# SUBFRAME SWITCHING IN DATA COMMUNICATIONS

**G. J. Foschini, B. Gopinath and J. F. Hayes**
**Bell Telephone Laboratories, Incorporated**
**Holmdel, New Jersey 07733**

## ABSTRACT

An alternative to packet switching, designated as subframe switching is studied. In this technique the traffic flowing between a pair of nodes in the transport network is given a fixed capacity by means of time division multiplexing. This flow between two nodes, viewed as a single commodity, may consist of packets from many different terminals or ports connected to a node. The advantage of this technique over conventional packet switching lies in simplicity of switching at tandem nodes and in simplicity of operation, particularly in flow control. The disadvantage of subframe switching is that less efficient use is made of transmission capacity. An analysis of the technique is made with the aim of quantifying delay performance and buffering required. A preliminary study of the relative switching complexity of packet and subframe switching is undertaken. Numerical results indicate the relative merits of each method.

## I. INTRODUCTION

In considering switched networks devoted exclusively to data, it is informative to view the network as consisting of two distinct parts. One component encompasses the local distribution networks, typically terminal to node and return, while the second component includes the internodal transport network (see Figure 1). In hierarchical networks one can associate the second component with a higher level network that connects regions. This second component which is often called the communications subnet is the subject of our investigation. We shall be concerned with design tradeoffs between transmission capacity on the one hand and switching and operational complexity on the other.

The study of data switching and multiplexing in this subnet is usually couched in terms of packet or message switching vs line switching (see for example References 1-3). Broadly speaking, packet switching may be viewed as a technique for sharing facilities among many users whose individual demands are low. Each user is allocated only the amount of transmission capacity that is instantaneously required. Overhead in the form of packet addressing is incurred and each packet must be handled individually. In contrast, line switching, by handling all packets produced in a call as an aggregate, reduces addressing overhead and processing. The penalty is less efficient use of transmission capacity. In the sequel we shall examine a technique whose motivation is to trade transmission capacity for processing and operational complexities. We designate this technique as subframe switching.

In subframe switching, transmission capacity is dedicated to all of the traffic between a particular source-sink pair of nodes. This traffic is composed of packets addressed to individual terminals at the destination nodes. The capacity allocated to a pair of nodes is slowly varying, responding, not to the originations and terminations of individual data sessions, but to large scale modifications of traffic patterns. Such changes occur because of the progression of the busy hour across time zones and because of node and link failures. We may think of the rate of change of transmission capacity allocations as being similar to the rate of change of routing tables in packet switched networks.

Since subframe switching does not respond to instantaneous user demand it is less efficient than packet switching with regard to the utilization of transmission capacity. In order to assess this effect, a mathematical model of subframe switching was developed and analyzed. Performance in terms of delay and storage required is calculated for a particular set of examples of capacity and user requirements. For purposes of comparison, performance in a packet switched network is calculated for corresponding cases. (Specifically we assume that mean delay performance is dominated by those transactions involving three hop paths.) In order to gain insight into processing complexity the operations required at a particular node are broken down into rudimentary components. The number of these basic operations that must be carried out in a second form the basis of comparison.

## II. PACKET SWITCHING (DATAGRAM AND VIRTUAL CIRCUIT)

Consider packet switching as it is implemented in the ARPA network, [4] which is the archetype of packet switched networks. Messages generated by terminals attached to HOST computers enter the network through an Interface Message Processor (IMP). This HOST computer may be an adjunct to the IMP which does local distribution functions. The HOST computer formats the message into packets with a certain maximum length. Appended to each of the packets is address information. The packets travel through the network in a queue-and-forward fashion from IMP-to-IMP. Each IMP on a packet's itinerary examines the packet address and determines the next IMP. The receiving IMP checks a newly arrived packet for errors and buffers it for processing. If both of these operations are satisfactory, a positive acknowledgment is sent back to the transmitting IMP. The transmitting IMP buffers a packet until this acknowledgment has been received. Flow control is accomplished through end-to-end acknowledgments.

In routing the packet through the network ARPA employs a dynamic routing scheme where all packets are routed independently. Fixed routing is another possible scheme where all packets which are part of the same call take the same path. In the case of dynamic routing one must avoid looping of the packets and provision must be made to reassemble packets that arrive out of order. In the case of fixed routing, information on calls in existence must be part of the routing information that is stored at intermediate switching nodes. A further embellishment is the possibility of variable length packets. In this case overhead information indicating packet length must be carried along either explicitly or implicitly through an end of message character.

The virtue of packet switching is manifest when many bursty sources share the same transmission facilities, since these facilities are given to a user only upon demand. This stands in contrast to line switching where bandwidth is dedicated to a user for the duration of his call.

## III. SUBFRAME SWITCHING ALTERNATIVE

The savings in transmission achieved by packet switching is not without some cost. Since each packet is treated as a separate entity it is switched through all nodes that it passes through on the way to its destination. Again there is the contrast to line switching for which entire calls, which may consist of a number of packets, are treated as separate entities. We propose a data multiplexing technique which is intermediate between packet and line switching. Begin by considering all of the data sources at a node to be grouped into disjoint classes. Members of the same class have the same destination node in the transport network. Within a class, user data is blocked into fixed size packets to which are appended addresses indicating the destination terminal or port at the destination node. All of the user classes share transmission facilities by using time division multiplexing. Flow on outgoing trunks is blocked into fixed size frames. A user class is assigned a fixed portion of the frame giving rise to the name subframe. The technique is illustrated on Figure 2a for four user classes.

At intermediate switching nodes time slot interchange is used to switch data onto the appropriate trunks (see Figure 2b). The addressing accompanying packets is not used until the destination node is reached. At the destination node the addressing allows the distribution of packets to individual terminals.

