Traffic Analysis Tools for Integrated Digital Time-Division Link Level Multiplexing of Synchronous and Asynchronous Message Streams

EDWARD ARTHURS AND BARTON W. STUCK, MEMBER, IEEE

Abstract—**Information is transmitted over a digital link in repetitive time intervals called frames; each frame consists of one or more time slots. Theoretical performance limitations of a variety of time-division multiplexing policies for models of synchronous and asynchronous message streams are studied.**

I. INTRODUCTION

THE performance of a theoretical model of a link level digital time-division multiplexer for synchronous and asynchronous messages streams is studied. This has potential use in a variety of applications such as voice telephony, on-line data processing, graphics and facsimile transmission, computer-to-computer communication, electronic office communication, heating and ventilation sensor communication, security sensor communication, and many others.

Two types of sources use the link for communications: *synchronous* sources that generate messages at regularly spaced time intervals, and *asynchronous* sources that do so at irregular points in time. Fig. 1 is a block diagram of a model of a link level multiplexer we wish to analyze. The fundamental logical periodic time unit is called a *frame*. A frame is subdivided into *slots* and each slot is available for transmission of a *chunk* of bits. The design question is to decide which slots will be assigned to synchronous and which to asynchronous message streams. Fig. 2 shows a representative frame. Three cases arise.

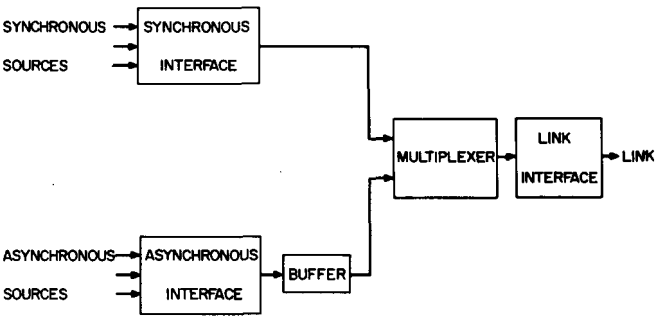

Fig. 1. Hardware block diagram of link multiplexer.

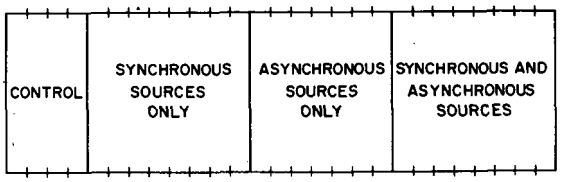

Fig. 2. A general frame: control, dedicated, and shared slots.

A. Dedicating Time Slots to Each Session

The first policy is dedicating time slots during a frame to each session so that each message stream has its own digital transmission system. Each transmission system can be engineered separately with no interaction between different types of message streams. This allows sharing and hence a potential economy of scale of hardware, software, and maintenance costs. Synchronous sources can be multiplexed by a generalization of circuit switching: one or more slots per frame are dedicated to a communication session, while conventional circuit switching has only 1 slot/frame/session. For example, with a frame rate of 1200 frames/s, 4 slots/frame, and 2 bits/slot, one 2400 bit/s terminal would require 1 slot/frame, while one 4800 bit/s terminal would require 2 slots/frame. This has been handled elsewhere [14], [19], [22], [25]. In order to transmit data from asynchronous sources via synchronous transmission facilities, one common practice is to always transmit an idle character if no data character is present, so that the asynchronous stream has the transmission characteristics of a synchronous bit stream.

B. Sharing Time Slots Among All Sessions

Allowing all sessions equal access to any time slot within a frame is a different policy. This allows sharing and, hence, a potential economy of scale for hardware, software, and maintenance costs, as well as the transmission costs. A priority arbitration rule is employed to determine which message stream gains access to the link. The priority might be chosen according to the urgency of the message stream. This has been addressed by many other workers (e.g., [9], [12], [13], [20], [21], [24], [30], [31], [35]). Careful systematic

Manuscript received April 19, 1983; revised May 3, 1983. Parts of this paper were presented at the International Conference on Communications, Boston, MA, 1979, the International Conference on Communications, Seattle, WA, 1980, and the National Telecommunications Conference, Houston, TX, 1980.

The authors are with Bell Laboratories, Murray Hill, NJ 07974.

0733-8716/83/1200-1112$01.00 ©1983 IEEE

simulation studies [26], [39] helped stimulate a great deal of analytic activity in this area (cf. [2], [17], [27], [28]).

C. Dedicating Time Slots to Some Sessions and Sharing the Remainder

The remaining case, a hybrid of the previous two, involves dedicating some transmission capacity to each session, with a common shared transmission capacity pool that is used by all sources in event that the dedicated capacity is completely exhausted.

D. Overview

Our purpose is to study the most difficult case for analysis: totally shared transmission capacity for all communication sessions. The analytic techniques used are well known to mathematicians but, in our opinion, need popularization among mathematically sophisticated engineers. Because the message streams can have radically different characteristics, which will have a profound impact on our analysis, we digress to discuss each traffic in more detail.

E. Synchronous Message Streams

First, we focus on a very common type of synchronous message stream, a voice telephone call. In practice analog voice signals are sampled, encoded into a digital bit stream, and then transmitted over a link. The voice samples are delivered at regularly spaced time intervals for each conversation, with at least one and at most two sampling intervals (due to clock jitter and skew) between successive samples. All the samples must be buffered and transmitted before the next arrivals. This suggests inserting a fixed delay, hopefully negligible compared to voice time scales, which is the time required to load and unload a voice sample during a conversation. Typically voice conversations last for 100–300 s, while the initiation and completion lasts for 1–5 s; this suggests attempting to dedicate transmission capacity for the duration of a conversation, and if no capacity is available at the instant a telephone call is attempted, new arrivals or attempts should be blocked or rejected because it is highly unlikely that transmission capacity would become available within the next few seconds. Voice bit stream utilization per conversation is typically on the order of 30–50 percent (e.g., [4], [5], [7], [8]), a further argument for demanding that transmission capacity be dedicated to each conversation for the duration of a call because it is quite likely that the transmission capacity will, in fact, be needed. These conditions are felt to be sufficient for high quality speech reproduction, but may be unduly conservative when compared to other alternatives.

F. Asynchronous Message Streams

Second, we focus on data traffic generated by interactive terminals and computer systems. Messages within each conversation or session arrive at very irregularly spaced time intervals. In applications representative of on-line computer systems for support of telephone company operations activities, the authors have found traffic is typically 1–2 bytes/s per session for a stream generated by a terminal interacting with a computer, with the interarrival times between samples being much less regular than voice. Furthermore, the utilization of a typical dedicated link for one terminal interacting with a computer is often well under 1 percent. Since the message arrival statistics are highly irregular, and the link utilization is quite low, this suggests pooling or buffering messages from a number of sources (with the pooling strategy based on the statistics of the message streams and their urgencies) for subsequent transmission over a link. This allows both the transmitter and the link to be simultaneously busy, and hence offers a higher total mean throughput rate, at the expense of additional storage. Communication session initiation and completion clean-up times can be comparable to data transmission message times, unlike for the synchronous message stream. If a controller is capable of setting up and taking down a circuit in a time much shorter than the session, this type of switch might be chosen: less time would be spent in overhead versus in useful communications. If this is not the case, then a packet switch might be the choice. In addition, a variety of control bits are required to govern flow control, addressing, error handling, and so forth, that must be transmited along with the actual data bits, further reducing link utilization.

G. Outline

In the next section we present a *naive* analysis of the benefits of *total* sharing of transmission capacity that leads to completely *erroneous* insight into the gain of such a policy. The reason for the error is that a mean value analysis of first-order statistics for the arrival rates and holding times of voice and data conversations ignores secondary statistics such as *fluctuations* about the mean values and more importantly *correlations* from one frame to the next, which *cannot* be ignored here. The following section gives a more rigorous analysis that quantifies these statements. Subsequent sections present two independent highly sophisticated analyses to show that great care is required in engineering a transmission system that pools messages from sources with radically different characteristics. We stress that we have perturbed not only the *parameters* in the models but the underlying modeling *assumptions* to investigate the *robustness* of our findings. The closing section summarizes our findings and presents a number of options or alternatives for effective use of a single link by both synchronous (e.g., voice) and asynchronous (e.g., data) message streams.

