DATA NETWORKS: ANALYSIS AND DESIGN
THIRD DATA COMMUNICATIONS SYMPOSIUM

Sponsored by
IEEE Computer Society
and
The Association for Computing Machinery (SIGCOMM)
in cooperation with
IEEE Communications Society

St. Petersburg, Florida
November 13-15, 1973

DATACOMM 73

DATA NETWORKS: ANALYSIS AND DESIGN
THIRD DATA COMMUNICATIONS SYMPOSIUM

Sponsored by
IEEE Computer Society
and
The Association for Computing Machinery (SIGCOMM)
in cooperation with
IEEE Communications Society

St. Petersburg, Florida
November 13-15, 1973

Copies available from:
IEEE Computer Society
9017 Reseda Boulevard, #212
Northridge, California 91324

ACM
Order Department
1133 Avenue of the Americas
New York, New York 10036

IEEE
345 East 47th Street
New York, New York 10017

IEEE Catalog Number 73 CHO828-4C

Committee

CHAIRMAN

Dr. Raymond L. Pickholtz
Dept of Electrical Engineering
and Computer Science
George Washington University
Washington, D.C. 20006
202/296-1529

PROGRAM CHAIRMAN

Mr. Thomas Pyke
Bureau of Standards
Washington, D.C. 20234

VICE CHAIRMAN

Dr. Peter E. Jackson
Bell Laboratories
Room 2B434
Holmdel, N.J. 07733
201/949-2231

FINANCIAL MANAGER

Mr. Edward Fuchs
Bell Laboratories
Room 2C436
Holmdel, N.J. 07733
201/949-3959

TREASURER

Mr. Harry Hayman
IEEE Computer Society
Box 639
Silver Spring, M.D. 20901
301/439-7007

LOCAL ARRANGEMENTS CHAIRMAN

Prof. Richard Filipowski
University of South Florida
Tampa, Fla. 33620
813/974-2581 X273

STEERING COMMITTEE

Dr. Maurice Karnaugh
T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, N.Y. 10598
914/945-2238

Table of Contents

MODELING AN EXPERIMENTAL COMPUTER COMMUNICATION NETWORK

J. F. Hayes
Bell Telephone Laboratories, Incorporated
Holmdel, New Jersey 07733

Abstract

This paper reports the results of a performance study of an experimental computer communication network. The network is currently being designed and built in order to test concepts and techniques that may find future application. The network consists of synchronous digital transmission lines connected in loops to a Central Switch. User traffic enters the system through multiplexers connected to the synchronous lines. The Central Switch has the two-fold function of routing and controlling traffic.

Two multiplexing techniques were examined, Demand Multiplexing (DM) and Synchronous Time Division Multiplexing (STDM). In both techniques user messages are blocked into fixed size packets, prior to multiplexing on the line. The synchronous line can carry these packets at a minimum rate of 4000 packet slots per second. In STDM each terminal is assigned a packet slot which recurs periodically. In contrast, for DM, packets are multiplexed on the line asynchronously into unoccupied packet slots. Alternative implementations of the DM technique were studied, one where each terminal transmits and receives at a maximum rate of 4000 packets per second and another where the maximum rate is 2000 packets per second.

As part of its message handling function the Central Switch buffers messages in transit. This allows User Terminals to transmit and receive messages with a degree of independence from one another. However the terminal strategy affects the amount of storage required in the Central Switch. In order to prevent the loss of information when there is insufficient buffering there is a mechanism to inhibit traffic from User Terminals when the Central Switch's buffer is near overflow. Due to this control of traffic, there is a relationship between the amount of data that flows through the Switch and the amount of buffering in the Switch.

Simulation results showed that there was little difference in delay performance between the two implementations of DM. However an analysis comparing DM and STDM showed a great difference in performance for all but the very heaviest line loadings. This difference increases as the number of terminals sharing the T1 line increases.

Our study concentrated on two aspects of buffering in the Central Switch. We examined the relationship between throughput and the amount of storage available in the switch. The results of a simulation study showed that throughput can be quite high for all but minimal storage in the switch. Moveover, a strategy that dedicates buffers does quite well compared to common buffering. The second aspect of the study concentrated upon the User Terminal's strategy. Since each terminal acts independently, there may be strategies that make particularly high demands upon storage capacity in the Central Switch. An analysis showed that at the loadings where the system would be expected to operate, the user strategy in transmitting and receiving messages has little effect.

I. Introduction

An experimental computer communications network is currently being designed and built. The function of this network is to provide a flexible communications medium between computers, users, and peripheral devices. The network can accommodate sources with varying input-output rates and varying activity. Many of the components of the system employ techniques that are new. In order to gain insight into the operation of these components and thereby aid in design decisions, mathematical models were developed. The study of these models involved both analysis and simulation. The results are presented in the form of sets of curves.

The system under study consists of several T1 carrier lines,* configured as loops, connected to a Central Switch (see Figure 1). The system is accessed through Terminal Interface Units (TIU) connected between User Terminals and the T1 line. In addition to forming an interface between the user and the T1 line, the TIU also does signaling. This signaling plays a role in switching calls and controlling the traffic flow.

There may be a wide variation of users accessing the system, ranging from teletypes to high speed computer systems. The switch receives messages from all terminals and delivers messages to all addressed terminals so that any terminal in the system may communicate with any other terminal. All data passes through the Switch even when two terminals are on the same T1 line.

The T1 line operates at a rate of 1.544×10^6 bps. For purposes of synchronization and timing the bit flow is divided into frames of 193 bits, with a flow of 8000 frames per second. The multiplexing arrangement in the system under study is such that a "network frame" consists of two adjacent T1 frames. Figure 2 indicates schematically how the 386 bits of the pair of T1 frames are allocated. Of greatest interest to us is the 280 bits assigned to carry data. Of these 24 bits are set aside for header. The 24 bit signal packets convey control and routing information between the TIU and the Switch.

From the foregoing we see that the information carrying capability of the T1 loop is 4000 packets per second, each packet bearing 256 data bits, yielding a total information capacity of 1.024 Mbits per second. There are many strategies that can be used to divide this capacity among the terminals connected to the loop. We shall evaluate the performance of two of these, Synchronous Time Division Multiplexing and Demand Multiplexing. In Synchronous Time Division Multiplexing (STDM) each user terminal is assigned a particular packet slot which recurs periodically. The terminal may multiplex data into its slot and receives data only in this same slot. For example, if there are ten terminals on the T1 loop and each terminal receives the same service, a terminal may multiplex packets on the line at a constant rate of 400 packets per second.

---

* The T1 carrier line is a digital synchronous short haul transmission system operating at 1.544 M bit/sec rate.

In Demand Multiplexing (DM), packet slots are not assigned to a particular terminal. If a terminal has a packet to transmit to the Switch, it inserts the packet into the first slot that is empty. So that the Switch can sort packets according to their originating terminal, each packets has an address label in its 24 bit header. Similarly, information packets going from the Switch to the terminal are not ordered and a header is required for each packet. Furthermore, each T1U must be able to recognize packets addressed to it.

Once a packet has been multiplexed on the loop either from the Switch or from a terminal, it has priority over incoming traffic until it reaches its destination. A terminal must wait for an empty data packet slot before it can place a waiting packet onto the line. A terminal can place an outgoing packet into a slot from which it is removing an incoming packet.

We consider two implementations of the DM system, corresponding to the maximum speed at which terminals can transmit or receive. In the adjacent slot seizure implementation, terminals can transmit and receive at a 4000 packet per second rate. We consider an alternate implementation where the terminal is constrained to operate at a 2000 packet per second rate. In this case a terminal can only write into or read from alternate packet slots. This implementation is designated as alternate slot seizure.

A major component of the system is the Central Switch. The function of the Switch is to route and to control the flow of information. All messages generated at User Terminals pass through the Switch where they are passed on to their destination terminals. Now the operation of the system is such that as it may not be possible to deliver a message to its destination immediately, hence messages are temporarily stored in the Switch. Also destination terminals have some control over the way that these stored messages are read out of the Switch's buffer.

The storage capability of the Switch is not unlimited; therefore, the flow of information packets into the Switch must be controlled. The Switch does this by informing terminal T1U of the amount of storage currently available in the Switch. The terminal does not transmit information packets when there is no room in the Switch, but holds them until storage is available.