Although one pays a price in transmission rate for subframe multiplexing in comparison to the packet multiplexing schemes of the previous section, switching is simplified. (We shall quantify these statements presently.) In the case of packet switching discussed earlier each packet is handled as a separate entity. Moreover, prior to its arrival the switch has no knowledge of its destination. Thus the switch must read the address header for each incoming packet and in real time decide upon the disposition of the packet. In contrast, for frame multiplexing, an entire block of packets can be switched at one time. Furthermore the switch knows the disposition of each packet before it arrives and no real time processing of addresses is required. In the case of subframe switching we assume that link acknowledgments are not required since there is no uncertainty as to the availability of buffering at a node. This stands in contrast to packet switching where flow is random and buffering may not be available at a particular node. We assume also that transmission is reliable enough so that link-by-link error detection is unnecessary.

Perhaps the main advantage of subframe multiplexing lies in the area of flow control. In packet switched networks, flow is random and routing decisions can be made in a distributed fashion throughout the network. Thus congestion at a node or on a link can be caused by traffic originating at distant nodes. This congestion which may be caused by momentary surges, may propagate to cause serious disruptions of network operations. In the case of subframe multiplexing, flow throughout the transport network is entirely controllable. The random buffer levels due to random traffic flow is due solely to traffic locally generated. Thus congestion control is facilitated since the source of congestion is easily pin-pointed and its effect confined.

While high usage trunks can be used to provide dedicated paths for commodities, subframe switching is advantageous because the capacities required for commodities can respond to slow traffic fluctuations as well as status changes. Moreover, with subframe switching there is an additional resource sharing advantage associated with the flexible allocation of TDM channels as opposed to using trunks of standard speeds.

With respect to the utilization of transmission capacity, subframe multiplexing becomes less effective as the number of user classes increases since transmission capacity is spread among too many users (see Section IV below). For classes that do not generate much traffic, as an alternative to allocation of a TDM channel, provision can be made for conventional packet switching. A part of a frame can be dedicated to this packet traffic. This subframe can be switched to a portion of the node where packets are handled in the conventional fashion of a packet switched network. Addresses are examined and the next step of the itinerary is chosen. The usual acknowledgment protocols can be implemented on a time division multiplexed system.

As the traffic changes, the sizes of frames and the portions of frames that are allocated to a particular user class can be changed. These changes can be conveyed around the transport network so that synchronism is maintained. Similarly changes in bandwidth allocation would follow changes in the topology of the operational network due to planned evolution of facilities or malfunction. <u>Distributed intelligence is not required at transit nodes to effect</u>

the transport of packets during quiescent operation. With subframe switching the need for direct intelligent response to data session origination or termination is obviated.

## IV. MATHEMATICAL MODELS

In this section we shall determine the transmission capacity and switching complexity required as a function of traffic levels and performance requirements. We first consider the performance as measured by delay and buffer requirements as a function of transmission capacity for subframe multiplexing. Then we consider the requirements for ordinary packet switching by means of a particular mathematical model.

### A: Distribution of Buffer Space - Subframe Multiplexing

The analysis of subframe multiplexing is based on a queuing model studied by Boudreau, Griffin and Kac[5] (see Figure 3). The model that they treated concerns a helicopter that departs from a landing pad at fixed, equally spaced intervals. The helicopter always arrives empty and has a fixed maximum capacity n. It is assumed that the arrival rate of customers is random; the only restriction being that arrivals are statistically independent from interval-to-interval. The problem addressed by Boudreau et al. is to find the distribution of the number of customers waiting when the helicopter departs. It can be shown that the probability generating function for the number of customers waiting in line just before boarding the helicopter is given by

$$G(z) = \frac{f(z)(z-1)(n-f'(1)) \prod_{j=2}^{n} (z-C_j)}{(z^n - f(z)) \prod_{j=2}^{n} (1-C_j)} \qquad (1)$$

where f(z) is the probability generating function of the customer arrival process and f'(1) denotes the first derivative of f(z) at z = 1. $C_i$; i = 2,3,..n are the n - 1 roots of the equation $z^n$ f(z) = 0 inside the unit disk. One of the contributions of Boudreau et al is to show that this equation has n distinct roots within the unit disk.

In order to find these roots we transform the forgoing equation into the set of n equations

$$z - \exp(-(\lambda/n)(P(z) - 1) + 2\pi ik)\ 0; \quad k = 0,1,...,n - 1$$

where $i = \sqrt{-1}$ . Each of these equations has a simple root within the unit disk. In order to find these roots we have used a simple Newton-Raphson iteration beginning at the origin.

We have found that for a large range of cases a solution is reached in a small number of steps. One suspects that it is generally true that Newton-Raphson converges in this case. This remains a point for further study.

This model can be used in the study of the performance of subframe multiplexing. We assume that messages arrive for multiplexing at a Poisson rate. (As mentioned above more general arrival processes can be accommodated.) Further we assume that these messages have a random number of packets. We assume that messages that arrive after the beginning of a frame are held over until the next frame (see Figure 2). The probability generating function of the packet arrivals under this model is

$$f(z) = \exp(\lambda_F(P(z) - 1)) \tag{2}$$

where $\lambda$ is the arrival rate of messages, $T_F$ is the duration of a multiplexing frame and P(z) is the probability generating function for the length of a message. We shall measure the length of a message in terms of a general quantity which we shall designate as a data unit. For example, data units may be in terms of bits, characters or 1000 bit packets. If a message length is geometrically distributed in terms of data units, its probability generating function is

$$P(z) = \frac{(1-p)z}{1-pz} \tag{3}$$

so that Pr[i data units in a message] = $(1-p)p^{i-1}$.

The foregoing gives the number of data units from a particular user class that are in the buffer when the portion of the frame allocated to that class commences. Note that this portion of the frame can accommodate n data units. This gives the maximum amount of buffer space that is required by a particular class. We are also interested in the delay suffered by a message i.e., the time interval between message arrivals and its complete multiplexing on the outgoing line.

## B: The Delay Distribution - Subframe Multiplexing

We may distinguish four basic components comprising the delay of a message. Assume that a frame starts at time t = 0, and that a message arrives at time $t = \tau$. Even if there are no previously arrived data units in the system, the message delay is $T_F - \tau$ plus the time required to multiplex a message. In general the delay of a message will be longer since there may be data units remaining from the previous cycle and data units newly arrived in the time interval $\tau$. The total number of new packets that must be multiplexed before a particular message can be multiplexed is given by

$$P_T = P_R + P_N + P_M \tag{4}$$

where $P_R$ is the number of packets remaining from a previous frame, $P_N$ is the number of packets arriving since the start of a frame and $P_M$ is the number of packets in the message. These quantities are statistically independent of each another. The distribution of $P_R$ can be calculated from the generating function in equation (1) (see below). The generating function for $P_N$ is given by Equation (2) with $\tau$ replacing $T_F$. The generating function for $P_M$ is given in Equation (3) for the case of geometrically distributed messages.

