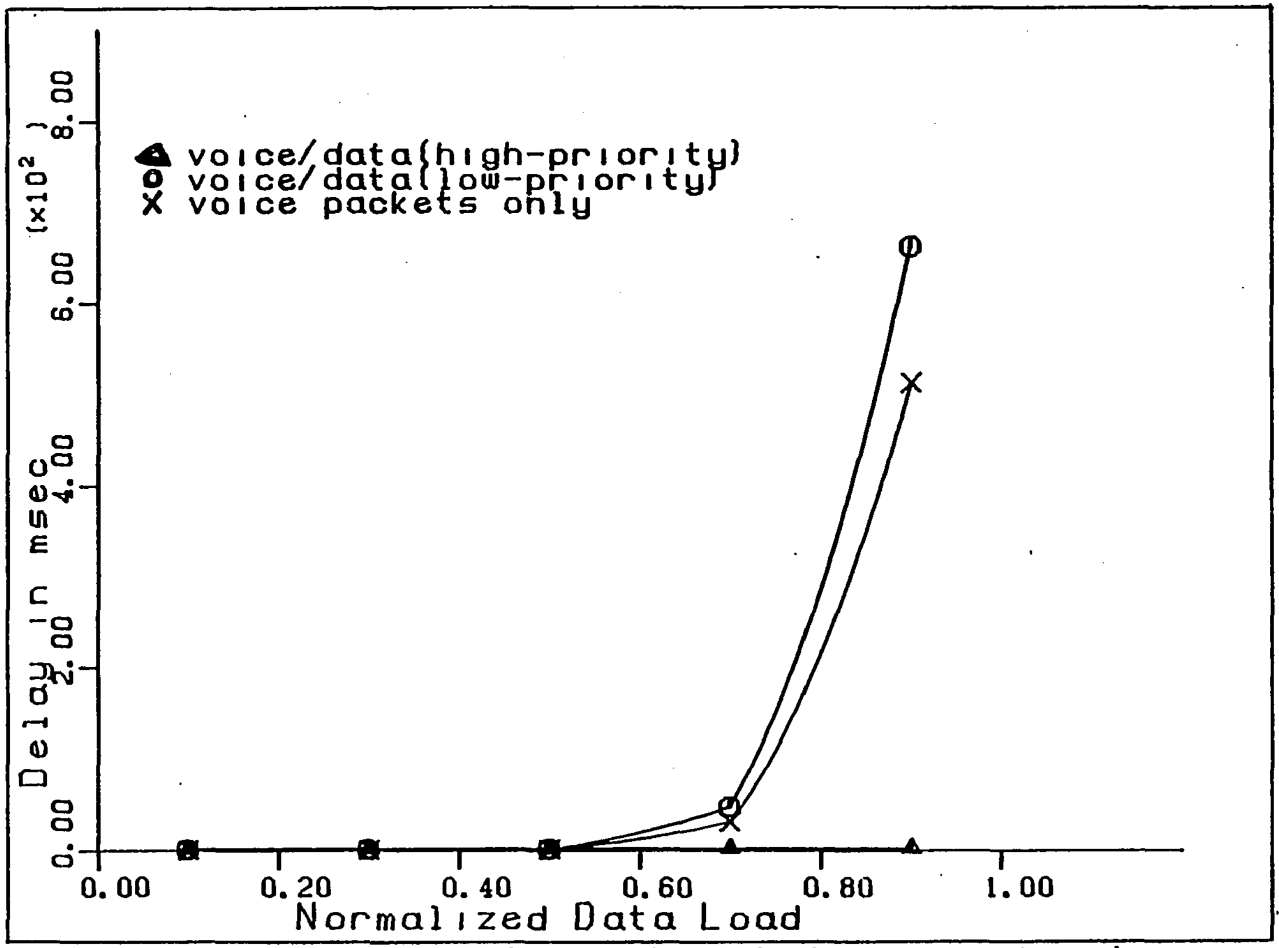


FIGURE 34. Delay with/without high-priority data packets in voice cycles

Figure 35 shows the throughput of two cases: when high-priority packets are allowed during voice cycles, and when voice packets only are allowed during voice cycles. Up to Rho=0.7, the throughput is same for these two cases. In the heavy traffic region, the throughput of the proposed protocol slightly decreases. One thing noted here is that the throughput decreases in spite of the sending of high-priority packets during voice cycles. This is because high-priority packets occupy the channel most of the time, and low-priority packets cannot access the network. Since the throughput is only slightly degraded in the heavy traffic region and the delay is much improved in all traffic levels, high-priority packets are allowed in voice cycles in the proposed protocol.