II. Problem Statement

Information is transmitted over a digital link in repetitive time intervals called frames. Frame $n=1,2,\cdots$ consists of S_n slots. Each slot contains a fixed number of bits. Messages arrive from synchronous sources and are either accepted or rejected, and presumably will retry later. The synchronous message streams that have been accepted for transmission demand V_n time slots in frame $n=1,2,\cdots$.

From this point on, in the interest of brevity, we will refer to the synchronous message stream as *voice*. Each remaining slot within a frame can be used to transmit messages from asynchronous sources. Asynchronous messages arrive during frame $n=1,2,\cdots$ and require D_n time slots of transmission capacity. From this point on, in the interest of brevity, we will refer to the asynchronous message stream as *data*. $S_n - V_n$ is the amount of transmission capacity (measured in time slots) available during frame $n=1,2,\cdots$ for transmission of asynchronous traffic. The random variable R_n denotes the amount of asynchronous chunks waiting to be transmitted at the start of frame n. From this discussion, R_n is governed by the following recursion:

$$R_{n+1} = \max(0, R_n + D_n + V_n - S_n) \quad n = 0,1,2,\cdots. \quad (1)$$

In the interest of brevity, we fix $S_n = S$. If we allowed the number of available slots to vary from frame to frame, this would correspond to the case of partially shared transmission capacity for synchronous and asynchronous message streams. Our goal is to study the statistics of R_n as a function of the statistics of D_n and V_n for a given choice of S.

A. Summary of Analysis

Granted that $D_n + V_n - S_n$ obeys Markovian statistics, our analysis shows that the long term time averaged distribution of R_n, $n \to \infty$, is a weighted sum of geometric distributions. The weights and modes of the different geometric distributions are dependent upon the particular modeling assumptions for the arrival and holding time statistics for the synchronous and asynchronous message streams as well as the policy for allocating time slots within a frame. One of the modes of the long term time averaged distribution for R_n, $n \to \infty$ will decay slower than all the other modes, and will dominate the asynchronous traffic delay and buffering requirements under load. The geometric decay parameter for the slowest decaying mode of the distribution of R_n, $n \to \infty$ will be called the *critical exponent* because it will be critical in determining the fraction of time the buffer contains greater than a given amount of data chunks.

B. Summary of Results

From an engineering point of view, knowing that the long term time averaged distribution for data decays geometrically suggests that the slowest decaying mode of all the distributions should be measured. In a well engineered system, this should be close to zero, i.e., there should be relatively little data buffering. If this mode is close to one, then there is a *potential* for buffering significant amounts of data. If voice and data are multiplexed together over a common link, the analysis presented here suggests that the voice can demand all the transmission capacity for a period of time that is brief relative to the duration of a voice telephone call, yet is much longer than typical data transmission times. Effectively voice locks out data for time intervals unacceptable to data. This suggests that prudence is required in assessing the benefits of dedicating transmission capacity versus totally sharing transmission capacity for each type of message stream. Engineering practice based on mean values, average loadings, and the like appear to give excellent insight into traffic handling characteristics for *dedicated* transmission systems handling only *one* type of service. When transmission capacity is *shared*, average loading and mean value analysis can be misleading, and much greater caution and sophistication is required. Here, timing between stations is quite controlled and regular; other schemes have been proposed (e.g., [29]) that involve uncontrolled and irregular timing, and hence should do *worse* than the approach described here for handling data.

TABLE I
IDLE TRANSMISSION CAPACITY VERSUS VOICE BLOCKING
24 Time Slots/Frame, 8000 Frames/Sec Transmission Rate

Fraction of Attempts Blocked	Mean Excess Transmission Capacity Available for Data	Mean Time to Enter Voice Blocking State
0.001	762 KBPS	4166.67 sec=69.44 min
0.002	704 KBPS	2083.33 sec=34.72 min
0.005	621 KBPS	833.33 sec=13.89 min
0.010	557 KBPS	416.67 sec=6.94 min
0.020	467 KBPS	208.33 sec=3.47 min
0.050	320 KBPS	83.33 sec=1.39 min
0.100	211 KBPS	41.67 sec=0.69 min

III. A Naive First Cut Analysis (e.g., [29])

Consider the following example: $S = 24$ time slots are transmitted 8000 times/s, with each time slot containing 8 bits. Each voice call requires 1 slot/frame, or 64 kbits/s. Suppose that the link is engineered to handle sufficient voice telephone calls such that no more than 1 percent of the voice traffic is blocked or rejected. Using standard Erlang blocking analysis techniques [11, pp. 89–93], we find the link will handle on the average 15.3 voice telephone calls, i.e.,

$$\text{prob}[\text{all } S \text{ slots filled with voice}] = B(S,A) < 0.01 \to S = 24,\ A = 15.3 \quad (2)$$

where $B(S, A)$ is the Erlang blocking function for S servers and an offered load of A Erlangs. This implies we have $S - A = 24 - 15.3$ or 8.7 time slots/frame available for data, or 557 kbits/s. If we only use 64 kbits/s for data, for example, always transmit 1 byte of data every frame, then the total (both voice and data) link utilization will be (16.3/24) or 67.9 percent, the data delay may be roughly two frames, or 250 μs, and hence the mean data buffering may be 64 kbits/s multiplied by 250 μs or 16 bits.

Table I summarizes the mean number of time slots filled with voice out of 24 ($S = 24$) for a given level of blocking B, the transmission capacity that is idle for handling surges of voice and data.

This type of analysis is based on mean values. Unfortunately, this chain of reasoning ignores *both* the *fluctuations* of the voice and data about its mean values, and the *correlation* present from one frame to the next for voice. The combination of these phenomena makes the mean value analysis sketched here *much* too optimistic: we will need *much* greater buffering for data than expected, data delays will be *much* greater than expected, with *much* less margin for overload surges.

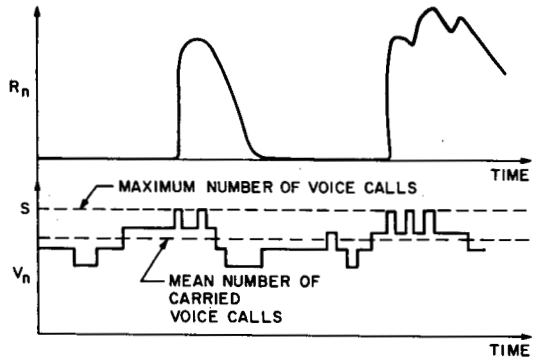

Fig. 3. Illustrative sample path of shared voice/data states.

In hindsight, based on careful systematic simulation studies [26], [39], the reason for this is clear on physical grounds: voice telephone calls last for 2–3 min, or 100–200 s. Data messages can last for 1–10 ms. The time scale for voice is hundreds of seconds, which is 4–5 orders of magnitude greater than that for data. A relatively short surge of voice traffic might last for 1–10 s, but this short surge may demand *all* the available transmission capacity, i.e., none is available for data. As far as data is concerned, the link has just *failed*: no messages can be transmitted. With the link engineered for a voice service grade of 1 percent blocking, this will occur 1 percent of the time during an hour, or a total of 36 s, scattered throughout the hour: 5 s here, 2 s there, 10 s there. Data will be arriving throughout the time when all the transmission capacity is dedicated to voice, potentially leading to thousands of data packets arriving during such an interval. Once this occurs, the validity of the model is now in question: higher level data communications protocols, which are ignored here, come into play, flow control procedures are invoked, time-outs occur, retries, and reinitialization of link control procedures take place, compounding the data delay even more. Fig. 3 is an illustrative simulation of this phenomenon: an illustrative sample path of the stochastic processes we wish to study, that is typical of that encountered in simulation studies (e.g., [26], [39]). The voice traffic rarely surges to seize all the available transmission capacity. Once the voice blocks all data transmission, enormous data queues arise, that persist long after the voice blocking has ended. Put differently, most of the time the data are *not* delayed at all, but if the data are delayed, they are delayed a *long* time. Our intent is to quantify these intuitive notions in later sections.

