Buffer Management in a Packet Switch

MAREK I. IRLAND

Abstract—Consider a single packet switch with a finite number of packet buffers shared between several output queues. An arriving packet is lost if no free buffer is available, as in the CIGALE network. It has been observed by simulation that if load increases too much, congestion may occur, i.e., throughput declines; it appears that the busiest link's queue tends to hog the buffers. Therefore, we will limit the queue length and when the queue is full the packet will be dropped. We expect that this restricted buffer sharing policy will avoid congestion under conditions of heavy load.

A queueing model of a packet switch is defined and solved by local balance. Loss probability is evaluated, and values of queue limit to minimize loss are found; they depend on load. A Square-Root rule is introduced to make the choice of queue limit independent of load. For a sample switch, with three output links, a comparison is made between performance under different buffer sharing policies; it is shown that restricted sharing prevents congestion by making throughput an increasing function of load.

I. INTRODUCTION

Packet-switched communications networks were modeled by Kleinrock [7], making the simplifying assumption that an unbounded number of buffers is available in each packet switch. As a result, there is no limit on the size of waiting queues and no packets are lost.

Recently we have investigated the French CIGALE packet-switched network [12]. In this network each packet switch has a fixed number of buffers. A buffer can contain one packet up to the maximum of 255 characters, although only the filled part of a buffer is transmitted. The buffers are shared between all queues in the switch on a first-come-first-served basis. If a packet arrives at a switch when all its buffers are occupied, it is lost. An efficient simulator [2] was used to investigate CIGALE; in particular, to find the network throughput, packet loss rate, link utilization, and average end-to-end delay. The results [3] indicated that a new mathematical model of a packet switch with a finite storage assumption was needed to explain the behavior of the above performance measures. It was observed that unbalanced traffic in a packet switch caused hogging of most buffers by a single output link, thus preventing packets for less utilized links from getting through. As load for one link was increased beyond a certain point the total switch throughput declined, i.e., the switch became congested. Buffer management policies restricting the sharing of buffers are proposed and incorporated into the model in order to reduce loss probability and to ensure that throughput increases with increasing load.

The problem of buffer overflow in a packet-switched network has also been considered by Lam and Schweitzer [9, 13]; however, they have not dealt with restricted buffer sharing. An analysis of shared storage in a special case of balanced load has been presented by Kamoun and Kleinrock [6]. Pennotti and Schwartz have treated control of buffers in tandem networks [10].

II. PACKET SWITCH MODEL

A packet switch is shown in Figure 1. The fixed routing procedure [8] is used; execution of this procedure is assumed to be much faster than link transmissions and is omitted from the queueing model in Figure 2. In general there are N output links; a packet whose routing is via output link n is called an n-packet (index n always varies over integers 1, ..., N). It is assumed that the total number of buffers in the switch is K < ∞ with each output queue limited to M; the same M applies to all queues. A buffer can hold one packet of arbitrary length. When all K buffers are busy, an incoming packet is lost; an n-packet is also dropped if the queue for output link n is full. The following stochastic assumptions are used in the analysis.

A) Arrivals are Poisson with the total mean rate \( \lambda = \sum_{n=1}^{N} \lambda_n \) where \( \lambda_n \) is the mean arrival rate of n-packets.

B) The packet length is an exponential random variable; since the transmission time is proportional to packet length, the service time for output link n is also an exponential random variable with mean \( 1/\mu_n \).

The number of packets in the queue for output link n, including the packet in transmission, is denoted by \( k_n \). The following conditions must be satisfied:

C) \( \sum_{n=1}^{N} k_n \leq K \)

D) \( k_n \leq M_n \) for all \( n = 1, ..., N \).

The Unrestricted Buffer Sharing policy corresponds to setting \( M = K \). The purpose of the following analysis is to find the values of \( M \) to minimize the loss probability for the
Optimal Restricted Buffer Sharing policy. Two other restricted 
sharing policies are also considered: No-Sharing (or fixed store-
partition) where $M = K/N$ (rounded down), and Square-Root 
Sharing with $M = K/\sqrt{N}$ (rounded to the closest integer).