To examine the effects of voice on the performance of data, the throughput and delay of data packets are observed when the proposed protocol operates either data traffic only, or with both voice/data traffic. To operate the proposed protocol for data stations only, all voice stations are set idle not to generate any voice packets. For studying integrated services of the proposed protocol, all 40 voice stations are connected to the network for conversation and 20 data stations generate traffic according to the traffic level of the channel. The throughput of these two cases are compared in Figure 36. In the proposed

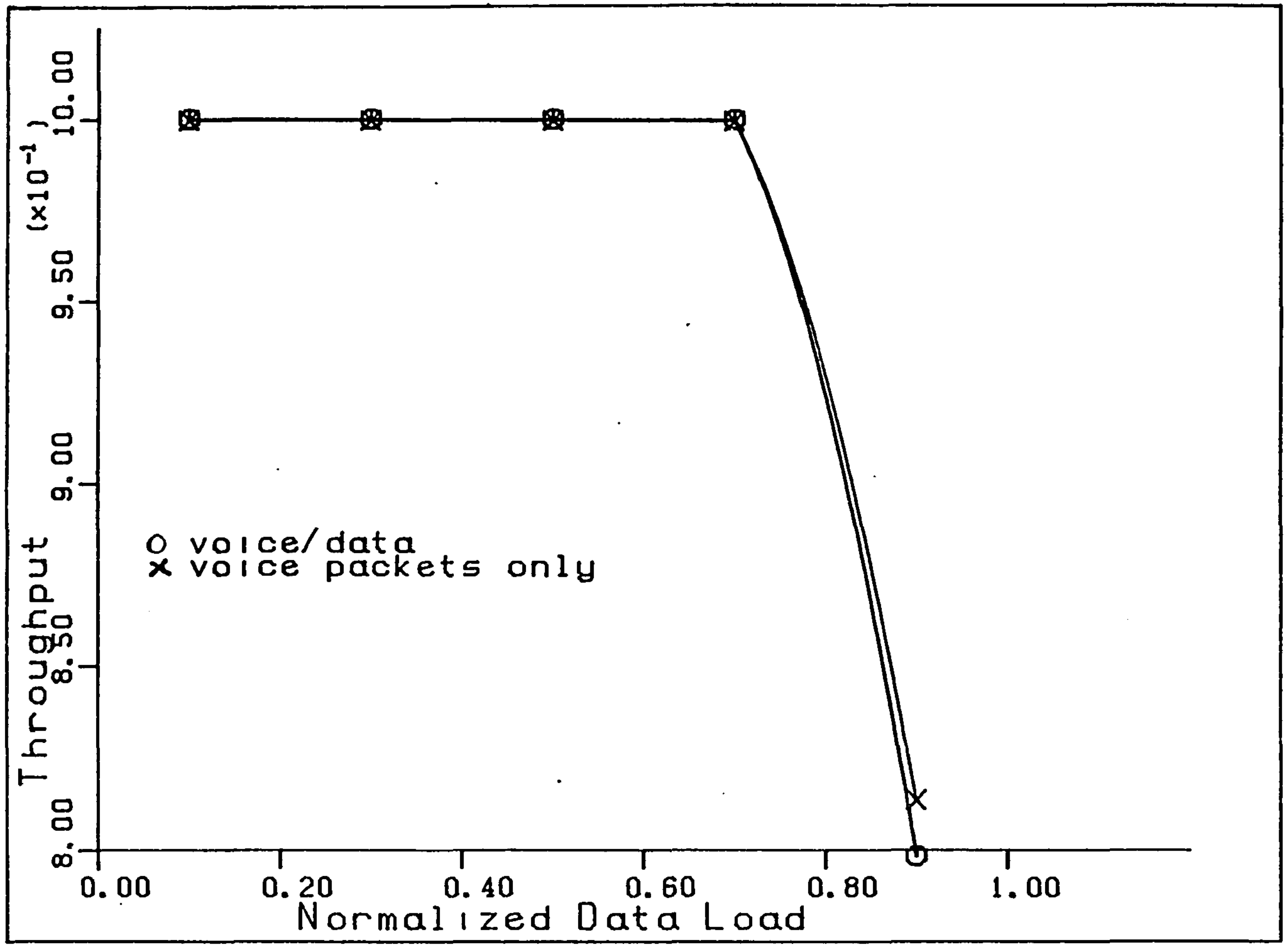


FIGURE 35. Throughput with/wihtout high-priority data packets in voice cycles

protocol, since voice traffic occupies a fraction of channel as designed, it would be expected that throughput of data would decrease. As shown in this figure, the throughput of the proposed protocol is not degraded significantly in the light and medium traffic regions and only slightly degraded in the heavy traffic region. Although the throughput of the protocol is slightly degraded, the performance of the proposed protocol for integrated service still remains high considering the fact that the throughput is 0.80 even in the heaviest traffic, i.e., 0.88.

The queueing delay of the data-only and integrated voice/data cases of the proposed protocol is shown in Figure 37. The queueing delay of low-priority packets increases significantly from the medium traffic region when it is compared with the queueing delay of the data-only case. The delay of the high-priority packets shows very low delay (1.7msec at Rho=0.88). Considering that messages of low-priority data type requires high throughput but not necessarily fast response, while messages of the high-priority data type require fast response (low queueing delay), the delay performance output of the proposed protocol shows a desirable performance.