IV. A More Sophisticated Analysis

In order to get more insight into the behavior of the example in the previous section, we study the statistical behavior of the voice loss system in greater detail. The sequence of nonblocked voice telephone calls holding times are assumed to be i.i.d. exponential random variables with mean $1/\mu$. The modeling assumptions here is that the duration of a voice call is much longer than a frame, and that we require a geometrically distributed number of frames to be dedicated to each telephone call. Fig. 4 shows states and transition rates for the voice telephone calls. We denote by $\lambda = \lambda_{\text{offered}}$ the mean arrival rate of the *offered*

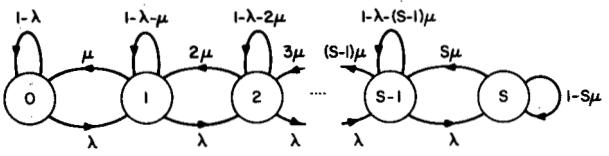

Fig. 4. Voice Markovian state space and transition rates.

voice traffic load, while λ_{carried} is the mean throughput rate of the *carried* voice traffic load, given by

$$\lambda_{\text{carried}} = \lambda_{\text{offered}}\left[1 - B(S, A = \lambda_{\text{offered}}/\mu)\right]. \quad (3)$$

A level is fixed, denoted by L, for the maximum number of simultaneous active voice telephone conversations. When the number of calls in progress exceeds L, newly arriving voice calls will be rejected or blocked. Hence, $0 < L \leqslant S$. $\pi(K)$, $K = 0, \cdots, S$ denotes the fraction of time there are K simultaneously active voice calls in progress. The rate of entering the state of having L calls in progress from the state of having $L-1$ calls in progress is denoted by $r_{L-1,L}$. This implies using standard techniques that

$$r_{L-1,L} = \lambda_{\text{offered}}\pi(L-1). \quad (4)$$

In words, the fraction of time there are $L-1$ simultaneously voice telephone calls multiplied by the total arrival rate of voice calls yields the rate of entering state L from state $L-1$.

$T_{L,L-1}$ denotes the mean time to move from state L to state $L-1$, i.e., to move out of the blocking state of having L simultaneously active voice telephone calls. This implies

$$r_{L-1,L}T_{L,L-1} = \sum_{K=L}^{S} \pi(K). \quad (5)$$

In other words, to find the mean time to move from the state of L to $L-1$ active telephone calls, the total fraction of time the Markov process modeling voice calls is in state $K = L, \cdots, S$ must be divided by the rate of entering state L from $L-1$

$$T_{L-1,L} = \frac{\sum_{K=L}^{S} \pi(K)}{\lambda_{\text{offered}}\pi(L-1)}. \quad (6)$$

Choosing $L = S$ simplifies all of this

$$r_{S-1,S} = \lambda_{\text{offered}}\pi(S-1) = \mu SB[S, \lambda_{\text{offered}}/\mu]. \quad (7)$$

For example, with $S = 24$ time slots/frame, and with $1/\mu = 100$ s for a voice call holding time of 100 s, and the voice call blocking engineering to be no more than 1 percent of the time ($B(S, A) = 0.01$) the mean time to enter the state of having all $S = 24$ time slots filled from the state of having of having $S-1 = 23$ time slots filled with voice, is given by

$$\frac{1}{r_{S,S-1}} = \frac{1}{\mu SB(S, \lambda_{\text{offered}}/\mu)}$$

$$\approx 416\frac{2}{3}\text{s} \quad S = 24 \text{ slots/frame} \quad (8)$$

TABLE II
MEAN TIME TO VOICE BLOCKING STATE VERSUS VOICE BLOCKING

24 Time Slots/Frame, 8000 Frames/Sec Transmission Rate			
Fraction of Attempts Blocked	*Mean Number of Voice Filled Slots*	*Mean Number of Idle Slots*	*Excess Transmission Capacity for Data*
0.001	*12.1 Slots*	*11.9 Slots*	*762 KBPS*
0.002	*13.0 Slots*	*11.0 Slots*	*704 KBPS*
0.005	*14.3 Slots*	*9.7 Slots*	*621 KBPS*
0.010	*15.3 Slots*	*9.3 Slots*	*557 KBPS*
0.020	*16.7 Slots*	*7.3 Slots*	*467 KBPS*
0.050	*19.0 Slots*	*5.0 Slots*	*320 KBPS*
0.100	*20.7 Slots*	*3.3 Slots*	*211 KBPS*

In words, on the average every $416\frac{2}{3}$ s the link will be completely filled with voice calls in progress. How long will this last on the average? For a mean time denoted by $T_{S,S-1}$, where

$$T_{S,S-1} = \frac{1}{\mu S} \approx 4.16667 \text{ s} \quad S = 24. \tag{9}$$

As far as the data are concerned, on the average, every $416\frac{2}{3}$ s the voice will demand all the transmission capacity for a mean time interval of $4\frac{1}{6}$ s. If data arrive at an average rate of 64 kbits/s, then at least 256 kbits of data on the average must be buffered when the link is busy handling nothing but voice, or 32 kbytes. Furthermore, the transmission capacity for data once the system leaves the voice blocking state is now only 64 kbits/s, which only keeps up with arriving data but does *not* empty the buffer, and the very strong correlation suggests that the link might block within the next second or two with nothing but voice, as before.

Table II is a summary of similar calculations for the same parameters described in the earlier section.

The sojourn time in the all blocking state is 100/24 s or 4.16 s. Would a customer choose a system with voice blocking only 1 percent of the time, while 557 kbits/s of idle transmission capacity is available, knowing that on the average every 6.94 min the idle transmission capacity would drop to *zero* for an average 4.16 s? Many knowledgeable engineers would argue that it is difficult enough to get such systems to work at all, without having to handle *interactions* between different services like voice and data such as this is.

V. 2 SLOTS/FRAME

Here is a summary of a highly sophisticated analysis (presented in a later section), done by means of an illustrative case study: there are 2 slots/frame, $S = 2$, with either zero, 1, or 2 slots occupied with synchronous traffic. For simplicity of exposition, we assume $D_n = 1$, i.e., there is always one data chunk arriving every frame, which would be a worst case analysis. The basic recursion for the number of data chunks waiting to be transmitted at the start of frame $n+1$, denoted by R_{n+1}, is given by

$$\begin{aligned} R_{n+1} &= \max[0, R_n + D_n + V_n - S_n] \\ &= \max[0, R_n + V_n - 1] \qquad n = 0,1,2. \end{aligned} \tag{10}$$

The voice traffic obeys birth and death process statistics, with mean voice holding time of $1/\mu$ and mean voice call arrival rate of λ. In a later section, we present the underlying mathematical analysis; here we merely summarize the results and discuss how to interpret the results.

In order for a nontrivial long term time averaged distribution of R_n, $n \to \infty$ to exist, we demand that

$$\frac{\lambda}{\mu} < \sqrt{2}. \tag{11}$$

Granted that this condition is satisfied, the long term time averaged fraction of time there are R_n, $n \to \infty = K$ chunks in the buffer is given by

$$\text{fraction of time data does not wait} = \text{prob}[R = 0] = 1 - \omega G \tag{12}$$

$$\text{fraction of time } K \text{ chunks in data buffer} = \text{prob}[R = K] = 1 - G(1-\omega)\omega^K. \tag{13}$$

The factor G is directly proportional to the voice blocking. The mean amount of data buffered is simply

$$E[R] = \sum_{K=0}^{\infty} K \,\text{prob}[R = K] = \frac{G}{1-\omega}. \tag{14}$$

In words, the mean amount of data buffered is the ratio of two terms. The numerator is proportional to the fraction of time voice is blocked, i.e., the voice grade of service. The denominator is approximated by the total number of slots/frame over the mean voice holding time, measured in slots. Both the numerator and the denominator are close to zero, but, for numbers of practical interest, the denominator is significantly smaller, and hence the mean data delay is much larger than might be expected (cf. the simulation sample path in Fig. 3).

