
Retrospective Theses and Dissertations

1987

A Simulation Study of a Limited Sensing Random Access Algorithm for a Local Area Network with Voice Users

Robert M. Spillers
University of Central Florida

 Part of the [Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/rtd>

University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Spillers, Robert M., "A Simulation Study of a Limited Sensing Random Access Algorithm for a Local Area Network with Voice Users" (1987). *Retrospective Theses and Dissertations*. 5079.

<https://stars.library.ucf.edu/rtd/5079>

A SIMULATION STUDY OF A LIMITED SENSING RANDOM ACCESS
ALGORITHM FOR A LOCAL AREA NETWORK WITH VOICE USERS

BY

ROBERT M. SPILLERS
B.S.E., United States Military Academy, 1979

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program
of the College of Engineering
University of Central Florida
Orlando, Florida

Fall Term
1987

ABSTRACT

The purpose of this work is to evaluate the performance of a limited sensing random access algorithm in a local area network with voice users. Random access algorithms have proven to be very efficient in local area network environments with data users. However, in contrast to data packets, voice packets cannot be allowed to experience long delays, because of the requirement that a voice "data stream" must be played out at the receiver. If a voice packet does exceed its established maximum delay, it is discarded. This simulation study finds the number of voice users that a network can support, provided the packet loss rate that can be tolerated by a customer does not exceed a certain threshold. Finally, a comparison is made with the simulation results of this algorithm with other commonly used protocols.

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	v
Chapter	
I. INTRODUCTION	1
II. THE MODEL	4
III. THE ALGORITHM	7
First-Time Transmission Rule	9
IV. PERFORMANCE MEASURES - SIMULATION RESULTS	15
V. DISCUSSION	25
The Model	25
Simulation Results	26
Comparisons of LSAVU with CSMA/CD and GBRAM	27
VI. CONCLUSION	29
Appendices	
A. DERIVATION OF P AND T	31
B. SIMULATION PROGRAM LISTING	32
REFERENCES	38

LIST OF TABLES

1. Packet Loss Rate for a 1MBPS Network	19
2. Packet Loss Rate for a 10MBPS Network	20
3. Comparison of Circuit Capacity	24

LIST OF FIGURES

1. Packet Movement with a Success	11
2. Packet Movement with a Collision	12
3. Packet Movement with an Idle	13
4. Packet Loss Rate Versus Number of Circuits as a Function of Packet Length for a 1MBPS Network	17
5. Packet Loss Rate Versus Number of Circuits as a Function of Packet Length for a 10MBPS Network	18
6. Comparison of Packet Loss for LSAVU Versus CSMA/CD and GBRAM on a 1MBPS Network With a Packet Length of 96 Bytes	22
7. Comparison of Packet Loss for LSAVU Versus CSMA/CD and GBRAM on a 10MBPS Network With a Packet Length of 96 Bytes	23

CHAPTER I

INTRODUCTION

Local area networks (LANs) have been used extensively in the past few years for data communications. Using a single, multi-access channel, the LAN can support a large number of users at very high data rates. Musser shows the technical feasibility of utilizing a LAN as a multi-drop local subscriber loop for a Private Automatic Branch Exchange (PABX).[1] His objective was to replace the multiple twisted pairs being pulled from the PABX with a single coaxial cable. Subscriber terminals (voice users) may then simply tap into the cable at each location. Indications show that as the cost of the tap and other electronics involved in digital telephony decrease, while labor costs of pulling wires increase, implementation of the above arrangement will become cost-effective.

It is worth noting that the "LAN arrangement," proposed by Musser, can support a variety of users (i.e., voice users, data users, etc.).[1] In other words, any user (voice, data, or other) who wants to access the channel, simply taps onto the cable. Considering the fact, that future communication networks are expected to handle a variety of traffic types, and that an enormous effort is currently being undertaken to incorporate voice

and data on the existing telephone network, the "LAN arrangement" is a step towards the right direction.

Musser examined the performance of two well known protocols. [1] A carrier sense multiple access with collision detection (CSMA/CD) protocol (specifically Ethernet) [2], which is a random access (contention) scheme and the group broadcast recognizing access method (GBRAM) protocol [3], which is a decentralized demand-assignment (contention-free) scheme. In contrast to data packets, voice packets cannot be allowed to experience long delays, because of the requirement that the receiver buffers not empty, so that "stream data" can be played out at the receiver (the telephone earpiece).[1] Therefore, voice networks must be operated on a packet-loss basis. The performance curves corresponding to CSMA/CD and GBRAM by Musser are plots of the packet loss rate versus the number of voice circuits, which are active on the channel.[1]

This thesis examines the performance of a limited sensing random access algorithm for the "LAN arrangement" proposed by Musser.[1] This random access algorithm (RAA) was first proposed by Merakos, who analyzed its performance in a LAN environment with data users, under the assumption of a slotted channel.[4] The same RAA was analyzed by Georgiopoulos for the more realistic case of an unslotted channel.[5] This RAA has a number of advantages. First, it is a limited sensing RAA, which implies that a voice user does not have to sense the channel unless it has a packet to

transmit. Secondly, it has been proven to be very effective in a LAN environment with data users for both slotted and unslotted channels.[4][5] Thirdly, it is a stable algorithm for the infinite population user model.[4][5] Fourthly, it has last-come-first-serve characteristics, which is desirable in LANs with voice users because voice packets cannot experience long delays.[4] Finally, as the results of this thesis will show, it outperforms CSMA/CD and GBRAM in a LAN environment with voice users.

CHAPTER II

THE MODEL

This model assumes that the two ends of a voice circuit generate R bits/second of traffic into the system. Voice packets of constant size L bits are assembled at regular intervals and sent to the voice user buffer. A buffer size of one packet is required at each voice user site. A packet from an active voice user will be generated at every $F = L/R$ seconds. Since a buffer size of one packet is provided at each user site, a constraint of F seconds packet lifetime must be imposed at the transmitter. A packet with transmission delay longer than F seconds results in a packet loss. Packets generated from all active voice users are stored in the appropriate buffers and then transmitted over the common cable. The model assumes that the capacity of the cable is C bits/seconds; hence, a packet will require a slot length of $\Delta = L/C$ seconds for its transmission. The length of the cable is equal to d km. The end-to-end propagation delay (the time it takes for a packet to traverse the cable from one end to the other) is denoted by α and is equal to d/v , where v is the speed of light.

To facilitate the presentation, α is taken to be the unit of time ($\alpha = 1$). To express the parameters F and Δ in units of time, let:

$$F = T * \alpha \quad (1)$$

and

$$\Delta = P * \alpha \quad (2)$$

Now, the maximum packet lifetime equals T units of time and the packet transmission time equals P units of time. Without loss of generality, P and T are assumed to be integers.

During the simulation, the system generates N packets (N conversations are active) every T units of time, and these packets are uniformly distributed over the period of T units of time. The same packet generation model was also adopted by Musser.[1]

To simplify the simulation, the following assumptions were made:

- A1. The channel is divided into slots.
- A2. The length of a slot is equal to the end-to-end propagation delay α .
- A3. Voice users are allowed to initiate packet transmissions only at the beginnings of slots.