III. STABLE STATE SOLUTION

Let $(k_1, \cdots, k_N)$ denote a state of the system; a state is 
feasible if it satisfies conditions C) and D) stated in Section II.
Let $P(k_1, \cdots, k_N)$ be the stable state probability that the sys-
tem is in $(k_1, \cdots, k_N)$, where $P(k_1, \cdots, k_N) = 0$ for any state 
which is not feasible. The state transition diagram is presented 
in Figure 3 (for $N = 2$).

It is possible to write down the global balance equations for 
the system under consideration, but the number of equations 
is large due to the large number of boundary conditions. 
Instead, the method of local balance [1] is applied. The local 
balance equations are

$$
\mu_1 P(k_1 + 1, \cdots, k_n, \cdots, k_N) = \lambda_1 P(k_1, \cdots, k_n, \cdots, k_N)
$$

$$
\vdots
$$

$$
\mu_n P(k_1, \cdots, k_n + 1, \cdots, k_N) = \lambda_n P(k_1, \cdots, k_n, \cdots, k_N)
$$

$$
\vdots
$$

$$
\mu_N P(k_1, \cdots, k_n, \cdots, k_N + 1) = \lambda_N P(k_1, \cdots, k_n, \cdots, k_N).
$$

(1)

No special equations are needed for the boundary conditions

as they are covered by (1) above. We define the load of link $n$ 
to be $\rho_n = \lambda_n/\mu_n$. Then (1) has a solution of the form

$$
P(k_1, \cdots, k_N) = \begin{cases} 
C^{-1} \prod_{n=1}^{N} \rho_n^{k_n}, & \text{if the state is feasible} \\
0, & \text{otherwise}
\end{cases}
$$

(2)

where $C$ is a normalizing constant chosen so that the sum of 
the probabilities of all states is unity. This implies

$$
C = \sum_{n=1}^{N} \prod_{n=1}^{N} \rho_n^{k_n}
$$

(3)

where the sum is taken over all feasible states. When $\rho_1, \rho_2, \cdots, 
\rho_N$ and the number 1 are all distinct, $C$ is given by the following 
expression (4) derived in the Appendix; otherwise, $C$ can be 
produced from (4) by L'Hospital's rule. For the index set 
$I = \{1, \cdots, N\}$ denote by $P$ the set of all $j$-element subsets of $I$, 
and by $I_{n,j}$ the set of all $j$-element subsets of $I\backslash\{n\}$; also, for 
any $W \subseteq I$, let $\rho^W = \prod_{i \in W} \rho_i$. Then

$$
C = \frac{1 + \sum_{j=1}^{J} (-1)^j \sum_{W \subseteq I} \rho^W^{M+1}}{\prod_{n=1}^{N} (1 - \rho_n)}
$$

$$
= \frac{\rho^N K^{N - J(M + 1)} \sum_{W \subseteq I_n} \rho^W^{M+1}}{\prod_{n=1}^{N} (1 - \rho_n)}
$$

(4)
where \( J \) is the least integer greater or equal to \( K/M - 1 \).

IV. LOSS PROBABILITY

An arriving \( n \)-packet is dropped when all \( K \) buffers are used, or when output queue \( n \) already has \( M \) packets. Therefore, the loss probability for an \( n \)-packet is

\[
L_n = Pr \left\{ k_1 + \cdots + k_M = M \right\} + Pr \left\{ k_1 + \cdots + k_M < M \right\}
\]

where \( D_n \) is the family of states

\[
D_n = \{(k_1, \ldots, k_M) : k_m = M \text{ and } \sum_{m=1}^{M} k_m \leq M \}.
\]

Using the stable state solution (2) the loss formula is \( L_n = B_n/C \) where

\[
B_n = \sum_{k_1 + \cdots + k_M = M} \prod_{m=1}^{M} \rho_m k_m + \sum_{D_n} \prod_{m=1}^{M} \rho_m k_m
\]