We now turn to some illustrative numerical results. We assume that the frames are transmitted 8000/s, with 2 slots/frame, each capable of holding one chunk of 8 bits. Fig. 5 shows ω as a function of voice traffic holding time (for different levels of voice blocking): it is clear that when the voice traffic holding time is greater than 10 slots, ω is greater than 9/10, while for cases of practical interest, where the ratio of voice call holding time to data holding time is 1000 or more, ω can be arbitrarily close to unity! This is the impact of correlation on the amount of data buffering required. In addition, Fig. 5 shows an easy to calculate *lower* bound on ω that is evidently quite close to the *exact* value of ω for the numbers and assumptions chosen here. We will derive this lower bound in a later section.

In Fig. 6, we plot the fraction of time voice is blocked versus the fraction of time data are delayed at all, for

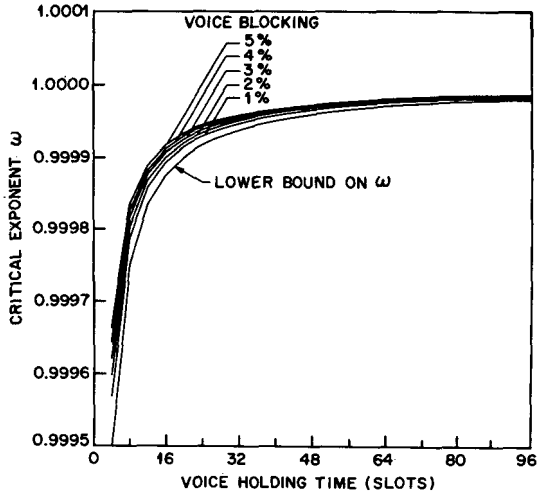

Fig. 5. Critical exponent versus voice holding time.

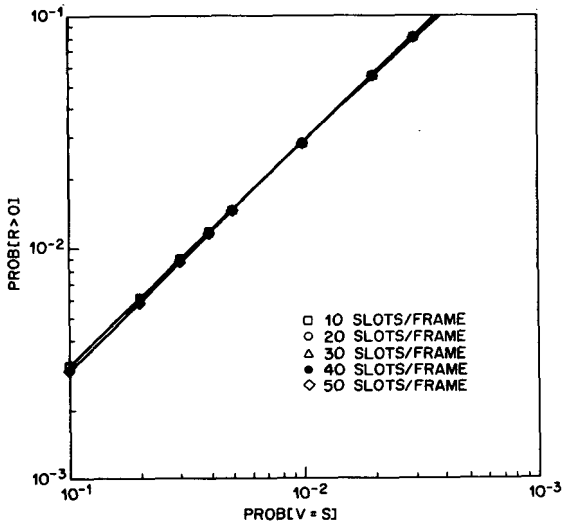

Fig. 6. Fraction of time $R > 0$ versus voice blocking.

different levels of voice holding time. Note that we can have quite small voice blocking and quite small data blocking (i.e., data blocking is the fraction of time data are delayed at least 1 frame).

On the other hand, while data blocking can be quite small, if data are delayed, they can be delayed a *long* time (measured in frames). This is shown by the mean amount of data buffered being proportional to $B/(1-\omega)$ and ω is quite close to unity, while the voice blocking B is close to zero. Fig. 7 plots mean data buffering for a fixed mean voice holding time, measured in time slots. As is evident, the data buffering requirements, and hence the mean data delay, can be far in excess what the naive analysis suggested earlier. Put differently, even though the service for voice is acceptable, the data simply cannot count on the transmission capacity being available.

VI. General Semi-Markov Queueing Model

In this section we summarize the analysis of a more general class of models called *semi-Markov* queueing systems [38].

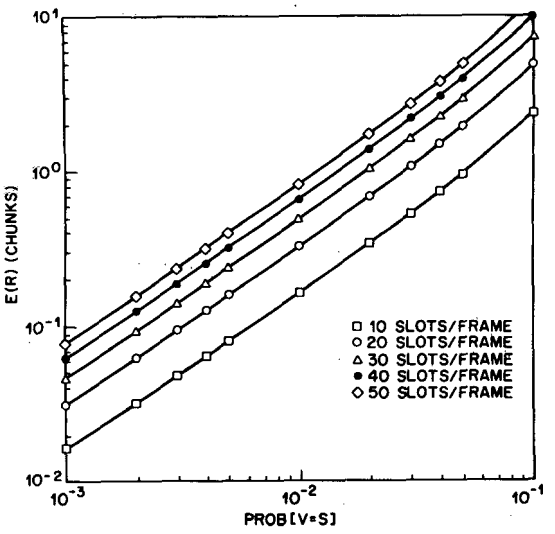

Fig. 7. Mean data in system versus voice blocking.

The amount of data and voice, measured in slots, arriving in frame n is denoted by D_n, and V_n, respectively. This stochastic process is *modulated* by an irreducible, aperiodic Markov chain with state space Ω; at the start of frame n, the state of the Markov chain is σ_n. Granted this notation, we define $H_{lk}(i, j)$ as the transition probability generator for the voice and data stochastic process

$$H_{lk}(i,j) = \text{prob}[D_n = i, S - V_n = j, \sigma_{n+1} = k | \sigma_n = l]$$
$$\sigma_j \in \Omega, \quad j = 1,2,\cdots \quad n = 0,1,2,\cdots \tag{15}$$

where $S - V_n$ is the amount of time slots available for data transmission during frame n. The amount of data remaining to be transmitted just prior to the start of frame n is denoted by R_n and this obeys the following recursion:

$$R_{n+1} = \max[0, R_n + D_n + V_n - S]. \tag{16}$$

The matrix moment generating function associated with the matrix H is denoted by η:

$$\eta_k(x) = \sum_{ij} H_{lk}(i,j) x^{i-j} \quad 0 \leqslant x \leqslant 1 \quad l,k \in \Omega \tag{17}$$

where x is a complex variable taking on values inside the unit disk.

The transient moment generating function for the amount of data remaining at the start of frame n with the underlying Markov chain in state j given the initial state of the Markov chain is state i will be denoted by P

$$P_{ij}(x,r) = \sum_{n=0}^{\infty} r^n E\left[x^{R_n}, \delta_{\sigma_n j} | \sigma_{n=0} = i\right] \tag{18}$$

where $r \in (0,1)$ takes on values inside the unit disk, and the moment generating function for the distribution of the initial amount of data in the system R_0, is given by $E[x^{R_0}]$.

Following earlier work [38], we carry out a Wiener–Hopf factorization of a matrix inverse closely related to η:

$$[I - r\eta(x)]^{-1} = \eta^+(x,r)\eta^-(x,r) \tag{19}$$

where $\eta^+(x,r), \eta^-(x,r)$ is analytic inside (respectively outside) the unit disk $x \in (0,1)$.

The transient moment generating function $P(x,r)$ is given by

$$P(x,r) = T\left[E\left[x^{R_0}\right]\eta^-(x,r)\right]\eta^+(x,r) \qquad x \in (0,1),\ r \in (0,1) \quad (20)$$

where T is a projection operator for a vector valued Laurent series

$$T\left[\sum_{k=-\infty}^{\infty} x^k M_k\right] = \sum_{k=-\infty}^{-1} M_k + \sum_{k=0}^{\infty} x^k M_k. \quad (21)$$

In principle, all that remains is to invert this moment generating function, via numerical approximations, since analytic methods appear to be intractable in all but the simplest cases.

