AD-A065 449 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 17/2
DATA TRAFFIC PERFORMANCE OF AN INTEGRATED CIRCUIT- AND PACKET-S--ETC(U)
OCT 78 C J WEINSTEIN, M L MALPASS F19628-78-C-0002
UNCLASSIFIED TN-1978-41 ESD-TR-78-276 NL

| OF |

AD
A065449

END
DATE
FILMED
6-79
DDC

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

DATA TRAFFIC PERFORMANCE OF AN INTEGRATED CIRCUIT- AND PACKET-SWITCHED MULTIPLEX STRUCTURE

C. J. WEINSTEIN
M. L. MALPASS
Group 24

M. J. FISCHER
Defense Communications Engineering Center

TECHNICAL NOTE 1978-41

26 OCTOBER 1978

LEXINGTON MASSACHUSETTS

ABSTRACT

New results are presented for data traffic performance in an integrated multiplex structure which includes circuit-switching for voice and packet-switching for data. The results are developed both through simulation and analysis, and show that excessive data queues and delays will build up under heavy loading conditions. These large data delays occur during periods of time when the voice traffic load through the multiplexer exceeds its statistical average. A variety of flow control mechanisms to reduce data packet delays are investigated. These mechanisms include control of voice bit rate, limitation of the data buffer, and combinations of voice rate and data buffer control. Simulations indicate that these flow control mechanisms are quite effective in improving system performance. A combination of data buffer limitation with data-queue-dependent voice rate control was the most effective flow control technique tested.

CONTENTS

I. INTRODUCTION

An analysis of the performance of an integrated, circuit- and packet-switched multiplexer structure was presented recently by Fischer and Harris.[1] The structure, introduced by Coviello and Vena,[2] is based on a time-slotted frame format where a certain portion of the frame is allocated to circuit-switched calls and the remaining capacity is reserved for data packets. To increase channel utilization, a "movable boundary" feature is included, so that data packets are allowed to use any residual circuit-switched capacity momentarily available due to voice traffic variations. The performance measures employed in the analysis were blocking probability for circuit-switched voice customers and average waiting time for data packets. Results indicated significant saving, in terms of channel requirements for the integrated system, as compared to a system with separate circuit- and packet-switched facilities handling the same traffic loads with the same blocking probability and average packet waiting time. However, an error has been discovered in the original derivation of average data packet waiting time. New results[3] indicate much larger data packet delays than previously predicted for the case where the service capacity reserved for data is smaller than the data traffic load. These results tend to indicate that the advantage of the integrated system in terms of channel utilization is much less than indicated previously.

However, these results should certainly not be considered in isolation in evaluating integrated system approaches. In particular, the analysis here assumes constant levels of average voice and data traffic load, and shows that a dynamically movable boundary is not very effective in allowing data packets to utilize free capacity due to momentary fluctuations in voice traffic. However, it is also shown in this paper that appropriate flow control mechanisms can appreciably improve data traffic performance with a dynamically movable boundary. In addition, an integrated system ought to allow effective adaptation to the more gradual variations in voice and data traffic loads that would typically occur over a 24-hour period. Finally, other criteria such as flexibility and overall hardware cost must be considered together with efficiency of channel utilization in comparing various system configurations.

The primary purpose of this paper is to develop modified results for the data performance of the multiplexer structure. These new results have been obtained through two approaches: (1) a simulation designed to match the original model as closely as possible; (2) an analysis of a somewhat modified system model, which is more analytically tractable yet retains the same essential performance characteristics. The simulation model and sample results will be presented first, and an attempt will be made to explain the basic cause of the large packet delays. This material is contained in Section II. Then the analytical approach together with some sample results is given in Section III. The second purpose of the paper is the investigation of techniques for controlling the flow of voice and/or data traffic through the multiplex structure, with the goal of decreasing data packet delay. This work is described in Section IV. Explanation of the error in the earlier analysis and a more detailed description of the new analytical approach as well as comparisons with the simulation runs are given in Appendices.

II. SIMULATION OF MULTIPLEXER PERFORMANCE

A. Frame Structure

The multiplexing technique under investigation is a Time Division Multiplex (TDM) scheme in which time slices of fixed duration, a frame period, are partitioned and allocated to the transmission of digitized voice and data packets. The voice component of a frame is further divided into slots allocated to ongoing line switching communications. In general the voice slots can be of varying size depending on the bit rates of different voice users, and the size of data packets is also variable. For simplicity, and in order to allow direct comparison with the earlier analysis,[1] the simulation here has assumed a multiplexer frame (taken to be of duration $b = 10$ ms) divided into $S + N$ equal-capacity slots. The nominal bit rate for the voice coders is taken as 8 kbps, which is accommodated with an 80 bit slot size and a 10 ms frame period. Voice traffic is allowed to occupy the first S slots, and the remaining N slots are reserved for data packets which are each assumed to occupy one slot. A voice call retains its slot for the duration of the connection; but if all S voice slots are busy when a new call is initiated, that call is blocked. Data traffic is permitted to use any voice slots which may be momentarily free due to statistical fluctuations in the voice traffic.

B. Voice Traffic Simulation Model

Voice traffic is modelled by a Poisson call arrival process of rate λ and exponentially-distributed call holding times with mean μ^{-1}. The average number of call arrivals, λb, in a frame interval is generally small enough so that the frame structure can be ignored for voice traffic analysis. Thus the voice multiplexer can be described as a classical Markov S-server loss system M/M/S/S whose state-transition-rate diagram is shown in Fig. 1.

Let $n_v(t)$ represent the number of calls in progress at a given time. A simulation of the variation of $n_v(t)$ with time, based on the state-transition diagram, has been developed as follows. Assume that $n_v(t)=k$ at a particular starting time. Then $n_v(t)$ is held at k for a time τ drawn from an exponential distribution

$$f_\tau(t) = \tau_k^{-1} e^{-t/\tau_k}, \tag{1}$$

where the mean holding time τ_k is determined as

$$\tau_k = 1/(\lambda + k\mu) \qquad k = 0, 1, \ldots, S-1 \tag{2}$$

$$\tau_S = 1/S\mu.$$

After a time τ, $n_v(t)$ is increased to k+1 with probability

$$P_{up}(k) = \lambda/(\lambda + k\mu) \qquad k = 0, 1, \ldots, S-1 \tag{3}$$

$$P_{up}(S) = 0$$

or decreased to k-1 with probability $1-P_{up}(k)$. This process is repeated as often as desired to generate sample functions of $n_v(t)$.

It is useful to note that the steady state blocking probability for the S-server loss system is given by the "Erlang-B formula"

$$P_L = \frac{(\lambda/\mu)^S/S!}{\sum_{k=0}^{S} (\lambda/\mu)^k/k!} \tag{4a}$$

18-2-13825

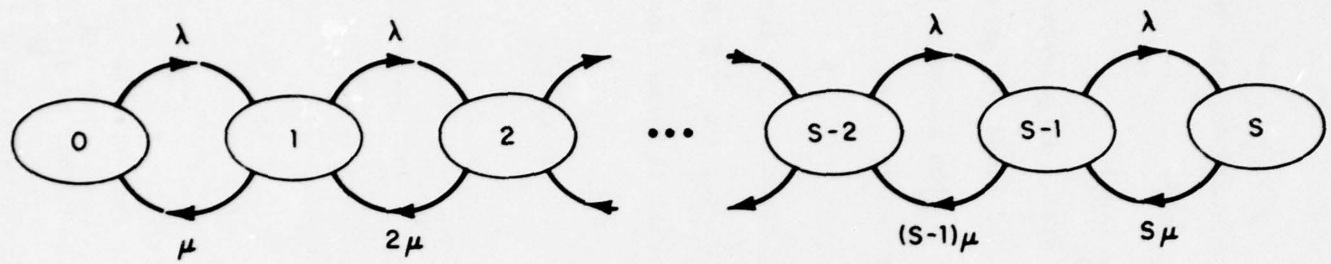

Fig.1. State-transition-rate diagram for S-server loss system M/M/S/S. When system is in state k, k calls are active and k voice slots are occupied $[n_v(t) = k]$.

and that the expected value of n_v, $\langle n_v \rangle$, is given as

$$\langle n_v \rangle = (\lambda/\mu)\ (1-P_L), \tag{4b}$$

where n_v is the random variable representing the number of voice calls in the system.

C. Data Traffic Simulation Model

Data packets are assumed to arrive in a Poisson process with rate θ packets/sec. Packet size is assumed fixed and equal to the number of bits in one slot. During any frame, the number of slots available for transmission of data packets is $N+S-n_v(t)$, the sum of N dedicated data slots and $S-n_v(t)$ unused voice slots.

Consider a period of time of duration t_i between the ith and (i+1)st transition in $n_v(t)$, during which $n_v(t) = n_v^{(i)}$; the number, n_+, of data packets arriving during this interval is drawn from a Poisson distribution

$$p(n_+) = \frac{(\theta t_i)^{n_+}\ e^{-\theta t_i}}{n_+!} \tag{5}$$

with mean θt_i. The maximum number of data packets which can be sent out during this time is

$$n_- = (N + S - n_v^{(i)})\ (t_i/b) \tag{6}$$

(since generally $t_i \gg b$, the fact that t_i/b is not necessarily an integer has been ignored.) Let the number of data packets waiting for service at the beginning of the ith interval be $n_d^{(i)}$. The evolution of this random variable to the next interval is simulated as

$$n_d^{(i+1)} = \max (n_d^{(i)} + n_+ - n_-, 0) \quad (7)$$

where the max insures that $n_d^{(i+1)}$ never becomes negative. This equation is strictly valid as long as the data queue does not empty during the interval t_i. If the data queue does empty during the interval, then $n_d^{(i+1)}$ may be slightly greater than predicted by (7), if the dispersion of packet arrival times causes some packet service slots to go unused. This slight discrepancy during periods when n_d is close to zero would have negligible effects on the results presented below, and has been ignored.

The model used for the combined voice/data simulation is described by equations (1)-(7). Note that n_d is updated only at times of transition in $n_v(t)$. For the purpose of display and of determining time averages, $n_d(t)$ is assumed to vary linearly between its sample values $n_d^{(i)}$. Suitable random number generators are used to determine τ_k, n_+, and the up/down transition decisions for $n_v(t)$. The average data packet waiting time is determined from measured values of $\langle n_d \rangle$ by Little's theorem as

$$W_d = \langle n_d \rangle / \theta , \quad (8)$$

where $\langle n_d \rangle$ is a time average of n_d.

D. Simulation Results

1. Analysis of Sample Runs

Sample functions of $n_v(t)$ and $n_d(t)$ are shown in Figure 2 for S = 10, N = 5. The average call holding time is taken as $1/\mu = 100$ sec, with $\lambda = .05\ \text{sec}^{-1}$ so that the offered voice traffic is λ/μ Erlangs.