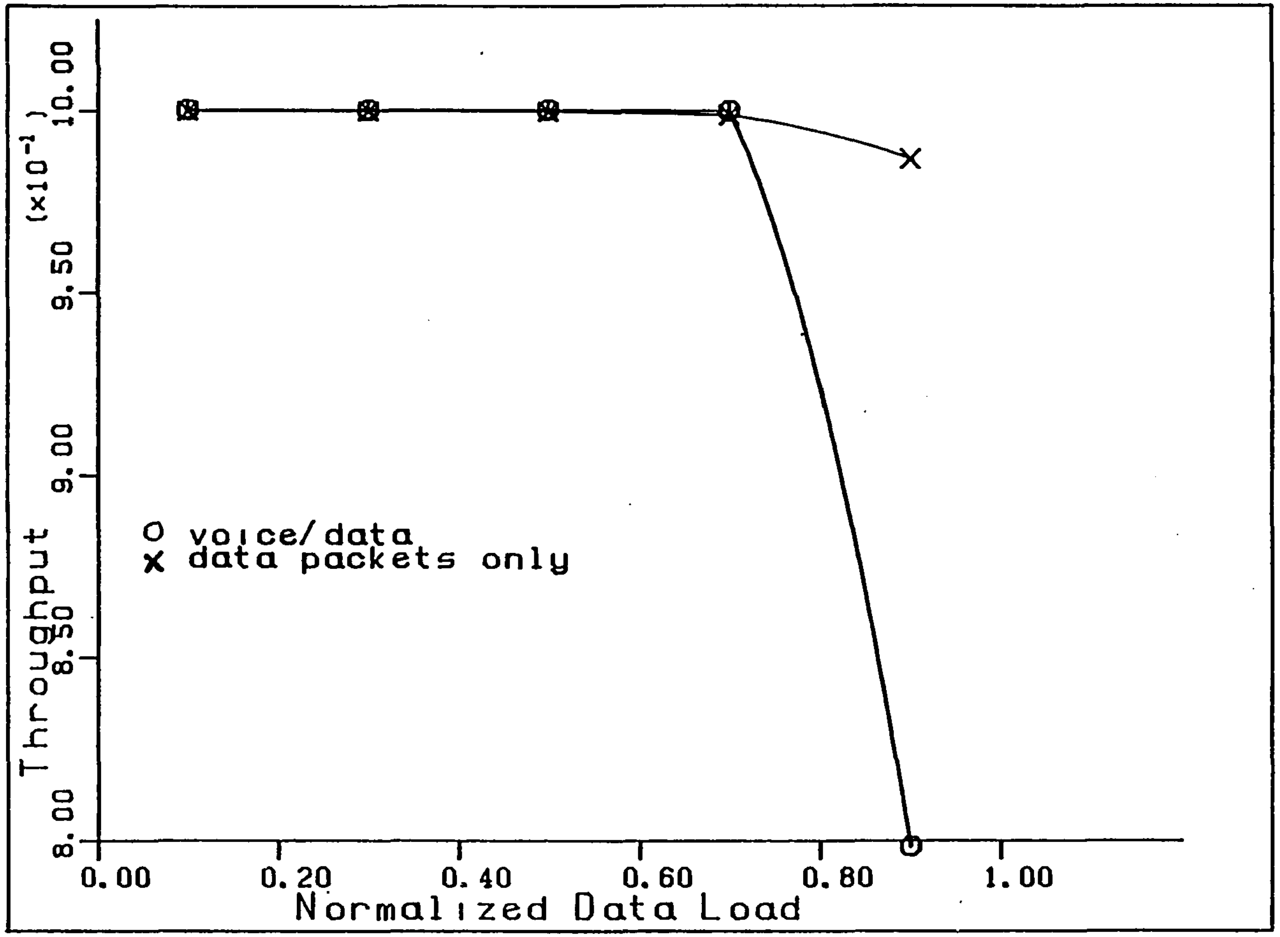


FIGURE 36. Throughput with/without voice of the proposed protocol

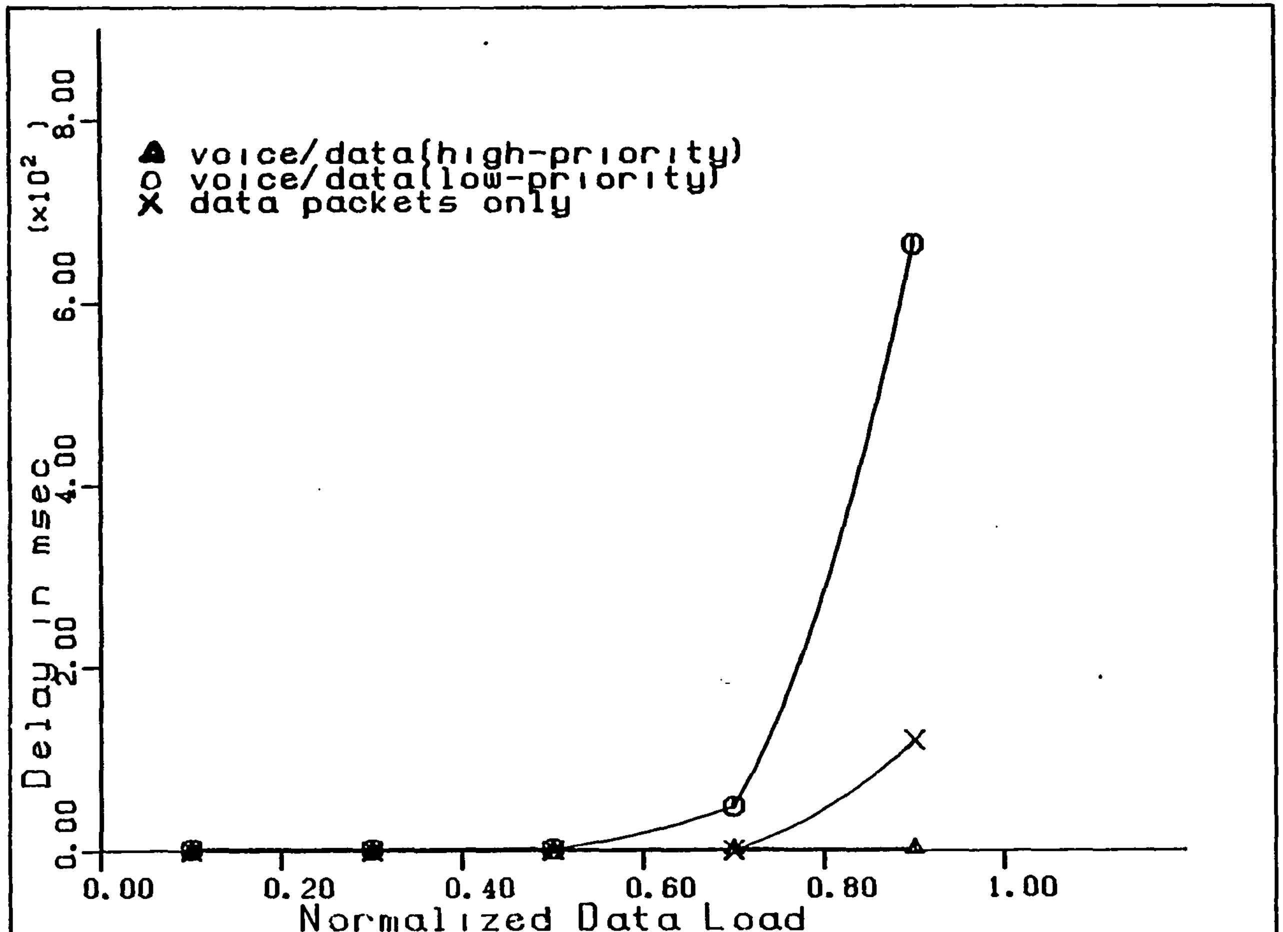


FIGURE 37. Delay with/without voice of the proposed protocol

CHAPTER 8. CONCLUSIONS AND FUTURE WORK

Conclusions

An IVD-token ring protocol which provides high performance for integrated voice and data transmission under various data traffic loads has been proposed.

To aid in the design of the proposed protocol, three LAN problems were clarified, viz, delay limit on a talkspurt, variable network access delay for voice packets, and queueing delay of data packets. Design issues were studied to achieve small loss of voice packets while preserving satisfactory performance of data packet transmission.

- To be compatible with IEEE 802.5, the packet format of IEEE 802.5 was selected as the packet format of the proposed protocol. In this packet format only one reserved bit was used as the TEB bit for voice stations. This bit was used by voice stations to reserve channel for voice cycles.
- To limit delay of a talkspurt, data packet size was limited to 4kbits by considering the packet size limit effects on performance. A condition for packetization period (Tp) was derived as in (4.15).
- It was found that allowing only one chance for voice stations to transmit a packet in each

packetization period resulted in unacceptable voice performance (PLOST=14-28%). To enhance voice performance, a second voice cycle was allowed after a first voice cycle for the packets just missing packet transmission. The effect of this voice cycle allocation method on data performance was considered. From design equation (4.6), 20% of channel capacity was allocated for the first voice cycle and about 10% of each data cycle was used by a potential second voice cycle for high quality of voice while preserving satisfactory data performance. With this allocation strategy, the maximum allowed number of stations were 27 conversational voice stations (total 54) and 19 data stations.

