

# Store-and-Forward Buffer Requirements in a Packet Switching Network

SIMON S. LAM, MEMBER, IEEE

**Abstract**—Previous analytic models for packet switching networks have always assumed infinite storage capacity in store-and-forward (S/F) nodes. In this paper, we relax this assumption and present a model for a packet switching network in which each node has a finite pool of S/F buffers. A packet arriving at a node in which all S/F buffers are temporarily filled is discarded. The channel transmission control mechanisms of positive acknowledgment and time-out of packets are included in this model. Individual S/F nodes are analyzed separately as queueing networks with different classes of packets. The single node results are interfaced by imposing a continuity of flow constraint. A heuristic algorithm for determining a balanced assignment of nodal S/F buffer capacities is proposed. Numerical results for the performance of a 19 node network are illustrated.

## INTRODUCTION

IN THE past, analytic models for packet switching networks have always assumed infinite storage capacity in store-and-forward (S/F) nodes [1], [2]. This, together with the independence assumption (due to Kleinrock [3]), reduce a very difficult problem to an open network of queues problem [4], [5]. The latter can then be decomposed into separate analyzable single-server problems which reflect the network structure and traffic flows [1]. Such analytic models have been used in conjunction with simulation models, heuristic procedures, and experimentation in the performance evaluation and design of actual networks [6]. It was concluded that these analytic models are valuable in providing insight into network behavior as well as providing keys to good heuristic design procedures and ideal performance bounds. Nevertheless, many important network operating features and practical constraints have been omitted in these models, such as finite nodal storage capacity, priority classes of packets, packetizing and reassembly of messages, adaptive routing and flow control schemes, etc. The analysis of a model which includes most of the aforementioned features and constraints is extremely difficult (if at all possible). In this paper, we relax the assumption of infinite nodal storage and present a model to study 1) the degradation in network performance due to this additional constraint of limited storage capacity, and 2) the S/F nodal buffer requirements in a packet switching network to achieve some small probability of nodal blocking. The problem of nodal blocking has been studied before in the narrower context of statistical multiplexing by Chu [7] or using more simplified models of a S/F node by Ziegler [8] and Closs [9]. In this paper, the S/F node model studied by Schweitzer and Lam [10] is generalized to a multinode network.

In the next section, our network model is first presented. The overall problem, instead of being decomposed into single-server problems as in earlier works [1]–[3], is decomposed into single-node problems; each such single-node problem corresponds to a queueing network model of a S/F node. This formulation enables us to incorporate into the model the constraints of finite nodal storage capacities, as well as the channel transmission control mechanisms of positive acknowledgment and time-out of packets [11], [12]. The routing of packets in the network is modeled by Markovian transition matrices. For each S/F node, steady-state queue length statistics are obtained by applying queueing network theory [5]. These single-node results are interfaced by imposing a continuity of flow constraint; an efficient computational procedure is given for iterative solution of the resulting set of nonlinear equations. Analytic results are presented for various network performance measures including average packet delay, network throughput rate, and nodal blocking probabilities. A heuristic algorithm is proposed for determining a balanced assignment of S/F buffer capacities to achieve some small probability of nodal blocking. Finally, numerical results illustrating the performance of a 19 node network are shown. We conclude that the model and analytic results in this paper may be used to supplement other network design and optimization techniques based upon the simpler analytic models which assume infinite nodal storage [1]–[3].

## THE NETWORK AND NODE MODELS

We consider a packet switching network consisting of  $M$  nodes as shown in Fig. 1. Each node has a finite number  $N_i$  ( $1 \leq i \leq M$ ) of S/F buffers. We assume that each packet of data can be stored in exactly one buffer. All S/F buffers in a node form a common pool for the storing of all packets being forwarded by the node.

As a packet moves through the network from its source node to its destination node, each intermediate node stores a copy of the packet until a positive acknowledgment (ack) is returned from the succeeding node. The ack indicates that the packet has been received without error and has been accepted. Once a node has accepted a packet and returned an ack, it holds onto a copy of that packet until it in turn receives an ack from the succeeding node. However, a node may refuse to accept a packet by not returning an ack. It may do so for the following reasons: 1) the packet has been received with one or more bits in error, and 2) all S/F buffers are temporarily filled. (This latter event will be referred to as *nodal blocking*). In the absence of an ack within some time-out interval, the transmitting node of the unsuccessful packet retransmits it.

Paper approved by the Associate Editor for Computer Communication of the IEEE Communications Society for publication without oral presentation. Manuscript received July 10, 1975; revised November 17, 1975.

The author is with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

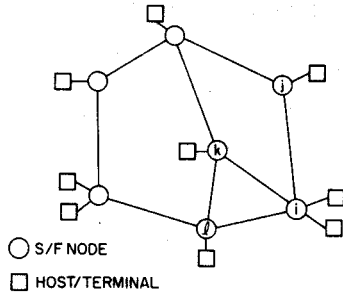


Fig. 1. The network model.