As mentioned earlier models of the system were studied in order to gain insight into performance and thereby guide design decisions. The models studied are approximations to actual system operation. We felt that the study of more exact, hence more complicated, models would have involved far more time and effort, without a corresponding increase in insight.

II. Loop Study

A basic consideration in the design of the system is the response time to interactive users. An important component in this response time is message delay, which is defined to be the time elapsing between the arrival of a message at a User Terminal and the departure of the last packet of the message from the terminal. Message delay is composed of multiplexing delay and queuing delay. Message delay is the criteria that we use to evaluate the multiplexing techniques under study. Recall that techniques are Synchronous Line Division Multiplexing, Demand Multiplexing with adjacent slot seizure and Demand Multiplexing with alternate slot seizure.

The message delay for STDM can be found through analytic techniques.* The delay in DM Systems with adjacent slot seizure has received considerable attention lately. Analytical solutions can be found in references 3, 4 and 5. For the case of DM with alternate slot seizure, no analytic solution is available and a simulation was necessary.† In the study of message delay incurred in loop multiplexing, we assume that messages arrive at user terminals at a Poisson rate of λ messages per second (see Figure 3). We also assume that each user terminal receives as much traffic as it transmits. In calculating delay the assumption is made that there are no restrictions on the amount of storage that is available at User Terminals. Thus it never occurs that arriving messages are turned away.

Although the analytical and simulation techniques used in our study are not restricted to a particular message distribution, we concentrated on the case where 30 percent of the messages are 32 packets long (8192 bits) and the remaining messages are one packet in duration (256 bits). This message distribution was our best guess at the actual distribution of messages in the system and reflects the fact that most terminals will, in fact, be computers. In the sequel we use the term variable message length to designate this distribution. The case where all messages are one packet in duration was also studied to some extent. In referring to this latter distribution we use the term constant message length.

Typical results of our study of multiplexing techniques are shown on Figures 4-7. Figures 4 and 5 examine the relative performance of alternate and adjacent slot seizure in the DM implementation. In Figures 6 and 7, the performances of STDM and DM with adjacent slot seizure are compared. In these figures the load is defined to be the portion of the time that the line between terminals on the loop is occupied. These curves apply to the variable message length distribution defined above. The resulting average message length is 10.3 packets per message. The message arrival rate at each station in the loop in terms of loading, ρ, is $\rho/(10.3N)$ messages per slot time. (Each slot time is 1/4000 sec.) Thus for .103 loading on a 20 terminal loop, messages arrive at a rate of .0005 messages per slot time or 2 messages per second.

The results shown on Figures 4 and 5 were obtained from simulation. The curves indicate delay as a function of line loading. For comparison the results of a theoretical calculation of the average and the standard deviation of delay is also shown on Figures 4 and 5 respectively.

The simulation results show that adjacent slot seizure yields somewhat better delay performance than alternate slot seizure. For almost all values of line loading adjacent slot seizure gives lower values of mean delay and standard deviation of delay. Measurements made on loops with 2, 10 and 64 terminals, not shown here, yield much the same result.

The reader will notice, however, that for most values of line loading, the difference between alternate and adjacent slot seizure is not large. The difference is small enough so that ease of implementation should probably determine the choice between the two.

---

* A general analysis for which STDM is a special case is given in reference 2.

† In this presentation details of analysis and simulation will be omitted. These details will be published elsewhere.

For each of the simulations care was taken to insure that statistical equilibrium had been reached. The duration of runs and the random sequences used in the simulation were varied. The standard deviation of the estimates of the mean values of delay shown on Figure 4 can be estimated. We assume that the meassured standard deviations are the true standard deviations. The standard deviation of the mean is then the measured standard deviation divided by the square root of the number of samples. The results indicate that the standard deviation of the mean is small compared to the mean value. The standard deviation is largest relative to the mean at light loadings on the 20 terminal loop where it is approximately 5 percent of the mean.

There is a basic difficulty in making measurements at light loadings on a 20 terminal loop. Due to the relatively low departure rate, fewer independent samples can be gathered. Except for the lighter values of line loading on the 20 terminal loop there is good correspondence between the results of simulation and theory. Even at these lighter loadings the results of simulation are not so far from theory so as to cast doubt on the simulation.

The comparison of STDM and DM with adjacent slot seizure was carried out using only analytical techniques. In order to simplify analysis we have ignored the fact that for DM information packets must contain the address of the transmitting terminal. No such addressing is required for STDM. However such addressing information is a negligible part of an information packet. For example, for a 64 terminal loop only 6 bits are necessary to specify the address of a terminal. We feel that the small improvement in accuracy that could be attained by considering addressing did not justify the complications introduced into the analysis.

Typical results are shown on Figure 6 where delay in both packet times and milliseconds is shown as a function of line loading for the variable length message case. As the curves show DM is clearly superior to STDM. This superiority is more pronounced at the lighter loading, where the multiplexing time is the strongest component of delay. In the absence of interferring traffic the time required to multiplex a message in DM is an average of 10.3 slot times. In contrast for an STDM system with N terminals the average time required to multiplex a message is 10.3×N slot times. As the loading increases the difference between the two systems decreases. Line traffic in the DM System interferes with message multiplexing and as the load increases so does the interference.

Similar results have been obtained for the constant length message distribution. Computations of the standard deviation of delay for both constant and variable message length distributions also show the same basic pattern.

Another view of the performance is indicated on Figure 7 where average delay is shown as a function of the number of terminals in the loop for fixed values of loading. In Figure 7 the dependence of delay in the STDM system on the number of terminals in the loop is marked. There is little of this dependence in the case of DM. However, DM is more sensitive to changes in load than STDM. Notice the large jump in delay from .5 loading to .9 loading in the case of DM. Although we have not shown them, similar results obtain for the constant message length distribution.

III. Switch Throughput

The second phase of our work involved a study of buffering in the Switch. Streams of data enter the Switch from the loops connected to it. System operation is such that, at any given time, each terminal in the system transmits to and receives from only one other terminal in the system. Information on which pairs of terminals are linked together is stored in the Switch. Therefore given the origin of an information packet, the Switch determines its destination by looking in a table.

A terminal can rapidly change the destination of the packets that it transmits. Stored in the Central Switch is a list of up to 64 possible correspondent terminals for each terminal. A terminal that is transmitting to terminal A for example, may select a new destination, say terminal B. By means of signal packets (see above) the Central Switch is notified of this change in destination. After the change all information packets transmitted from the originating terminal are routed to terminal B. A terminal can select only from the list of its 64 correspondent terminals stored in the Switch. However this list can be altered by the originating terminal when it wishes to make connection with a new terminal or drop connection with an old. Again signal packets are used to communicate between the terminal and the Switch. The process of altering the list requires much more time then switching between terminals already on the list.

At a given instant of time, a terminal transmits to and receives from the same terminal. Further each terminal in the system acts independently in selecting the correspondent terminal that is the destination of its packets. Thus a terminal may select a destination terminal that is, at that point in time, corresponding with a third terminal. In this event the packets that are transmitted are stored temporarily in the Central Switch. The Central Switch, again using signal packets, notifies the destination terminal that packets from a particular originating terminal are waiting to be delivered. It may happen that, for a particularly busy terminal, there may be messages from several different originating terminals stored in the Switch waiting to be delivered. The receiving terminal is free to choose the order in which these messages are read out of the Switch buffer.

As we have indicated, it may be necessary to store information packets in the Switch before they can be delivered. As a practical necessity, the amount of storage in the Switch is finite and under heavy loading conditions storage may be used up. In this situation the Switch sends signal packets to User Terminals which inhibit transmission until storage is available in the Switch.

Our study of packet storage capacity in the Switch focused on two aspects of the problem, throughput and user strategy. Given the random nature of the message flow in the system, there will be occasions when all of the storage assigned to a channel is used up and the transmitting terminal is inhibited. If this condition occurs often enough, there will be a significant effect on the total throughput of data. Secondly, the user through his strategy in reading out packets from the Switch can affect the amount of storage that is required. A certain amount of time is required to switch to a new correspondent terminal. If a terminal switches very often more storage is needed in the switch.

In order to study throughput and the effect of user strategy on buffer requirements, a simplified model was constructed. The model is shown in Figure 8. N independent data streams carrying λ messages per second flow into N buffers. These data streams represent traffic from correspondent terminals flowing to the same destination terminal. The destination terminal's changing correspondents is represented by the switch in Figure 8 moving from buffer to buffer. In the model the time required to switch buffers is taken to be either zero or eight packet times (1/4000 second).