(5)

and \( C \) is the normalizing constant. Notice that the second sum above is similar to expression (3) except for the state domain. If \( C \) is parameterized by \( C = C(K; M; \rho_1, \ldots, \rho_M) \) then this sum can be rewritten as \( \rho_n M C_n - \) where we define \( C_n = C(K-M-1; M; \rho_1, \ldots, \rho_{n-1}, \rho_n, \ldots, \rho_M) \) and thus it too may be computed using (4). Denote the first sum in (5) by \( G \). Using a procedure similar to that outlined for \( C \) in the Appendix, we derive

\[
G = (-1)^J \sum_{n=1}^{N} \rho_n^{K+M-1-J(M+1)} \sum_{w \in I_n} \rho_w^{M+1} \prod_{m=1}^{M} (\rho_n - \rho_m)
\]

(6)

where \( J \) is again chosen as in (4). Altogether, the loss formula for the \( n \)-packets becomes

\[
L_n = \frac{G + \rho_n M C_n -}{C}
\]

(7)

where \( G \) can be computed from (6), and \( C \) and \( C_n - \) from (4).

The overall loss probability for any packet arriving at the switch is a weighted sum of the above; namely \( L = \sum_{n=1}^{N} \rho_n L_n \) where \( \rho_n = \lambda_n/\lambda \) is the probability that an arriving packet is an \( n \)-packet. By (7) this loss probability is

\[
L = \frac{G + \sum_{n=1}^{N} \rho_n \rho_n M C_n -}{C}
\]

(8)

A simple interpretation of (8) is that \( G \) corresponds to loss when all waiting room is used, and \( \rho_n \rho_n M C_n - \) represents loss due to queue \( n \) being full.

For instance in the case of \( N = 2 \), for \( K/2 \leq M \leq K \), the loss probability is \( L = B/C \) where

\[
B = \frac{\rho_1^{K-M} \rho_2^{M+1} - \rho_2^{K-M} \rho_1^{M+1}}{\rho_2 - \rho_1} + \frac{\rho_1 \rho_2^{M+1}}{1 - \rho_2} + \frac{\rho_2 \rho_1^{M+1}}{1 - \rho_1}
\]

and

\[
C = \frac{1 - \rho_1^{K+1} - \rho_2^{M+1}}{(1 - \rho_1)(1 - \rho_2)} + \frac{\rho_1^{K-M+1} \rho_2^{M+1}}{(1 - \rho_1)(\rho_1 - \rho_2)} + \frac{\rho_2^{K-M+1} \rho_1^{M+1}}{(1 - \rho_2)(\rho_2 - \rho_1)}
\]

If \( \rho_1 = 1, \rho_2 = 1, \) or \( \rho_1 = \rho_2 \), L'Hospital's rule may be employed. For example when \( \rho_1 = \rho_2 = 1 \), for \( K/2 \leq M \leq K \),

\[
L = \frac{2(M+1)}{(K+1)(K+2) - 2(K-M)(K+1)}
\]

(9)

which may also be obtained directly from the geometrical view of the state space (Figure 3). Each state is equally probable in this case, therefore, the loss probability is the ratio of the length of the boundary \( k_1 = M, k_2 = M \) or \( k_1 + k_2 = K \) to the total area representing the state space.

V. OPTIMAL RESTRICTED BUFFER SHARING POLICY

Due to the large number of special cases, the closed-form algebraic formulae presented above are not very convenient for actual computations. In [4] we have developed computational algorithms based on the convolution approach (starting with equations (A1) and (A2) in the Appendix) for the normalizing constant, loss probability and mean queue length. The algorithms are efficient in the sense that the numbers of additions and multiplications are in the order of \( 2NK \). Implemented in APL, the resulting computer program has been used to evaluate \( L \) for integer values of \( M \), and to determine the best value to minimize probability of packet loss. As an example, the case of \( N = 2 \) is considered with the total number of buffers \( K = 10 \). The best values of \( M \) are tabulated in Table 1 as a function of \( \rho_1 \) and \( \rho_2 \), where \( \mu_1 = \mu_2 \) is assumed. Note that under a loss discipline the system remains stable even for \( \rho_n > 1 \). It may be observed in Table 1 that for reasonably small values of \( \rho_1 \) and \( \rho_2 \) the choice of best \( M \) depends primarily on the sum of \( \rho_1 \) and \( \rho_2 \).