The calculation of delay is complicated by the fact that the random variable $P_N$ is a function of the random variable $\tau$ which is itself uniformly distributed in the interval $(0, T_F)$.

Taking these facts into account, the mean, the variance and percentile points of delay were computed numerically. One could calculate an explicit formula for these quantities, however, little insight would be gained thereby. Instead we shall outline the computational procedure by which delay is computed. Recall that the probability generating function of the number of data units present before the server arrives is G(z) and let $g_i$ = Pr[i data units waiting]. It can be shown that F(z), the probability generating function for the number of data units remaining after n have been removed, is

$$F(z) = z^{-n}\left| \sum_{i=0}^{n} g_i (z^n - z^i) + G(z) \right| . \tag{4}$$

The values of $g_i$ can be found quite easily by means of the Fast Fourier Transform (FFT). Now suppose $\tau$ is fixed, the probability generating function of the number of data units to be multiplexed on the line is given by the product of the generating function for $P_R$ , $P_N$ and $P_M$ . Unfortunately, the time required to multiplex these data units is not directly expressible in terms of these generating functions. Thus for each value of $\tau$ we use the FFT to compute the inverse transform of this composite probability generating function.

From this we can compute the probability distribution of delay. Let $P_T$ be the total number of packets to be multiplexed. The time required to multiplex $P_T$ packets is

$$T_M = \left| \frac{P_T}{n} \right|^{-} T_F + \left\{ P_T - n \left| \frac{P_T}{n} \right|^{-} \right\} \frac{T_F}{num}$$

where $[X]^{-}$ is the largest integer less than X and where num is the number of data units per frame allocated to all users. The total delay for a given value of $\tau$ is

$$D = T_M + T_F - \tau.$$

The moments and the distribution of delay are given from the knowledge that $\tau$ is uniformly distributed in the interval $(O,T_F)$.

### C: The Average Delay Computation for Ordinary Packet Switching

The computation of average delay for a practical packet network is a difficult problem and the specifics of a networks operations are basic to a comprehensive study. It is far beyond the scope of this paper to catalogue various methods of operation and then undertake a thorough delay analysis of each. Instead, for illustrative purposes we settle here for a rough estimate of delay.

To obtain the estimate for average delay we assume that the first node in a path can be modeled as an M/G/1 queue with the service time distribution being the distribution of message lengths. Moreover, we assume that longest path in the network is a three hop path. We take three times the average waiting time in the first node as the estimate of path delay. While it is true that many messages will go through shorter paths, the time average delay of messages going over longer paths must be satisfactory. We do not consider averages taken over all paths as a valid measure of performance.

### D: Relative Processing Complexity

We consider now the complexity of switching that is required for packet multiplexing and for subframe multiplexing. While we have no difficulty using bits per second as a valid measure for transmission capacity it is difficult to find a comparable measure for switch or processor complexity. The difficulty is encountered whether one uses the subframe idea or not since the details of what should be done at a node and how it is to be done remain somewhat unsettled. A way of measuring processing complexity is to simply count the operations per second at a node for each technique. We shall employ this measure to assess the relative switching complexity of the two ways of multiplexing. The counting method may provide a pessimistic appraisal of subframe switching since it does not reduce the weight of routine periodic operations (that are likely candidates for relegation to hardware) relative to complex operations (requiring software implementation).

Consider a stream of packets entering a node in a packet multiplexing system. On each packet the following operations must be performed.

1. The address or header information must be accessed.
2. This address must be deciphered.
3. Either an internal counter signals the end of the packet or an end of message character is recognized.
4. The packet is loaded into a buffer.

5. Positive acknowledgment to transmitting node is transmitted.
6. After receiving a positive acknowledgment the corresponding stored packet is released.
7. Increment to next packet.

In the case of subframe multiplexing, action is taken by the processor only at the subframe intervals. At these points all that is required is to switch a user class in steps 4) and 7) above. In the sequel we shall use these estimates of the number of operations per switch in order to compare the complexity of switching in packet multiplexing and subframe multiplexing. We recognize the arbitrary nature of the foregoing measure. However, we felt that it would serve as a measure of complexity in a preliminary study such as this.

An approach to measuring switching complexity employing information theoretic notions could be used in future studies. For example the task of the nodes certainly includes relaying packets to their destination and providing confirmation of satisfactory packet transport. To provide these functions the node needs the intelligence to ask questions like:

> Where does this packet go?
>
> Has an acknowledgment of a transmitted packet arrived yet?

The questions can be construed as determinations of outcomes of chance experiments of known entropy. The rate such questions are asked, weighted by their respective entropies, assign a bit per second measure to processing power (needed or) provided.

## V. NUMERICAL EXAMPLES

In this section the relative merits of packet switching and subframe switching are evaluated with respect to transmission capacity and switching complexity by means of numerical examples. The primary focus is upon messages which have an average duration of 20 eight bit characters. This average seems to be typical of the kind of traffic that places the highest demands upon data networks of the class under consideration. With this average we consider two distributions of messages, constant length and geometric. In the former case we take a data unit to be 20 characters long. In the later case the length of a data unit is 4 characters and $p = 0.8$ (equation 3).

In the sequel we shall present curves of delay as a function of load. We shall compare the delay for subframe multiplexing with delay for packet multiplexing. Throughout we shall be concerned with the delay that is a function of traffic. Thus the analysis presented in Section III above is used to assess delay for subframe multiplexing. This delay is incurred at the point where data is multiplexed into a frame. As mentioned in the foregoing, it is assumed that delay in the packet switched network is computed under the assumption that a message traverse three nodes which behave as M/G/1 queues. Our focus is on traffic induced delays, consequently, we shall ignore delays such as propagation synchronization and timing. Our purpose is to make comparisons not to obtain absolute measures of delay.