In our study of Switch throughput two kinds of buffering were considered, dedicated and common. For dedicated buffering, each of the N buffers is a fixed size. When a buffer is filled, the transmitting terminal is informed and information packets are held at the User Terminal until there is room. In the case of common buffering, a fixed amount of storage is allocated for all N buffers. Each input line uses as much input capacity as it needs. Thus one input line can use up all of the common storage. Again when there is no more room in the buffer data flow is inhibited. In our study of throughput, we assume that the entire contents of a buffer are removed before moving on to a new buffer. The order in which buffers are examined is fixed and empty buffers skipped. We shall also compare this to a strategy where switching takes place after a single message has been read out of a buffer.

A good deal of previous work on buffer occupancy has been based on a Poisson model in which messages arrive instantaneously with an exponentially distributed interval of time between messages. A more realistic model, for our study of throughput, it one in which messages arrive over a time interval proportional to the message length with the time between the beginning of one message and the end of the previous message being exponentially distributed. This latter model is more appropriate to buffering in the Switch where the arrival and departure of messages is over T1 lines and the read in and write out rate of messages is the same.

The model represented by Figure 8 was studied by means of a simulation. The basic tasks of the simulation program is to measure the throughput as a function of storage capacity and to measure the average occupancy of the buffers. Input variables to the program determine the amount of storage available, the number of input lines and buffers, the time required to switch between buffers and the message arrival rate.

In a run of the simulation program the total number of packets that were fed into buffers in a fixed amount of time were measured. By varying the total amount of storage available, with all other parameters fixed, one obtains the relationship between throughput and storage. Simulation runs were made for the constant and variable length message distributions. Measurements were also made of the total number of messages in the buffers. The results of these latter measurements will be considered presently when we deal with user strategy.

Typical results of simulation are shown on Figures 9 and 10 for 5 and 20 input lines respectively. In obtaining the results shown on both figures the variable message length distribution was used. The switching time is 8 packet slots. If the line rate is 4000 packets per second, the time required to switch is 2 msec. The curves show normalized throughput as a function of the total packet storage with message arrival probability as a parameter. For each loading the throughput is normalized to the throughput measured at very large storage capacity. The message arrival probability is the probability of a message arriving on at least one input line in a packet slot time (1/4000 sec).

The basic configuration of the curves is as one might expect. As the storage capacity decreases, the throughput decreases. Further the normalized throughput decreases faster for the large values of loading.

The results show that, even for a limited amount of storage, the throughput is high. For example, if there are two packet slots for each buffer, the throughput is over 70 percent even for high loading. Results (not shown) for the case of zero switching time show that for this same amount of storage the throughput is over 90 percent.

A surprising result shown on Figures 9 and 10 is that dedicated storage shows better performance than common storage when there is a limited amount of buffering available. A combination of factors produces this result. First of all, even through storage may be held in common it is committed to input lines in a specific way that may be far from optimum. The preponderance of traffic is contained in messages that are 32 packets long. If the amount of storage held in common is limited, one channel may absorb all of the storage that is available in the Switch. We have also simulated models where all messages are one packet long. In this case common storage is superior to dedicated. However, even in this case the difference between common and dedicated storage is not great.

Storage and User Strategy

At any given point in time a User Terminal knows which of its 64 correspondent terminals has messages in the Switch waiting to be delivered. A terminal is free to read out these messages in any order it wishes. Presumably terminal will not interrupt the reading out of a message in order to switch to a new correspondent. Since the selection procedure is entirely in the hands of the User Terminal, we studied the effect of different strategies on system storage requirements. Accordingly a calculation of buffer occupancy statistics for different user strategies was performed.

As in the study of throughput we use the model shown on Figure 8. In order to make the analysis tractible, we assume that messages arrive at a Poisson rate of λ messages per second over each input line. Further we assume that eight packet slot times are required to switch buffers. In order to calculate bounds we also consider the case where no time is required to switch.

Now if it is known which of the buffers are not empty, the worst strategy, in terms of buffer occupancy, is to always switch after reading a message out of a buffer. Thus, even if messages remain in a buffer, time is wasted in switching to a new buffer. In the sequel we shall refer to this strategy as "1-by-1". In contrast the most efficient strategy is to cycle through the N buffers skipping empties and reading out the entire contents of non-empty buffers. This latter strategy is the one considered in the previous section on throughput. We shall refer to it as "empty before switch". An intermediate strategy involves switching at random. In this case, after a message has been read out of a buffer, one selects the next buffer at random from those having messages. It can be shown that, if there are N buffers, the probability of switching to a new buffer is 1-1/N. Thus with probability 1/N, two messages are read from the same buffer in succession.

The 1-by-1 and the random strategies have been studied by means of the theory of the M/G/1 queue. The messages arrive over the input lines at a Poisson rate. The service time is taken to be the time required to multiplex the message plus the time required to switch. In the case of random switching, the switching time is zero with probability 1/N.

The analysis of buffer occupancy in the "empty before switch" strategy is mathematically difficult. However an upper bound on storage requirements can be found. We consider a strategy in which each of the N buffers is examined in turn (even empty buffers). This is similar to a polling system which has been analyzed in reference 7.

The results of computations of buffer occupancy are shown in Figures 11 and 12 for 5 and 20 buffers respectively. In these figures the average occupancy of each buffer is shown as a function of load, which is the product of the message arrival rate and the average time required to read out a message, $N\lambda\bar{m}$. As expected the lowest buffer occupancy occurs in the case where no time is required to select a new correspondent. When an 8 slot switching is required, the technique with the lower occupancy depends upon the loading. At light loading the 'empty before switch' strategy shows poorer performance because time is wasted stopping at empty buffers. It must of course be remembered that this is only an upper bound for the 'empty before switch' that selects only non-empty buffers. As the loading increases there are fewer empty buffers and the performance of the 'empty before switch' strategy improves relative to the 1-by-1 strategy.

In the previous section we considered an 'empty before switch' strategy that skipped empty buffers. As we have mentioned above, the problem of calculating occupancy statistics for this technique is intractable. However the results shown on Figures 11 and 12 form bounds on the skipping empty technique. The shaded areas in the figures indicate the areas in which the statistics for this method lie.

If the system is operated below .5 loading the difference between the different channel switching strategies is not very large. For example, for 20 channels and .4 loading (see Figure 12) the average occupancy for 1-by-1 rotating strategy is 1.1 packets. For the 'empty before switch' strategy skipping empties, the average occupancy is between .4 and 1.0 packet. As the load increases beyond .5 loading, the 1-by-1 strategy leads to saturation and the cyclic system is clearly superior.

Results for the standard deviations of buffer occupancy have been obtained. These results support the foregoing conclusions.

The simulation program discussed in the previous section computed means and standard deviations of buffer occupancy for an empty before switch strategy with skipping of empty buffers. The results are shown on Figures 11 and 12. A comparison of analysis and simulation indicates that for the most part the analysis gives upper bounds to the simulation. This is not unexpected since in the simulation program messages arrive over an interval of time, whereas for the Poisson arrival model used in the analysis messages arrive instantaneously.

Acknowledgement

The system described in this paper was conceived and built by A. G. Fraser. The author would like to express his appreciation to Dr. Fraser for many enlightening discussions on the system while our work was being carried out. We would also like to thank Brian W. Kernighan who, with patience and unfailing good humor, answered many questions on the computer programming involved in the work.

References

1. Fraser, A. G., "Interconnecting Computers and Digital Equipment," Internal report available upon request.

2. Kohneim, A. G., "Service Epochs in a Loop System" Presented at the 22nd Int. Symp. on Computer-Communications Networks and Teletraffic Polytech. Inst. Brooklyn, Brooklyn, N. Y., April 1972.

3. Hayes, J. F. and Sherman, D. N., "Traffic Analysis of a Ring Switched Data Transmission System" Bell Syst. Tech. J., Vol. 50, No. 9, pp. 2947-2978, November 1971.

4. Anderson, R. R., Hayes, J. F. and Sherman, D. N., "Simulated Performance of a Ring Switched Data Network" IEEE Transactions on Communications, Vol. COM-20, No. 3, pp. 576-591, June 1972.