VI. THE SQUARE-ROOT RULE

In general the optimal choice of \( M \) depends on \( \rho_n \) for all \( n = 1, \ldots, N \) and therefore the buffer management parameter \( M \) requires readjustment as traffic characteristics through the
switch change. This may be difficult to do in practice because it involves measuring the load and leads to adaptive control problems. Instead, a fixed value may be chosen which although it involves measuring the load and leads to adaptive control policies, it is not optimal but allows the switch to operate reasonably well. If we choose a value of $M$ which minimizes the function $L$, we can consider a corresponding function defined on the real interval. This function is continuous, differentiable and convex. (It is easy to check that its values at the end-points of the interval are equal.) By differentiating with respect to $M$ the function is minimized by

$$M = \sqrt{\frac{K^2 + 3K + 2}{2} - 1}$$

which is closely approximated by $M = K/\sqrt{2}$. Thus $L$ will be minimized by one of the integers surrounding the value produced by (10). As we are interested here only in a reasonable setting of $M$, we will use the simper expression $M = K/\sqrt{2}$ rounded to the nearest integer.

A geometrical interpretation is obtained by observing that for $\rho_1 = \rho_2 = 1$ all states $(k_1, k_2)$ in the feasible-state space (Figure 3) are equally probable. Therefore, $L$ is the ratio of the length of the boundary $k_1 = M, k_2 = M$ or $k_1 + k_2 = K$ to the total area representing the state space. On geometrical grounds, this ratio is smallest when the boundary approximates an arc of a circle, which happens for $M = K/\sqrt{2}$. This geometrical interpretation extends to a multi-dimensional case for an arbitrary $N$, with $M$ becoming $M = K/\sqrt{N}$.

We call the buffer management policy with $M = K/\sqrt{N}$ rounded to the nearest integer the Square-Root rule. In the following section it is compared with the Optimal Sharing rule ($M$ set to the optimal value), and with the No-Sharing ($M = K/N$ rounded down) and Unrestricted Sharing ($M = K$) rules. Again, the Square-Root rule is attractive as it provides a good approximation to the optimal policy, but is much easier to implement in a real packet switch.

VII. SAMPLE RESULTS

In this section numerical results for the performance of a single switch are illustrated under the four buffer management policies. It is assumed that the switch has three output links of equal transmission capacity $\mu_1 = \mu_2 = \mu_3 = \mu$. We consider three different storage capacities: $K = 10, 20$ or $30$ packets. Two kinds of traffic patterns are investigated. The first is a balanced-load situation in which variable traffic rate, $\lambda$, is split evenly between three output links (i.e., $\lambda_1 = \lambda_2 = \lambda_3 = \lambda/3$); the second is an unbalanced-load situation in which the traffic rate for only one link is varied, while load for the other two links remains constant at $\rho_2 = 0.5$ and $\rho_3 = 0.7$.

In the performance graphs referring to the balanced-load situation, the horizontal axis represents load $\rho_1 = \rho_2 = \rho_3$; for the unbalanced-load situation the horizontal axis represents the variable $\rho_1$.

First we discuss the switch throughput. In Figure 4a, b and c, a comparison is made of throughput under the four buffer management policies in the balanced-load situation (for $K = 10, 20$ and $30$ buffers, respectively); it is shown in a normalized form $(1 - L)(\rho_1 + \rho_2 + \rho_3)$. The Optimal Sharing rule is uniformly superior, followed by Square-Root and Unrestricted Sharing. As expected, for light loads below $50\%$ of link capacity, all policies produce essentially the same results. For loads above $50\%$ and below $100\%$, the No-Sharing rule is substantially worse than the others, although it approaches Optimal Sharing as load increases to infinity.