If a long term time averaged distribution exists for R, then standard asymptotic techniques allow us to show

$$\lim_{n\to\infty} E\left[x^{R_n}\delta_{\sigma_n j} | \infty = i\right] = \lim_{r\to 1-} (1-r)P_{ij}(x,r). \quad (22)$$

A. An Alternative Derivation [15], [16]

Here is an alternative exposition of the same set of manipulations. The associated matrix moment generating function for the R_n process is given by

$$P_{ij}(r,x) = \sum_{k=0}^{\infty} r^k E\left(x^{R_k}, \sigma_k = j | \infty = i, R_0 = 0\right). \quad (23)$$

It is useful to rewrite this matrix as the product of two matrices, one with only nonnegative powers of x denoted by $M_+(r,x)$ and the other with only nonpositive powers of x denoted by $M_-(r,x)$

$$P(r,x) = \left[M_-(r,x=1)\right]^{-1} M_+(r,x). \quad (24)$$

The matrices M_-, M_+ are the Wiener–Hopf factors associated with the matrix $I - rQ(x)$, i.e.,

$$M_+(r,x)\left[I - rQ(x)\right] = M_-(r,x) \quad (25)$$

where

$$\tilde{Q}_{l,k}(x) = E\left[x^{D_n}\right]\text{prob}\left[\sigma_{n+1} = l | \sigma_n = k\right] x^{-(s-v_n)}. \quad (26)$$

The Wiener–Hopf factorization requires numerical work; clean illustrative analytic answers appear to be rare. The critical exponent ω is the unique real root in the interval $(0,1)$ of

$$\omega = \text{largest eigenvalue of } \tilde{Q}(x=\omega). \quad (27)$$

In other words, we find the spectral radius or Frobenius root of the operator $Q(x)$, as a function of x, and then search for the unique real root on the open interval such that the root equals the spectral radius evaluated at that root. Excellent numerical techniques are available for carrying out these steps quickly [18], [23], [37].

B. Another Alternate Derivation

Here is an alternate direct derivation of these results that may be easier to comprehend. We confine attention from this point on to the long term time averaged behavior. We define $F(i,j)$ as the fraction of time that $R_n < j$, $n \to \infty$ and the state of the underlying Markov chain, denoted by σ, is given by $\sigma = i$. Then, using the Q matrix associated with the moment generating function $\tilde{Q}$ above, we see

$$F(j,k) = \sum_{i=0}^{S} q_{ij} F(i, j+k-1) \quad k > 1 \quad (28)$$

$$F(i,1) = q_{1i}F(1,1) + q_{2i}F(2,2) + \cdots. \quad (29)$$

Let us try a solution of the form

$$F(j,k) = A(j) + B(j)\omega^k. \quad (30)$$

If it works, we are done due to invocation of existence and uniqueness theorems for this class of equations. Substituting, we see

$$A(j) + B(j)\omega^k = \sum_{i=0}^{S} q_{ij}\left[A(i) + B(i)\omega^{l+k-1}\right] \quad (31)$$

and, hence, we see that

$$A(j) = \sum_{i=0}^{S} A(i) q_{ij} \quad (32)$$

or in other words $A(j)$ is the left eigenvector of the matrix Q with associated eigenvalue unity, while

$$B(j)\omega = \sum_{i=0}^{S} B(i)\omega^l q_{ij}. \quad (33)$$

ω must be the solution in the unit circle of

$$\det\left[\omega I - \tilde{Q}(\omega)\right] = 0 \quad (34)$$

where

$$\tilde{Q}(\omega)_{ij} = \omega^l q_{ij}. \quad (35)$$

It remains to check the case $k = 1$:

$$A(j) + B(j)\omega = \sum_{k=1}^{S}\left[A(k) + B(k)\omega^k\right] q_{kl} \quad (36)$$

and thus we see that

$$\sum_{k=0}^{S} A(k) q_{ki} + \sum_{k=0}^{S} B(k)\omega^k q_{ki} = \sum_{k=1}^{S}\left[A(k) + B(k)\omega^k\right] q_{ki} \quad (37)$$

or in other words $A(0) = -B(0)$, and so forth for the remaining components of B. The components of $A(j)$ are the invariant measure associated with q_{ij}. The solution in general will be a linear combination of $(\omega_l)^k$ where the ω_l are roots inside the unit disk of the above equation, and the largest root on the interval $(0,1)$ will dominate asymptotically, $R \to \infty$, as claimed.

C. Example: 2 Slots/Frame

We now perform the explicit calculations that gave the results described earlier, for 2 slots/frame, with one chunk of data arriving every frame.

The transition matrix for the Markov chain associated with the synchronous traffic, denoted by Q, is given by

$$Q=\begin{bmatrix} Q_{00} & Q_{01} & Q_{02} \\ Q_{10} & Q_{11} & Q_{12} \\ Q_{20} & Q_{21} & Q_{22} \end{bmatrix}=\begin{bmatrix} 1-\lambda & \lambda & 0 \\ \mu & 1-\lambda-\mu & \lambda \\ 0 & 2\mu & 1-2\mu \end{bmatrix}. \tag{38}$$

The long term time averaged distribution for the number of slots occupied with synchronous traffic $\pi=(\pi_0,\pi_1,\pi_2)$ is given by

$$\pi_0=\frac{1}{1+\dfrac{Q_{01}}{Q_{10}}+\dfrac{Q_{01}}{Q_{10}}\dfrac{Q_{12}}{Q_{21}}}=\frac{1}{1+\dfrac{\lambda}{\mu}+\dfrac{\lambda}{\mu}\dfrac{\lambda}{2\mu}} \tag{39}$$

$$\pi_1=\pi_0\frac{Q_{01}}{Q_{10}}=\pi_0\frac{\lambda}{\mu} \tag{40}$$

$$\pi_2=\pi_0\frac{Q_{01}}{Q_{10}}\frac{Q_{12}}{Q_{21}}=\pi_0\frac{\lambda}{\mu}\frac{\lambda}{2\mu}$$

$$=\text{fraction of time voice calls blocked.} \tag{41}$$

In order for a nontrivial long term time averaged distribution for R to exist, we must have the mean number of synchronous calls plus the mean number of data slots occupied per frame be less than S

$$\sum_{K=0}^{S} K\pi(K)+E(D)<S. \tag{42}$$

For this particular problem, we can relate this to given quantities λ, μ and we find that

$$\frac{\lambda}{\mu}<\sqrt{2} \tag{43}$$

which we maintain is not obvious *a priori*!

The long term time averaged distribution of R_n, $n\to\infty$ is given by

$$\lim_{n\to\infty}\text{prob}[R_n=0]=1-\omega G \tag{44}$$

$$\lim_{n\to\infty}\text{prob}[R_n=K>0]=G(1-\omega)\omega^K \quad K=1,2,3\cdots \tag{45}$$

where ω is given by

$$\omega=\frac{Q_{22}(1-Q_{11})+Q_{12}Q_{21}}{Q_{00}(1-Q_{11})+Q_{10}Q_{01}}=\frac{(1-2\mu)(\mu+\lambda)+2\mu\lambda}{(1-\lambda)(\lambda+\mu)+\mu\lambda} \tag{46}$$

and G is given by

$$G=\pi_2\frac{v_0+v_1+v_2}{v_2} \tag{47}$$

where $v=(v_0,v_1,v_2)$ is the left eigenvector associated with $\eta_0(x)Q$ and is given by

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix}=\begin{pmatrix} Q_{11}(1-Q_{22}/\omega)-Q_{12}Q_{21}/\omega \\ Q_{01}\omega-Q_{01}Q_{22} \\ \omega Q_{01}Q_{12} \end{pmatrix}$$

$$=\begin{pmatrix} (1-\lambda-\mu)(1-(1-2\mu)/\omega)-2\mu\lambda/\omega \\ \lambda(\omega-1+2\mu) \\ \omega\lambda^2 \end{pmatrix}. \tag{48}$$