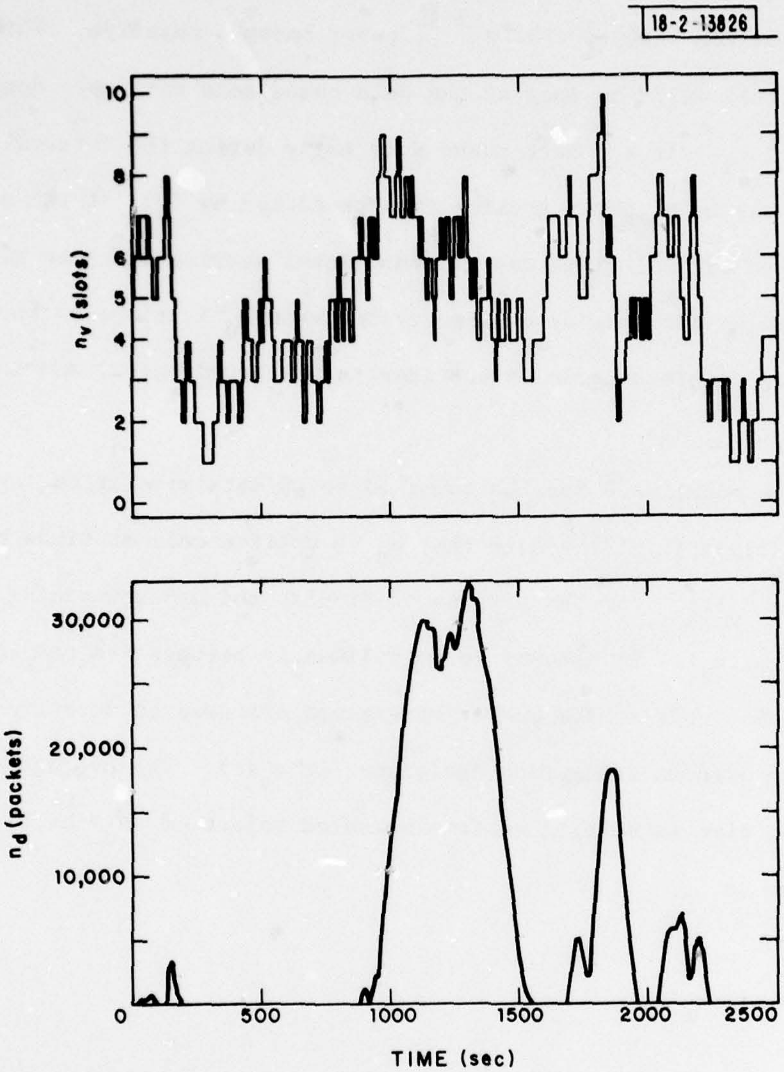

Fig. 2. Sample functions of $n_v(t)$ and $n_d(t)$ for S = 10, N = 5, $\lambda = 0.05\ sec^{-1}$, $\mu = 0.01\ sec^{-1}$ and $\theta = 900$ packets/sec ($\theta b = 9$ packets/frame).

Application of (4a) yields $P_L = .018$ and $\langle n_v \rangle = 4.91$ so that the average number of slots available for data packets is

$$\overline{n_p} = N + S - \langle n_v \rangle = 10.09.$$

The data packet arrival rate is $\theta = 900\ \mathrm{sec}^{-1}$ for an average data utilization factor of

$$\overline{\rho_d} = \theta b/\overline{n_p} = .89.$$

The plots represent 2500 sec of real time, during which approximately 230 voice calls either entered or left the system. It is clear that $n_v(t)$ is a highly correlated (in time) random process which exhibits swings of long duration above and below $\langle n_v \rangle$. The error (see appendix A1) in the previous analysis effectively caused the time correlation of $n_v(t)$ to be neglected, leading to large underestimates of data packet waiting time.

During "idle" periods where $n_v(t)$ is low enough so that $\theta b < N + S - n_v(t)$ more than enough capacity is available to handle incoming data traffic and an initially empty data queue $n_d(t)$ will remain essentially empty. But $n_d(t)$ will build up significantly during busy periods where $\theta b > N + S - n_v(t)$. For this example, a long busy period during which $n_v(t) \geq 6$ for about 500 sec begins at about $t = 900$ sec. During this period the data queue builds up to about 30,000 packets. The data queue is eventually emptied during a subsequent idle period of the voice channel, but then builds up again during the next busy period. The average buffer size during a run of which the first half is depicted in

Fig. 2 is $<n_d> = 9200$ packets, corresponding to a mean delay of $W_d =$ 10.2 sec.

The behavior depicted in Fig. 2 is typical of that exhibited in many similar runs. A segment of another pair of sample functions $n_v(t)$, $n_d(t)$ is shown in Fig. 3 for $S = 50$, $N = 25$, $\lambda = .4$, $\mu = .01$, $\theta = 3400$. Here $<n_v> = 39.2$ and $p_L = .02$. There are $S-<n_v> = 10.8$ residual voice channels on the average in steady state, and the average data load fills 9 of these residual channels. The plots cover 500 sec of real time and about 390 voice transitions. The long busy period ending near $t = 400$ leads to a maximum data buffer of about 63,000 packets. Average buffer size for this run is $<n_d> = 12{,}852$ packets, and average delay is $W_d = <n_d>/\theta \simeq 3.78$ sec.

Figure 4 illustrates what happens in the extreme case when only one slot per frame ($S=1$, $N=0$) is available. Here $\lambda = .01$, $\mu = .01$, and $<n_v> = .5$ so that the slot is occupied by voice half the time. The data load θb is .4 packets/slot so that

$$\overline{\rho_d} = \theta b/\overline{n}_p = .4/.5 = .8.$$

The data buffer $n_d(t)$ obviously must increase while $n_v(t) = 1$ and these increases must be worked off while $n_v(t) = 0$. For this example, $<n_d> = 7047$ packets and $W_d = 176$ sec. The effect of the average voice holding time $1/\mu$ should be apparent from this example. If $1/\mu$ were doubled to 200 sec with λ/μ fixed, a typical sample function of $n_v(t)$ would be as shown in Fig. 4, except that twice as much elapsed time would be represented. With θ fixed, this time scale change would lead to approximately a doubling of the $n_d(t)$ values plotted in Fig. 4, and

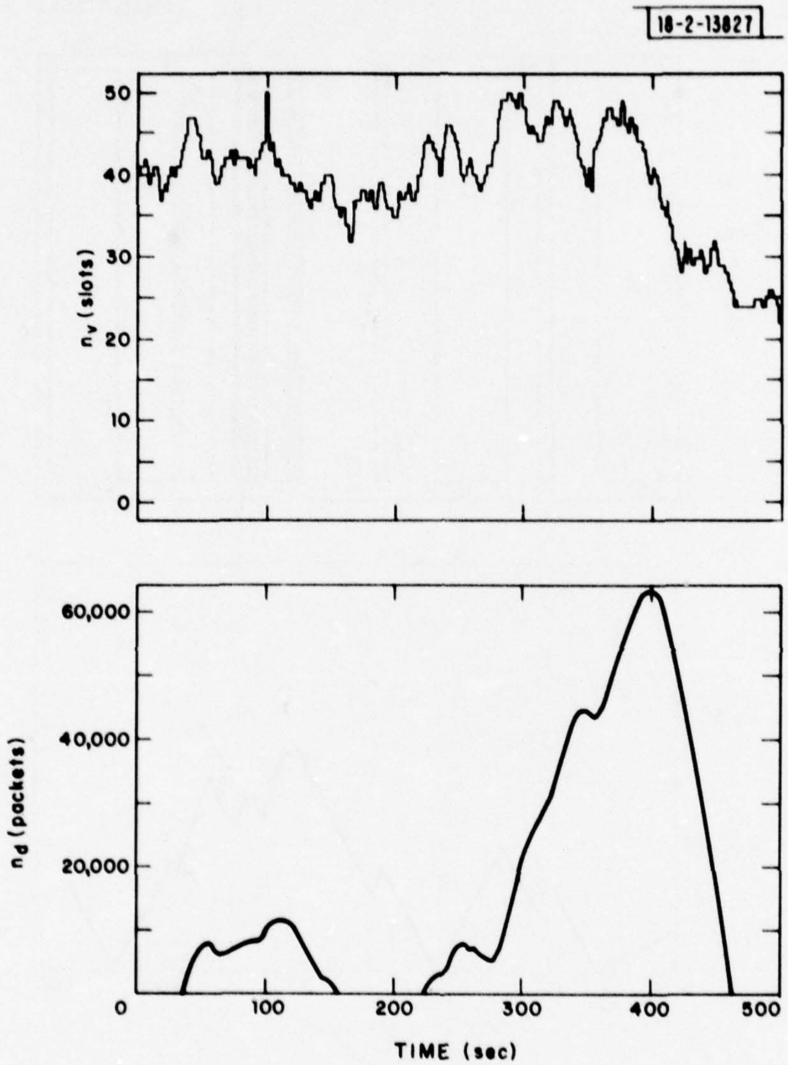

Fig. 3. Sample functions of $n_v(t)$ and $n_d(t)$ for S = 50, N = 25, λ = 0.4 sec^{-1}, μ = 0.01 sec^{-1}, and θ = 3400 packets/sec (θb = 34 packets/frame).

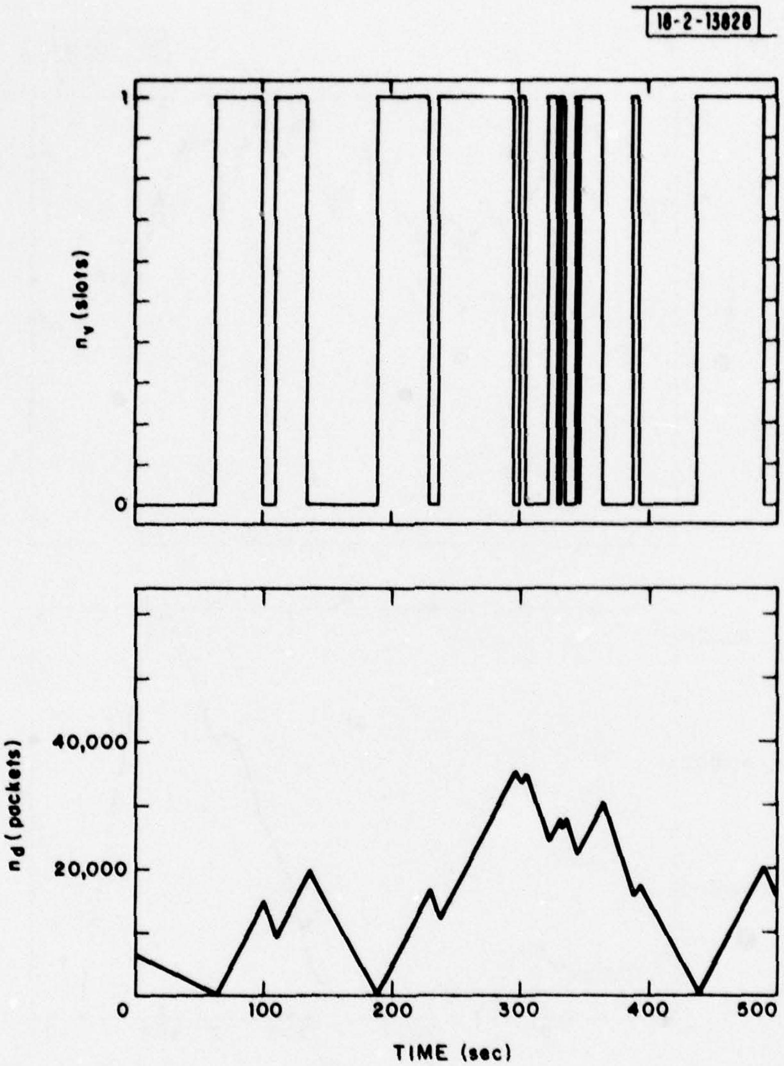

Fig. 4. Sample functions of $n_v(t)$ and $n_d(t)$ for S = 1, N = 0, $\lambda = 0.01\ sec^{-1}$, $\mu = 0.01\ sec^{-1}$, and $\theta = 40$ packets/sec ($\theta b = 0.4$ packet/frame).

therefore a doubling of $\langle n_d \rangle$ and W_d. The increased W_d is an obvious consequence of the fact that data packets (arriving when $n_v(t)=1$) must wait twice as long on the average for the ongoing conversation to terminate.