This model considers limited channel sensing and ternary feedback. That is, it assumes that the voice users sense the channel only when they have a packet to transmit, and they can determine which one of the following occurs:

- a). no transmission (idle period)
- b). a single transmission (success)
- c). two or more transmissions (collision)

An idle period corresponds to the end-to-end propagation delay and lasts one unit of time (one slot). The length of a successful transmission corresponds to the packet transmission time plus one unit of time to inform all the users that the channel is clear. That is, $P + 1$ units of time, or $P + 1$ slots. In the case of a collision, let β denote the fraction of each packet (in units of time) that gets transmitted during the collision before the transmitting users abort their transmissions by detecting the interference. A collision lasts until all monitoring users are aware of the collision and the channel clears. That is, $\beta + 1$ units of time. For local networks using a cable, where users can listen to their own transmission, it is commonly assumed that $1 \leq \beta < P$. The users have early collision detection capabilities, and β is referred to as the collision detect time. This model assumes that $\beta = 1$.

CHAPTER III
THE ALGORITHM

The execution of the algorithm governing the accessing of the channel is divided into a series of algorithm steps. Let t_i ($i = 0, 1, 2, \dots$) denote the instants at which consecutive algorithm steps begin. These algorithm instants always coincide with the beginning of some slot. At the beginning of the operation of the system let $t_0 = 0$, $t_1 = 1$. For $i \geq 1$, let:

$$t_{i+1} - t_i = \begin{cases} 1 & \text{if the } i\text{th algorithm step is idle} \\ P+1 & \text{if a successful transmission occurs} \\ & \text{at the } i\text{th algorithm step} \\ \beta+1 & \text{if a collision occurs at the } i\text{th} \\ & \text{algorithm step} \end{cases}$$

All voice users in the network are active (see model in Chapter II). A packet may either be new or blocked. A new user at time t_i is one whose packet arrived during step $i-1$. A blocked user at t_i is one whose packet has collided before step i . Since each voice user in the network has a buffer with a capacity of one packet, the terms user and packet can be used interchangeably.

Definition: A packet in the system is called "legitimate" if its delay is smaller than the maximum packet lifetime, T .

A packet whose delay is larger than the maximum packet lifetime, T , is discarded from the system. Each "legitimate" packet has a counter, which assumes non-negative integer values. A "legitimate" packet is in the system if the counter value of the packet has already assumed a non-negative value. Let CI_i denote the counter indication of an arbitrary packet at algorithm instant t_i . The following operational rules are defined:

1. At instant t_i , all "legitimate" packets with $CI_i = 0$ are transmitted.

2. All users with "legitimate" packets in the system, sense the channel and act as follows:

- a. If a successful transmission occurred at step i , then the "legitimate" packet with $CI_i = 0$ leaves the system. All "legitimate" packets with $CI_i = r$ ($r \geq 1$) increment their counters by $m-1$ ($m \geq 1$) at instant t_{i+1} , and set $CI_{i+1} = r+m-1$, where m is an integer parameter.

- b. If a collision occurred at step i , then every "legitimate" packet with $CI_i = 0$, independently of the others, sets its counter value to $m-1+N$, where N is an integer random variable uniformly distributed on $\{1, 2, \dots, n\}$, and n is an integer parameter such that $n \geq 2$. Each of the "legitimate" packets with $CI_i = r$ ($r \geq 1$) increment their counter by $m+n-1$. Thus, $CI_{i+1} = r+m+n-1$.

c. If algorithm step i is idle, then all "legitimate" packets with counter values $CI_i > 1$ decrement their counter values by one ($CI_{i+1} = CI_i - 1$).

The integers m and n are design parameters to be optimized. To complete the description of the algorithm, the rule by which a new "legitimate" packet will determine the instant for its initial transmission attempt will now be specified.

First-Time Transmission Rule

When a new "legitimate" packet arrives during a slot at a voice user site, the user senses the channel at the beginning of the next slot. If the channel is idle, the packet sets its counter value to 0, and therefore, attempts transmission at the same instant. If the channel is sensed busy, the user waits until the channel is sensed idle for the first time (at the beginning of some slot), and only then the user sets the counter value of its packet to M , where M is an integer random variable uniformly distributed on $\{0, 1, \dots, m-1\}$.

As it can be seen from the description above, for the implementation of the algorithm in a distributed fashion, it suffices for each packet to have one counter and two random number generators.

The general operation of the algorithm is perhaps better illustrated by introducing the concept of a "stack" as it was first done by Tsybakov.[6] A stack will be understood to mean an

abstract storage device, consisting of an infinite number of cells, labeled 0, 1, 2, The number of packets that a cell can accommodate is unrestricted. At each algorithm instant, t_i , the k th cell of the stack contains the packets with $CI_i = k$ ($k \geq 0$). Packets are eventually successfully transmitted (unless they are discarded) after moving through the cells of the stack in accordance with the rules of the algorithm.

In figures 1, 2, and 3, by using the concept of the stack, a successful, a collision, and an idle step, respectively, is shown.

In Figure 1, there is one packet in the transmission cell ($X_1 = 1$) at time t_i . Therefore, the algorithm steps ahead to $t_i + P + 1$, and any new packet arrivals during the successful transmission enter the first m cells of the stack (note that new packets arriving in the slot immediately before $t_i + P + 1$ see the channel idle and enter the transmission cell). All previous "legitimate" packets in the stack increase their counter values by $m-1$, to make room for the new packet arrivals.

In Figure 2, there are K ($K \geq 2$) packets in the transmission cell at time t_i . These packets collide, and are distributed in n cells of the stack, after the algorithm steps ahead to $t_i + 2$. New packet arrivals are distributed as in Figure 1 into the first m cells of the stack, while old packets in the stack move up $m + n - 1$ cells.

In Figure 3, at time t_i , there are no packets in the transmission cell ($X_1 = 0$), and there are no new arrivals before

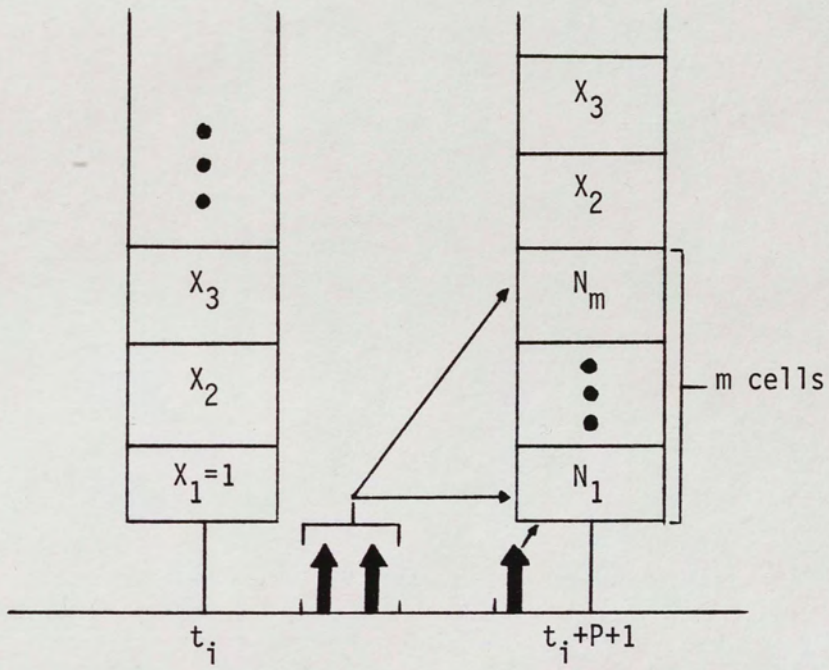


Figure 1. Packet Movement with a Success.