D. A Lower Bound on Critical Exponent

We now present a *lower* bound on the critical exponent ω that illustrates the impact that correlation can have on performance. This is easier to calculate than the exact analysis, and the techniques are felt to be of general interest and, hence, should be popularized. To simplify notation, we define a new random variable Y_n which is the difference between the voice and data arriving in frame n and the total available transmission capacity, in slots: $Y_n=D_n+V_n-S$. As in the previous section, we model correlation by assuming that Y_n is modulated by a Markov chain. This means that there is an underlying Markov chain with state σ_n for frame n taking values in a discrete state space denoted by Ω, and with transition probability generator for the voice and data given by $H_{lk}(i,j)$:

$$H_{lk}(i,j)=\text{prob}[D_n=i, S-V_n=j, \sigma_{n+1}=k|\sigma_n=l]. \tag{49}$$

1) Fundamental Analysis: As one example, we assume the voice traffic is generated by a birth and death process, with λ denoting the probability of an arrival and $i\mu$ denoting the probability of a departure given there are i active voice calls. The state of the underlying Markov chain associated with the start of frame n is denoted by σ_n. Given these assumptions, we see that

$$R_{n+1}=\max(0,R_n+Y_n) \quad n=0,1,2,\cdots. \tag{50}$$

From this recursion, we can rewrite the fraction of time that R_n, $n\to\infty$ exceeds a finite threshold, say K

$$\lim_{n\to\infty}\text{prob}[R_n>K]$$

$$=\sum_{J=0}^{S}\pi(J)\,\text{prob}\left[\sup_n[Y_1+Y_2+\cdots+Y_n]>K|\infty=i\right]. \tag{51}$$

If we focus only on the state where the synchronous traffic has seized all available time slots, i.e., and drop the other $(S-1)$ states from consideration, then we obtain a *lower* bound on our expression of interest

$$\lim_{n\to\infty}\text{prob}[R_n>K]$$

$$>\pi(S)\,\text{prob}[Y_1+Y_2+\cdots+Y_m>K|\infty=S]. \tag{52}$$

Since the process is Markovian, the holding time distribution in this state is *geometric*, with the number of time slots in this state, say M, having a moment generating function given by

$$E[X^M] = \frac{X(1-S\mu)}{1-SX\mu}. \tag{53}$$

The lower bound on the fraction of time $R_n, n \to \infty$ exceeds K can be written as

$$\lim_{n\to\infty} \text{prob}[R_n > K] = \pi(S)\,\text{prob}[D_1 + \cdots + D_M > K | \infty = S]. \tag{54}$$

For example, with one data chunk arriving every frame, $D_n = 1$ (a.s.),

$$\lim_{n\to\infty} \text{prob}[R_n > K] > \pi(S)(1-S\mu)^K \tag{55}$$

where

$$\pi(S) = B(S, \lambda/\mu) = \frac{(\lambda/\mu)^S/S!}{\sum_{k=0}^{S} (\lambda/\mu)^k/k!} \tag{56}$$

and hence

$$\omega > 1 - S\mu \tag{57}$$

which is independent of the arrival rate. This is in the same spirit as *heavy traffic* limit theorems of queueing theory. For example, choosing $S = 24$ and 100 s voice holding time, so $\mu = 1/(100\text{ s})(8000\text{ frames/s})$, we find

$$\omega > 1 - \frac{24}{100\times 8000} = 1 - \frac{1}{40000} = 0.999975. \tag{58}$$

On the other hand, if D_n obeys a geometric distribution, so

$$\lim_{n\to\infty} E[x^{D_n}] = \frac{(1-\alpha)x}{1-\alpha x} \tag{59}$$

then the same chain of arguments can be used to show

$$\omega > 1-(1-S\mu)(1-\alpha) > 1-(1-S\mu) \tag{60}$$

which is even closer to unity than ignoring the bursty nature of the data arrivals, i.e., the shape of the data arrival distribution statistics matter!

2) The Impact of Time Assignment Speech Interpolation: As a second example of how to use this model, what if we wish to multiplex data during speech silence intervals, e.g., inbetween words and pauses in conversation, (e.g., [4]–[6])? Again, we can refine the above argument as follows: let (i, j) denote i active calls and j calls actively talking at a given time epoch, where $i = 0, \cdots, S$; $j \leqslant i$. We model this via a Markov chain as follows:

$$\text{prob}[i+1, j+1 | i, j] = \lambda \quad 0 \leqslant i < S \tag{61}$$

$$\text{prob}[i, j+1 | i, j] = \beta(i-j) \quad 0 < i \leqslant S,\ j < i \tag{62}$$

$$\text{prob}[i, j | i+1, j+1] = \mu(j+1) \quad 0 \leqslant i < S \tag{63}$$

$$\text{prob}[i, j | i, j+1] = \gamma(j+1) \quad 0 \leqslant i \leqslant S. \tag{64}$$

Then, paralleling the previous arguments, we see that the fraction of time the long term time averaged amount of data buffered exceeds a threshold K is lower bounded by

$$\lim_{n\to\infty} \text{prob}[R_n > K] = \pi(S, S)\,\text{prob}[D_1 + \cdots + D_m > K] \tag{65}$$

where the random variable m is drawn from a geometric distribution,

$$\text{prob}[m = J] = \gamma S(1-\gamma S)^{J-1} \quad J > 0 \tag{66}$$

and hence

$$\omega > 1 - S\gamma. \tag{67}$$

To summarize both these exercises, granted these assumptions, we have shown

$$\omega > 1 - \frac{\text{number of slots/frame}}{\text{mean holding time (in slots)/call}} \tag{68}$$

which is independent of the arrival rate. As the arrival rate approaches zero, it is not the exponent that approaches zero but rather the constant multiplying the exponential term that approaches zero.

VII. A Markovian Model for Multiplexing Voice and Data over a Single Digital Link

In this section, we analyze a Markovian model of a totally shared voice and data link multiplexer. Both the assumptions and method of analysis are different from that described earlier.

A. Model

The total transmission capacity of the link is C bits/s. Each voice call is transmitted at a fixed bit rate, denoted by R_v, for the duration of a voice call. The duration of each voice call forms a sequence of i.i.d. exponential random variables with mean duration $1/\beta$ s. The interarrival times of voice call attempts form a sequence of i.i.d. exponential random variables with mean interarrival time $1/\alpha$.

The interarrival times for data messages form a sequence of i.i.d. exponential random variables with mean interarrival time $1/\lambda$ s. The length of each data message forms a sequence of i.i.d. exponential random variables with mean message length B bits.

The maximum number of voice calls permitted to simultaneously use the channel is S calls. We see that

$$C \geqslant SR_v. \tag{69}$$

The transmission capacity (in bits/s) available for data at any given instant of time, given that there are $K = 0, \cdots, S$ simultaneous voice calls in progress, is

$$\text{data transmission capacity} = C - KR_v \quad K = 0, 1, \cdots, S. \tag{70}$$

The rate (in messages/s) at which messages are transmitted at any instant of time is the data transmission capacity divided by the mean message length (in bits)

$$\text{message transmission rate} = \mu_K = \frac{C - KR_v}{B} \qquad K = 0,1,\cdots,S. \quad (71)$$

B. Analysis [32] – [34]

The number of voice calls in progress at time t is denoted by $V(t)$. The number of data messages buffered in the queue at time t is denoted by $N(t)$. Granted the previous assumptions, $[V(t), N(t)]$ forms a two-dimensional Markov chain with a discrete set of states and a continuous time parameter. This Markov chain has a nondegenerate long term time averaged distribution if and only if

$$\lambda < \sum_{K=0}^{S} \pi_K \mu_K \quad (72)$$

where π_K, $K = 0,\cdots,S$ is the invariant measure for the number of voice calls in progress. Let $\boldsymbol{A}$ denote an $(S+1) \times (S+1)$ matrix, where

$$\boldsymbol{A} = \text{diag}[\mu_0,\cdots,\mu_S]. \quad (73)$$

Let $\boldsymbol{B}$ be an $(S+1)\times(S+1)$ matrix that is the minimal solution of the following matrix equation:

$$\boldsymbol{B}^2\boldsymbol{A} + \boldsymbol{B}[\boldsymbol{P} - \lambda\boldsymbol{I} - \boldsymbol{A}] + \lambda\boldsymbol{I} = 0 \quad (74)$$

where $\boldsymbol{P}$ is the generator of the voice call Markov chain and $\boldsymbol{I}$ is the $(S+1)\times(S+1)$ identity matrix. By this, we mean that $\boldsymbol{B}$ can be calculated from the following iteration:

$$\boldsymbol{B} = \lim_{J\to\infty} \boldsymbol{B}_J$$

$$\boldsymbol{B}_{J+1} = [\boldsymbol{B}_J^2\boldsymbol{A} + \lambda\boldsymbol{I}][\lambda\boldsymbol{I} + \boldsymbol{A} - \boldsymbol{P}]^{-1}$$