Changes in μ for fixed λ/μ will have similar effects on the results depicted in Figures 2 and 3. Actually the holding time $1/\mu = 100$ sec is shorter than typically encountered in telephone traffic, so that the results given here on data delays are probably optimistic by about a factor of 2 or 3.

2. Average Data Performance Statistics

Average data packet delays as measured by the simulation are depicted in Fig. 5 for $S=10$, $N=5$, $\lambda/\mu = 5$, and $\mu = .01$. Each plotted point represents an average of 4 runs, with each run comprising 5000 transitions in $n_v(t)$, or about 50,000 sec of real time. Actually, the average buffer size $\langle n_d \rangle$ was measured, and (8) used to obtain W_d. For example, at $\theta b = 9$, $W_d \approx 10$ sec and $\langle n_d \rangle \sim 9000$ packets. In each run, the maximum as well as the average of $n_d(t)$ was measured. Generally $(n_d(t))_{max} >> \langle n_d \rangle$, as indicated in Table 1. (Packet size has been taken as 80 bits as in the previous analysis.[1]) Enough storage must be allocated to handle $(n_d(t))_{max}$. For example, with $\theta b = 9.0$, about 800,000 words of storage would be needed to prevent data buffer overflow.

Results were presented previously[1] for data waiting times for the same values of N, S, and λ/μ as utilized in obtaining Fig. 5. Assuming $b = 10$ ms, the predicted waiting time was $W_d \approx 20$ ms for $\theta b = 9$. This predicted delay is about a factor of 500 below the simulation results obtained here.

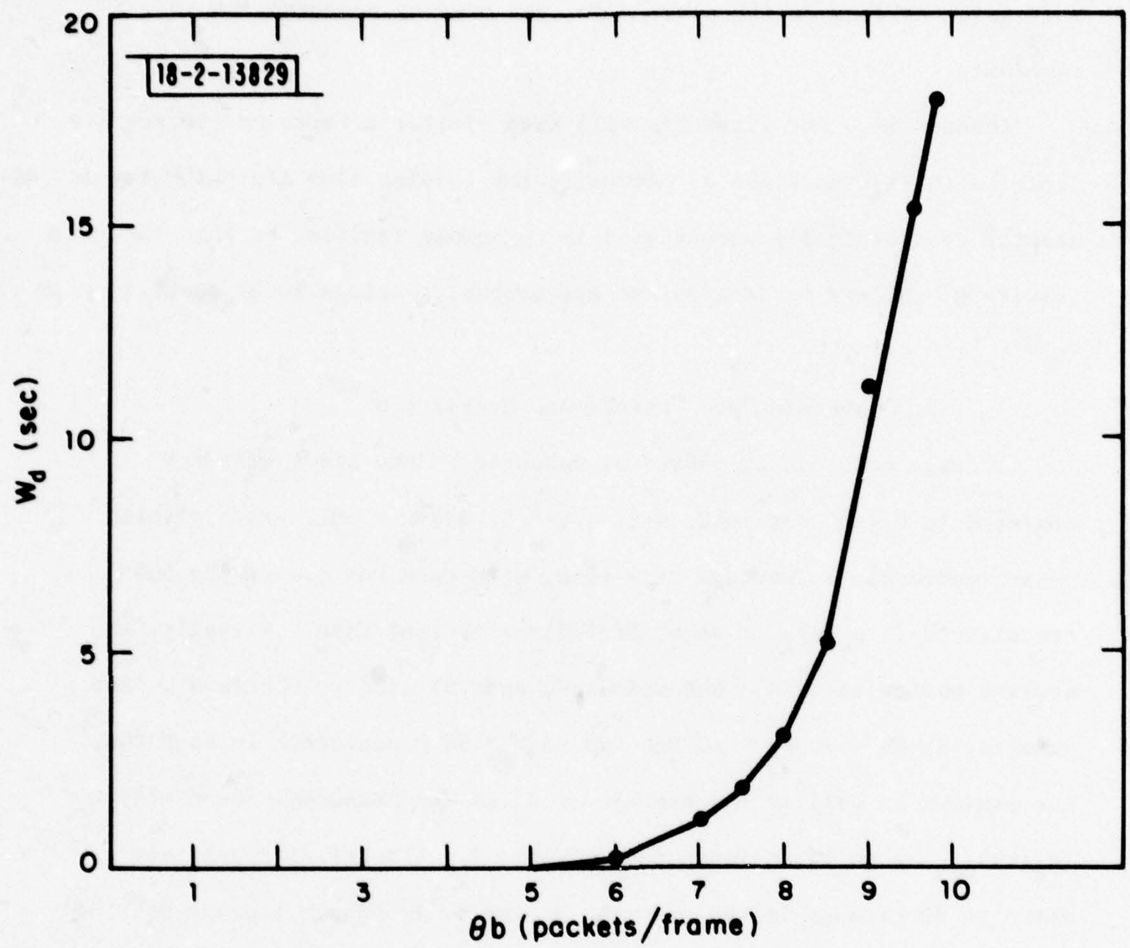

Fig. 5. Average data packet delays as function of data load for S = 10, N = 5, $\lambda = 0.05\ sec^{-1}$, $\mu = 0.01\ sec^{-1}$. Each plotted point represents an average of 4 runs, with each run comprising 5000 transitions in $n_v(t)$ and about 50,000 sec of real time.

Average data packet delays for $S = 50$, $N = 25$, $\lambda/\mu = 40.0$ and $\mu = .01$ are shown in Figure 6. Delays are slightly less than in Fig. 5 for the same values of $\overline{\rho_d} = \theta b/(N + S - \langle n_v \rangle)$, partially because the larger value of λ/μ implies that transitions in $n_v(t)$ occur at a faster rate. However, the larger values of θ imply that buffer sizes are similar to those shown in Table 1. Again, for $\overline{\rho_d} > .9$, average waiting times on the order of 10 sec are exhibited.

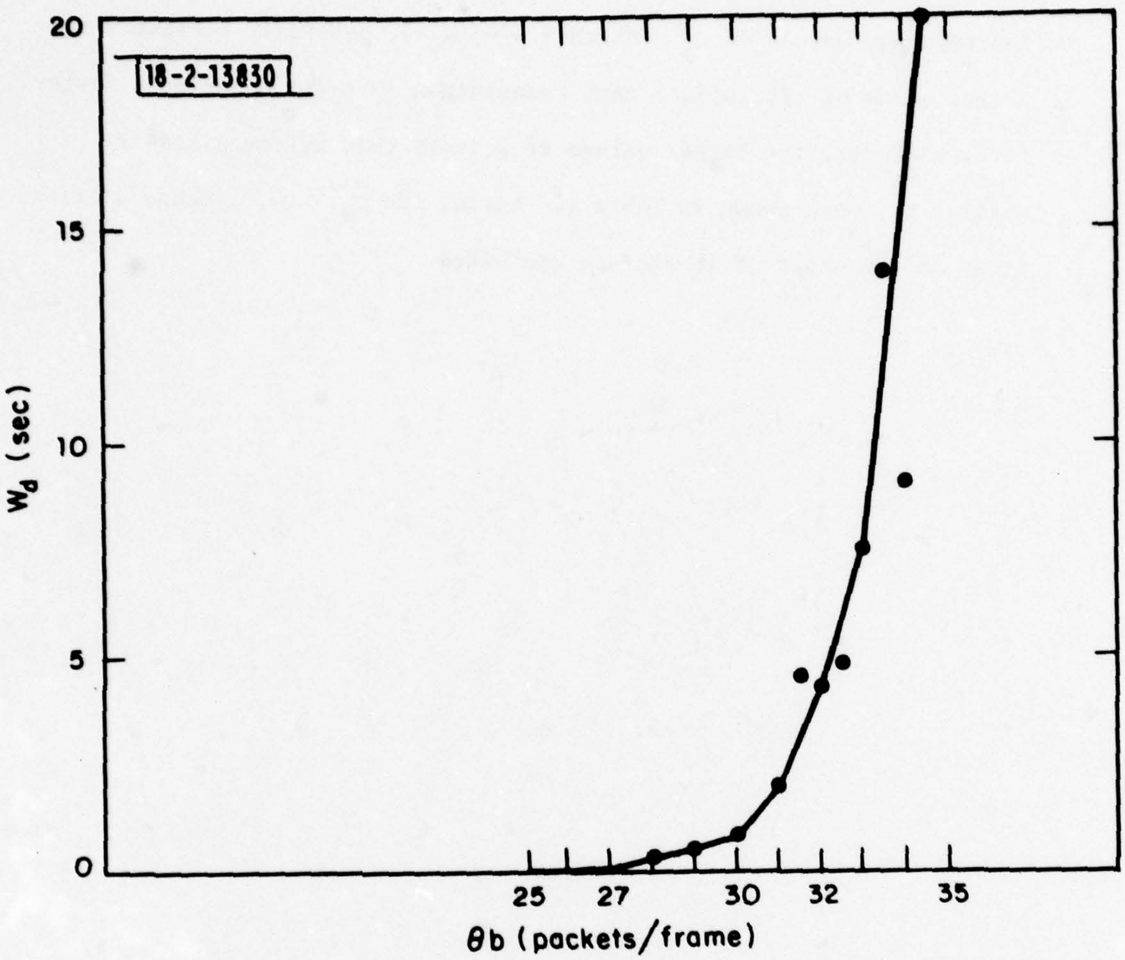

Fig. 6. Average data packet delays as function of data load for S = 50, N = 25, λ = 0.4 sec^{-1}, and μ = 0.01 sec^{-1}. Each plotted point represents an average of 2 to 4 runs, with each run comprising 5000 transitions in $n_v(t)$, and about 6500 sec of real time.

TABLE 1

AVERAGE AND MAXIMUM DATA BUFFER SIZE, IN THOUSANDS OF 16-BIT WORDS, AS FUNCTION OF DATA LOAD

[Each entry represents an average of $\langle n_d \rangle$ or $(n_d)_{max}$ over 4 runs, with each run comprising 5000 transitions in $n_v(d)$ and about 50,000 sec of real time.]