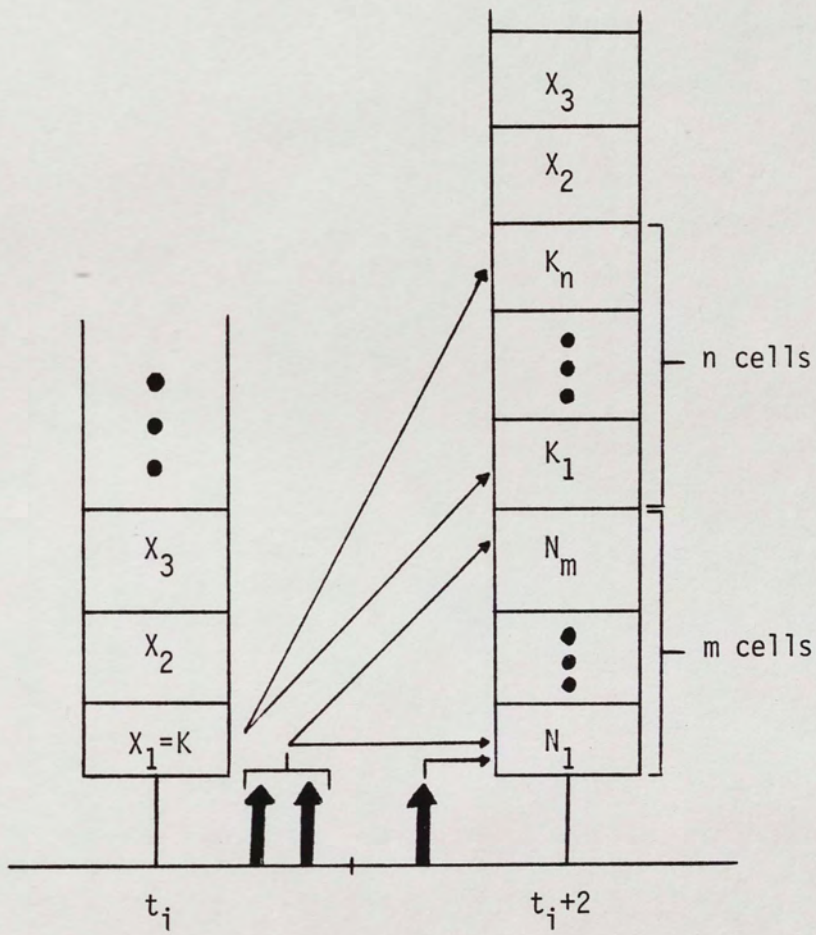


Figure 2. Packet Movement with a Collision.

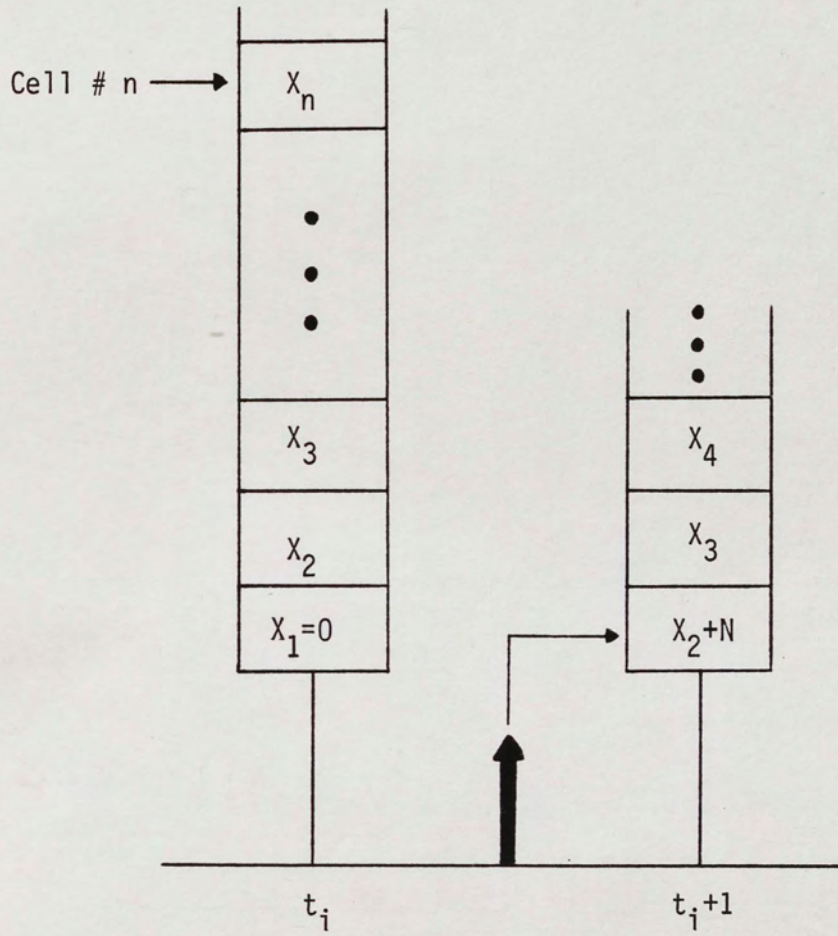


Figure 3. Packet Movement with an Idle.

t_i . The algorithm steps ahead one slot ($t_i + 1$), and all the packets in the stack (if any) move down one cell, while N ($N \geq 0$) new packet arrivals enter the transmission cell.

The above described algorithm will be referred to as LSAVU (Limited Sensing Algorithm for Voice Users) in this thesis.

CHAPTER IV
PERFORMANCE MEASURES - SIMULATION RESULTS

The most important performance measure of the effectiveness of LSAVU is the packet loss rate (averaged over all active voice circuits) versus the number of active voice circuits. The packet loss rate is defined to be the percentage of voice packets discarded by LSAVU. This performance measure was also adopted by Musser to test the effectiveness of the CSMA/CD and the GBRAM protocol.[1]

For the simulation, the model of Chapter II was adopted. The values of α , β , P , T are needed to perform the simulations. According to the model in Chapter II, $\alpha = \beta = \frac{\Delta}{1}$ unit of time. For the values of P and T , the following cases are considered:

- Case 1: $P = 231, T = 3600$
- Case 2: $P = 116, T = 1800$
- Case 3: $P = 58, T = 900$
- Case 4: $P = 29, T = 450$
- Case 5: $P = 24, T = 3600$
- Case 6: $P = 12, T = 1800$
- Case 7: $P = 6, T = 900$
- Case 8: $P = 3, T = 450$

In cases 1 through 8, the length, d , of the cable is taken to be equal to 1.0 km. Furthermore, in cases 1 through 8 each voice circuit generates $R = 64,000$ bits/second of traffic into the system. In cases 1 through 4, the capacity of the cable is $C = 1.0$ Mbps. In cases 5 through 8, $C = 10.0$ Mbps. In cases 1 and 5, the

packet length is $L = 768$ bits (96 bytes); in cases 2 and 6, $L = 384$ bits (48 bytes); in cases 3 and 7, $L = 192$ bits (24 bytes); and in cases 4 and 8, $L = 96$ bits (12 bytes). The same d , R , C , and L values were also adopted by Musser.[1] In Appendix A, the derivation for the values of P and T for cases 3 and 7 is shown.