An intuitive reason why the Restricted (Optimal or Square-Root) Sharing rules outperform the Unrestricted one is the following. Under heavy-load conditions the buffers become a critical resource in a packet switch; one way to increase availability of buffers is to reduce buffer occupancy time per packet. Restricted Sharing does so by limiting queue length, thus making buffers available for shorter queues where they will be turned around faster. We should keep in mind that even in the balanced-load situation queue lengths fluctuate.

Notice that throughput does not fall under excessive load in this case, even with Unrestricted Sharing. In Figure 5a, b and c, the throughput curves are shown for the unbalanced-load situation. Notice that with Unrestricted Sharing throughput falls sharply under excessive load, over $100\%$ capacity of the busiest link. This result of our analysis confirms earlier simulation-based observations [3].

Intuitively, the busiest link monopolizes most of the buffers although it cannot increase its utilization beyond $100\%$. At the same time, traffic for less used links is rejected due to lack of buffers, although spare transmission capacity is wasted. Although, normally, the steady traffic is expected to remain well below $100\%$ capacity of any link, temporary overload—for example—will cause serious degradation of
Figure 4. Throughput: balanced load. (a) $K = 10$. (b) $K = 20$. (c) $K = 30$. 
Figure 5. Throughput: unbalanced load. (a) $K = 10$.
(b) $K = 20$. (c) $K = 30$. 
throughput. Thus congestion is possible if Unrestricted Sharing is in force.

We further observe that with Square-Root Sharing throughput declines moderately, but only under extremely high over-load. It does not decline when our Optimal Sharing policy is applied. Thus either the Optimal or Square-Root rules will protect the switch from congestion.

Next let us look at loss probability and its dependence on the number of buffers in the switch. In Figures 6 and 7, loss probability $L$ is shown for a “normal operating range” (load between 60% and 100% of link capacity). For clarity, only Optimal Sharing and Unrestricted Sharing curves are shown. The No-Sharing rule has loss probability well above the others, and outside the graph scale. Curves for the Square-Root rule would lie between those shown for the load range 70% to 100%, and slightly above for 60%. In this example, if only 10 buffers are provided the loss probability will be unacceptably high (more than one packet in a hundred dropped) for traffic levels exceeding 50% of link capacity, no matter which buffer management scheme is used. Twenty buffers allow an acceptable level of performance for up to 75% capacity in the balanced case, and up to 85% for the most loaded link in the unbalanced case considered. The availability of 30 buffers pushes these numbers to about 85% and 95%, respectively. It can be observed that our Optimal Sharing rule reduces the loss probability by up to one-half in comparison with Unrestricted Sharing.

When traffic is balanced the loss probabilities for all links are the same, i.e., $L_1 = L_2 = L_3 = L$. The advantage of Optimal Sharing is seen even more clearly for Unbalanced-Load in Figure 8 which separates loss probability for the individual links (when $K = 20$); this time $L_1, L_2$ and $L_3$ may not be equal. Under Unrestricted Sharing the loss probabilities are the same for all links, independent of their relative load. On the other hand, the Optimal rule produces much lower losses for less loaded links, while the busiest link suffers higher losses. We can say that this rule protects other traffic from saturation of one link.

Another important performance measure is mean delay (queueing plus transmission) shown in Figures 9 and 10 in units of mean packet transmission time (for $K = 20$ only). It too is improved by Restricted Sharing. Note that the irregular character of the delay curves for the Optimal rule is due to discrete changes in the queue limit parameter. For reference, the mean delay curve for a switch with infinite storage is also shown.

VIII. CONCLUSIONS

In this paper a finite storage model of a packet switch was introduced and analyzed. It helped us to explain behavior of performance parameters measured in our earlier simulation work, in particular, the decline of throughput while load increased beyond a certain point under Unrestricted Sharing. Two Restricted Buffer Sharing policies, Optimal and Square-Root, were proposed. Numerical results point out that both
improve switch performance. Optimal Sharing made throughput an increasing function of load, and Square-Root Sharing almost did so. In this sense these policies provide congestion control. We also considered the No-Sharing policy; although simple to implement, it results in unnecessarily large loss probability.