θb (packets/frame)	Average Data Buffer (K words)	Maximum Data Buffer (K words)
5	0.002	1.2
6	0.55	42
7	4.16	147
7.5	7.24	189
8	12.3	242
8.25	21.2	380
8.5	22.4	343
8.75	34.5	484
9.0	52.4	798
9.25	54.2	575
9.5	73.9	904
9.75	87.9	846

III. MODIFIED ANALYTICAL MODEL

A. Model and Notation

If one ignores the effect of the time quantization introduced by the frame structure and assumes exponentially-distributed rather than constant data packet lengths, the entire system can be analyzed as a two-dimensional Markov chain. It is useful to introduce new notation where the voice customers are denoted by class 1 and the data customers are class 2. Assume the arrival process for each is Poisson with rate $\lambda_i (i = 1,2)$ and the service time is exponential with rate μ_i. Let Q_i denote the number of class i customers in the system and P_{ij} represent the steady-state probability that $Q_1 = i$ and $Q_2 = j$. Referring to the simulation model described above, λ_1 corresponds to λ, μ_1 to μ, λ_2 to θ, μ_2 to b^{-1}, Q_1 to n_v, and Q_2 to n_d.

There are S + N server channels with voice calls having priority on S of the channels. If a voice call shows up and $Q_1 < S$, the call will seize a free channel if there is one, otherwise it will preempt a data packet which returns to the head of the data queue. No queueing is allowed for voice. In Appendix II, a general set of two-dimensional difference equations for P_{ij} is written, and a solution procedure is outlined. The performance parameter of interest is the expected waiting time for data packets, which is related to P_{ij} according to

$$W_d = \frac{1}{\lambda_2} E[Q_2] = \frac{1}{\lambda_2} \sum_{i=0}^{S} \sum_{j=0}^{\infty} j \, P_{ij} . \qquad (9)$$

B. Single-Channel Example

For the special case of one voice channel and no data channels (S = 1, N = 0), a reasonably simple expression for W_d emerges, in the form

$$W_d = \frac{\rho_2(1 + \rho_1)^2 + \rho_1 \lambda_2/\mu_1}{\lambda_2(1 + \rho_1)(1 - \rho_2 - \rho_1\rho_2)} \qquad (10)$$

where $\rho_1 = \lambda_1/\mu_1$ and $\rho_2 = \lambda_2/\mu_2$. The factor λ_2/μ_1 in the numerator represents the expected number of data packets to arrive during an average voice holding time. This factor can be quite large and accounts in large part for the very large data delays noted in the simulation. Applying (10) to the case illustrated in Fig. 4 where S = 1, N = 0, $\lambda_1 = .01$, $\mu_1 = .01$, $\lambda_2 = 40$, and $\mu_2 = 100$ results in $W_d \approx 250$ sec. This analytic result, based on exponentially-distributed packet lengths, is reasonably consistent with the result of the simulation, where with constant data packet lengths W_d was estimated at 176 sec.

IV. INVESTIGATION OF FLOW CONTROL TECHNIQUES

The conclusion that can be drawn from the simulation and analytic results shown above is that attempts to achieve high channel utilization can lead to data packet queues so large as to overflow any reasonable amount of storage in the multiplexer. Even assuming infinite storage, the mean data delays are large because of the buffer buildup during periods when voice channel occupancy is high. The need for a flow-control mechanism is apparent, and two types of flow control have been investigated: (1) voice rate control, where voice coders are assumed to be capable of operating at a variety of rates and a new voice call is assigned (at dial-up) one of several available bit rates, based on the current voice utilization and/or data queue size; and (2) data flow control, where a fixed limit is imposed on the size of the data queue.

A set of simulations has been run to gather statistics on multiplexer performance with different flow-control strategies. Generally total channel capacity was fixed at 120 kbps, with voice capacity limited to 80 (sufficient to accommodate $N = 10$ users at the nominal 8 kbps rate), and 40 kbps dedicated to data. For comparison, one of the flow control schemes was tested with 600 kbps channel. A movable boundary was employed to allow data packets to be sent in temporarily unused portions of the voice capacity. The voice traffic was modelled as above, with λ set at .05 sec^{-1} and $\mu = .01$ sec^{-1}, corresponding to 5 Erlangs of offered voice traffic. The maximum number of simultaneous voice calls was limited to 10, with blocked calls cleared. Data packets, with fixed lengths of 80 bits, were assumed to arrive in a Poisson process. The purpose of the simulations was to investigate system performance for data packet arrival rates greater than 500 per second (or equivalently 5.0 per frame), when a portion of the voice capacity must be utilized for data. For the cases where the data packet arrival rate is less than 500 per second no large data queues build up. The performance measures of interest for data include average delay, average queue size, maximum queue size, and fraction of packets arriving when the data queue is full (when data flow control is employed). With voice rate control, the distribution of callers among the available bit rates and the average voice bit rate assigned serve as performance measures for voice users.

A. Experiments with Voice Rate Control but No Data Flow Control

Up to now, all voice users were assumed to operate at 8 kbps. The voice channel utilization exhibited large variations around its mean, and data queues would build up during the peaks of voice utilization. The idea behind the voice rate control techniques studies here was to cut down the peaks of voice channel utilization by assigning lower bit rates to callers who enter the system when utilization is high. This scheme of course assumes that each voice user has a flexible vocoder capable of a variety of rates. The presumption is that the assigned voice bit rate and the corresponding speech coder performance will vary depending on the traffic load in the system.

The results for four voice rate control schemes are reported here. To define these schemes, the following notation is introduced:

R_V = sum of bit rates of all voice users active at a given time;

C_V = maximum channel capacity allocated for voice;

Q_d = number of packets currently in the data queue;

R_{new} = bit rate assigned to a new voice caller at dial-up.

Unless otherwise noted, C_V was fixed at 80 kbps. The first voice rate control scheme that was tried is illustrated in Fig. 7. New calls were assigned at 2, 4, 8 or 16 kbps as a function of current total voice channel utilization R_V/C_V, with lower rates assigned during higher utilization periods. The inclusion of a 16 kbps rate will provide better voice service for some users, but has the effect of filling in the valleys of voice utilization that are present when voice rate control was not employed. The second scheme was similar to the first except that the voice rate of 16 kbps was not used, but 8 kbps was assigned to callers arriving when R_V/C_V was less than .25. The third approach used voice rate control combined with monitoring of the data queue was included. When the data queue was empty, voice rates were assigned as in Fig. 7; however voice callers arriving when the data queue was not empty were assigned a rate of 2 kbps. In the fourth technique, the rate of a new call was governed solely by the size of the data queue. If the queue was empty, a rate of 8 kbps was assigned. If the queue was non-empty but did not exceed 150 packets, 4 kbps was assigned. Otherwise, 2 kbps was used. The four approaches just outlined will be referred to below as schemes v1, v2, v3 and v4; v0 will denote the situation where no control is imposed. These voice-rate control schemes are summarized as follows:

v0: R_{new} = constant = 8 kbps

$$\text{v1:}\quad R_{new} = \begin{cases} 16 \text{ kbps} & R_V/C_V \le .25 \\ 8 \text{ kbps} & .25 < R_V/C_V \le .5 \\ 4 \text{ kbps} & .5 < R_V/C_V \le .75 \\ 2 \text{ kbps} & .75 < R_V/C_V \le 1.0 \end{cases}$$

$$\text{v2:}\quad R_{new} = \begin{cases} 8 \text{ kbps} & R_V/C_V \le .5 \\ 4 \text{ kbps} & .5 < R_V/C_V \le .75 \\ 2 \text{ kbps} & .75 < R_V/C_V \le 1.0 \end{cases}$$

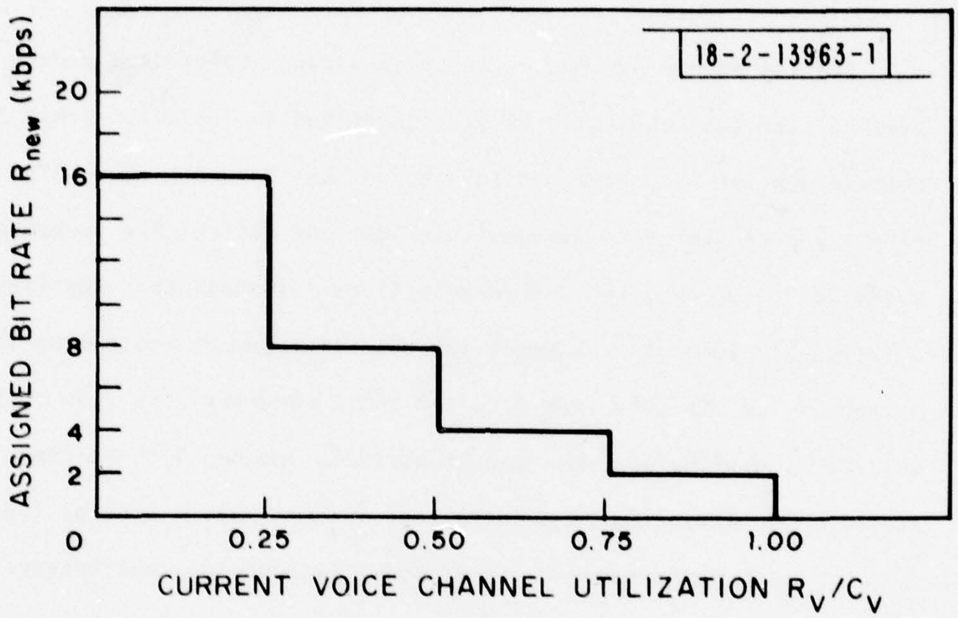

Fig. 7. Example of voice-rate control based on current voice channel utilization. The bit rate assigned to a new user is plotted as a function of the voice channel utilization R_V/C_V.

$$v3: \quad R_{new} = \begin{cases} \text{Assigned as in v1} & Q_d = 0 \\ 2 \text{ kbps} & Q_d \neq 0 \end{cases}$$

$$v4: \quad R_{new} = \begin{cases} 8 \text{ kbps} & Q_d = 0 \\ 4 \text{ kbps} & 0 < Q_d \leq 150 \\ 2 \text{ kbps} & Q_d > 150. \end{cases}$$

Results on average packet delay as a function of data packet arrival rate for schemes v0-v4 are summarized in Table 2. Table 3 shows the average bit rate assigned to a voice user for each case. The first scheme represents an improvement over no flow control for packet arrival rates Θb = 6.0, 7.0, and 8.0 packets/frame. (Recall that the frame duration b = 10 ms.) The price for this improvement was a drop in average voice bit rate from 8 to 5.7 kbps. However, the flow control worsened the situation when packet arrivals reached 9.0 per frame. When 16 kbps was eliminated as a voice rate (scheme v2), a general improvement in delay performance was realized. In addition, the average voice bit rate increased to 6.6 kbps because fewer users had to be assigned to the lower rates. Scheme v3 (the same as v1 except that some data queue monitoring was introduced) represented a significant improvement over v1 in terms of data delay. Scheme v4, where voice rate control was based strictly on data queue size, was generally the most effective of the voice rate control techniques.