To implement the proposed IVD protocol in a token ring network, a timer (VTMR) and two flags (TMRF and ICF) were added to each voice stations, and priority (P), reservation (R) and TEB packet bits were used to support:

- design issues described in Chapter 4
- minimizing variation of voice cycle
- skipping voice cycles when no voice packets are resident in output buffers after VTMR expiration.

The proposed protocol was specified with state transition diagrams and an example was used for the explanation of the proposed protocol operation.

In modeling the proposed protocol, a discrete event simulation model was built using the SLAM simulation language for performance evaluation. Since random time differences in arrival time of tokens and packets affects voice performance significantly, all factors affecting the state of the system were carefully represented in the simulation model. These include:

- location of stations
- station delay (assumed 1 bit)
- propagation delay to be 0.1 microsec/station
- packet overhead (64 bits)
- token passing overhead (24 bits)

The performance of the proposed protocol was simulated with Tp=10msec, Nv=40, Nd=20, and Rc=10Mbps. Voice load was fixed at high value of 0.74 and data load was varied from 0.1 to 0.9 to observe the effects of voice on data and vice versa. The simulation results of the proposed protocol can be summarized as follows:

- High quality of voice could be obtained with the proposed protocol. The percentage of lost packets (PLOST) was only 0.52% in worst case, which is much smaller than the allowable limit (1-2%).
- The voice effects on data throughput was almost negligible. The delay performance of data packets

was satisfactory regarding transfer type does not necessarily fast response. (The interactive type data packets were delayed only by 1.1713msec while low-priority packets suffered 662msec delay when Rho=0.8837.)