In most of the cases that we shall consider the transmission rate between nodes on the transport network is 56 Kbps. We shall also consider a case where the rate is 1.544Mbps. We divide the frame in the case of subframe multiplexing into a basic component which we designate as a data unit. This is the minimum unit that can be switched at a node. We note also that it is not necessary that the portion of a frame dedicated to a particular commodity need not be greater then a packet.

The results of our computations of delay for subframe multiplexing are shown on Figures 4, 6 and 8. On all of these figures the abscissa is load (or equivalently message arrival rate) and the ordinate is average message delay. The parameter distinguishing different curves is frame structure. Consider for example the curve labeled (8,16) on Figure 4. Eight users are given 16 slots each consisting of 4 bytes. Thus the frame duration is 512 bytes or 4096 bits. Since the line speed is 56 kbps the duration of a frame is 73.14 msec.

The results are shown for average delay. We have also computed percentile of delay. A rough rule of thumb is that the 90 percent point is twice the average delay and the 99 percent point is four times the average delay.

On Figure 5 results are shown for a geometric distribution of messages with p = .8 (see Equation 3). The curves exhibit the shape that is characteristic of queuing systems. In carrying out the calculations it was assumed that 10 percent of the packet is devoted to overhead. As the load increases toward 1.0 the system becomes unstable. As the load decreases toward 0 the delay is asymptotic to a nonzero value which is the average time required to put a message on the line plus half the duration of a frame. In Figure 5 results are shown for packet multiplexing comparable to that shown on Figure 4 for frame multiplexing. The results are shown for different values of fixed packet size. Notice that there is a rather severe penalty for mismatching message length and packet size. For example, in the case of 32 byte packets, many packets are not entirely filled with data but with idle code. Nevertheless such packets require the same transmission resource as if they were filled with data.

A comparison of frame multiplexing and packet multiplexing for constant length messages is shown on Figures 6 and 7. The same phenomena govern delay in this case as in the case of constant length messages.

The values of delay of interest to us are in the range 100-200 msec for average delay. Generally speaking the packet switching technique yields lower values of average delay. Nevertheless, the advantage of the packet switching technique is not so great as to preclude other considerations. In fact with a mismatch between constant length packets and message lengths subframe multiplexing holds the advantage.

On Figure 8 results are shown for a 1.544 MBPS line which carry twenty four 56 kbps channels. The packets are assumed to be a constant 128 bytes in duration. This case or other line speeds are easily obtained from the basic computations.

Calculations of storage requirements for the two techniques were made for both types of multiplexing. For subframe multiplexing the method given in Section III was used with variable length messages (p=.8). For packet multiplexing the model used was an M/M/1 queue with the same message size and delay. Since the coefficient of variation of the message length is .89 (close to 1), we felt that it was not unreasonable to approximate its distribution as exponential. The M/M/1 model allowed us to easily calculate probabilities of overflow. Storage requirements for three tandem nodes were calculated with the distribution assumed to be independent from node-to-node. The results are shown in Figure 9 in the form of the capacity required to prevent overflow with probabilities .01 and .001. Again the results shown an advantage to packet multiplexing.

We come now to the comparison of the relative switching complexity for subframe and packet multiplexing. For subframe multiplexing the number of switches per second at a node is a simple function of the structure of the frame and the line rate. On Figure 10 we show the relationship of the number of switches per second as a function of the number of data units assigned to a user with the number of bytes per data unit as a parameter. Also shown are the number of operations that must be performed in a second under the assumptions on the number of operations per switch that were explained in the foregoing.

The switching rate for packet switching is a linear function of the rate of traffic flow which in turn depends upon the message rate generated by users and the number of packets per user. This is shown on Figure 11. Again the number of operations per second implied by the switching rate is also shown on Figure 11. Notice that the scaling is different than that on Figure 10 since we presume that packet multiplexing is inherently more complicated than packet switching.

On Figure 12 the results on delay and switching complexity are summarized. The absissa is average delay and the ordinate is number of processor operations at a single node. The curves illustrate the tradeoff between transmission capacity and switching complexity. The curves for subframe multiplexing tend to lie along the absissa indicating simpler switching gained at the expense of transmission capacity. Packet switching shows a contrary tendency with efficient utilization of transmission capacity achieved by more complex switching.

## VI. DISCUSSION

An alternative to packet switching for the transport network has been examined. The alternative, subframe multiplexing, is shown to be simpler than packet multiplexing in terms of switching complexity. This advantage is achieved at the cost of less efficient use of transmission capacity than packet switching.

Perhaps the most significant advantage of subframe switching lies in the realm of congestion and flow control. All of the queuing of packets for subframe switching takes place at the node where a packet enters the transport network. Thus controlling traffic causing congestion is considerably simplified. This contrasts with packet switching where congestion at a node can be caused by traffic originating at distant nodes.

In a sense subframe switching and packet switching represent two extremes in the multiplexing of data. There are intermediate techniques.[*] For example consider a variable frame technique. As in the case of subframe switching a fixed maximum capacity is assigned to each commodity. However, if a commodity has no traffic at a particular moment its data slots are assigned to an adjacent commodity. Additional overhead bits indicating a switch to a new commodity are required. Switching is more complex. However, more efficient use is made of transmission capacity. In future work in this area this technique can be compared to subframe and packet multiplexing.

## REFERENCES


1. Kümmerle K. and Rudin H., “Packet and Circuit Switching: Cost Performance Boundaries.” IBM Report RZ-805 Zürich Switzerland.

2. Miyahara H. et al. “A Comparative Evaluation of Switching Methods in Computer Communication Networks” Proc ICC, June 1975 San Francisco, Calif. pp. 6-6 to 6-10.

3. Itch K. et al. “An Analysis of Traffic Handling Capacity of Packet Switched and Circuit Switched Networks” Proc Third Data Communications Symposium, Florida, November 1973, pp. 29-37.

4. Roberts, L. G., “Resource Sharing Computer Networks,” Proc IEEE International Convention, New York 1969.


---

[*] For example see Reference 9.