Intuition suggests that our policies could be further improved by allowing different limits for each queue rather than the same limit $M$, as we did. Our computational algorithms could be modified to handle this generalization but we have felt that performance improvement would be slight, while determining the individual limits and practical implementation in a real switch would be more expensive. Some work in this direction has recently been undertaken by Potier [11].

Finally, in [5] we extend the analysis of a single switch to the communications subnetwork of a packet-switched network.

APPENDIX

We derive expression (4) here. Define

$$S_n = 1, \rho_n, \rho_n^2, ..., \rho_n^M, \rho_n^{M+1}, \rho_n^{M+2}, ...$$

$$T_n^M = 1, \rho_n, \rho_n^2, ..., \rho_n^M, 0, 0, ...$$

$$U_n^M = 1, 0, 0, ..., 0, -\rho_n^{M+1}, 0, ...$$

$(-\rho_n^{M+1}$ occupies position $M+1)$

to be infinite sequences. Let * denote the convolution operator. It is easy to check that

$$T_n^M = S_n * U_n^M.$$  (A1)
By (3) $C$ can be conveniently expressed as

$$C = \sum_{k=0}^{K} (T_1^M * \cdots * T_N^M)(k). \quad \text{(A2)}$$

Convolution yields nicely to the $z$-transform approach. Denote the $z$-transform of any sequence $X$ by $(X)'$. We have $(S_n)' = z/(z - \rho_n)$ and $(U_n^M)' = 1 - (\rho_n/z)^{M+1}$. Hence by (A1),

$$(T_n^M)' = (S_n)'(U_n^M)' = z^{-M} \frac{2^M + 1 - \rho_n^{M+1}}{z - \rho_n} . \quad \text{(A3)}$$

With $M$ fixed, consider $C$ in (A2) as an infinite sequence with regard to parameter $K$. Then (A2) implies that

$$C' = \frac{z}{z-1} (T_1^M * \cdots * T_N^M)'$$

$$= \frac{z}{z-1} (T_1^M)' \cdots (T_N^M)' .$$

Substituting (A3) the $z$-transform of $C = C(K)$ becomes

$$C' = z^{-NM} \frac{z}{z-1} \frac{2^M + 1 - \rho_n^{M+1}}{z - \rho_n}$$

$$\cdots \frac{z^{M+1} - \rho_n^{M+1}}{z - \rho_n} . \quad \text{(A4)}$$

Below we show how to find the inverse $z$-transform of $C'$. Multiplying out the right-hand side of (A4) yields

$$C' = \frac{z^N}{(z-1)(z-\rho_1) \cdots (z-\rho_N)}$$

$$\frac{z^{N-1}}{(z-1)(z-\rho_1) \cdots (z-\rho_N)} \frac{N}{z} \sum_{n=1}^{N} \rho_n^{M+1}$$

$$\frac{z^{N-2}}{(z-1)(z-\rho_1) \cdots (z-\rho_N)} \frac{N}{z} \sum_{n=1}^{N} \sum_{m=n+1}^{N} \rho_n^{M+1} \rho_m^{M+1}$$

$$\cdots$$

$$\frac{z}{(z-1)(z-\rho_1) \cdots (z-\rho_N)}$$

$$\frac{N}{z} \prod_{n=1}^{N} \rho_n^{M+1} .$$

Partial-fraction expansion produces

$$C' = \left( \frac{\phi}{z-1} - \sum_{n=1}^{N} \frac{z}{z-\rho_n} \psi_n \rho_n^{N} \right)$$

$$- \left( \frac{\phi}{z-1} - \sum_{n=1}^{N} \frac{z}{z-\rho_n} \psi_n \rho_n^{N-1} \right) z^{-M} \sum_{n=1}^{N} \rho_n^{M+1}$$

$$+ \left( \frac{\phi}{z-1} - \sum_{n=1}^{N} \frac{z}{z-\rho_n} \psi_n \rho_n^{N-2} \right)$$

$$\cdots$$

$$\frac{N}{z} \prod_{n=1}^{N} \rho_n^{M+1} .$$

where

$$\phi^{-1} = \prod_{n=1}^{N} (1 - \rho_n) \text{ and } \psi^{-1} = \prod_{m=1}^{N} (\rho_n - \rho_m) .$$