- The effect on voice performance of including high-priority data packets during voice cycles was negligible.

In conclusion, both high quality of voice transmission and high performance data transmission (especially for interactive type of data) could be achieved with the proposed IVD-token ring protocol.

Future Work

Simulation technique development

Two principal constraints were encountered in simulating the proposed protocol. One was the limit on the number of files in the SLAM simulation language. Since each station was represented as a file in SLAM, and the maximum number of files is limited to 100, there was a fundamental problem in simulating a large number of stations. This limit inhibited the study of the effects of increasing channel bit rate, packetization period, and voice cycle time. Another constraint was the long run time in simulating the proposed protocol. For example, it took about 100 hours on an AT&T 3B5 computer to simulate about 5 seconds of real time for a given traffic level. This long run time results from including station location information in the discrete event simulation model, which was necessary for the evaluation of IVD protocol. Better simulation language modeling techniques for the analysis of communication networks are needed to overcome these limits.

Study on the effects of maintenance functions

For reliable and flexible operation of a LAN, maintenance functions should be reflected in the MAC layer protocol. The maintenance functions described in IEEE 802.5 are:

- initialization
- fault recovery
- insertion of stations
- deletion of stations

If there is a guarantee that these functions are not performed frequently with respect to frequency of conversations, these same maintenance functions can probably be applied to voice stations without modification. However, if the network environment requires frequent deletion and insertion of phones and stations, the performance of voice may be affected because the maintenance procedures require some fraction of the channel. If this is the case, development and analysis of separate maintenance functions for voice stations are needed.

Interconnection problems

When LANs supporting integrated voice/data are interconnected via gateways or bridges [49], voice stations in each LAN may need to communicate with voice/data stations in other LANs. When these stations are interconnected, the end-to-end delay limit problem may become more serious because of the time spent in gateways or bridges. To support high quality voice, existing gateway protocols or bridges may need modification. Another interconnection problem will occur when there is a need to connect voice stations in LANs to the existing telephone network. This need may be spurred by active research work on ISDN. For LAN-ISDN connection, a study on the interface between LAN stations and the ISDN is needed.

Extension of study for fiber-optic token rings

It is well known that the token ring is best suited for fiberoptic applications because it requires no tapping [50]. When a fiber-optic medium is used in a token ring network, integration of services has greater justification because of the high bandwidth. IVD protocol development and performance evaluation for fiber-optic token ring thus requires further study.

ACKNOWLEDGEMENT

I wish to express my sincere thanks to my major professor, Dr. Terry A. Smay and co-major professor, Dr. Douglas W. Jacobson for their guidance and encouragement during this research.

Thanks are also due to Dr. Arthur V. Pohm, Dr. Grover R. Brown, Dr. Charles T. Wright, and Dr. Arthur E. Oldehoeft for their suggestions and guidance as my graduate committee.

Special thanks are extended to Electrical and Computer Engineering Department and Rockwell International Co. for providing opportunities to do this research and financial support.

Finally, but not the least, I thank my wife, Jin Hee, for her selfless sacrifice, confidence, and encouragement, and my children, Yoon and Eun Joo for their smile, and our parents for encouragement and support throught this research.

BIBLIOGRAPHY

1. Schwartz, Mischa. _Telecommunications Networks_. Reading, MA: Addison-Wesley Publishing Company, 1986.

2. Ruiz, Antonio. "Voice and Telephony Applications for the Office Workstations." _IEEE 1985 International Conference on Computer Workstations_. San Jose, California: IEEE Computer Society Press, 1985, pp. 158-163.

3. Swineheart, D. C., Stewart, L. C., and Ornstein, S. M. "Adding Voice to an Office Computer Network." _IEEE Global Telecom. Conference_. San Diago, California: IEEE Communications Society Press, 1983, pp. 392-398.

4. Kenedi, R. and Wong, C. L. "Architectures for Implementation." _IEEE Commun. Magazine_, 24, No. 3 (March 1986), 18-31.

5. Decina, M. and Scace, E. L. "CCITT Recommendation on the ISDN: A Review." _IEEE Selected Areas in Commun._, SAC-34, No. 3 (May 1986), 320-325.

6. Stalling, W. _Data and Computer Communications_. New York, NY: Macmillan Publishing Company, 1985.

7. Stalling, W. _Local Networks: An Introduction_. New York, NY: Macmillan Publishing Company, 1984.

8. Pfister, G. M. and O'Brien, B. V. "Comparing the CBX to the Local Network - and the Winner Is?" _Data Communications_, 11 (July 1982) 103-110.