5. Avi-Itzhak, B., "Heavy Traffic Characteristics of a Circular Data Network," Bell System. Tech. J., Vol. 50, No. 8, pp. 2521-2549, October 1971.

6. Cox, D. R. and Smith, W. L., Queues, Methuen and Co., London, 1961.

7. Hayes, J. F. and Sherman, D. N., "A Study of Data Multiplexing Techniques and Delay Performance," Bell Syst. Tech. J., Vol. 51, No. 9, pp. 1983-2011, November 1971.

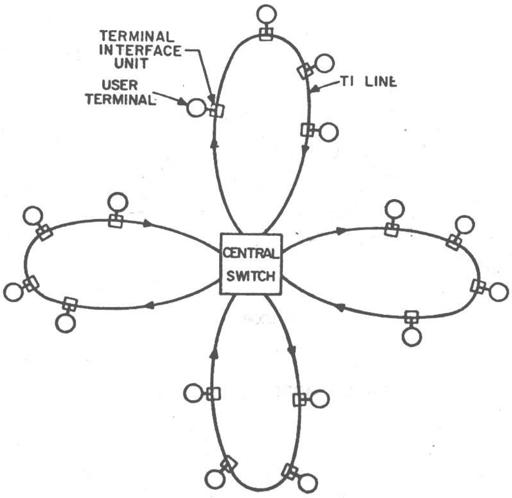

FIG.1 COMPUTER COMMUNICATIONS NETWORK

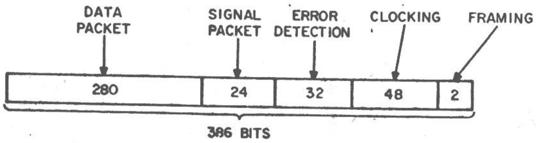

FIG.2 BIT ALLOCATION OF TI FRAME PAIR

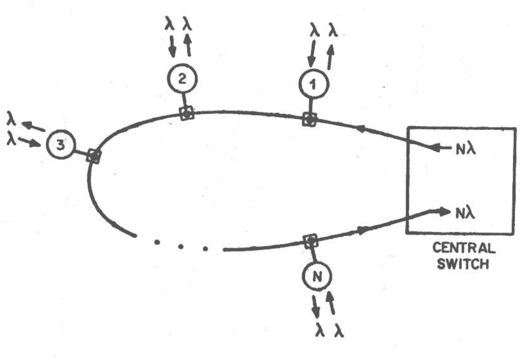

FIG.3 LOOP CONFIGURATION

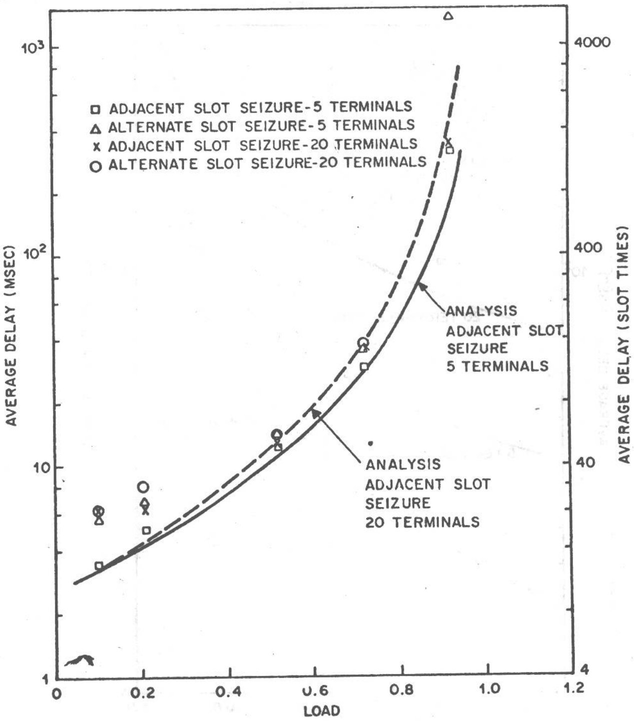

FIG.4 SIMULATION RESULTS AVERAGE DELAY

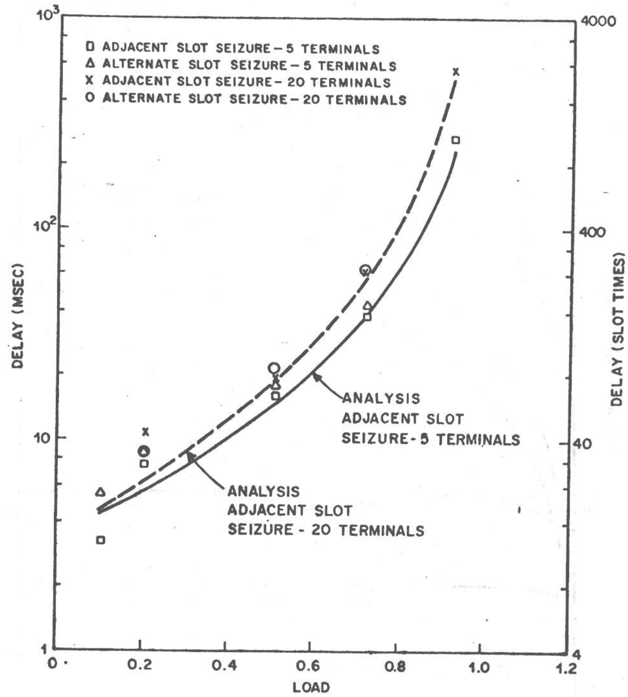

FIG.5 SIMULATION RESULTS STANDARD DEVIATION OF DELAY

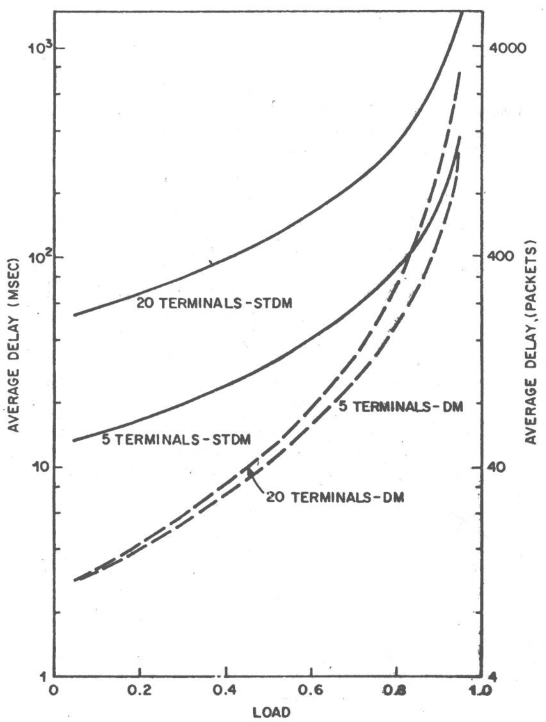

FIG. 6 AVERAGE DELAY VS LOADING
DM AND STDM SYSTEMS
30 % OF MESSAGES 32 PACKETS LONG

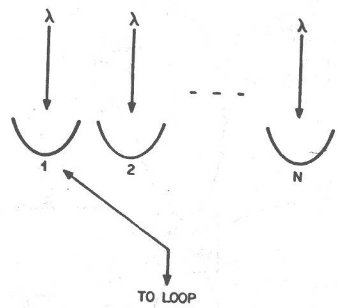

FIG. 8 MODEL OF SWITCH

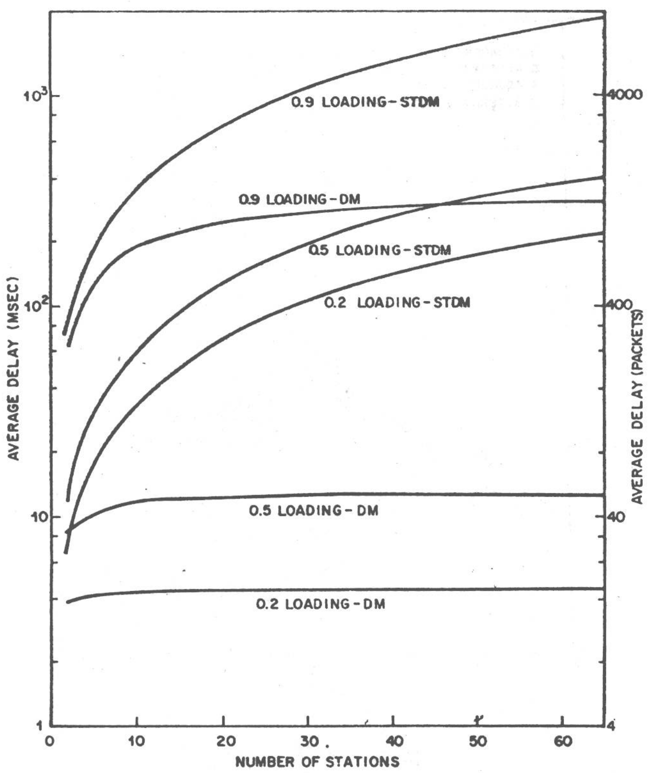

FIG. 7 MESSAGE DELAY VS NUMBER OF STATIONS
30 % OF MESSAGES 32 PACKETS LONG

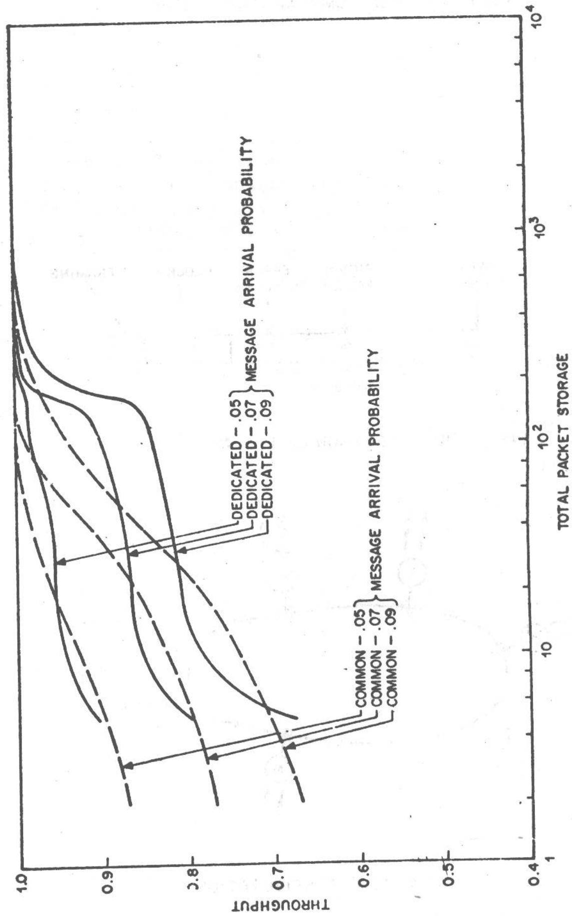

FIG. 9 THROUGHPUT VS PACKET STORAGE - 5 INPUT LINES
SWITCHING TIME = 8 SLOT TIMES

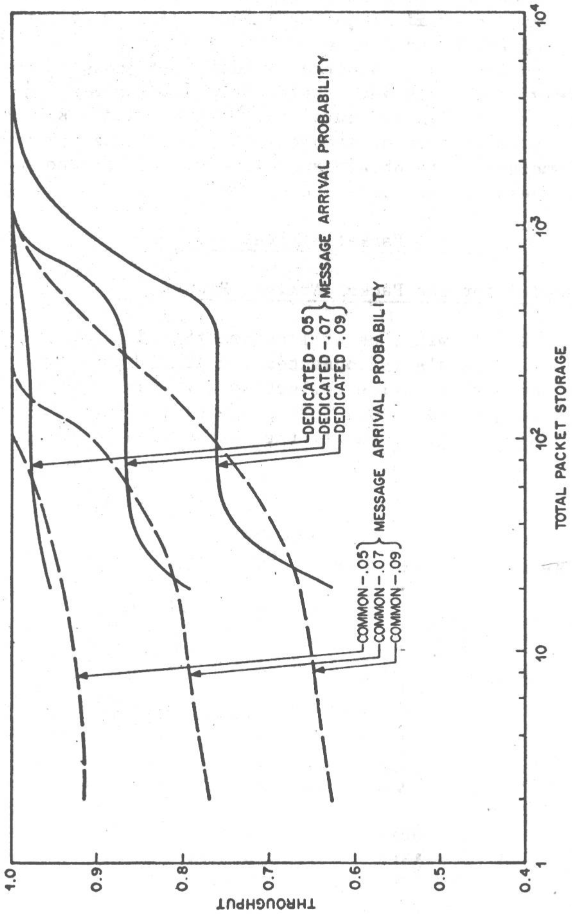

FIG. 10 THROUGHPUT VS PACKET STORAGE-20 INPUT LINES