However, none of the performance results shown in Table 2 are really satisfactory at the high packet arrival rates. Even for scheme v4, the delay of .63 sec at 9.0 packets/frame is unacceptable, and corresponds to an average queue size of 571 packets. Observed maximum queue sizes were significantly higher. The conclusion is that, although voice rate control alone can enhance data performance, some form of data

TABLE 2

AVERAGE DATA PACKET DELAY AS A FUNCTION OF DATA PACKET ARRIVAL RATE FOR VOICE-RATE CONTROL SCHEMES (v0 - v4 as described in text)

Packet Arrival Rate (θb in packets/frame) \ Flow Control Scheme	v0	v1	v2	v3	v4
	Average Delay (sec)				
6	0.18	0.0	0.0	0.0	0.0
7	1.18	0.0	0.0	0.0	.12
8	3.08	1.19	.14	.33	.30
9	11.6	39.0	1.6	2.55	.63

TABLE 3

AVERAGE BIT RATE ASSIGNED TO VOICE USERS FOR VOICE-RATE CONTROL SCHEMES v0 - v4

Packet Arrival Rate (θb in packets/frame) \ Flow Control Scheme	v0	v1	v2	v3	v4
	Average Voice Bit Rate (kbps)				
6	8.0	5.7	6.6	6.0	7.9
7	8.0	5.7	6.6	6.0	7.6
8	8.0	5.7	6.6	5.5	7.2
9	8.0	5.7	6.6	4.8	6.5

flow control is necessary to keep delays and queue sizes within reasonable limits.

The emphasis here has been on the effectiveness of voice rate control in reducing data packet delay. However, the ability of voice users to operate at a variety of rates depending on traffic loads would enhance the voice-traffic handling capability of the system, independent of its effect on data traffic. For example, the assignment of lower bit rate to new voice users as the number of active calls increases, could be utilized as a means for reducing the blocking probability for voice calls during busy periods.

B. Experiments with Combined Data Flow Control and Voice Flow Control

The first experiment including data flow control involved limiting the data buffer to a fixed maximum size Q_{max}, but not including any voice rate control. Q_{max} was (somewhat arbitrarily) set equal to 150 packets. Data packets were denied entry to the multiplexer when its queue was full. This represents added delay since these packets have to be retransmitted to the multiplexer. If the packets enter the multiplexer directly from a user terminal, then the terminal would retransmit when no acknowledgement was received. Otherwise, a store-and-forward node feeding the multiplexer could handle the retransmission. The second experiment combined the same data flow control procedure with the data-queue-independent voice rate control employed in scheme v2 above. The third test combined limitation of the data buffer to 150 packets with data-queue-dependent voice rate control similar to the method used in scheme v4 above. If the queue was empty, a rate of 8 kbps was assigned. If the queue was non-empty but did not exceed 75 packets, 4 kbps assigned. Otherwise, 2 kbps was used. The three combined data/voice flow control techniques just described will be referred to as d1, d2, and d3, and are summarized as follows:

d1, d2, d3: all impose fixed limit $Q_{max} = 150$ packets on data buffer size.

d1: $R_{new} =$ constant = 8 kbps

$$\text{d2:}\quad R_{new} = \begin{cases} 8 \text{ kbps} & R_V/C_V \le .5 \\ 4 \text{ kbps} & .5 < R_V/C_V \le .75 \\ 2 \text{ kbps} & .75 < R_V/C_V \le 1.0 \end{cases}$$

$$\text{d3:}\quad R_{new} = \begin{cases} 8 \text{ kbps} & Q_d = 0 \\ 4 \text{ kbps} & 0 < Q_d \le Q_{max}/2 \\ 2 \text{ kbps} & Q_{max}/2 < Q_d \le Q_{max}. \end{cases}$$

The average packet delays for the three cases are plotted in Fig. 8. In all cases, the improvement in data packet delay is dramatic as compared to the situation where no data flow control is imposed. For example, the worst case in Fig. 8 is an average delay of 49 msec for scheme d1 at 9.0 packets/frame arrival rate. The corresponding best result without data flow control is an average delay of 630 msec for v4 as shown in Table 2.

Tables 4 and 5 show the average voice bit rates and the percentages of data packets arriving when the queue is full, as a function of packet arrival rate for the three data flow control schemes. The tables include some measurements for packet arrival rates greater than 9.0 packets/frame. Table 5 shows that the percentage of packets finding a full data queue is rather low in all cases. This implies that the additional delay seen by the user who must retransmit these packets also ought to be modest.

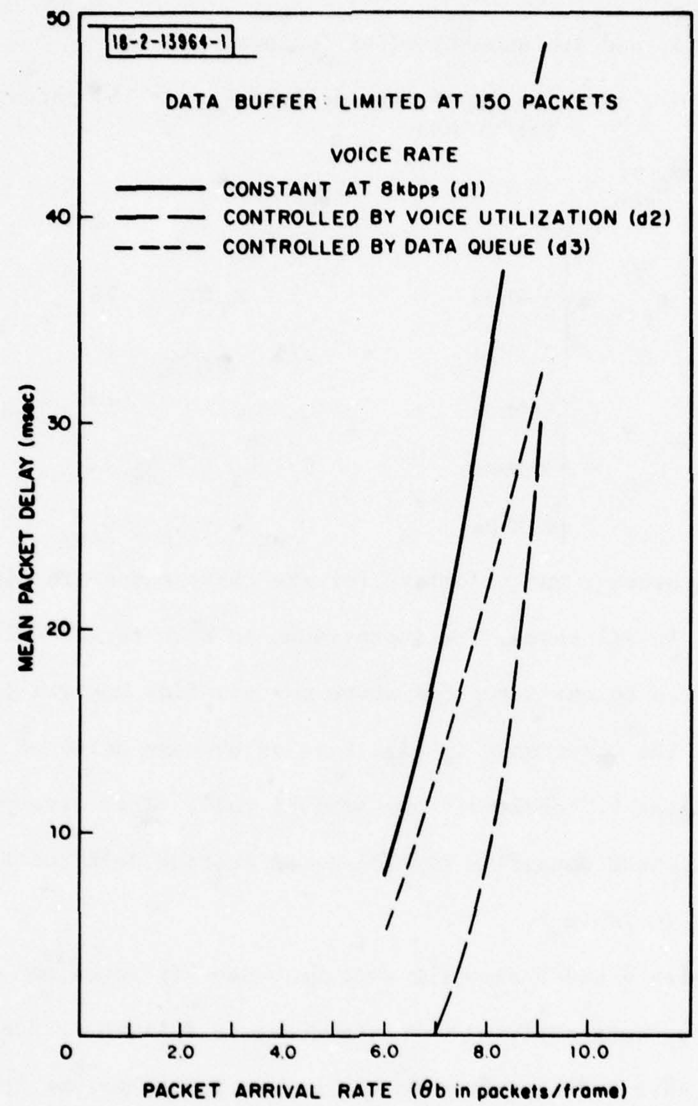

Fig.8. Mean data packet delay as a function of data packet arrival rate for different flow control experiments d1-d3 described in the text.

TABLE 4

AVERAGE BIT RATE ASSIGNED TO VOICE USERS FOR FLOW CONTROL SCHEMES d1-d3			
Flow Control Scheme / Packet Arrival Rate (θb in packets/frame)	d1	d2	d3
	Average Voice Bit Rate (kbps)		
6.0	8.0	6.6	7.9
7.0	8.0	6.6	7.7
8.0	8.0	6.6	7.3
9.0	8.0	6.6	6.8
9.5	8.0	6.6	6.4
10.0		6.6	6.0
10.5		6.6	5.5
11.0			4.5

TABLE 5

PERCENTAGE OF PACKETS ARRIVING WHEN QUEUE IS FULL UNDER FLOW CONTROL SCHEMES d1-d3

Packet Arrival Rate (θb in packets/frame) \ Flow Control Scheme	d1	d2	d3
	Packets Arriving When Data Queue is Full (percent)		
6.0	0.3	0.0	0.1
7.0	1.1	0.0	0.4
8.0	2.4	0.2	0.7
9.0	4.2	1.4	1.1
9.5	6.8	2.8	1.5
10.0		4.6	1.8
10.5		6.9	2.5
11.0			3.3

If schemes d1-d3 are compared on the basis of data packet delay alone, it is clear from Fig. 8 that the best performance is achieved with d2, which includes voice rate control independent of the data queue. However, the improvement in delay performance over d3 is counterbalanced by the fact that higher average voice bit rates were assigned in scheme d3 at all data packet arrival rates tested.

Thus far, performance of the three data flow control schemes has been displayed on the basis of packet arrival rate for convenient comparison with the voice rate control schemes. In order to take into account the effects of packet discard due to data flow control, and to provide a convenient means for comparing systems with different capacity, it is useful to present the results as a function of utilization. It is also desirable to focus on utilization of the variable capacity available to data due to fluctuations in voice traffic, rather than on utilization of fixed channel capacity dedicated to data. A correspondence between utilization of variable data capacity and packet arrival rate is shown in Table 6. Variable data capacity is defined as the difference between the total capacity allotted for voice (80 kbps) and the average capacity actually utilized by voice. The average packet delays for the three schemes as a function of variable data capacity utilization are plotted in Fig. 9. It will be noticed that scheme d3 performs more favorably than scheme d2 in terms of packet delay as utilization exceeds 70%. Similarly, the percentages of data packets finding the queue full, which are plotted in Fig. 10, show that d3 again has the best performance for higher utilization of variable data capacity. However, there is a price to be paid by using scheme d3. As utilization is increased by greater packet arrival rates, the data queue

TABLE 6

PERCENT UTILIZATION OF VARIABLE CAPACITY AVAILABLE FOR DATA FOR FLOW CONTROL SCHEMES d1-d3			
Flow Control Scheme / Packet Arrival Rate (θb in packets/frame)	d1	d2	d3
	Utilization of Variable Data Capacity (percent)		
600	19	18	19
700	39	37	37
800	55	55	55
900	70	71	70
950	78	78	76
1000		84	81
1050		88	87
1100			92

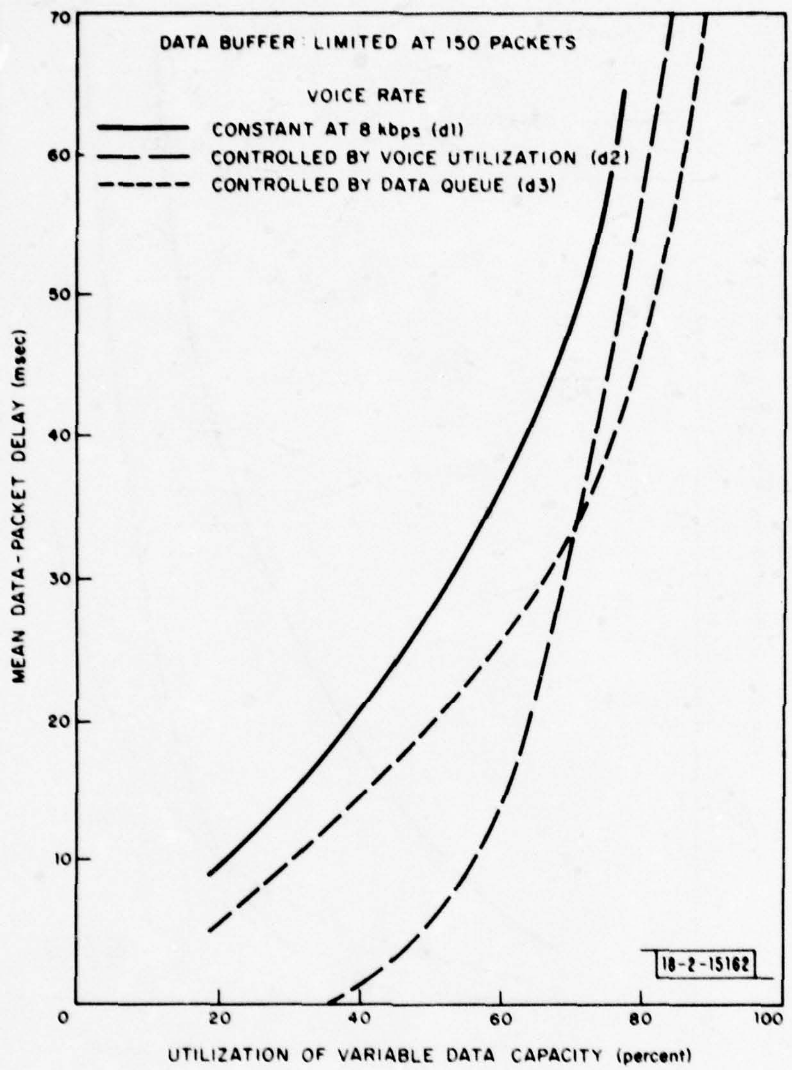

Fig.9. Mean packet delay as a function of utilization of variable data capacity for schemes d1-d3.