9. Richer, I., Steiner, M., and Sengoku, M. "Office Communications and the Digital PBX." _Computer Networks_, 5, No. 6 (Dec. 1981), 411-422.

10. IEEE. _IEEE Standard 802.3: CSMA/CD Access Method and Physical Layer Specification_. Silver Spring, MD: IEEE Computer Society Press, 1984.

11. IEEE. _IEEE Standard 802.4: Token Passing Bus Access Method and Physical Layer Specifications_. Silver Spring, MD: IEEE Computer Society Press, 1984.

12. IEEE. Standard 802.5: Token Ring Access Method and Physical Layer Specifications. Silver Spring, MD: IEEE Computer Society Press, 1984.

13. Stuck, B. "Which local net bus access is most sensitive to traffic congestion?" Data Communications, 12 (Jan. 1983), 107-120.

14. Bux, W. "Performance Issues in Local Area Network." IBM Systems Journal, 23, No. 4 (1984), 351-374.

15. Bux, W. "Local Area Subnetworks: A performance Comparison." IEEE Trans. on Communications, COM-29, No. 10 (Oct. 1981), 1465-1473.

16. Stallings, W. "Local Network Performance." IEEE Commun. Magazine, 22, No. 2 (Feb. 1984), 27-36.

17. Sachs, S. R., and Kan, K. "Performance Analysis of a Token Bus Protocol and Comparison with other LAN Protocols." IEEE 10th Conference on LCNs. Minneapolis, MN: IEEE Press, 1985, pp. 46-51.

18. Liu, M. T. and Hilal, W. "Performance Evaluation of Channel Access Protocols for Local Computer Networks." Proceedings of the COMPCON Fall 1982 Conference. IEEE Society Press, 1982, pp. 417-426.

19. Bux, W., Closs, F., Janson, P., Kummerle, K., Muller, H. R., and Rothauser, E. H. "A Local Communication Network Based on a Reliable Token-Ring System." Local Computer Network Symposium. Florence, Italy: IEEE Society Press, 1982, pp. 69-82.

20. Mark, J. W. "Priority Scheduling for Integrated Voice/Data Services in Local Area Networks." CCNG Report E-111. Waterloo, Canada: Computer Communications Networks Group, Univ. of Waterloo, 1983.

21. Saydam, T. and Sethi, A. S. "Performance Evaluation of Voice/Data Token Ring LANs wirh Random Priorities." IEEE INFOCOM 85. Washington, D.C.: IEEE Computer Society Press, 1985, pp. 326-332.

22. Ibe, O. C. and Gibson, D. T. "Synchronous Communications Protocols for Local Networks." IEEE 10th Conference on LCNs. Minneapolis, MN: IEEE Press, 1985, pp. 4-9.

23. Ibe, O. C. and Gibson, D. T. "Protocols for Integrated Voice and Data Local Area Networks." IEEE Commun. Magazine, 24, No. 7 (July 1986), 31-36.

24. Mark, J. W. and Limb, J. O. "Integrated Voice/Data Services on Fasnet." AT&T Bell Lab. Tech. Journal, 63, No. 2 (Feb. 1984), 307-336.

25. Hilal, W. and Liu, M. T. "Local Area Networks Supporting Speech Traffic." Computer Network Symposium. Silver Spring, Maryland: IEEE Society Press, 1983, pp. 107-110.

26. Wong, J. W. and Gopal, P. M. " Analysis of a Token Ring Protocol for Voice Transmission." Computer Network Symposium. Silver Spring, Maryland: IEEE Society Press, 1983, pp. 113-117.

27. Hafner, R., Nenadal, Z., and Tschanz, M. "A Digital Loop Communications System." IEEE Trans. on Communications, COM-22, No. 6 (June 1974), 877-880.

28. Pritsker, A. A. B. Introduction to Simulation and SLAM II. Second Edition. New York, NY: Halsted Press, 1984.

29. Brady, P. T. "A Technique for Investigating On-Off Patterns of Speech." The Bell Sys. Tech. Journal, 44, No. 1 (Jan. 1965), 1-22.

30. Brady, P. T. "A Statistical Analysis of On-Off Patterns in 16 Conversations." The Bell Sys. Tech. Journal, 47, No. 1 (Jan. 1968), 73-91.

31. Bullington, K. and Fraser, J. M. "Engineering Aspects of TASI." The Bell Sys. Tech. Journal, 38, No. 2 (March 1959), 353-364.

32. Huggins, A. W. F. "Effects of Lost Packets on Speech Intelligibility." NSC Note 78. Cambridge, MA: BB&N, Feb. 1976.

33. Gruber, J. G. "Delay Related Issues in Integrated Voice and Data Network." IEEE Trans. on Communications, COM-29, No. 6 (June 1981), 786-800.

34. Forgie, J. W. "Network Speech: System Implications of Packetized Voice." Annual Report ESD-TR-77-178. Cambridge, MA: Defense Communications Agency, MIT Lincoln Lab., 1976.

35. Wolf, D. "Speech Coding." IEEE Int'l Zurich Seminars on Digital Communications. Zurich, Switzerland: IEEE Society Press, 1984, pp. 1-5.