SWITCHING TIME = 8 SLOT TIMES

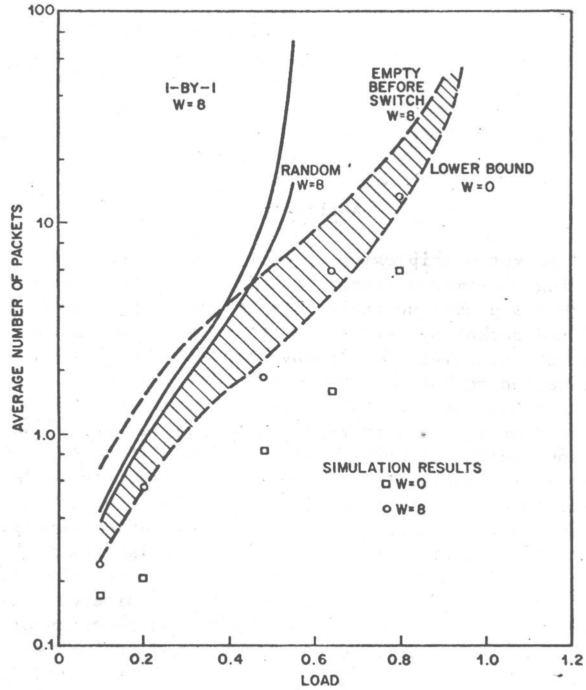

FIG. 11 5 BUFFERS
AVERAGE NUMBER OF PACKETS
IN EACH BUFFER
30% OF MESSAGES 32 PACKETS
LONG

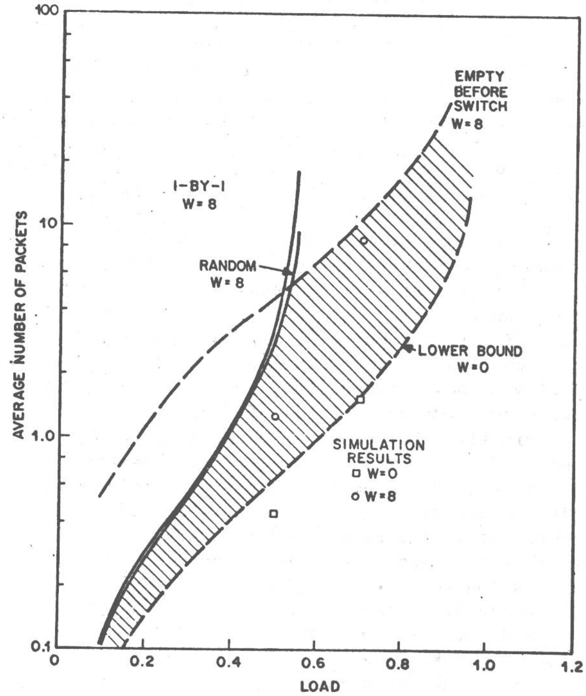

FIG. 12 20 BUFFERS
AVERAGE NUMBER OF PACKETS
IN EACH BUFFER
30% OF MESSAGES 32 PACKETS
LONG

PACKET ARRIVAL AND BUFFER STATISTICS IN A PACKET SWITCHING NODE

Felix Closs

Data Communications Center
IBM Research Laboratory
Zurich, Switzerland

Abstract

Packet multiplexing or switching systems split incoming customer messages into fixed length packets. Within a packet switching system information is transferred packet by packet. In this paper a model for a packet switching node is investigated. Terminals are connected to the node by local loops. In the model incoming messages are quantized into packets and completed packets are transferred to the desired outgoing trunk. Buffering techniques are applied to obtain high trunk utilization. In the model each trunk has a dedicated buffer which stores packets queueing up for transmission. First the statistics of packet arrival at a trunk buffer is analyzed by investigating the process of quantizing messages which arrive dispersed and overlapping in time. Then buffer requirements and overflow probabilities are derived as a function of local loop and trunk speeds and for various packet sizes.