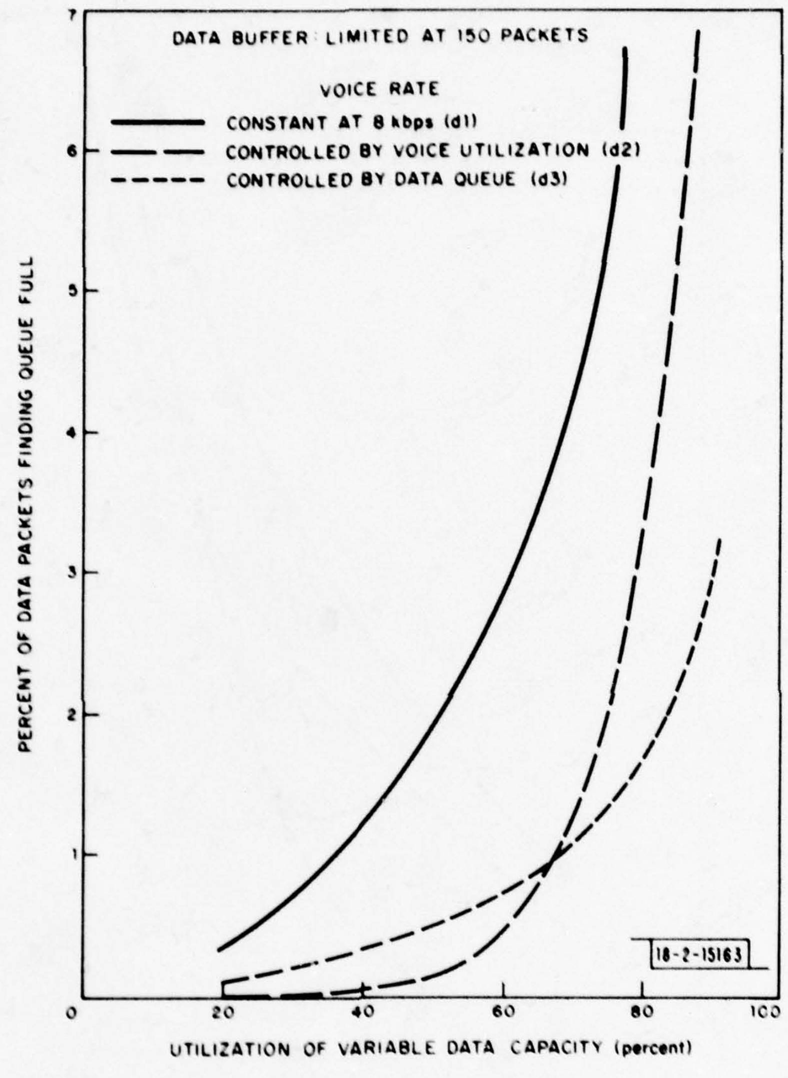

Fig.10. Percent of packets finding queue full as a function of utilization of variable data capacity for schemes d1-d3.

tends to remain more than half full during periods of higher voice utilization. Consequently, more calls are assigned at 2 kbps; and, lest the average voice bit rates shown in Table 4 be misleading, it should be mentioned that for all packet arrival rates tested almost all calls were assigned at either 8 kbps or 2 kbps using scheme d3. This is in contrast to scheme d2 where all calls were assigned at 8 kbps or 4 kbps with a ratio of slightly less than two to one, regardless of packet arrival rate. If voice users could tolerate a greater overall incidence of 2 kbps voice communication with a better chance of being assigned at 8 kbps, it can be argued that scheme d3 is slightly better than scheme d2. The percentages of calls assigned at 8 kbps are plotted in Fig. 11. The area of particular interest is that corresponding to a utilization of variable data capacity just over 70% which is where scheme d3 shows improvement over scheme d2 in both Figs. 9 and 10. In general, comparable performances could be achieved with both schemes d2 and d3.

Scheme d3 was tested for a larger population of users by increasing the channel capacity from 120 kbps to 600 kbps, with a voice capacity of 400 kbps and a dedicated data capacity of 200 kbps. The offered voice traffic for the larger population was set at 40 Erlangs. (With 50 8 kbps channels of voice capacity and 40 Erlangs of traffic, the blocking probability computed on the basis of fixed voice rate operation at 8 kbps is P_L = .0186, approximately the same as the blocking probability for the 10-channel, 5 Erlang case considered up to now.) The performance of d3 with this larger population was compared to that with the smaller population considered above, where 80 kbps was allocated for voice and the offered voice traffic was 5 Erlangs. As shown in the plots of average data packet delay, percentages of packets finding the queue

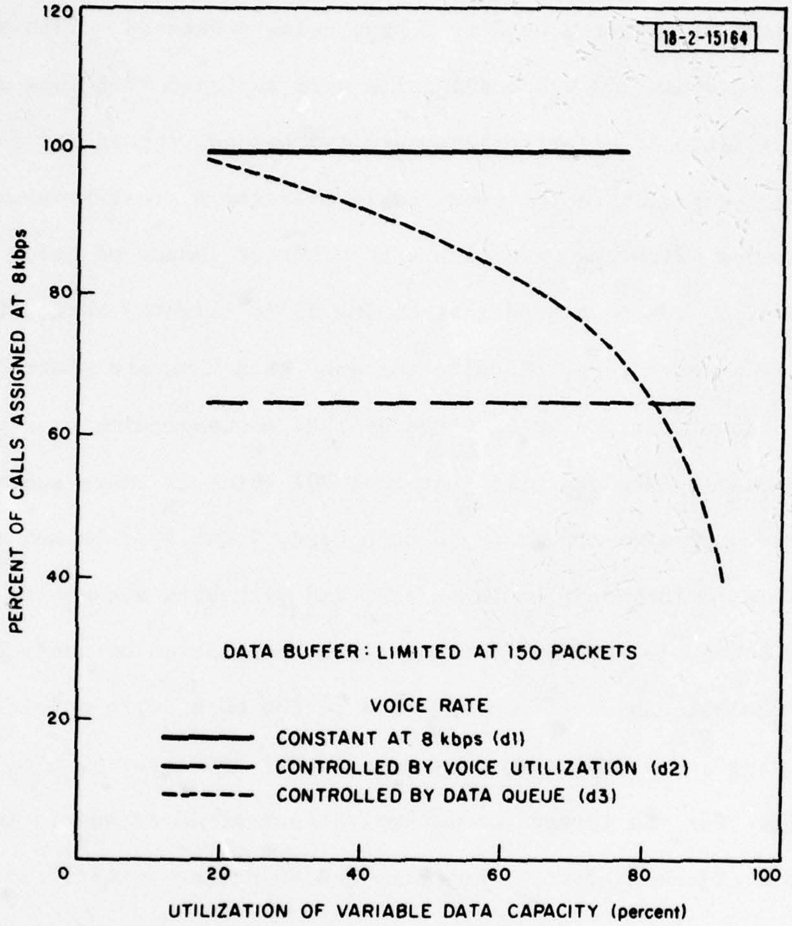

Fig.11. Percent of calls assigned at 8 kbps as a function of utilization of variable data capacity for schemes d1-d3.

full, and the percentages of calls assigned at 8 kbps in Figs. 12, 13, and 14, respectively, the performance was improved for the larger population. This can be attributed in part to the more rapid fluctuation in voice utilization, which causes the peak of voice utilization to be shorter in duration.

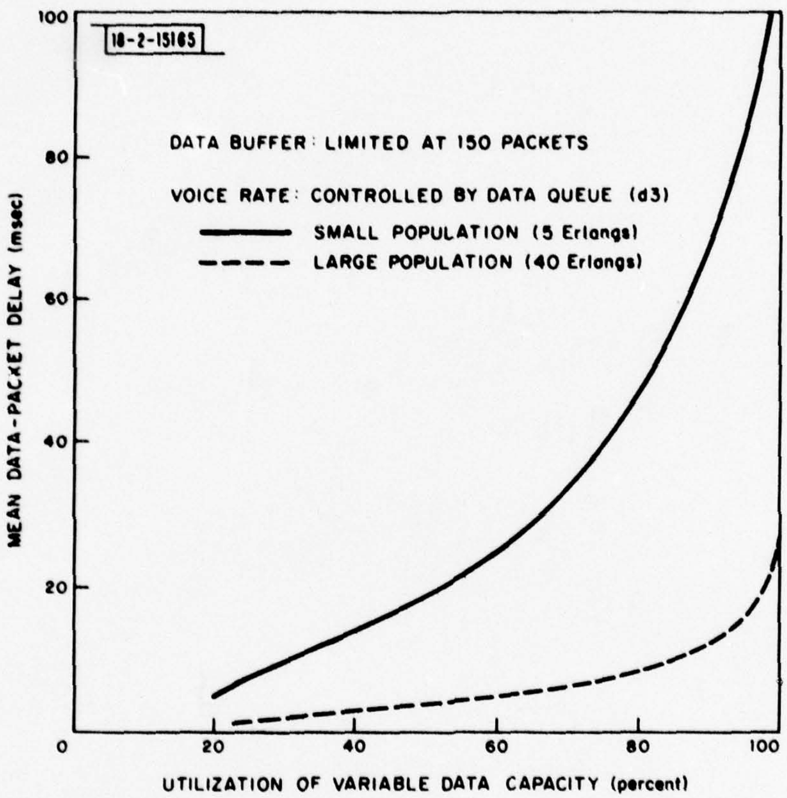

Fig.12. Mean packet delay for different user populations under flow control scheme d3.