$$J = 0,1,2,\cdots; \quad \boldsymbol{B}_0 = \boldsymbol{O}. \quad (75)$$

Let the long term time averaged distribution of the Markov chain associated with $[V(t),N(t)]$ be denoted by Q_{KJ}, $J = 0,\cdots,S$; $K = 0,1,\cdots$ where

$$\lim_{t\to\infty} \text{prob}[N(t) = K, V(t) = J] = Q_{KJ}$$

$$K = 0,1,2\cdots; \quad J = 0,1,\cdots,S. \quad (76)$$

The row vectors associated with this matrix

$$\tilde{Q}_K = [Q_{K0}, Q_{K1},\cdots,Q_{KS}] \quad K = 0,1,\cdots \quad (77)$$

are the fraction of time, averaged over a suitably long time interval, or the probability, of finding K messages in the data buffer. Granted the above assumptions, it is straightforward to show that

$$\tilde{Q}_K = \pi[\boldsymbol{I} - \boldsymbol{B}]\boldsymbol{B}^K \quad K = 0,1,\cdots,S \quad (78)$$

where π is the invariant measure associated with $\boldsymbol{P}$

$$\pi = \pi\boldsymbol{P} \quad \sum_{K=0}^{S} \pi_K = 1. \quad (79)$$

We find it useful to define a related quantity, denoted ρ_K, which is the long term fraction of time there are K messages in the data buffer

$$\rho_K = \sum_{I=0}^{S} Q_{KI} \quad K = 0,1,\cdots,S. \quad (80)$$

The moment generating function for the long term time averaged number of messages in the buffer is given by

$$\lim_{t\to\infty} E[X^{N(t)}] = \sum_{K=0}^{S} \rho_K X^K = \sum_{K=0}^{S}\sum_{I=0}^{S} Q_{KI} X^K = \eta(X). \quad (81)$$

The random variable T_{delay} denotes the message delay, measured from the time it enters the system until it is completely transmitted over the link. The long term time averaged distribution for the time in system of a message is denoted by $G_{T_{\text{delay}}}(Y)$ with associated moment generating function

$$E[\exp(-zT_{\text{delay}})] = \int_0^\infty e^{-zY} dG_{T_{\text{delay}}}(Y) \quad (82)$$

which in turn must satisfy [10, p. 156, eq. 20]

$$E[\exp(-zT_{\text{delay}})] = \eta\left[\frac{\lambda - z}{\lambda}\right]. \quad (83)$$

The mean message delay is given by

$$E[T_{\text{delay}}] = \frac{1}{\lambda}\frac{d\eta(X)}{dX}\bigg|_{x=1} = \frac{\sum_{K=0}^{S} \rho_K}{\lambda}. \quad (84)$$

Different techniques are available for numerical approximations to desired measures of performance. These are omitted in the interest of brevity. The findings and interpretations are identical to that described earlier, lending further credence to these results.

VIII. Dedicated Transmission Capacity: Two Special Cases

Before we conclude, we summarize illustrative available results for dedicating transmission capacity for data or asynchronous message streams. Furthermore, the models are analytically *tractable* which can be a great advantage in practice. The first case is a bulk service queueing system, while the second involves assigned 1 slot/frame to asynchronous traffic and one or more slots to the synchronous traffic.

A. No Synchronous Traffic

If there is no synchronous traffic, data arrivals are multiplexed onto the transmission link in order of arrival. R_n denotes the number of data chunks which were availa-

ble for transmission during frame n but which were blocked from so doing. Then, R_n obeys the following recursion:

$$R_{n+1} = \max(0, R_n + D_n - S) \quad R_0 = 0 \tag{85}$$

where D_n is the number of data chunks arriving during frame n.

If $S=1$, granted certain reasonable assumptions, this can be handled by standard techniques. In fact, if there are different data streams with different urgencies, the data delay statistics for each type of data stream can be found using static priority arbitration [10].

What if $S>1$? In a pioneering paper [3], the long term time averaged distribution for the number of data chunks in the system at the start of a frame was found, provided the sequence D_n are i.i.d. random variables. Since the pair of random variables for the duration of busy period and number of tasks executed in a busy period is a sequence of i.i.d. random variables, the mean data delay is estimated consistently by dividing the number of data chunks in the system by the mean data arrival rate.

B. One Slot for Asynchronous Traffic and the Rest for Synchronous Traffic

If 1 slot/frame is available for transmission of messages generated by asynchronous traffic and the remaining $S-1$ time slots are dedicated to transmitting synchronous traffic, the data delay statistics can be calculated exactly (cf. [1]). The random variable T_{sync} denotes the time the link is denoted to synchronous traffic, while the random variable T_{async} denotes the time required to transmit one chunk of asynchronous traffic. We assume the asynchronous traffic interarrival times are i.i.d. exponential random variables with mean interarrival rate λ, and each asynchronous message requires one slot to be transmitted. Given these assumptions, the mean asynchronous traffic delay, defined as the time from arrival until transmission, has a moment generating function given by

$$E\left[\exp\left(-zT_{\text{delay}}\right)\right] = E\left[\exp\left(-zT_{\text{async}}\right)\right] \cdot \frac{1-E\left[\exp\left(-zT_{\text{sync}}\right)\right]}{zE\left(T_{\text{sync}}\right)} \cdot \frac{z\left(1-\lambda E\left(T_{\text{sync}}\right)\right)}{z-\lambda\left(1-E\left[\exp\left(-zT_{\text{sync}}\right)\right]\right)} \tag{86}$$

with associated mean delay given by

$$E\left(T_{\text{delay}}\right) = T_{\text{async}} + \frac{1}{2}T_{\text{sync}} + \frac{\lambda T_{\text{sync}}^2}{2\left(1-\lambda T_{\text{sync}}\right)}. \tag{87}$$

For example, if $T_{\text{sync}} = (S-1)T_{\text{async}}$, then

$$E\left(T_{\text{delay}}\right) = T_{\text{async}}\left[1+\frac{S-1}{2}\left(+\frac{\lambda(S-1)T_{\text{async}}}{1-\lambda(S-1)T_{\text{async}}}\right)\right]. \tag{88}$$

Note that for light traffic, $\lambda(S-1)T_{\text{async}} \ll 1$, the mean data transmission time delay is augmented by a factor of roughly $\frac{1}{2}S$ which could be considerable.

IX. Summary and Closing Comments

Our goal was to study properties of the statistics of R_n, the amount of data waiting to be transmitted at the start of frame n, which was governed by the recursion

$$R_{n+1} = \max[0, R_n + D_n + V_n - S]. \tag{89}$$

Two phenomena are present which can impact traffic handling characteristics

- fluctuations in D_n, V_n, or, in other words, the *shape* of the distribution
- correlation from frame to frame of D_n, V_n, or, in other words, the relative *time scale* of the different types of traffic.

Under certain specific assumptions, we showed that

$$\lim_{n\to\infty} \text{prob}[R_n > k] \approx \text{constant} \times \omega^k \tag{90}$$

and explicitly evaluated the constant and ω. The transient behavior of the system is related to time scales of the order of $1/\omega$ which is also of great interest.

How can we combat the phenomena here? There are a variety of methods. One approach is to dedicate a given amount of transmission capacity to voice and a given amount to data, and engineer these two systems separately. The voice transmission capacity could be chosen to meet a lost calls cleared grade of service, while the data transmission capacity could be chosen to meet a delayed message grade of service. The problem in the example here was the transmission capacity was shared or pooled, and the voice swamped the data. A second approach is as follows. In many applications, the amount of bits transmitted due to voice is much much greater than that due to data: rather than demand the relatively small amount of data be delayed by the relatively large amount of voice, why not reverse the priorities? Simply give the data higher priority over the voice, and if there is a surge in voice, drop the voice bits and not the data bits. This is possible because *people* are generating the voice, and will detect problems (clicks, unusual sounds, spurious signals, and so on) and will retry ("What did you say?" or "Could you repeat that, we seem to have a bad connection!")