1. Introduction

Packet switching is one of several possible techniques to implement future switched data networks. The basic idea behind a packet switching network is to quantize incoming customer messages into packets of a certain length and to transfer information between nodes packet by packet.[1] As in a message switching system, buffering techniques are employed to achieve high utilization of internodal communication channels. Packet switching promises some operational advantages over message switching, e.g., a saving in transmission delay.

The buffer space provided in the switching nodes allows queues of packets to build-up in front of channels just engaged in packet transmission. Naturally, buffer space will always be limited and so will be the number of packets which can be queued up. Even though buffer cost will decrease significantly in the future, buffer size, nevertheless, is an important parameter from the point of view of proper network operation. Insufficient capacity may lead to frequent overflow or nodal congestion, both seriously affecting network performance or grade of service.

Obviously, the number of packets competing or queueing for transmission to a certain destination is closely related to the time pattern in which packets arrive at the waiting line in front of the desired channel. It is the purpose of this study to investigate this time sequence in which packets arrive for transmission to a certain destination given the characteristics of customer messages, their length distribution and interarrival periods. It will be shown that the speed of the local loops connecting terminals to the node has an important influence on this (non-poisson) packet arrival process. The results of this analysis will then be used to analyze the number of packets simultaneously competing or waiting for transmission, or, in other words, to compute the required buffer space for given traffic parameters.

This particular packet arrival process is of more general significance. It can be observed in systems splitting-up incoming messages into packets or blocks and where these packets or blocks then compete for access to a single facility serving one packet after another. This facility may be, e.g., a communication channel connected to a buffered (or statistical) multiplexor, or a central processor equipped with a packet assembling front-end device.

The analysis was inspired by Rudin and Chang[2,3] and will rely on methods and results of renewal theory[4,5] particularly on P.A.W. Lewis' work.[6] While Rudin's work is exclusively based on simulation, we will use the diffusion method[7-9] to obtain approximate analytic and more general results for buffer requirements.

2. The Packet Arrival Process

2.1 A Model for the Packet Arrival Process

In the following, we will assume that a (large) number of terminals is connected via local loops of finite speed to a device collecting a certain number of bits of incoming messages into packets or blocks of fixed length, Fig. 1. Due to the finite line speed, messages arrive at the packet assembly circuit not instantaneously but more or less dispersed in time. Message arrivals from different terminals may also overlap in time.

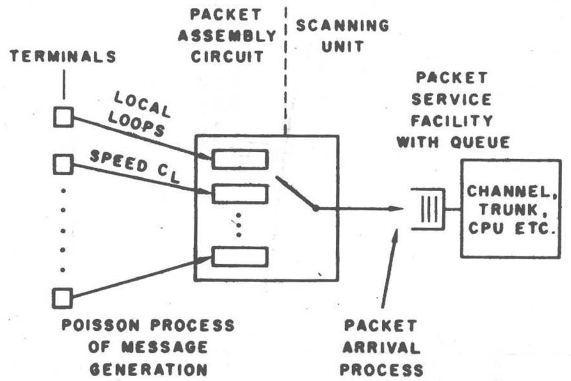

Fig. 1. Transformation of the message arrival process into a packet arrival process.

A scanning unit detects completely assembled packets and immediately notifies the packet handling facility. Then the packet is transferred to that facility and it will have to queue for service in case the facility is engaged (it is assumed that a packet can be transferred from the scanning unit to the facility or queue, in negligible time). It should be emphasized that packet transfers (or arrivals at the facility or queue) appear in random time intervals and mixed from various messages (technically packets can be identified by address information).

2.2 Assumptions Concerning Terminal Traffic

We will assume that all terminals generate messages at the same average rate and that the time intervals between two consecutive messages are exponentially distributed. The message lenths are assumed to be independent and discrete so that all messages can be broken down into an integer number of packets at the packet assembly device. The problem area of incompletely filled packets will not be discussed here. We take sufficiently many terminals to justify the assumption that the overall message generation is a poisson process.

Figure 2 shows the message flow from three terminals and illustrates the resulting sequence of packet transfers or arrivals. Figure 2(a) shows that packet arrivals have a tendency to cluster. In Fig. 2(b), line speed is 1/4 of that in Fig. 2(a) and the arrival of packets is more uniformly distributed over the time axis. For very low speed lines the packet arrival looks like a poisson process. This behavior was outlined qualitatively by Rudin.[2]

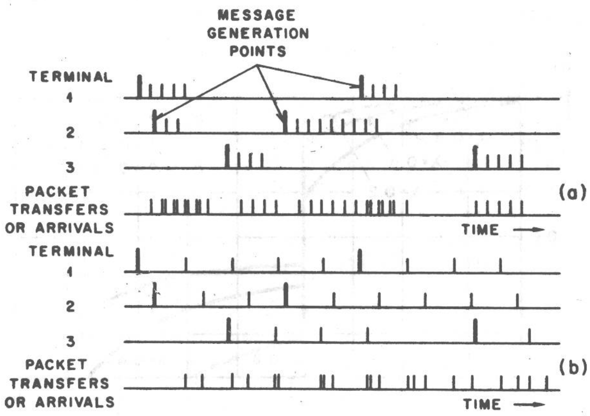

Fig. 2. The packet arrival process.

2.3 The Analysis

To analyze this packet arrival process we use the concept of a branching poisson process.[5,6] In such a process there is a series of primary events - the message generation points - which follow each other in random intervals $X_1, X_2, \ldots$. According to the assumption in Section 2.2, the X_i are independent and taken from a negative exponential distribution. Each primary event generates a sequence of secondary events - packet generation points. For the analysis we assume first that secondary events follow each other in random intervals $Y_1, Y_2, \ldots$, the Y_i being independent and identically distributed. The general results will then be applied to the special case of constant packet length, i.e., Y_i = constant. One generally assumes independent message lengths which means here that secondary processes are independent of each other. The packet arrival process in Fig. 1 is a superposition of all primary and secondary processes where primary and secondary events are indistinguishable.

We denote the average generation rate of messages from all terminals together by λ and the average number of packets per message by N.

2.4 Interarrival Statistics

For conciseness, we will derive the distribution of interarrival times between packets directly in a heuristic way. A rigorous derivation is given by Lewis.[6]

It is assumed that an event - a packet arrival - occurs at time t_o, long after the process has started. This can be a primary event - a message generation point - the probability for that being 1/N, or a secondary event, the probability for that being (N-1)/N. The probability that it takes more time than, say t, until the next event occurs is equal to the probability that no event can be observed in $(t_o, t_o+t]$. This implies that there is neither a primary nor a secondary event in $(t_o, t_o+t]$. There could be a secondary event belonging to a message generated during $(0, t_o]$. Denote by $P(t_o, t)$ the probability that there is no such secondary event in $(t_o, t_o+t]$. Recall from Section 2.2 that message interarrival times are assumed to be negative exponentially distributed. Then, if there is a primary event at t_o

$$\text{Prob}\,\{\text{no event in } (t_o, t_o+t] \,|\, \text{prim. event at } t_o\} = e^{-\lambda t}\, P(t_o,t)$$

If there is a secondary event at t_o

$$\text{Prob}\,\{\text{no event in } (t_o, t_o+t] \,|\, \text{secondary event at } t_o\} = e^{-\lambda t}\, P(t_o,t)\ \text{Prob}\,\{\text{next sec. event, belonging to the one at } t_o, \text{ more than } t \text{ apart}\}\ .$$

Denote the right most term in this expression by $R_y(t)$. It is the distribution of time intervals Y_i between events in a secondary process. Combining both probabilities gives:

$$R(t) = \text{Prob}\,\{\text{no event in } (t_o, t_o+t]\} = \frac{1 + (N-1)\, R_y(t)}{N}\, P(t_o,t)\, e^{-\lambda t}\ , \qquad (1)$$

$R(t)$ is the desired interarrival distribution.

$P(t_o, t)$ can be derived using properties of the poisson process.[5] For $t_o \to \infty$:

$$P(t_o, t) = \text{Prob.}\,\{\text{no sec. event in } (t_o, t_o+t]\} = \exp\,\{-\lambda(N-1) \int_0^t R_y(u)\, du\}\ . \qquad (2)$$

Therefore,

$$R(t) = \frac{1+(N-1)R_y(t)}{N} \exp\,\{-\lambda t - \lambda(N-1) \int_0^t R_y(u)\, du\}\ . \qquad (3)$$

This expression for the interarrival distribution can be evaluated when the distribution $R_y(t)$ of time intervals between secondary events is known.