It was determined from experimentation that 10,000 voice packets were sufficient to produce reliable simulation results. Different values of m and n were also checked, and it was found that for all cases (1-8), the optimum values were:

$$m_{\text{opt}} = 1$$

$$n_{\text{opt}} = 3$$

The optimum values m_{opt} and n_{opt} of m and n were the ones which produced the smallest packet loss rates for each case simulated. The LSAVU algorithm with $m = m_{\text{opt}} = 1$ and $n = n_{\text{opt}} = 3$ is denoted as $\text{LSAVU}_{\text{opt}}$.

In Figure 4, the packet loss rate versus the number of active voice circuits curve corresponding to the $\text{LSAVU}_{\text{opt}}$ algorithm, when the cable has a capacity of $C=1.0$ Mbps (cases 1-4) is drawn. In Figure 5, the packet loss rate versus the number of active voice circuits curve corresponding to the $\text{LSAVU}_{\text{opt}}$ algorithm, when the cable has a capacity of $C = 10.0$ Mbps (cases 5-8) is shown. Tables 1 and 2 give the numerical data corresponding to figures 3 and 4, respectively.

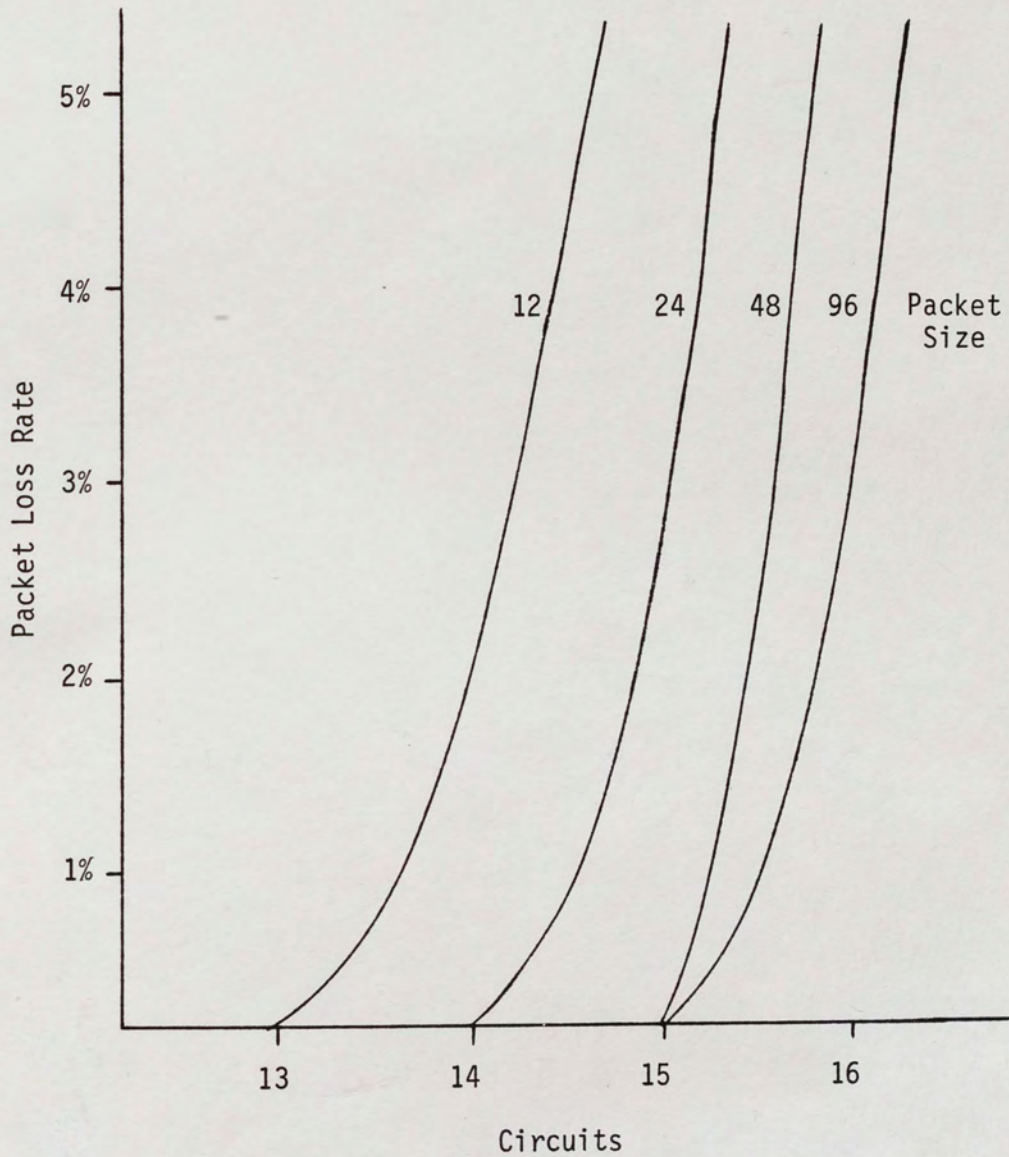


Figure 4. Packet Loss Rate Versus Number of Circuits as a Function of Packet Length for a 1 MBPS Network.