5. Boudreau P. E. et al. "An Elementary Queuing Problem" Am. Math Monthly Vol. 69, No. 8, 1962, PP. 713-724.

6. Konheim A. G. and Meister B. "Service in Loop System" Journal of the ACM Vol. 19, No. 1, January 1972, pp. 92-108.

7. Cox D. R. and Smith W. L. Queues Chapman and Hall 1961.

8. Kleinrock, L., Communication Nets Stochastic Message Flow Delay, Dover, 1972.

9. de Amet, J. and Senders, R. W. "'Pacuit' Switching Combines Two Techniques in One Network" Computer Design, June 1976.

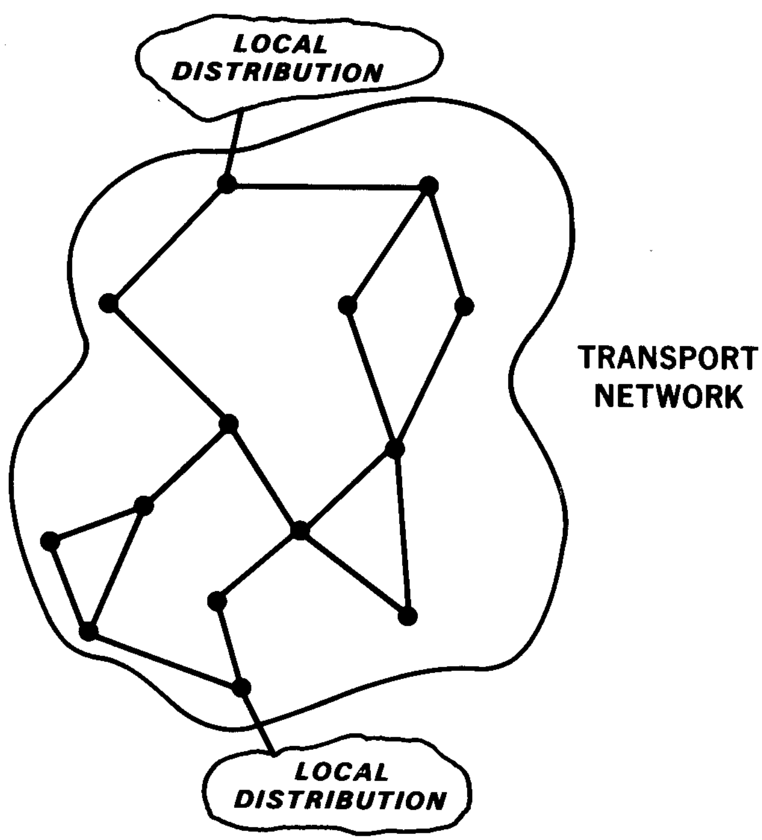



**Figure 1**

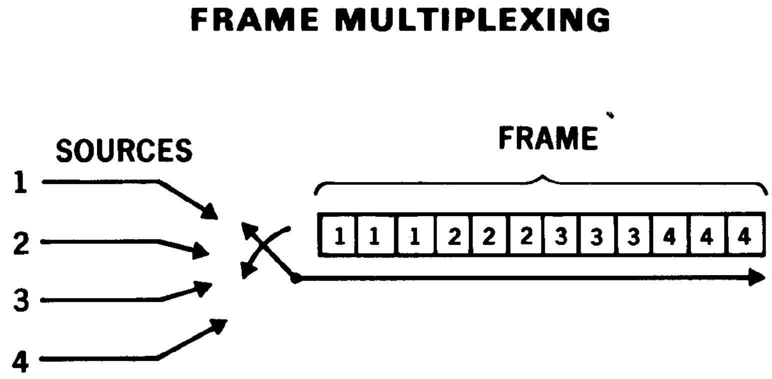



Figure 2A

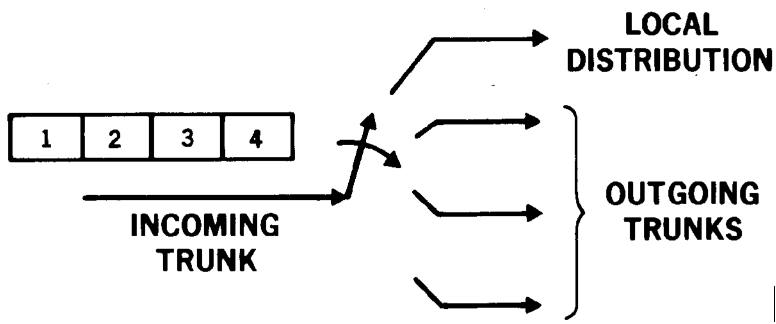



Figure 2B

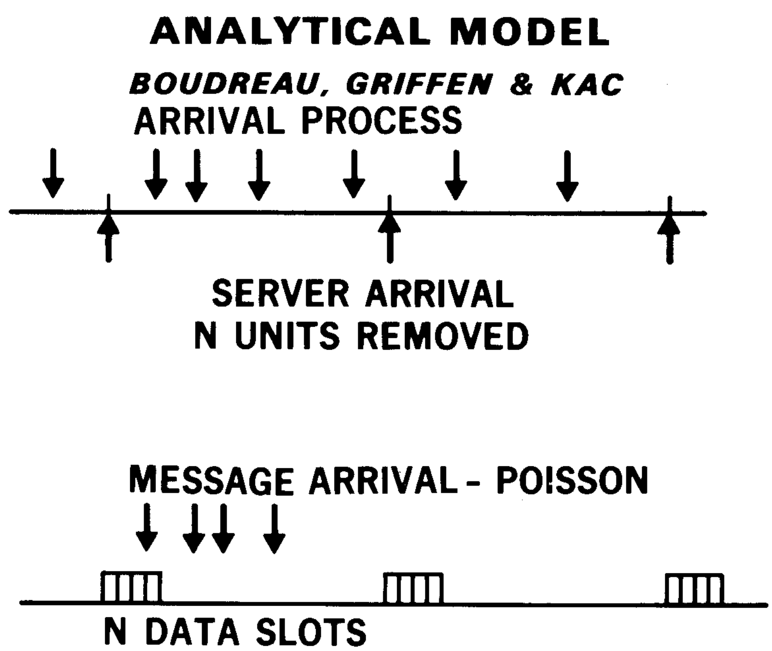



Figure 3

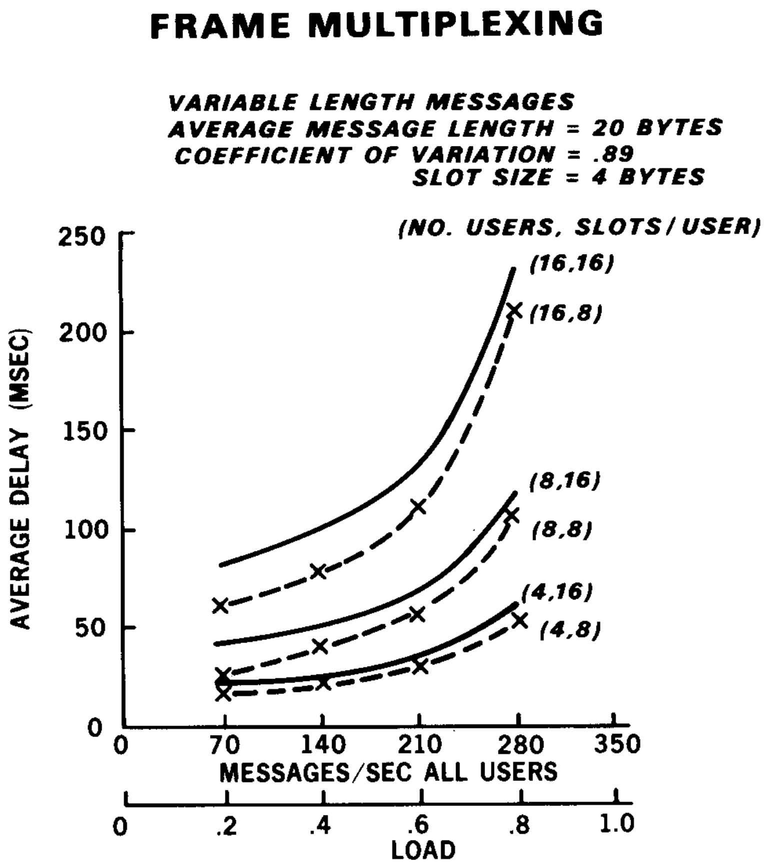