These are only a few of the possible approaches for multiplexing voice and data together over a shared link and providing acceptable service for each. The purpose of our analysis is to uncover the problem, which might not be obvious *a priori*.

Acknowledgment

The authors gratefully acknowledge the encouragement shown by Prof. M. Decina of the University of Rome toward pursuing this work. All errors, omissions, and other oversights are the sole responsibility of the authors.

REFERENCES

[1] R. R. Anderson, G. J. Foschini, and B. Gopinath, "A queuing model for a hybrid data multiplexer," *Bell Syst. Tech. J.*, vol. 58, no. 2, pp. 279–300, 1979.

[2] E. Arthurs and B. W. Stuck, "A theoretical traffic performance analysis of an integrated voice data virtual circuit packet switch," *IEEE Trans. Commun.*, vol. COM-27, no. 7, pp. 1104–1111, 1979.

[3] P. E. Boudreau, J. S. Griffin, Jr., and M. Kac, "An elementary queuing problem," *Amer. Math. Mon.*, vol. 69, pp. 713–724, 1962.

[4] P. T. Brady, "A technique for investigating on-off patterns of speech," *Bell Syst. Tech. J.*, vol. 44, no. 1, pp. 1–22, 1965.

[5] —, "A statistical analysis of on-off patterns in 16 conversations," *Bell Syst. Tech. J.*, vol. 47, no. 1, pp. 73–91, 1968.

[6] —, "A model for generating on-off speech patterns in two-way conversation," *Bell Syst. Tech. J.*, vol. 48, no. 7, pp. 2445–2472, 1969.

[7] K. Bullington and I. M. Fraser, "Engineering aspects of TASI," *Bell Syst. Tech. J.*, vol. 38, no. 3, pp. 353–364, 1959.

[8] S. I. Campanella, "Digital speech interpolation," *COMSAT Tech. Rev.*, vol. 6, no. 1, pp. 127–158, 1976.

[9] D. Cohen, "A protocol for packet switching voice communication," *Comput. Networks*, vol. 2, pp. 320–331, 1978.

[10] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*. Reading, MA: Addison-Wesley, 1967, pp. 159–190; eq. (35a), p. 174.

[11] R. B. Cooper, *Introduction to Queueing Theory*, 2nd ed. Amsterdam, The Netherlands: North Holland, 1981.

[12] G. Coviello and P. A. Vena, "Integration of circuit packet switching in a SENET (slotted envelope network) concept," in *Conf. Rec., Nat. Telecommun. Conf.*, New Orleans, LA, Dec. 1975, pp. 42.12–42.17.

[13] G. J. Coviello, "Comparative discussion of circuit versus packet switched voice," *IEEE Trans. Commun.*, vol. COM-27, no. 8, pp. 1153–1159, 1979.

[14] O. Enomoto and H. Miyamoto, "An analysis of mixtures of multiple bandwith traffic on time division switching networks," in *Proc. 7th Int. Teletraffic Conf.*, Stockholm, Sweden, June 13–20, 1973, pp. 635/1–635/8.

[15] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. I, 3rd ed. New York: Wiley, 1968, p. 393.

[16] —, *An Introduction to Probability Theory and Its Applications*, Vol. II, 2nd ed. New York: Wiley, 1971, pp. 408–412.

[17] M. J. Fischer, "Data performance in a system where data packets are transmitted during voice silent periods—Single channel case," *IEEE Trans. Commun.*, vol. COM-27, no. 9, pp. 1371–1375, 1979.

[18] B. S. Garbow, J. M. Boyle, J. J. Dongarra, and C. B. Moler, *Matrix Eigensystem Routines — EISPACK Guide Extension* (Lectures Notes in Computer Science, 6). New York: Springer-Verlag, 1977, Sect. 2.1.9.

[19] L. A. Gimpelson, "Analysis of mixtures of wide- and narrow-band traffic," *IEEE Trans. Commun.*, vol. COM-13, no. 3, pp. 258–266, 1965.

[20] I. Gitman, W-N Hsieh, and B. J. Occhiogrosso, "Analysis and design of hybrid switching networks," *IEEE Trans. Commun.*, vol. 29, no. 9, pp. 1290–1300, 1981.

[21] J. G. Gruber, "Delay issues in integrated voice and data transmission," *IEEE Trans. Commun.*, vol. COM-29, no. 6, pp. 786–800, 1981.

[22] E. Harrington, "Voice/data integration using circuit-switched networks," *IEEE Trans. Commun.*, vol. COM-28, no. 6, pp. 781–793, 1980.

[23] The IMSL Library, "Routine ZANLYT," IMSL, Inc., Houston, TX, 1979.

[24] D. H. Johnson and G. C. O'Leary, "A local access network for packetized digital voice communication," *IEEE Trans. Commun.*, vol. COM-29, no. 5, pp. 679–688, 1981.

[25] L. Katzschner and R. Scheller, "Probability of loss of data traffics with different bit rates hunting one common PCM channel," in *Proc. 8th Int. Teletraffic Conf.*, Melbourne, Australia, 1976, pp. 525-1–525-8.

[26] K. Kuemmerle, "Multiplexer performance for integrated line and packet switched traffic," in *Proc. 2nd Int. Comput. Commun. Conf.*, Stockholm, Sweden, Aug. 12–14, 1974, pp. 505–515.

[27] B. Maglaris and M. Schwartz, "Performance evaluation of a variable frame multiplexer for integrated switched networks," *IEEE Trans. Commun.*, vol. COM-29, no. 6, pp. 800–807, 1981.

[28] —, "Optimal fixed frame multiplexing in integrated line- and packet-switched communication networks," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 2, pp. 263–273, 1982.

[29] N. F. Maxemchuk, "A variation on CSMA/CD that yields movable TDM slots in integrated voice/data local networks," *Bell Syst Tech. J.*, vol. 61, no. 7, pp. 1527–1550, 1982.

[30] H. Miyahara and T. Hasegawa, "Integrated switching with variable frame and packet," in *Conf. Rec., Int. Conf. Commun.*, Toronto, Ont., Canada, June 1978, pp. 20.3.1–20.2.5.

[31] O. A. Mowafi and W. J. Kelly, "Integrated voice/data packet switching techniques for future military networks," *IEEE Trans. Commun.*, vol. COM-28, no. 9, pp. 1655–1662, 1980.

[32] M. Neuts, "Some explicit formulas for the steady state behavior of the queue with semi-Markovian service times, *Adv. Appl. Prob.*, vol. 9, pp. 141–157, 1977.

[33] —, "Queues solvable without Rouche's theorem," *Oper. Res.*, vol. 27, no. 4, pp. 767–781, 1979.

[34] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models — An Algorithmic Approach*. Baltimore, MD: Johns Hopkins Univ. Press, 1980.

[35] E. Port, K. Kuemmerle, H. Rudin, C. Jenny, and P. Zafiropulo, "A network architecture for the integration of circuit and packet switching," in *Proc. Int. Conf. Comput. Commun.*, Toronto, Ont., Canada, Aug. 1976, pp. 505–514.

[36] M. J. Ross, A. C. Tabbot, and J. A. Waite, "Design approaches and performance criteria for integrated voice/data switching," *Proc. IEEE*, vol. 65, no. 9, pp. 1283–1295, 1978.

[37] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigen System Routines — EISPACK Guide*, 2nd ed. (Lecture Notes in Computer Science, 6) New York: Springer-Verlag, 1976, sec. 2.1.10.

[38] L. Takacs, "On fluctuation problems in the theory of queues," *Adv. Appl. Prob.*, vol. 8, no. 3, pp. 548–583, 1976.

[39] C. Weinstein, M. Malpass, and M. J. Fischer, "Data traffic performance of an integrated circuit and packet switched multiplex structure," *IEEE Trans. Commun.*, vol. COM-28, no. 6, pp. 873–878, 1980.

Edward Arthurs, for a biography, see p. 959 of the November 1983 issue of this JOURNAL.

Barton W. Stuck (S'67–M'72), for a photograph and biography, see p. 701 of the November 1983 issue of this JOURNAL.