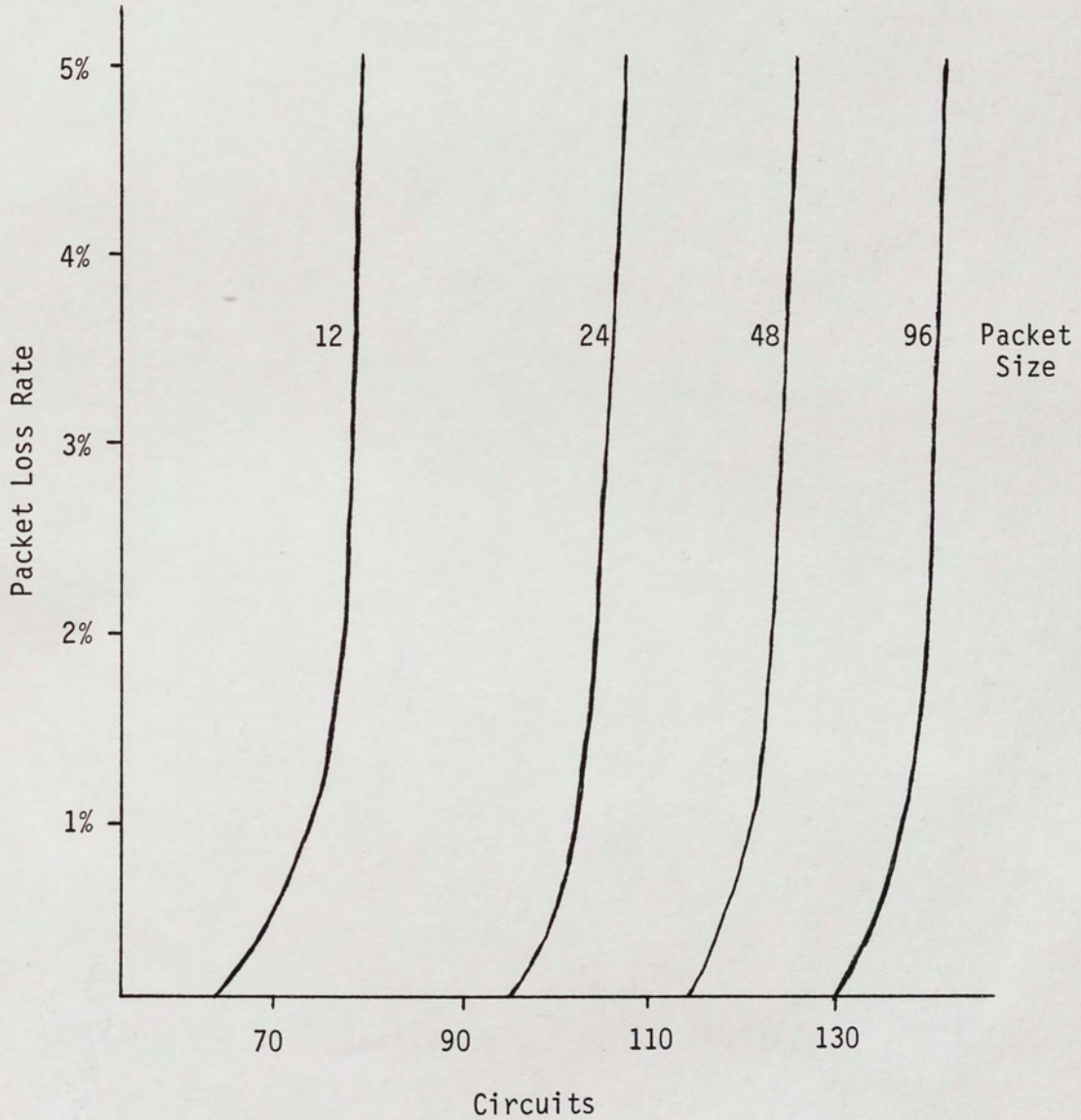


Figure 5. Packet Loss Rate Versus Number of Circuits as a Function of Packet Length for a 10 MBPS Network.

TABLE 1
 PACKET LOSS RATE FOR A 1MBPS NETWORK

CIRCUITS:		13	14	15	16	17
P	T	PACKET LOSS RATE (%)				
231	3600	0	0	0	4.03	9.71
116	1800	0	0	0	6.08	10.24
58	900	0	0	2.45	8.59	13.11
29	450	0	1.88	7.68	11.78	21.05

TABLE 2
 PACKET LOSS RATE FOR A 10MBPS NETWORK

P	T	PACKET LOSS RATE (%)				
CIRCUITS:		130	132	133	134	135
24	3600	0	0.17	0.86	1.83	2.28
CIRCUITS:		115	117	188	119	120
12	1800	0.01	0.11	0.68	1.60	2.68
CIRCUITS:		90	92	95	97	100
6	900	0	0	0.05	0.25	2.06
CIRCUITS:		68	70	72	73	75
3	450	0.13	0.56	1.02	2.15	3.63

In Figure 6, the packet loss rate versus the number of active voice circuits curves corresponding to CSMA/CD [1], GBRAM [1], and LSAVU_{opt} algorithms, when the cable has a capacity of $C = 1.0$ Mbps and the packet length is $L = 768$ bits (case 1) is shown. Finally, in Figure 7, the packet loss rate versus the number of active voice circuits curves corresponding to CSMA/CD [1], GBRAM [1], and LSAVU_{opt} algorithms, when the cable has a capacity of $C = 10.0$ Mbps and $L = 768$ bits (case 5) is shown.

Figures 6 and 7 exhibit the superior performance of LSAVU compared to CSMA/CD or GBRAM. To get a better feeling of the superiority of LSAVU_{opt}, Table 3 shows the number of voice circuits supported by LSAVU_{opt}, CSMA/CD [1], and GBRAM [1] at a packet loss rate of 2% for a 1.0 Mbps and a 10.0 Mbps cable and a packet length of 768 bits. Note that at a packet loss rate of 2%, LSAVU_{opt} can support 134 voice circuits, while GBRAM can support only 125 voice circuits, and CSMA/CD can support only 94 voice circuits on a 10.0 Mbps cable.

In Appendix B, the computer program which produced the simulation results in figures 4 and 5 and tables 1 and 2 is listed.