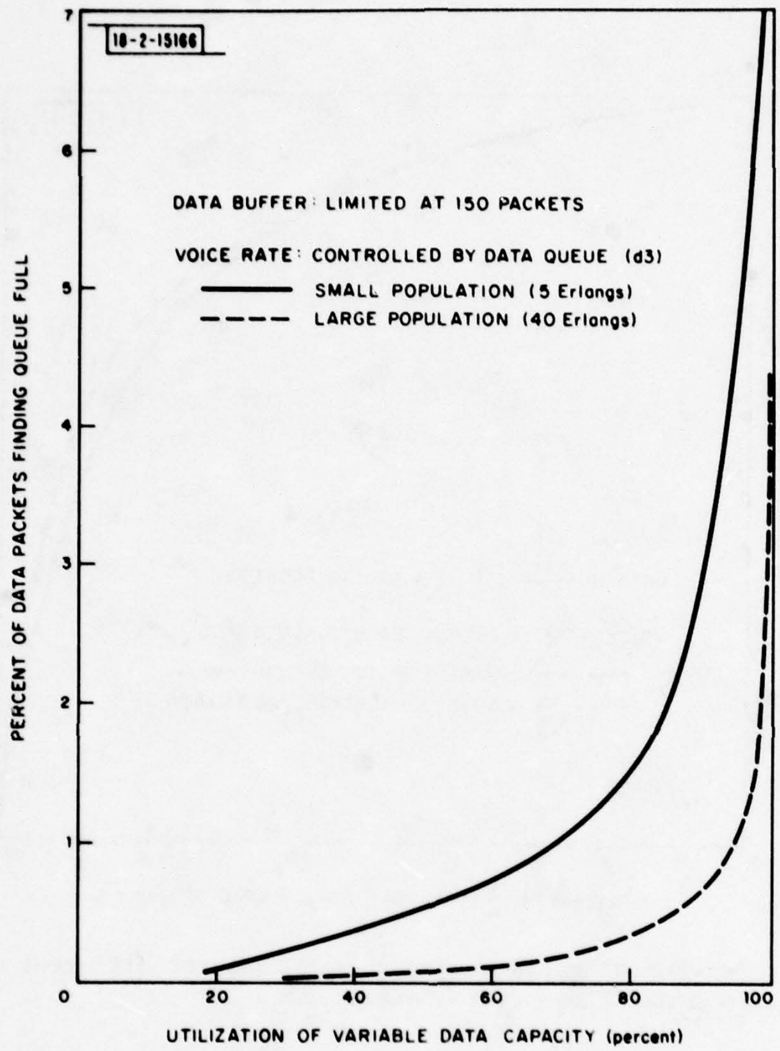

Fig.13. Percent of packets finding queue full for different user populations under flow control scheme d3.

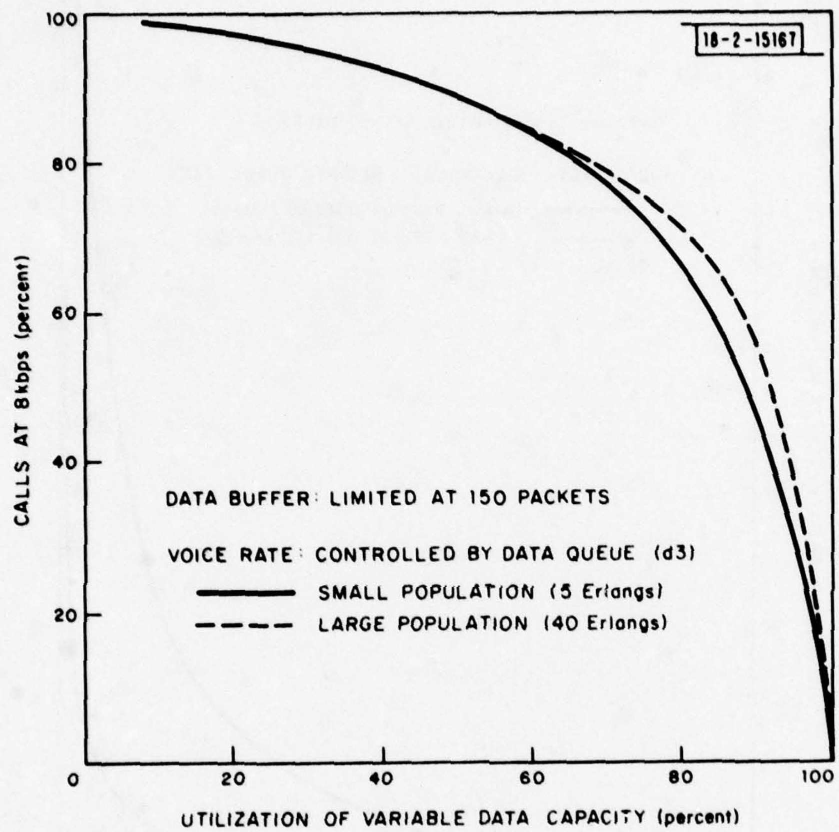

Fig.14. Percent of calls assigned at 8 kbps for different user populations under flow control scheme d3.

V. CONCLUSIONS

An error in a previous analysis has been identified, and modified results have been presented for the data packet delay performance of an integrated voice/data multiplexer structure. The new results were developed both through a simulation and analysis, and showed that excessive data queues and delays could build up under heavy loading conditions. A variety of flow control mechanisms to prevent these large delays were investigated. These mechanisms included voice rate control, limitation of the data buffer, and combinations of voice and data flow control. The best results were obtained with limitation of the data buffer combined with data-queue-dependent voice rate control.

APPENDIX I. Error in Previous Analysis

The analysis at issue was developed in Equations (9)-(13) of [1]. Q_n^V and Q_n^D were defined as the number of occupied voice channels and number of data customers, respectively, in the system just after the nth gate opening (where gate openings are time instants occurring at b = 10 ms intervals when data and voice customers are admitted to the system). An expression was written for the z-transform of the conditional probability distribution

$$\Pr\,[Q_{n+1}^D = i \mid Q_n^D = j,\ Q_n^V = S - k]$$

and this probability was "unconditioned" in two stages according to

$$\Pr[Q_{n+1}^D = i]$$

$$= \sum_k \Pr[Q_n^V = S-k] \sum_j \Pr[Q_n^D = j] \Pr[Q_{n+1} = i \mid Q_n^D = j,\ Q_n^V = S - k] \quad \text{(AI.1)}$$

However this unconditioning procedure would be valid only if Q_n^D and Q_n^V were statistically independent. One should not assume such independence to hold, particularly since large Q_n^D tends to be correlated with large Q_n^V. The correct unconditioning relation is

$$\Pr[Q_{n+1}^D = i]$$

$$= \sum_{k,j} \Pr[Q_n^V = S-k,\ Q_n^D = j] \Pr[Q_{n+1} = i \mid Q_n^D = j,\ Q_n^V = S-k] \quad \text{(AI.2)}$$

but it is difficult to proceed beyond (AI.2) since the required joint probability distribution of Q_n^V and Q_n^D is unknown. In proceeding from (AI.2), it was assumed that $Pr[Q_n^v = k] = Pr[Q^v = k]$ for all n. This assumption, together with the incorrect unconditioning procedure, has the effect of treating Q_n^v as a random process which is independent between adjacent 10 msec frames. Actually, as discussed above and illustrated for example in Fig. 2, Q_n^v changes only with call initiation or termination and is very highly correlated across frame-length time intervals.

APPENDIX II. Description of Modified Analytic Approach

In order to develop an analytic model for the SENET multiplexer structure, as defined in [2], we need to assume exponentially distributed rather than constant data packet lengths and ignore the effect of time quantization introduced by the frame structure. Under these assumptions the system can be modelled as a two-dimensional Markov chain. The solution then is found by the use of generating functions. This method is standard for such problems and has been used on problems similar to the one addressed here.[4,5,6,7,8] In fact Chang[8] has analyzed the same system as the one addressed here. We include here a formulation of the two-dimensional difference equations for the system state probabilities and a discussion of the general solution procedure for the infinite buffer case. An explicit solution is presented for the single-channel case. Finally, results of direct numerical solution of the system difference equations for the finite buffer case are presented and compared with simulation results.

The difference equations for the system state probabilities are formulated as follows. Let class 1 be voice and class 2 be data. We assume that the arrival process for each is Poisson with parameter λ_i, $i = 1,2$; and that the service distribution for each class is exponential with rate μ_i. If the service and arrival distributions are independent and Q_i is the steady state number of class i customers in the system, let $\Pr\{Q_1 = i,\ Q_2 = j\} = P_{i,j}$. There are S + N channels with S set aside for the voice calls (class 1), with priority. In other words if a voice call shows up and there are less than S voice calls occupying the channels the call will seize a free channel if there is one, otherwise it will preempt a data customer, who might be using one of them. The preempted data customer returns to the buffer, and is serviced on a first-come, first-served. Arriving data customers can use any free channel for the special case where $N = 0$ (see [7]). We assume that there are M buffer spaces available for the data customers. The steady state equations for $P_{i,j}$ are for $i = 0,1,\ldots,S - 1$

$$(\lambda_1 + \lambda_2 + i\mu_1 + j\mu_2)P_{i,j} = \lambda_1 P_{i-1,j} + (i + 1)\mu_1 P_{i+1,j}$$

$$+ \lambda_2 P_{i,j-1} + (j + 1)\mu_2 P_{i,j+1}: \quad j = 0,1,\ldots, N + S - i - 1 \qquad \text{(AII.1a)}$$

$$(\lambda_1 + \lambda_2 + i\mu_1 + (N + S - i)\mu_2)P_{i,j} = \lambda_1 P_{i-1,j}$$

$$+ (i + 1)\mu_1 P_{i+1,j} + \lambda_2 P_{i,j-1} + (N + S - i)\mu_2 P_{i,j+1}: \qquad \text{(AII.1b)}$$

$$N + S - i \leq j \leq M + N + S - i - 1$$

and for $j = M + N + S - i$

$$(\lambda_1 + i\mu_1 + (N + S - i)\mu_2)P_{i,j} = \lambda_1 P_{i-1,j} + (i + 1)\mu_1 P_{i+1,j}$$

$$+ \lambda_2 P_{i,j-1} \,. \qquad \text{(AII.1b')}$$

For $i = S$, we have

$$(\lambda_2 + S\mu_1 + j\mu_2)P_{S,j} = \lambda_1 P_{S-1,j} + \lambda_2 P_{S,j-1}$$

$$+ (j + 1)\mu_2 P_{S,j+1}: \quad j = 0,1,\ldots,N - 1 \; ; \qquad \text{(AII.2a)}$$

$$(\lambda_2 + S\mu_1 + N\mu_2)P_{S,j} = \lambda_1 P_{S-1,j} + \lambda_2 P_{S,j-1} + N\mu_2 P_{S,j+1}:$$

$$N \leq j \leq M + N - 1; \qquad \text{(AII.2b)}$$

and for $j = M + N$

$$(S\mu_1 + N\mu_2)P_{S,j} = \lambda_1 P_{S-1,j} + \lambda_2 P_{S,j-1} \, . \qquad \text{(AII.2b')}$$

In these equations $P_{-1,j} \equiv P_{S+1,j} \equiv P_{i-1} \equiv P_{i,M+N+S+1-i} = 0$, and when $N = 0$ equation (AII.2a) is meaningless.

The solution of this system of equations can be carried out formally using transform methods for the case where $M = \infty$. The procedure is to define

$$P_i(z) = \sum_{j=0}^{\infty} P_{i,j} z^j \qquad i = 0,1,\ldots,S \qquad \text{(AII.3)}$$

and use AII.1 and AII.2 to develop a set of linear equations of the form

$$A(z)\; P(z) = B(z) \qquad \text{(AII.4)}$$

where $A(z)$ is an $(S + 1) \times (S + 1)$ matrix depending on the known system parameters λ_1, λ_2, μ_1, μ_2, N, and S; $P(z)$ is an $(S + 1) \times 1$ vector with components $P_0(z)$, $P_1(z), \ldots, P_S(z)$; and $B(z)$ is an $(S + 1) \times 1$ vector which depends on the unknown quantities $P_{i,j}$ for $i = 0,1,\ldots,S$ and $j = 0,1,\ldots,S$ and $j = 0,1,\ldots,N + S$. Equation (AII.1a) can be used to evaluate recursively these $P_{i,j}$ so that $B(z)$ depends only on the $N + S$ unknowns, $P_{0,j}$ for $j = 0,1,N + S - 1$. To determine these unknowns, $N + S$ equations are needed. One equation can be found by the condition that the carried load must equal the expected number of busy servers.[9] Another $N - 1$ equations can be obtained from (AII.2a). The remaining equations are obtained by finding the S unique roots[4] in (0,1). Then we require that $\det(A_i(z))$ also vanish at these points, where $A_i(z)$ is obtained by replacing the ith column in $A(z)$ with $B(z)$.