36. Geihs, K. "Modeling and Analysis of Silence Suppression Strategies for Packet Voice Transmission." In Performance of Computer Communication Systems. Amsterdam, Netherlands: North-Holland, 1984, pp. 139-142.

37. Montgomery, M. A. "Techniques for Packet Voice Synchronization." IEEE Journal on Selected Areas in Commun., SAC-1, No. 6 (Dec. 1983), 1022-1028.

38. Naylor, W. E. and Kleinrock, L. "Stream Traffic Communication in Packet-Switched Networks: Destination Buffering Considerations." IEEE Trans. on Communications, COM-30, No. 12 (Dec 1982), 2527-2534.

39. Gopal, P. M., Wong, J. W., and Majithia, J. C. "Analysis of Playout Strategies for Voice Transmission Using Packet Switching Systems." Performance Evaluation, 4, No. 1 (Feb. 1984), 11-18.

40. Garcia, L. A., Kwong, R. H., and William, G. F. "Performance Evaluation Methods for an Integrated Voice/Data Link." IEEE Trans. on Communications, COM-30, No. 8 (Aug. 1982), 1848-1857.

41. Little, J. D. C. "A Proof of Queueing Formula $L=\lambda W$, Operations Research, 9 (1961), 383-387.

42. Shoch, J. F. "Carrying Voice Traffic Through an Ethernet Local Network: A General Overview." In Local Networks for Computer Communications. Amsterdam, Netherlands: North-Holland, 1981, pp. 429-447.

43. Johnson, D. H. and O'leary, G. C. "A Local Access Network for Packetized Voice Communication." IEEE Trans. on Communications, COM-28, No. 5 (May 1981), 679-688.

44. Nutt, G. J. and Bayer, D. L. "Performance of CSMA/CD Networks under Combined Voice and Data Loads." IEEE Trans. on Communications, COM-30, No. 1 (Jan. 1982), 6-11.

45. Chlamtac, I. and Eisinger, M. "Performance of Integrated (Voice/Data) CSMA/CD Networks." Communications of ACM, 28, No. 1 (1985), 87-93.

46. Kleinrock, L. Queueing Systems, Vol. II: Computer Applications. New York, NY: John Wiley, 1975.

47. Marthe, M. and Hawe, B. "Predicted Capacity of Ethernet in a University Environment." In Introduction to Local Area Networks. Maynard, MA: Digital Equipment Corp., 1982.

48. Shoch, J. F. and Hupp, J. A. "Measured Performance of an Ethernet Local Network." Communications of the ACM, 23, No. 12 (Dec. 1980), 711-721.

49. Bux, W. and Grillio, D. "End-to-End Performance in Local Area Networks of Interconnected Token Rings." IEEE INFOCOM 84. San Fransisco, CA: IEEE Computer Society Press, 1984, pp. 60-68.

50. Ross, F. E. "FDDI - a Tutorial." IEEE Commun. Magazine, 24, No. 5 (May 1986), 10-17.

APPENDIX

For the evaluation of the voice performance of the proposed protocol, statistics were collected for each voice station over various data traffic levels. In this Appendix, the following information were collected for each voice station:

- NPCKT(i): number of packets generated by a voice station i.
- NPSC(i): number of packets served in a second voice cycle for a voice station i.
- NLOST(i): number of packets lost in a station i.
- PLOST(i): percentage of packets lost in a station i.

These statistics were collected when

Tv/Tp=0.2,

Tp=10msec,

Max(PSd)=4000 bits,

PSv=640 bits,

Nv=40,

Nd=20,

and Rc=10Mbps.

TABLE 11. Rho = 0.1

I	NPCKT(I)	NPSC(I)	NLOST(I)	PLOST(I)(%)
2	283	0	0	0.0
4	452	0	0	0.0
6	320	181	0	0.0
8	356	0	0	0.0
10	239	210	1	0.42
12	364	0	0	0.0
14	432	0	0	0.0
16	343	0	0	0.0
18	345	0	0	0.0
20	245	0	0	0.0
22	270	0	0	0.0
24	245	184	1	0.41
26	191	169	1	0.52
28	321	0	0	0.0
30	76	0	0	0.0
32	324	0	0	0.0
34	178	0	0	0.0
36	439	0	0	0.0
38	249	0	0	0.0
40	359	0	0	0.0
41	215	0	0	0.0
42	46	0	0	0.0
43	179	0	0	0.0
44	142	0	0	0.0
45	260	92	0	0.0
46	134	0	0	0.0
47	66	0	0	0.0
48	155	0	0	0.0
49	153	0	0	0.0
50	253	0	0	0.0
51	228	0	0	0.0
52	254	15	0	0.0
53	308	19	0	0.0
54	177	0	0	0.0
55	422	0	0	0.0
56	173	0	0	0.0
57	320	0	0	0.0
58	59	0	0	0.0
59	249	0	0	0.0
60	139	0	0	0.0