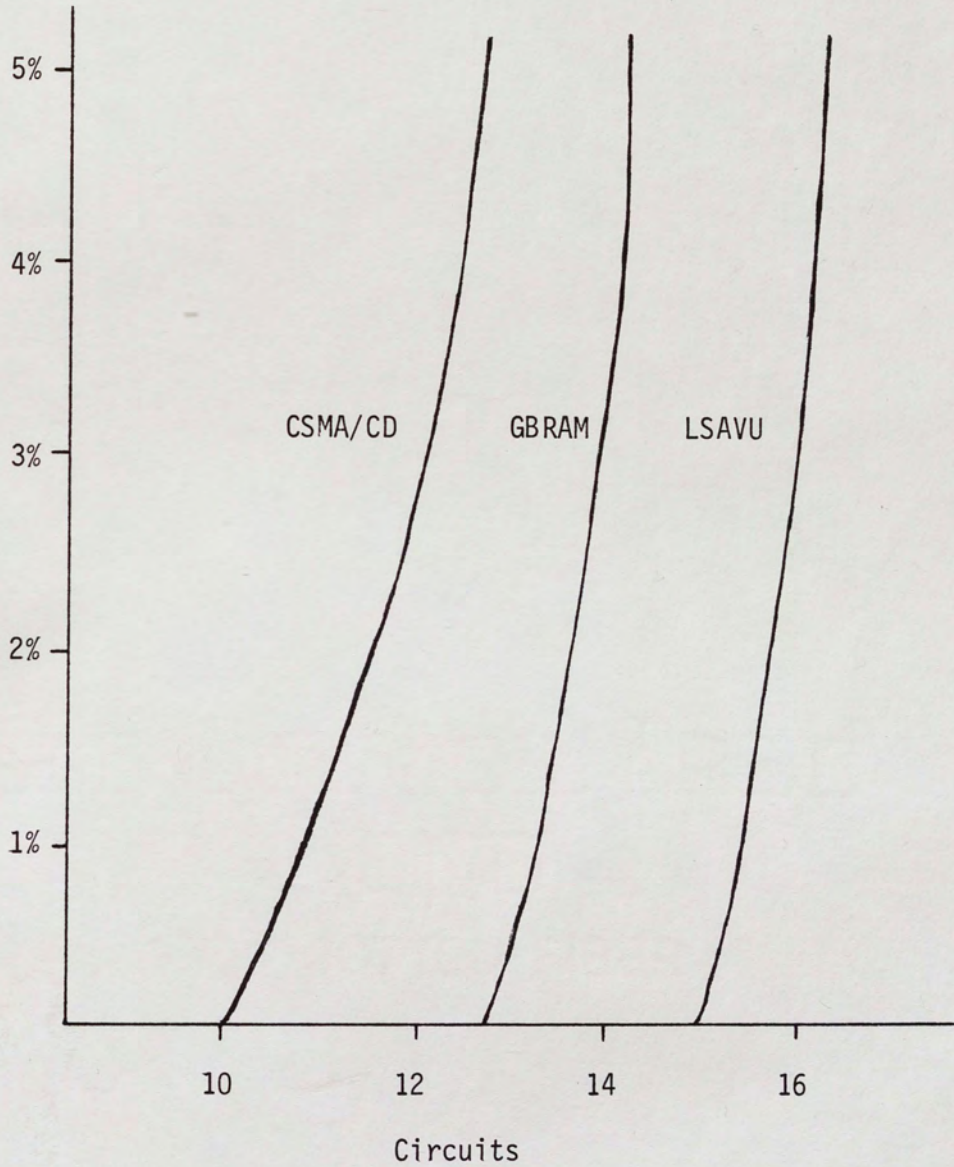


Figure 6. Comparison of Packet Loss for LSAVU Versus CSMA/CD and GBRAM on a 1 MBPS Network with a Packet Length of 96 Bytes.

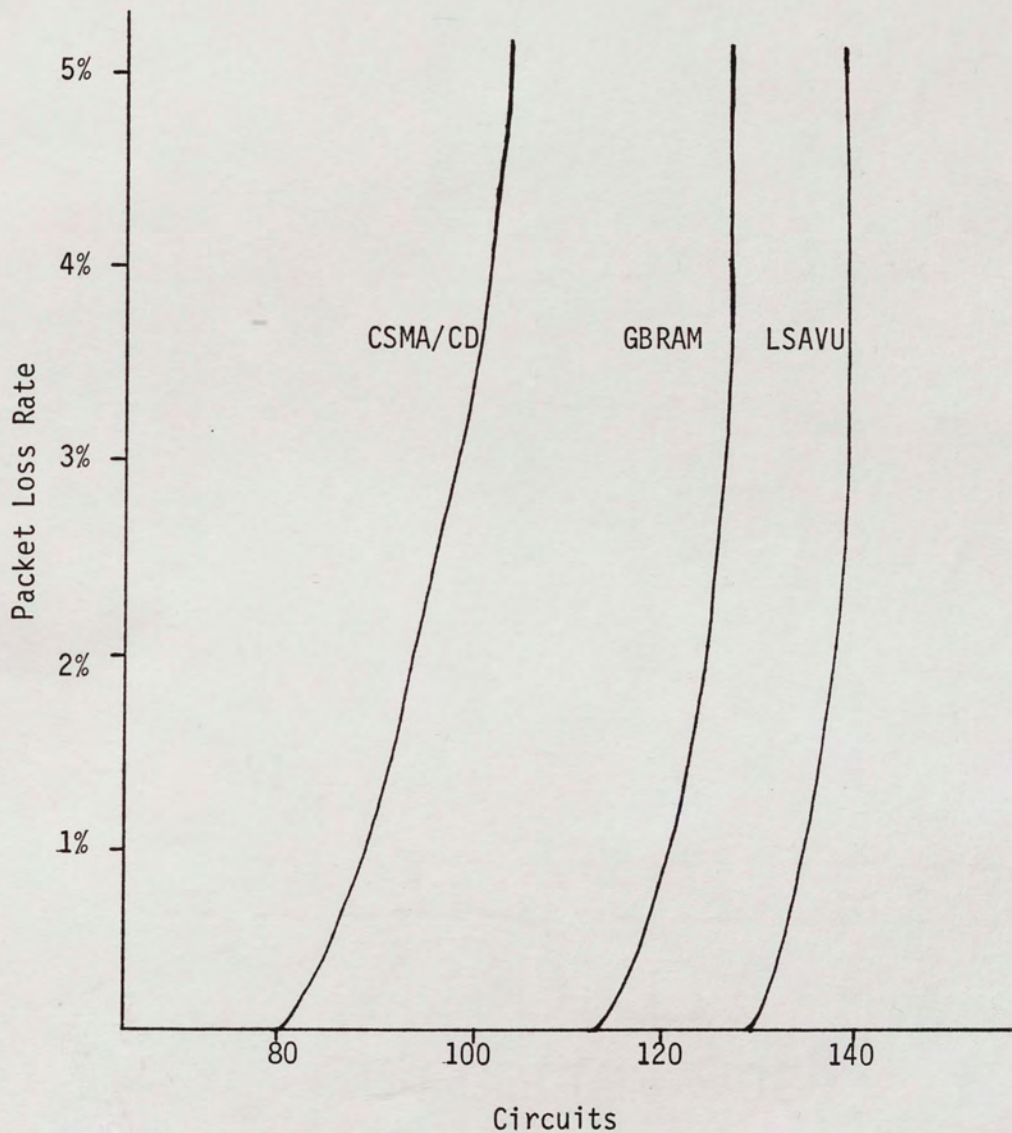


Figure 7. Comparison of Packet Loss for LSAVU Versus CSMA/CD and GBRAM on a 10 MBPS Network with a Packet Length of 96 Bytes.

TABLE 3
COMPARISON OF CIRCUIT CAPACITY

	CSMA/CD	1 MBPS GBRAM	LSAVU	CSMA/CD	10 MBPS GBRAM	LSAVU
Circuits	12	14	15	94	125	134

CHAPTER V

DISCUSSION

The Model

The model assumed in Chapter II implies that N is the maximum number of voice circuits that can be active at the same time. Furthermore, it implies that a voice circuit is continuously in a talkspurt. Experimental results have shown that in an active conversation, a talkspurt alternates with a silence period and vice versa.[7] It has been found that talkspurts and silence periods are exponentially distributed with a mean = 1.34 and 1.67 seconds, respectively.[8] Based on the above observations, it concludes that the simulation results of Chapter IV are pessimistic. In other words, $LSAVU_{opt}$ can support more voice circuits than shown in figures 4 and 5 or tables 1 and 2.

The assumption that the voice packets are uniformly distributed over the maximum packet lifetime (T) is also realistic. Experimental results have shown that the voice calling generation process is Poisson.[9] Once the assumption (as in the model) is made, that the number of active voice circuits is equal to N , the N packets generated every T units of time are uniformly distributed in this interval. This is a well-known property of the Poisson process.[9]

The assumption of the slotted channel need not be made. It was adopted to simplify the simulations. Actually, it is shown that the RAA of Chapter III performs better in the unslotted LAN environment than in the slotted one.[5] The main reason for the improvement in the performance is, that in the unslotted version, the uncertainty interval (the interval over which collisions can happen) is generally smaller than the end-to-end propagation delay, while in the slotted version, the uncertainty interval is exactly equal to the end-to-end propagation delay.