As stated in Section 2.1 messages arrive at the packet assembly device dispersed in time because of finite local loop speed C_L. The time to assemble a packet from incoming bits is therefore,

$$Y_o = \frac{L_P}{C_L}\ ,$$

L_P = packet length in bits.

This means that the events in a secondary process - the points in time where packets of a certain message are completed - are separated by a constant time interval Y_o.

From Eq. (3) we get for the probability that two consecutive packet arrivals are separated by more than t :

$$R(t) = \begin{cases} e^{-\lambda N t} & 0 \leq t < Y_o \\ \frac{1}{N} e^{-\lambda Y_o (N-1)} e^{-\lambda t} & t \geq Y_o \end{cases} \quad (4)$$

The average time between packet arrivals is, of course,

$$E[t] = \frac{1}{\lambda N} \quad (5)$$

and the variance becomes

$$Var[t] = \frac{1}{\lambda^2 N^2} \left[1+2(N-1)\, e^{-\lambda N L_P / C_L}\right]. \quad (6)$$

The coefficient of variation is

$$k = Var[t]/\{E[t]\}^2 = 1+2(N-1) \exp\left[-\lambda N L_P/C_L\right]. \quad (7)$$

2.5 Discussion and Results

Equations (4) to (7) describe the statistical behavior of a stream of fixed length packets or blocks into which customer messages, arriving randomly and overlapping in time, are split-up. It is this arrival process a packetwise-operating facility has to deal with. The facility may be a trunk to another node, a central processing unit, etc. Of particular interest is the influence of three parameters: line speed C_L, average number N of packets in a message, and total call rate λ.

For given N and Lp it follows from Eq. (7) that for very fast local loops ($C_L \to \infty$) $k \to 2N-1$. This large coefficient of variation indicates that packets arrive in clusters. Figure 2 illustrates this behavior. For slow local loops ($C_L \to 0$) the coefficient of variation approaches unity and packets arrive in a poisson fashion. This agrees with intuitive reasoning and was qualitatively stated by Rudin.[2]

Figure 3 shows the interarrival time distribution for various local loop speeds. The distribution has a jump at L_P/C_L - the packet duration.

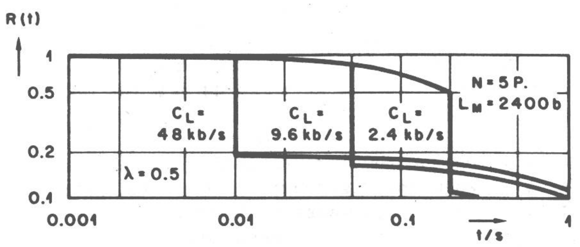

Fig. 3. Distribution of packet interarrival times for different local loop speeds C_L. Average message length L_M = 2400 bits, N = 5 packets.

For Fig. 4 it is assumed that incoming messages have an average length L_M. The curves show R(t) when L_M is split up into 3, 10, and 30 packets, or, in other words, for different packet sizes.

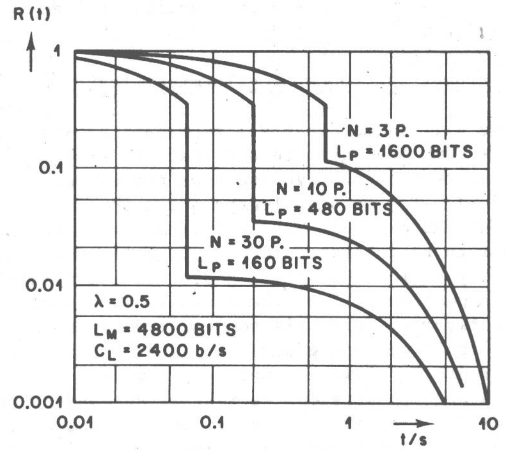

Fig. 4. Distribution of packet interarrival times for different packet sizes. Average message length L_M = 4800 bits. Local loop speed C_L = 2400 b/s.

Figure 5 finally illustrates the effect of call rate λ on R(t).

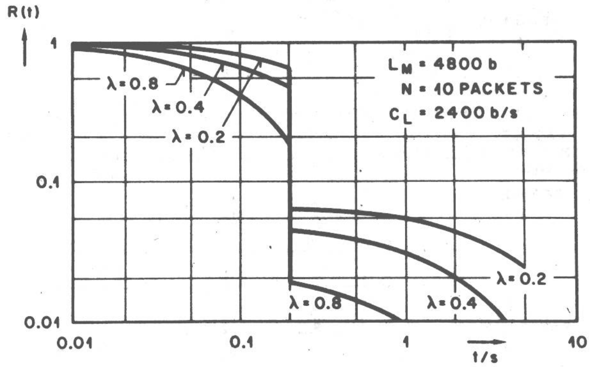

Fig. 5. Distribution of packet interarrival times for different call rates. Average message length L_M = 4800 bits. N = 10 packets. Local loop speed C_L = 2400 b/s.

3. Buffer Analysis

3.1 A Model for a Packet-Switching Node

In this section the model given in Section 2.1 will be expanded to build a structure for a packet switching node. Our interest will concentrate on the length of lines of waiting packets as indicated in Fig. 1, and on the required buffer space to store these packets.

Figure 6 shows a packet assembly device (there could be more than a single one) on the left and outgoing trunks on the right side. Each packet is directed to a certain outgoing trunk, depending on the desired destination of the packet. The trunk selector circuit in Fig. 6 performs this task using destination information stored in the node. Each outgoing trunk has a memory module and control logic to store packets queueing-up for transmission. Again it is assumed that packet transfer through the trunk selector in Fig. 6 is fast, compared to trunk and local loop speeds.

At this point some explanation should be made. The above model is not intended to describe the operation of a presently existing store and forward node. In our model, buffering techniques are applied to improve trunk utilization when customer messages are separated by relatively long idle periods. This type of traffic can be observed, e.g., in interactive systems.

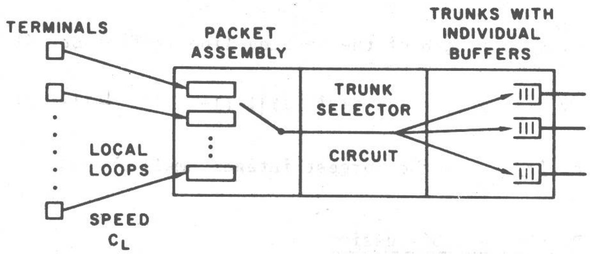

Fig. 6. A packet switching node with individual trunk buffers.

A separate buffer for each trunk is clearly a waste of memory space compared to existing systems with pooled or shared buffers.[3] This argument may lose some of its importance in view of rapidly decreasing memory cost.

3.2 Queue Analysis by Diffusion Approximation

In the following, it is assumed that incoming messages (and their packets) are evenly and randomly distributed to all outgoing trunks (Fig. 6). When the number of terminals is large and packet transfer from the assembly circuit to a trunk is fast compared to local loop and trunk speeds, then packet arrival at a trunk is given by Eqs. (4) to (7). λ is then the average call rate (message generation rate) to a certain destination.

The primary objective of buffering is high trunk utilization. We will, therefore, concentrate the analysis on waiting lines for heavily loaded trunks. Under heavy load conditions the diffusion approximation used by Gaver, Shedler and Kobayashi[7-9] is an elegant method to obtain approximate results for the queue length distribution.

We will briefly outline the basic idea of the diffusion approximation method. Let

$Q(t)$ = number of packets at a certain trunk at time t. This includes those waiting and the one being transmitted,

$\Delta Q(\tau)$ = change in the number of packets over $(t, t+\tau)$
$= Q(t+\tau) - Q(t)$,

denoting by $A(\tau)$, $D(\tau)$ the number of arrivals and departures during τ

$$\Delta Q(\tau) = A(\tau) - D(\tau) .$$

When the interarrival times a_i and service time intervals d_i are both independent and identically distributed, then, for sufficiently large τ to allow many arrivals and departures in τ, ΔQ should be approximately normally distributed. The normal distribution satisfies the diffusion equation. Solving the steady state form of this equation with the boundary condition $Q(t) \geq 0$ gives the probability that the waiting line is longer than n packets.[8]

$$P\{W > n\} \approx \rho \exp[-2\mu(n+1)/\sigma^2] \tag{8}$$

where

$$\mu = \lim_{t\to\infty} \tau^{-1} (E[D(\tau)] - E[A(\tau)]) \tag{9}$$

and

$$\sigma^2 = \lim_{t\to\infty} \tau^{-1} (Var[D(\tau)] + Var[A(\tau)]) . \tag{10}$$

ρ is the utilization of the service facility and will be given in Section 3.5. A detailed derivation is given in Refs. 7,8 and 9.