TABLE 12. Rho = 0.3

I	NPCKT(I)	NPSC(I)	NLOST(I)	PLOST(I)(%)
2	300	0	0	0.0
4	386	0	0	0.0
6	314	141	0	0.0
8	280	0	0	0.0
10	298	226	1	0.34
12	314	0	0	0.0
14	317	0	0	0.0
16	347	0	0	0.0
18	222	0	0	0.0
20	119	0	0	0.0
22	386	0	0	0.0
24	417	292	1	0.24
26	307	216	1	0.33
28	375	0	0	0.0
30	398	0	0	0.0
32	408	0	0	0.0
34	89	0	0	0.0
36	263	0	0	0.0
38	451	0	0	0.0
40	435	0	0	0.0
41	198	0	0	0.0
42	112	0	0	0.0
43	185	3	0	0.0
44	218	0	0	0.0
45	201	72	0	0.0
46	183	0	0	0.0
47	181	0	0	0.0
48	151	0	0	0.0
49	276	0	0	0.0
50	378	0	0	0.0
51	112	0	0	0.0
52	82	13	0	0.0
53	192	13	0	0.0
54	123	0	0	0.0
55	100	0	0	0.0
56	90	0	0	0.0
57	409	0	0	0.0
58	235	0	0	0.0
59	47	0	0	0.0
60	63	0	0	0.0

TABLE 13. Rho = 0.5

I	NPCKT(I)	NPSC(I)	NLOST(I)	PLOST(I)(%)
2	188	1	0	0.0
4	141	0	0	0.0
6	375	153	0	0.0
8	252	10	0	0.0
10	355	175	1	0.28
12	394	0	0	0.0
14	254	0	0	0.0
16	232	0	0	0.0
18	393	0	0	0.0
20	183	0	0	0.0
22	296	0	0	0.0
24	446	281	1	0.22
26	242	159	1	0.41
28	232	0	0	0.0
30	223	0	0	0.0
32	313	0	0	0.0
34	226	0	0	0.0
36	457	0	0	0.0
38	221	0	0	0.0
40	284	0	0	0.0
41	310	0	0	0.0
42	357	0	0	0.0
43	124	2	0	0.0
44	246	4	0	0.0
45	144	70	0	0.0
46	104	0	0	0.0
47	243	0	0	0.0
48	266	0	0	0.0
49	104	0	0	0.0
50	315	0	0	0.0
51	202	0	0	0.0
52	53	6	0	0.0
53	257	42	0	0.0
54	266	1	0	0.0
55	275	0	0	0.0
56	185	0	0	0.0
57	271	0	0	0.0
58	41	0	0	0.0
59	276	0	0	0.0
60	214	0	0	0.0

TABLE 14. Rho = 0.7

I	NPCKT(I)	NPSC(I)	NLOST(I)	PLOST(I)(%)
2	279	8	0	0.0
4	234	0	0	0.0
6	397	39	0	0.0
8	285	16	1	0.35
10	317	93	0	0.0
12	494	0	0	0.0
14	353	2	0	0.0
16	283	2	0	0.0
18	398	12	0	0.0
20	443	0	0	0.0
22	381	0	0	0.0
24	403	166	0	0.0
26	291	122	0	0.0
28	84	1	0	0.0
30	334	0	0	0.0
32	396	0	0	0.0
34	288	1	0	0.0
36	352	7	0	0.0
38	382	0	0	0.0
40	304	0	0	0.0
41	219	18	0	0.0
42	264	0	0	0.0
43	102	2	0	0.0
44	213	44	0	0.0
45	181	72	0	0.0
46	4	0	0	0.0
47	145	3	0	0.0
48	215	4	0	0.0
49	100	8	0	0.0
50	55	0	0	0.0
51	117	0	0	0.0
52	96	37	0	0.0
53	208	46	0	0.0
54	414	18	0	0.0
55	164	0	0	0.0
56	102	2	0	0.0
57	210	2	0	0.0
58	145	2	0	0.0
59	116	0	0	0.0
60	194	0	0	0.0

TABLE 15. Rho = 0.9

I	NPCKT(I)	NPSC(I)	NLOST(I)	PLOST(I)(%)
2	474	29	0	0.0
4	420	0	0	0.0
6	224	11	0	0.0
8	367	75	0	0.0
10	401	93	0	0.0
12	345	0	0	0.0
14	214	4	0	0.0
16	364	5	0	0.0
18	283	25	0	0.0
20	401	0	0	0.0
22	444	0	0	0.0
24	457	142	0	0.0
26	354	115	0	0.0
28	198	12	0	0.0
30	390	0	0	0.0
32	310	4	0	0.0
34	83	0	0	0.0
36	402	14	0	0.0
38	426	0	0	0.0
40	329	0	0	0.0
41	25	8	0	0.0
42	78	0	0	0.0
43	275	12	0	0.0
44	132	26	0	0.0
45	98	46	0	0.0
46	153	0	0	0.0
47	284	2	0	0.0
48	134	3	0	0.0
49	216	8	0	0.0
50	97	0	0	0.0
51	54	0	0	0.0
52	42	6	0	0.0
53	145	35	0	0.0
54	301	12	0	0.0
55	108	0	0	0.0
56	188	1	0	0.0
57	415	9	0	0.0
58	96	4	0	0.0
59	72	0	0	0.0
60	169	0	0	0.0