Simulation Results

The simulation results show that values of m and n near the optimum values $m_{opt} = 1$ and $n_{opt} = 3$ did not affect the performance of LSAVU. The simulation results also show that for a cable of constant capacity, LSAVU_{opt} performed better (supported larger numbers of voice circuits) for the large packet size ($L = 768$ bits = 96 bytes). This is a common characteristic of random access (contention) schemes in a LAN. They perform better when the ratio of the end-to-end propagation delay to the packet length becomes smaller. Finally, the simulation results show that LSAVU_{opt} operates near 0% packet loss rate up to a point and then there is a sharp increase in the packet loss rate. Therefore, the cutoff for the number of voice circuits supported is very abrupt.

The number of voice circuits that LSAVU_{opt} supports, such that the maximum individual (per voice circuit) packet loss rate

is smaller than 2%, is almost identical to the number of voice circuits that $LSAVU_{opt}$ supports, such that the packet loss rate (averaged over all active voice circuits) is smaller than 2%. In only one case did the individual packet loss rate lower the number of voice circuits that $LSAVU_{opt}$ supports. In case 4 (see Chapter IV), the number of voice circuits that $LSAVU_{opt}$ supports was reduced from 14 to 13.

Comparisons of LSAVU with CSMA/CD and GBRAM

Musser's simulation results were conducted with slightly different parameters than the parameters used to simulate $LSAVU_{opt}$. In fact, Musser assumes:

- MA1. A jam time of $4.8 \mu s$ is enforced after each collision.
[1]
- MA2. A $9.6 \mu s$ transmit/receive turnaround time is imposed.
[1]
- MA3. 6 bytes of control overhead and synchronization are appended to each voice packet before it is sent over the channel.[1]

For the simulations with LSAVU, the jam time and transmit/receive turnaround time is assumed to equal zero. Also, it is assumed that the voice packets consist of information bytes only.

Incorporating MA1 in the model of Chapter II, you simply have to increase β by the appropriate amount instead of letting $\beta = \alpha$. Incorporating MA2 and MA3 in the model, you have to increase the packet length (P) by the appropriate amount.

From experience with the RAA described in Chapter III, and the results of Table 1 in Merakos [4] , it is concluded that the simulation results of figures 4 and 5 and tables 1 and 2 will not be significantly changed. Hence, the claim that LSAVU outperforms both CSMA/CD and GBRAM remains valid, especially for the 10.0 Mbps cable (see also figures 6 and 7).

CHAPTER VI

CONCLUSION

The simulation study shows that LSAVU is a viable protocol for a LAN with voice users. In fact, there are strong indications that LSAVU outperforms both CSMA/CD and GBRAM.

The next research effort in this area is to conduct a simulation study with a mixture of data users and voice users. This would be in line with the ultimate goal of integrating voice with data on a LAN using a random access algorithm (RAA).

APPENDICES

APPENDIX A
DERIVATION OF P AND T

An example on how the values for P and T were found follows:

Case 3

α = cable length/speed of light

α = 1.0 KM/300,000 KM/sec

Define: $\alpha = 1$

For a packet length of 24 bytes ... = 24 * 8 = 192 bits

$P = (192/1 \text{ MBPS}) * (1/\alpha)$

$P = (192 * 300,000)/1,000,000 = 57.6$

Let $P = 58$ (next integer value)

$T = (192 * 300,000)/64,000 = 900$

Note that P and T are integer values, normalized to $\alpha = 1$.

Case 7

The only change from above is:

$P = (192/10.0 \text{ MBPS}) * (1/\alpha)$

$P = 5.76$, Let $P = 6$

$T = 900$ (as above)

APPENDIX B

SIMULATION PROGRAM LISTING

```

c fortran program to simulate voice packet loss rate
c using the random access algorithm developed by Merakos
c implemented on a local area network.
c a(i) = packet number
c r(i) = counter value
c s(i) = packet number in stack

      integer a(100000), r(1000), s(1000), b, ct, p, nm,
:nt, t, p1, i, p2, p3, m, n, nq, nq1, ns, op, nd
      integer j, x, ch, k, temp, oj, om, f, g, h, ix,
:nd1(200)
      real plr, rf, ri, rp1, z, y
c generate random packets uniformly distributed over
c period 0 to t.
c nm = number of packets generated
c p = packet length
c t = max packet lifetime
c f = number of active voice circuits

      nm=10000
      p=3
      t=450
      f=73
      g=f-1
      do 62 i=1,f
      call randa(x)
      a(i)=x
      nd1(i)=0
62  continue
c sort random numbers into numerical order
91  ch=0
      do 95 k=1,g
      if (a(k).le.a(k+1)) go to 95
      temp=a(k)
      a(k)=a(k+1)
      a(k+1)=temp
      ch=1
95  continue
      if (ch.eq.1) go to 91
      do 96 m=1,f
      write (*,*) 'a(i)=' ,a(m)

```

```

96  continue
    do 98 m=1,f
      do 64 j=f,(int(nm/f)*f),f
        a(j+m)=a(m)+((j/f)*t)
64  continue
98  continue
    do 93 i=nm-100,nm
      write(*,*)i,a(i)
93  continue
c  ct = current time
c  nst = number of packets in stack
c  nd = number of packets discarded
c  nt = number of packets in TX cell
c  b = collision detect time
c  ns = number of successfully TX packets
c  nq = number of packets that have entered stack
c  M and N are optimized stack values
    ct=0
    nst=0
    nd=0
    nt=0
    b=1
    ns=0
    nq=0
    m=1
    n=4
c
c  program checks for arrivals and new packets enter stack.

10  if ((ns+nd).eq.nm) go to 100
c  nq1 = next packet to enter stack
    nq1=nq+1 if (nq1.gt.nm) go to 13
    do 1 i=nq1, nm
      if (a(i).ge.ct) go to 13
      if (a(i).ge.(ct-1).and.a (i).lt.ct) go to 12
    go to 1
12  nq=nq+1
    nst=nst+1
    s(nst)=i
    r(nst)=0
    1  continue

c  no packets in stack

13  if (nst.eq.0) go to 15

c  packets in stack

    if (nst.gt.0) go to 20

```

c increase current time and check next interval.

```

15  nq1=nq+1
    if (nq1.le.nm) ct=a(nq1)+1
    if (nq1.gt.nm) ct=ct + 1
    go to 10

```

c discard packets in stack with delay > t.

```

20  p1=0
if (nst.lt.1) go to 30
    do 2 i=1, nst
        if ((ct-a(s(i))).ge.t) p1=i
    2  continue
        if (p1.ne.0) go to 25
c no packets discarded.