3.3 The Applicability of the Approximation

The diffusion approximation is based on the fact that the asymptotic distribution of the number of arrivals and departures is a normal distribution, provided interarrival times are independent and identically distributed. Furthermore, it requires sufficiently many events in a given observation time interval to justify the assumption of a normal distribution.

It is recognized that the use of the diffusion approximation can only partly be justified in view of the peculiarities of the packet arrival process with its interarrival dependencies. Nevertheless, it was felt that it would be useful to make an attempt with this method to obtain some analytic results to broaden the scope of pure simulation.

3.4 The Number of Packet Arrivals in a Time Interval

Equations (9) and (10) require average and variance of the number of packets arriving and departing during a certain time interval. Here an assumption for the message length distribution must be made. We will assume that each message contains at least one packet and that the number of additional packets is geometrically distributed. The average number of packets in a message is N.

Lewis[6] has computed the distribution of the number of arrivals in an interval taken long after the process has started. From his results we can derive $E[A(\tau)]$ and $Var[A(\tau)]$. The distribution is given as Laplace - Stieltjes transform of the probability generating function. From this we find by differentiating and inversion of the Laplace-Stieltjes transform:

$$E[A(\tau)] = \lambda N\tau \tag{11}$$

which is intuitively clear.

To obtain $Var[A(\tau)]$ we differentiate twice and invert again the Laplace-Stieltjes transform and obtain:

$$Var[A(\tau)] = \lambda N\tau + 2\lambda\tau N^2(1-\frac{1}{N})[1-(1-\frac{1}{N})^{\lfloor\tau/Y_o\rfloor}] \tag{12}$$

$\lfloor\tau/Y_o$ denotes the largest integer smaller than or equal to τ/Y_o.

We recall that Y_o is the "duration" of a packet (Section 2.4). The second term in (12) is due to the discrete nature of the secondary processes or, in other words, to the fact that all packets have a constant length.

We observe the following asymptotic behavior for $Var[A(\tau)]$. For fixed τ and $Y_o \to 0$, i.e., for $C_L \to \infty$

$$Var[A(\tau)] = \lambda N(2N-1)\tau \tag{13a}$$

which means that packets arrive in clusters. For $Y_o \to \infty$, i.e., for $C_L \to 0$

$$\mathrm{Var}[A(\tau)] = \lambda N\tau = E[A(\tau)] \ , \qquad (13b)$$

packet arrival has become a poisson process.

3.5 The Service Facility

The average and variance of the number of packets departing from the system can be obtained by considering the service facility. In Fig. 6 the service facility is a trunk with speed C_T. It transmits packet after packet. The constant service time is, therefore, L_P/C_T. Hence,

$$E[D(\tau)] = \frac{C_T}{L_P}\tau \quad \text{and} \quad \mathrm{Var}[D(\tau)] = 0 \ . \qquad (14)$$

The utilization ρ in Eq. (8) is:

$$\rho = \lambda N L_P/C_T = \lambda L_M/C_T \ ,$$

L_M is the average message length.

For simplicity L_M/C_T was put equal to unity giving:

$$\rho = \lambda \ . \qquad (15)$$

3.6 Queue Statistics

From (9) with (11) and (14) we obtain:

$$\mu = \frac{C_T}{L_P} - \lambda N \ .$$

For high speed local loops ($C_L \to \infty$; $Y_o \to 0$) we get for the probability that the number of waiting packets is greater than n:

$$P_\infty\{W > n\} \approx \lambda \exp\{-2(n+1)(1-\lambda)/\lambda(2N-1)\} \ .$$

This asymptotic result is obtained from Eq. (8) with (10) and (13) to (15).

Similarly we get for low speed local loops ($C_L \to 0$; $Y_o \to \infty$)

$$P_o\{W > n\} \approx \lambda \exp\left[-2(n+1)\,\frac{1-\lambda}{\lambda}\right] \ .$$

For finite, non-zero local loop speed C_L we get from Eqs. (10, (12), and (14)

$$\sigma^2 = \lambda N\,\{1+2(N-1)\,[1-(1-\frac{1}{N})^{\lfloor \tau/Y_o \rfloor}]\} \ .$$

σ^2 depends on the duration of the time interval τ during which arrivals are observed. τ must be sufficiently long to justify the assumption of normally distributed arrivals. No argument to derive an appropriate value for τ, depending on λ and N could be found. It was, therefore, decided to run a simulation program for large $\lambda N (N=30,\ \lambda=0.8)$ where the diffusion approximation should produce satisfactory results. Then τ was chosen such that analytic and simulation results were in good agreement. During this chosen interval τ an average number of $z=30$ packets will arrive at the queue. For all further analytic results this value of z was held constant. Apparently, a "window" allowing 30 arrivals (average) seems sufficient to approximate a normal distribution of the number of arrivals.

Now

$$P\{W>n\} \approx \lambda \exp\{-2(n+1)(1-\lambda)/\lambda(1+2(N-1)[1-(1-\frac{1}{N})^{\lfloor \tau/Y_o \rfloor}])\} \qquad (16)$$

Recall that L_M/C_T was put equal to unity. (L_M is the average message length.) Then

$$\frac{\tau}{Y_o} = \frac{z\,C_L}{\lambda\,C_T} \ .$$

The average length of the waiting line is from Eq. (16):

$$E[W] \approx \lambda/(\exp\{2(1-\lambda)/\lambda(1+2(N-1)[1-(1-\frac{1}{N})^{\lfloor \tau/Y_o \rfloor}])\}-1) \qquad (17)$$

where $\lfloor \tau/Y_o$ is the largest integer smaller than or equal to τ/Y_o.

3.7 Results and Discussion

We will first inspect the expression Eq. (17) for the average length of the waiting line of packets. It should produce well-known results for the asymptotic cases $C_L \to \infty$ and $C_L \to 0$ as well as for $N = 1$ (one packet per message).

At high local loop speeds ($C_L \to \infty$) messages arrive instantaneously. The message length was assumed to have a geometric distribution with an average number of N packets. λ is the call rate to a certain destination (Fig. 6). Equation (17) gives for the average length of the waiting line of packets which queue up for transmission

$$E_\infty[W] \approx \lambda/\{\exp[2(1-\lambda)/\lambda\ (2N-1)] - 1\} \ .$$

For sufficiently large N and λ close to unity E_∞ is a very good approximation of the excact result.[10]

For low speed local loops ($C_L \to 0$) messages arrive dispersed and overlapping in time, packet arrival then becomes a poisson process, Section 2.5. From Eq. (17)

$$E_o[W] \approx \lambda/\{\exp\,[2(1-\lambda)/\lambda] - 1\} \ .$$

This result is a good approximation for λ close to one. The same result is obtained for $N = 1$, which is intuitively clear.

$$E_\infty[W]/E_o[W] \approx 2N - 1 \ .$$

This ratio shows very clearly the potential savings in a packet switching system where messages are quantized into packets and transmission (or other types of service, e.g., processing) is carried out packet by packet. It must be emphasized, however, that savings can only be achieved when the local loops connecting terminals to a node are sufficiently slow. (Technically one can not choose N arbitrarily large because of addressing overhead accompanying each packet.)

Figure 7 illustrates the influence of local loop speed on the average buffer space occupied by packets waiting for service, i.e., transmission. The average message length $L_M = NL_P$ was held constant and L_M/C_T was put equal to unity. Curves are shown for various packet lengths Lp. The average occupied buffer space is obtained from Eq. (17) by multiplying by Lp. Figure 8 shows the asymptotic behavior for $C_L \to \infty$ and $C_L \to 0$. It is important to see that the potential savings in buffer space given above can only be expected when local loop speed is very low, roughly 1/40 of the trunk speed.