Once $P_i(z)$ has been found, the expected number of data customers $E\{Q_2\}$ and the expected waiting time $E\{W_2\}$ are derived from

$$E\{Q_2\} = \sum_{i=0}^{S} P_i'(z)\big|_{z = 1} \tag{AII.5}$$

and

$$E\{W_2\} = \lambda_2^{-1} E\{Q_2\}. \tag{AII.6}$$

This solution procedure, though formally tractable, is difficult to carry out in practice for reasonably large values of N and S. Chang[8] has worked out some specific examples and developed several approximation techniques.

We will not proceed further here with the general infinite buffer problem. However, it is instructive to consider the special case of

N = 0 and S = 1. For this case the birth and death equations for P_{ij} become

$$(\lambda_1 + \lambda_2)P_{0,0} = \mu_1 P_{1,0} + \mu_2 P_{0,1}$$

$$(\lambda_1 + \lambda_2 + \mu_2)P_{0,j} = \mu_1 P_{1,j} + \mu_2 P_{0,j+1} + \lambda_2 P_{0,j-1}: \qquad \text{(AII.7)}$$

$$j = 1,2,\ldots$$

and

$$(\lambda_2 + \mu_1)P_{1,0} = \lambda_1 P_{0,0}$$

$$(\lambda_2 + \mu_1)P_{1,j} = \lambda_1 P_{0,j} + \lambda_2 P_{1,j-1}: \; j = 1,2,\ldots \; . \qquad \text{(AII.8)}$$

Using these equations we have

$$(-\lambda_2 z^2 + (\lambda_1 + \lambda_2 + \mu_2)z - \mu_2)P_0(z)$$

$$= \mu_1 z P_1(z) + \mu_2(z - 1)P_{0,0} \qquad \text{(AII.9)}$$

and

$$\lambda_1 P_0(z) = [\lambda_2(1 - z) + \mu_1]P_1(z). \qquad \text{(AII.10)}$$

The only unknown, $P_{0,0}$, can be found via the carried load condition to be

$$P_{0,0} = (1 - \rho_2 - \rho_1\rho_2)/(1 + \rho_1). \tag{AII.11}$$

Returning to (AII.9) and (AII.10) to find $E(Q_2)$ we get

$$E(Q_2) = \frac{\rho_2(1 + \rho_1)^2 + \rho_1\lambda_2/\mu_1}{(1 + \rho_1)(1 - \rho_1 - \rho_1\rho_2)} \ . \tag{AII.12}$$

From this expression, and more specifically in the factor λ_2/μ_1, one can see the dependence of $E(Q_2)$ on the voice holding time $1/\mu_1$.

The main analysis tool that was used in this paper was the simulation model discussed in Section II. In Section III, a numerical comparison with the analytic results for the special case S = 1, N = 0 was made. We close this appendix with a comparison of the simulation results and a numerical solution of the birth and death equation for the case of N = 5, S = 10 and M = 150. This numerical solution was carried out by direct iteration of the two-dimensional difference equations presented above for the case of finite M, together with application of the condition that the state probabilities must sum to 1. The results are given in Figs. 15 and 16, and one can see there is reasonably good agreement between the simulation and numerical results. Exact agreement is not expected because of differences in the models used. In particular, we recall that exponentially-distributed packet lengths were assumed in the analytic model and fixed packet lengths were used in the simulation.

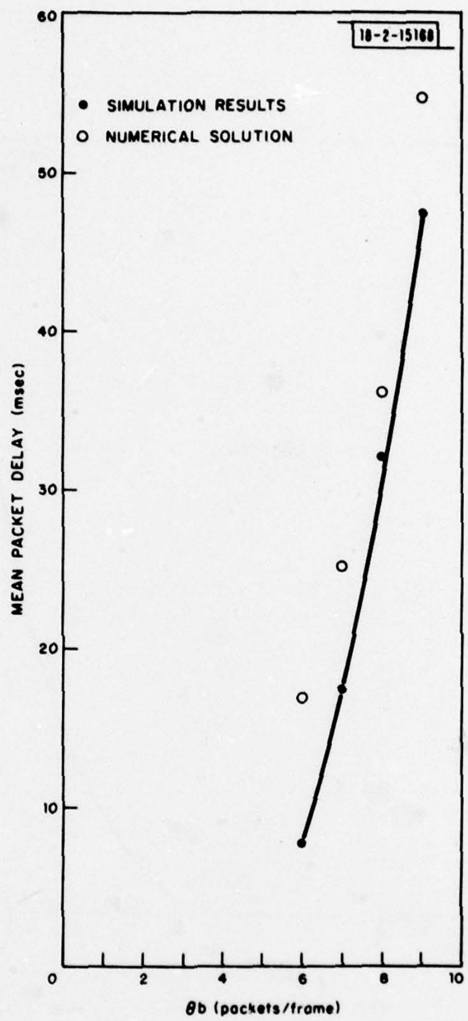

Fig.15. Comparison of simulation results with numerical solution of birth-death equations for mean packet delay with a finite buffer; S = 10, N = 5, λ = .05 sec^{-1}, data queue limited to M = 150 packets. The numerical solution assumes exponentially-distributed packet lengths of average size 80 bits, while the simulation used fixed-length 80-bit packets.

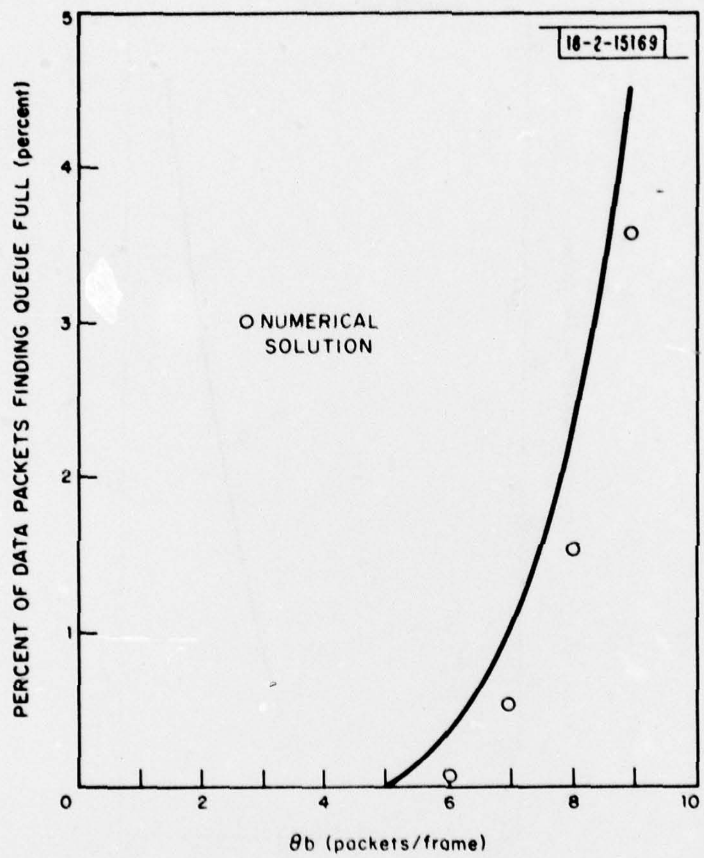

Fig.16. Comparison of simulation results with numerical solution of birth-death equations for percentage of data packets arriving when queue is full. System parameters are the same as in Fig.15.

ACKNOWLEDGMENT

The authors would like to acknowledge the efforts of William Hale of Defense Communications Engineering Center in obtaining the numerical birth and death equation solution results represented in Figs. 15-16.

REFERENCES

1. M. J. Fischer and T. C. Harris, "A Model for Evaluating the Performance of an Integrated Circuit- and Packet-Switched Multiplex Structure," IEEE Trans Communications, COM-24, No. 2, (February 1976).

2. G. Coviello and P. A. Vena, "Integration of Circuit/Packet Switching in a SENET (Slotted Envelope NETwork) Concept," National Telecommunications Conference, New Orleans, (December 1975), pp. 42-12 to 42-17.

3. MIT Lincoln Laboratory Annual Report on Network Speech Processing Program, DDC-AD-A053015, (30 September 1977), pp. 45-53.

4. I. L. Mitrany and B. Avi-Itzhak, "A Many-Server Queue with Service Interruptions," Operations Research, 16, No. 3, (1968).

5. V. Yechiale and P. Naor, "Queueing Problems with Heterogeneous Arrivals and Service," Operations Research, 19, No. 3, (1971).

6. V. A. Bhat and M. J. Fischer, "Multichannel Queueing Systems with Heterogeneous Classes of Arrivals," Naval Research Logistics Quarterly, 23, No. 2, (1976).

7. M. J. Fischer, "A Queueing Analysis of an Integrated Telecommunications System with Priorities," INFOR, 15, No. 3, (1977).

8. Lih-Hsing Chang, "Analysis of Integrated Voice and Data Communication Network," PhD. Thesis, Department of Electrical Engineering, Carnegie-Mellon University (November 1977).

9. R. B. Cooper, Introduction to Queueing Theory (Macmillan, 1972).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
ESD-TR-78-276

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Data Traffic Performance of an Integrated Circuit- and Packet-Switched Multiplex Structure

5. TYPE OF REPORT & PERIOD COVERED
Technical Note

6. PERFORMING ORG. REPORT NUMBER
Technical Note 1978-41

7. AUTHOR(s)
Clifford J. Weinstein
Martin J. Fischer
Marilyn L. Malpass

8. CONTRACT OR GRANT NUMBER(s)
F19628-78-C-0002

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Lincoln Laboratory, M.I.T.
P.O. Box 73
Lexington, MA 02173

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Program Element No. 33126K

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Communications Agency
8th Street & So. Courthouse Road
Arlington, VA 22204

12. REPORT DATE
26 Oct[redacted]978

13. NUMBER OF PAGES
60

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Electronic Systems Division
Hanscom AFB
Bedford, MA 01731

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES
None

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

packet-switching	simulation	analysis
data packets	data queue	traffic

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

New results are presented for data traffic performance in an integrated multiplex structure which includes circuit-switching for voice and packet-switching for data. The results are developed both through simulation and analysis, and show that excessive data queues and delays will build up under heavy loading conditions. These large data delays occur during periods of time when the voice traffic load through the multiplexer exceeds its statistical average. A variety of flow control mechanisms to reduce data packet delays are investigated. These mechanisms include control of voice bit rate, limitation of the data buffer, and combinations of voice rate and data buffer control. Simulations indicate that these flow control mechanisms are quite effective in improving system performance. A combination of data buffer limitation with data-queue-dependent voice rate control was the most effective flow control technique tested.

207650