```

```

    if (p1.eq.0) go to 30

```

c packet discarded and stack count adjusted.

c check individual packet loss rate

```

25  rf=f
    rp1=s(p1)
    do 1000 i=1,f
        ri=i
        z=(rp1-ri)/rf
        ix=z
        y=ix
        if (z.eq.y) nd1(i)=nd1(i)+1
1000 continue
    if (p1.eq.nst) go to 26
    if (p1.ne.nst) go to 27
26  nst=nst-1
    nd=nd+1
    write(*,*)'nd=',nd,'ct=',ct,'s(i)=' ,s(i)
    go to 20
27  p2=p1+1
    do 3 i=p2, nst
        s(i-1)=s(i)
        r(i-1)=r(i)
    3  continue
    nst=nst-1
    nd=nd+1
    write(*,*)'nd=',nd,'ct=',ct,'s(i)=' ,s(i)
    go to 20

```

c determine length of step forward.


```

30  nt=0
    if (nst.eq.0) go to 15
    p3=0
    do 4 i=1, nst
      if (r(i).eq.0) nt=nt+1
      if (r(i).eq.0) p3=i
4   continue
    if (nt.eq.0) go to 35
    if (nt.eq.1) go to 37
    if (nt.gt.1) go to 50

```

c no packets in tx cell, adjust stack, increase ct
c and check for new arrivals.

```

35  ct=ct+1
    do 5 i=1, nst
      r(i)=r(i)-1
5   continue
    go to 10
37  ns=ns+1
    if (p3.eq.nst) go to 39
    if (p3.ne.nst) go to 38

```

c

```

38  p2=p3+1
    do 6 i=p2, nst
      s(i-1)=s(i)
      r(i-1)=r(i)
6   continue

```

c

```

39  nst=nst-1
    if (nst.eq.0) go to 81
    do 7 i=1, nst
      r(i)=r(i)+m-1
7   continue

```

c

```

81  nq1=nq+1
    if (nq1.gt.nm) go to 85
    do 8 i=nq1, nm
      if (a(i).ge.(ct+p)) go to 85
      if (a(i).ge.ct.and.a(i).lt.(ct+p)) go to 42

```

```

42  nst=nst+1
    s(nst)=i

```

c generate a random number between 0 and (m-1) = op

c

```

    call randi(op)
    r(nst)=op
    nq=nq+1
8   continue

```

c increase time by packet length.

```
85 ct=ct+p+1
   go to 10
```

c

```
50 do 9 i=1, nst
   if (r(i).eq.0) go to 53
   if (r(i).ne.0) go to 55
```

c random number between 1 and n = oj.

```
53 call randj(oj)
   r(i)=r(i)+(m-1)+oj
   go to 9
55 r(i)=r(i)+(m-1)+n
   9 continue
   nq1=nq+1
   if (nq1.gt.nm) go to 75
   do 60 i=nq1, nm
   if (a(i).ge.(ct+b)) go to 75
   if (a(i).ge.ct.and.a(i).lt.(ct+b)) go to 70
```

c

```
70 nst=nst+1
   s(nst)=i
   nq=nq+1
```

c random number between 0 and (m-1) = om

```
   call randi(om)
   r(nst)=om
60 continue
```

c increase time by collision interval.

```
75 ct=ct+b+1
   go to 10
```

```
100 plr=(real(nd)/real(nm))*100.
   write(*,*) '# voice ckts =',f
   write(*,*) 'p=',p,'t=',t
   write(*,*) 'nm=',nm,'m=',m,'n=',n
   write(*,*) 'ns=',ns,'nd=',nd,'ct=',ct
   write(*,*) 'packet loss rate=',plr,'%'
```

```
write(1,*) '# voice ckts =',f
write(1,*) 'p=',p,'t=',t
write(1,*) 'nm=',nm,'m=',m,'n=',n
write(1,*) 'ns=',ns,'nd=',nd,'ct=',ct
write(1,*) 'packet loss rate =',plr,'%'
```

```
write(*,*) (nd1(i), i=1,f)
```

```

write(1,*) (nd1(i), i=1,f)
stop
end

```

```

c
c subroutine to generate uniformly distributed random numbers
c between 0 and m-1.

```

```

subroutine randi(x)
integer lo, hi, l, c, m, seed, x
real randu
lo=0
hi=0
l=29
c=217
m=1024
data seed /433/
seed=mod(seed*l+c,m)
randu=real(seed)/m
x=int(randu*(max(lo,hi)-min(lo,hi)+1))
end

```

```

c
c subroutine to generate uniformly distributed random numbers
c between 1 and n.

```

```

subroutine randj(x)
integer lo, hi, l, c, m, seed, x
real randu
lo=1
hi=4
l=29
c=217
m=1024
data seed /341/
seed=mod(seed*l+c,m)
randu=real(seed)/m
x=(int(randu*(max(lo,hi)-min(lo,hi)+1))+1)
end

```

```

c
c subroutine to generate packets, uniformly distributed over 0 to
c T.

```

```

subroutine randa(x)
integer lo, hi, l, c, m, seed, x
real randu
lo=0
hi=449
l=29
c=217
m=1024
data seed /9873/
seed=mod(seed*l+c,m)
randu=real(seed)/m
x=int(randu*(max(lo,hi)-min(lo,hi)+1))
end

```

TK 5105.7

REFERENCES

- [1] Musser, Jeffrey M.; Liu, Then Tang; Li, Liang; and Boggs, George J. "A Local Area Network as a Telephone Local Subscriber Loop." IEEE Journal on Selected Areas in Communications SAC-1, No. 6 (December 1983): 1046-1053. TK5101.A1 135
- [2] Metcalfe, R.M., and Boggs, D.R. "Ethernet: Distributed Packet Switching for Local Computer Networks." Commun. Ass. Comput. Mach. 19, No. 7 (1976): 395-403.
- [3] Liu, T.T; Li, L.; and Franta, W.R. "A Decentralized Conflict-Free Protocol, GBRAM for Large Scale Local Networks." In Proc. Comput. Network. Symp. (December 1981): 39-54.
- [4] Merakos, L. "Stack Algorithm for Local Area Networks Using Carrier Sense Multiple Access Channels." Technical Report, University of Connecticut, Storrs, CT, 1983.
- [5] Georgiopoulos, M.; Merakos, L.; and Papantoni-Kazakos, P. "High Performance Asynchronous Limited Sensing Algorithms for CSMA and CSMA/CD Channels." In Local Area and Multiple Access Networks, pp. 185-215. Edited by R.L. Pickholtz. New York: Computer Science Press, 1986.
- [6] Tsybakov, B.S., and Vvedenskaya, N.D. "Stack Algorithm for Random Multiple Access." Problemy Peredachi Informatsii 15, No. 3 (July/September 1980): 80-94.
- [7] Brady, P.T. "A Technique for Investigating On-Off Patterns of Speech." Bell Sys. Tech. Journal 44 (January 1965): 1-22.
- [8] Brady, P.T. "A Statistical Analysis of On-Off Patterns in Sixteen Conversations." Bell Sys. Tech. Journal 47 (January 1968): 73-91.
- [9] Schwartz, M. Telecommunication Networks. New York: Addison Wesley, 1987.