Figure 4

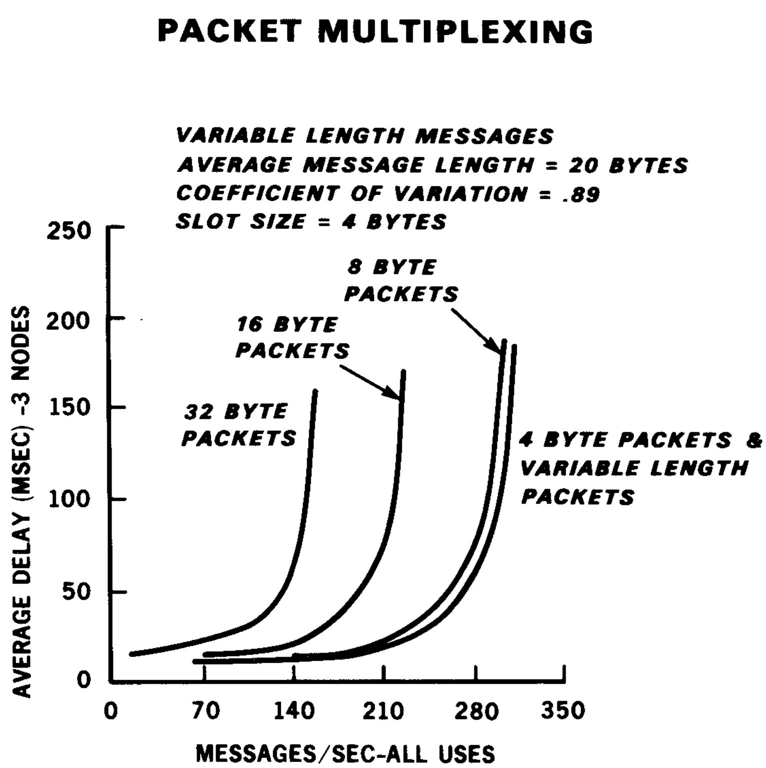



Figure 5

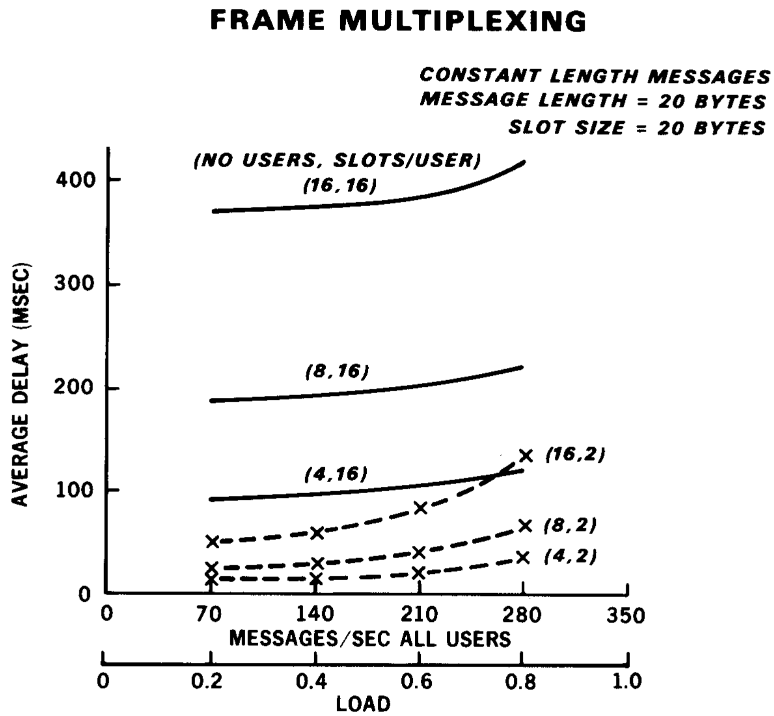



Figure 6

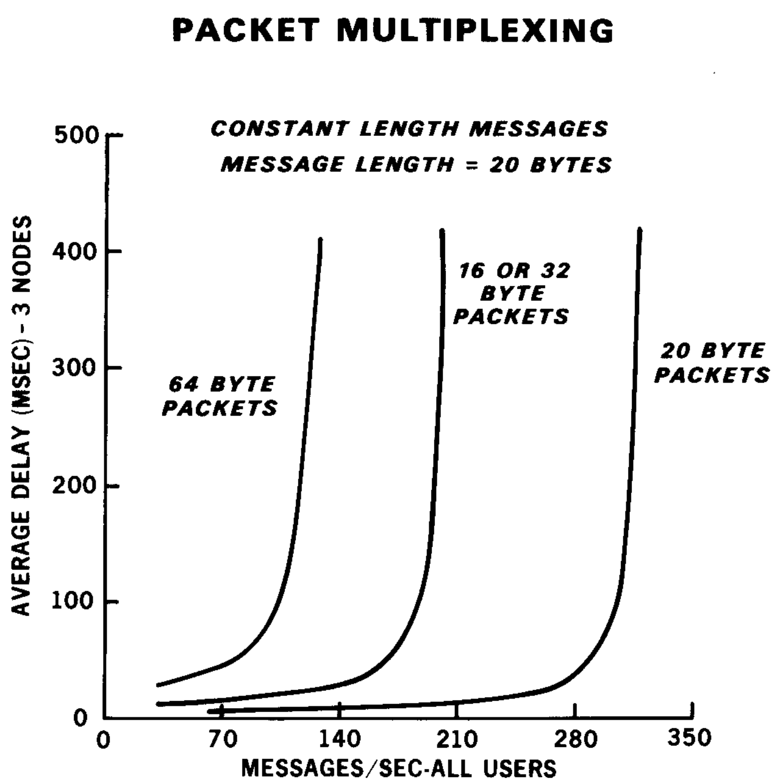



Figure 7

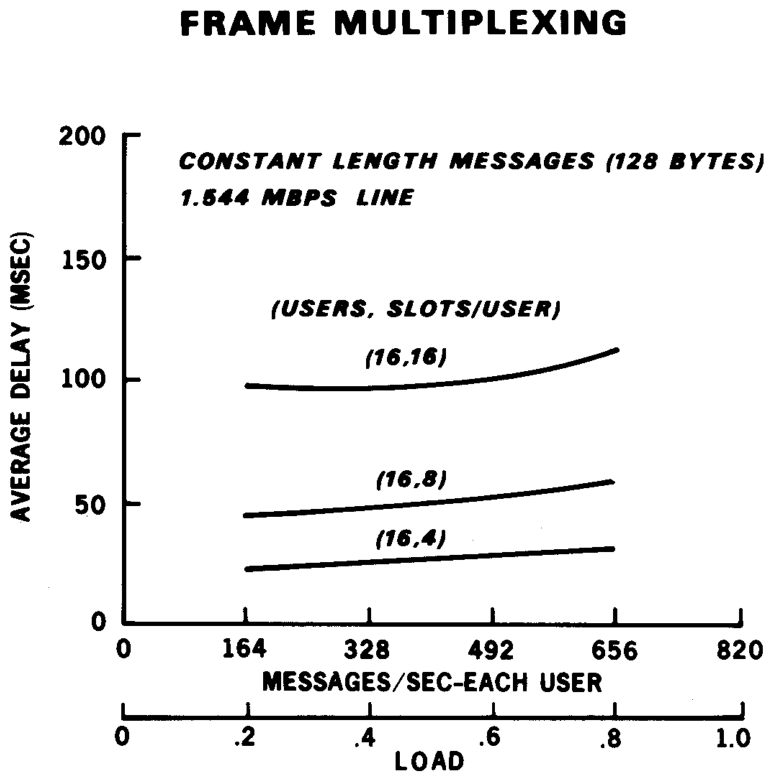



Figure 8

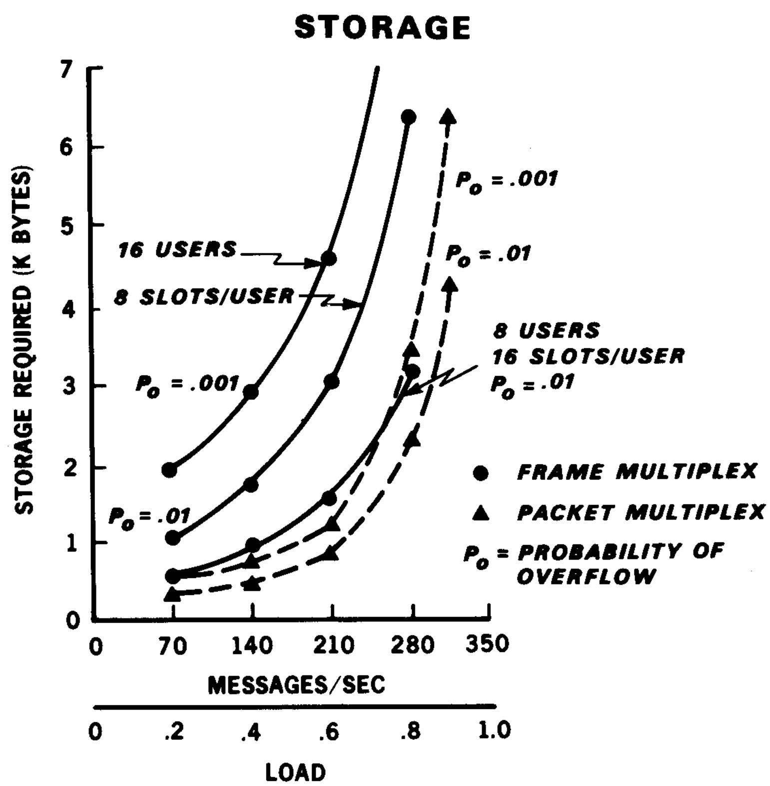



Figure 9

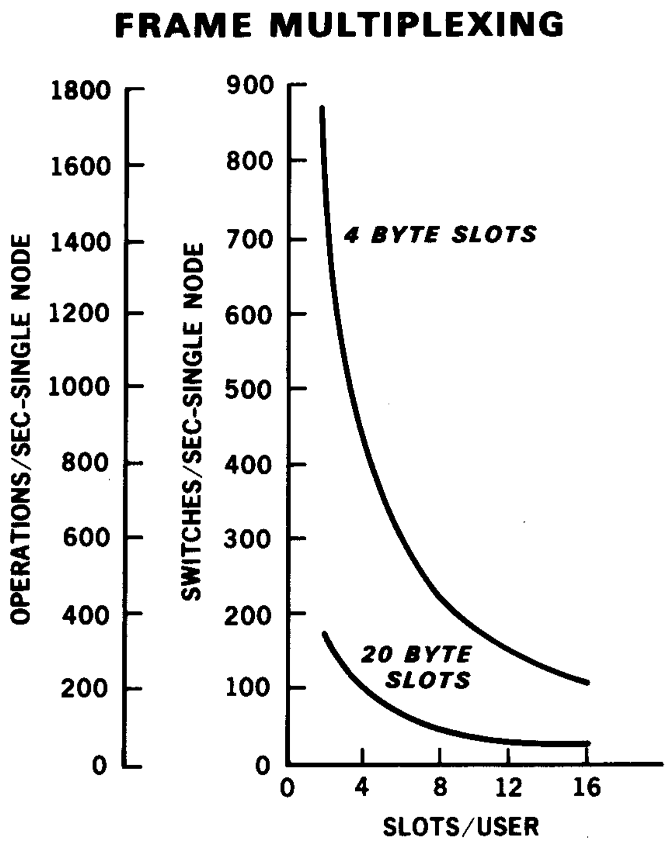



Figure 10