The inverse $z$-transform can be obtained by the right-shifting property

$$C = \left( \phi - \sum_{n=1}^{N} \psi_n \rho_n K + N \right)$$

$$- \left( \phi - \sum_{n=1}^{N} \psi_n \rho_n K + N - (M+1) \right) \sum_{n=1}^{N} \rho_n^{M+1}$$

$$+ \left( \phi - \sum_{n=1}^{N} \psi_n \rho_n K + N - 2(M+1) \right) \sum_{n=1}^{N} \sum_{m=n+1}^{N} \rho_n^{M+1} \rho_m^{M+1}$$

$$\cdots$$

$$\frac{N}{z} \prod_{n=1}^{N} \rho_n^{M+1} .$$

where only the first $J + 1$ rows are taken, for $J$ equal the least integer greater than or equal $K/M - 1$. By manipulating these first $J + 1$ rows a more compact equation (4) is derived.

ACKNOWLEDGMENT

The author thanks Dr. E. G. Manning for his continued advice and encouragement, and Drs. J. Mark and J. Wong for their critical help while preparing this manuscript. Valuable suggestions made by Drs. K. M. Chandy, M. Thompson and the referees are also gratefully acknowledged.
REFERENCES


Marek I. Irland was born in Poland on February 3, 1946. He studied mathematics at the Warsaw University and computer science at the University of Illinois, Urbana, where he received the M.Sc. degree in 1969. Subsequently he moved to Canada where, at the University of Waterloo, he completed the M.Phil degree in 1972 and Ph.D. in 1977, also in computer science.

Dr. Irland's early professional experience included work on an ALGOL compiler at the Institute for Mathematical Machines in Poland, and work on list processing programming languages at the University of Illinois. At Waterloo he was at first interested in computational complexity, but since 1973 his activities shifted to problems of computer communications. As a member of the Computer Communications Networks Group at the University of Waterloo he consulted, among others, to the Reseau CYCLADES project in France and he developed a distributed simulator of computer networks. He became a Lecturer of computer science in 1972 and an Assistant Professor in 1977.


Delay and Throughput Evaluation of Switching Methods in Computer Communication Networks

HIDEO MIYAHARA, MEMBER, IEEE, YOSHIMI TESHIGAWARA, MEMBER, IEEE, AND TOSHIHARU HASEGAWA, ASSOCIATE MEMBER, IEEE

Abstract—Message, packet and line switching in computer communication networks are analyzed by a queueing model. Message transmission delay time and network throughput between a source-destination node pair are obtained as a function of various parameters including message length, traffic arriving at the network, and the number of switching nodes existing between the nodes. A criterion to determine the most suitable switching method under the given conditions is offered. Also, the maximum length of a packet in packet switching is discussed.


H. Miyahara and T. Hasegawa are with the Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, Kyoto, Japan.

Y. Teshigawara is with EDP Systems Support Division, Nippon Electric Co., Ltd., Ninato-ku, Tokyo, Japan.

0090-6778/78/0300-0337$00.75 © 1978 IEEE

I. INTRODUCTION

 In attempts to use computers more effectively and more powerfully by sharing resources and leveling the load among computers, computer networks have been favored. These networks use advanced communication and computer technology. Speed and accuracy in the exchange of information between computers are requirements for computer networks to fulfill these functions. Store-and-forward and line switching procedures of transmission and switching of data in such computer networks are considered in this paper. Each method has different characteristics. Cost and performance vary widely depending on the external circumstances of a system. Therefore, criteria to find the optimal method under certain given conditions are required. The purpose of this paper is to evaluate three switching methods, i.e., message, packet and line switch-