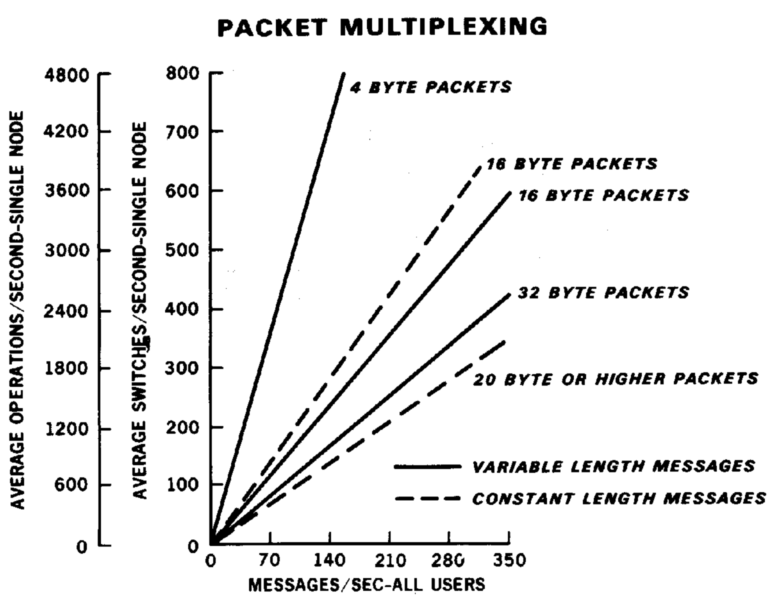
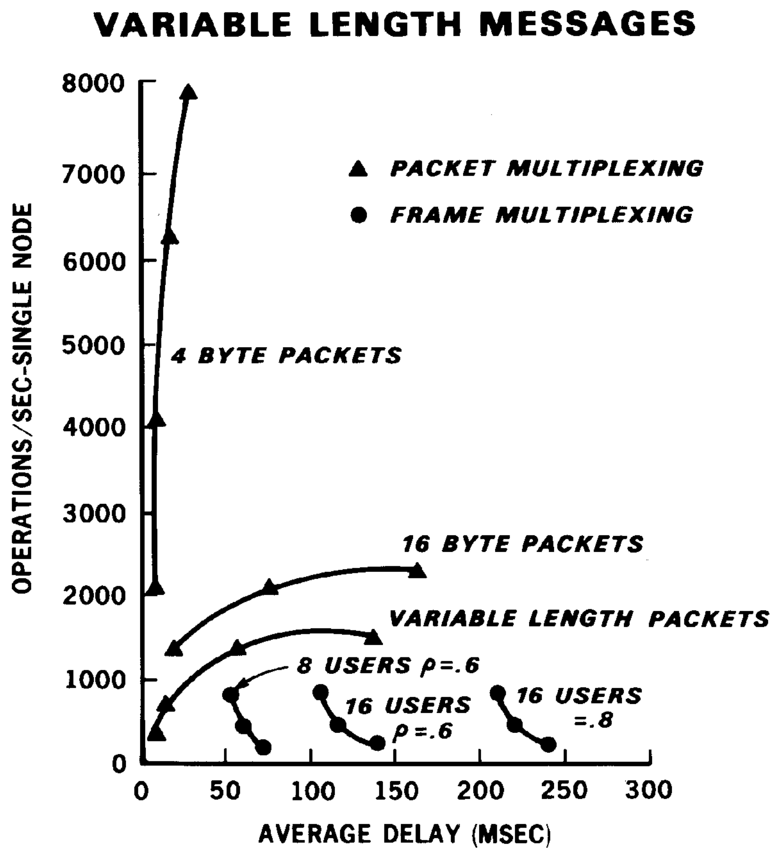



Figure 11



VARIABLE LENGTH MESSAGES
8000
7000
6000
5000
4000
3000
2000
1000
0
OPERATIONS/SEC-SINGLE NODE
PACKET MULTIPLEXING
FRAME MULTIPLEXING
4 BYTE PACKETS
16 BYTE PACKETS
VARIABLE LENGTH PACKETS
8 USERS ρ=.6
16 USERS
ρ =.6
16 USERS
=.8
0
50
100
150
200
250
300
AVERAGE DELAY (MSEC)


Figure 12