For the sake of mathematical tractability, we assume that all inputs from hosts and terminals into the S/F network consist of single-packet messages only. Thus, we have ignored the consideration of packetizing and reassembly of long messages [11], [12]. (In this context, packet switching becomes synonymous with message switching.) Furthermore, no priority structure is assumed. Also, since acknowledgments can be returned "piggy-backed" on normal network traffic in a set of acknowledgment bits [12], the overhead for transmitting acks is neglected in our model.

To model the routing of packets through the network,  $R$  classes of packets are distinguished. A specific class of packets may, for examples, consist of 1) all packets with a common destination node, 2) all packets between a given source-destination pair of network nodes, and 3) all packets between a given source-destination pair of hosts and terminals. The routing of class  $r$  ( $1 \leq r \leq R$ ) packets is specified by a set of Markovian transition probabilities  $\{P_{ij}^{(r)}, 1 \leq i \leq M, 1 \leq j \leq M+1\}$ ; a class  $r$  packet, currently at node  $i$ , is next routed to node  $j$  with probability  $P_{ij}^{(r)}$ . Note that  $\sum_{j=1}^{M+1} P_{ij}^{(r)} = 1$  for all pairs of  $i$  and  $r$ . The channel  $(i, M+1)$  models a local channel from node  $i$  to its external sink.<sup>1</sup> Thus, if the destination host of class  $r$  packets is attached to node  $i$ , we must have  $P_{i, M+1}^{(r)} = 1$ . The routing of packets in this model corresponds to nonadaptive routing policies and may be completely specified by a set of matrices,  $\mathbf{P}^{(r)} = [P_{ij}^{(r)}]$ ,  $1 \leq r \leq R$ . The special case of fixed (nonbifurcated) routing is obtained when  $P_{ij}^{(r)}$  takes on the values of 0 and 1 only for all  $i, j, r$ . Finally, we assume that  $P_{ii}^{(r)} = 0$  for all  $r$ , i.e., a node may not route a packet back to itself.

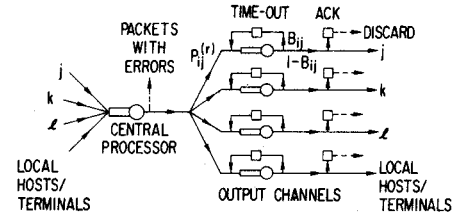
Both adaptive routing and flow control policies introduce complex state dependencies into the queueing analysis and will not be considered in this paper.

A fixed network input traffic pattern from external sources is assumed and is given by the vectors  $\sigma^{(r)}$ ,  $1 \leq r \leq R$ . The actual network input traffic  $S^{(r)}$  is specified by  $\sigma^{(r)}$  and a scaling factor  $\alpha$  referred to as the *network input level*, such that

$$S^{(r)} = \alpha \sigma^{(r)}, \quad 1 \leq r \leq R. \quad (1)$$

The elements of  $S^{(r)}$  are  $S_i^{(r)}$ ,  $1 \leq i \leq M$ , which are the input

<sup>1</sup> In a real network, the external sink (source) of node  $i$  may consist of a multiplicity of hosts and terminals. Without loss of generality, we assume that there is a single duplex channel [consisting of the simplex channels denoted by  $(i, M+1)$  and  $(0, i)$ ] connecting node  $i$  to its external sink (source).


 Fig. 2. The node model (node  $i$  in Fig. 1).

rates in packets per second of class  $r$  packets into the network nodes from their hosts and terminals. Define

$$S \triangleq \sum_{r=1}^R \sum_{i=1}^M S_i^{(r)}. \quad (2)$$

Under steady-state conditions,  $S$  is equal to the network throughput rate in packets per second.

We next describe our model of a S/F node depicted in Fig. 2. The central processor (which handles I/O channel interrupts, builds packet headers for its host/terminal inputs, checks errors, generates acks and routes packets, etc.) is modeled as a first-come-first-served (FCFS) queue. The output channels are also modeled as FCFS queues. The ack and time-out boxes shown in Fig. 2 model the storing of packets in node  $i$  before their acks return. Upon receipt of an ack, the buffer occupied by the acknowledged packet is freed and returned to the free buffer pool. Thus, the amount of time a packet spends in an ack box is a random variable equal to the interval between the end of its previous transmission and subsequent receipt of the ack for it. If a packet transmission is unsuccessful (which occurs in channel  $(i, j)$  with probability  $B_{ij}$ ), the packet spends a certain amount of time in the time-out box and is then put back on the output channel queue. Note that a packet, once stored in a buffer, does not have to be physically moved. Movement of packets depicted in Fig. 2 may be accomplished by software pointers.

We now state the assumptions needed for the analysis of individual S/F nodes in the next section.

1) The counting process of class  $r$  packets arriving at node  $i$ , including initial transmissions as well as retransmissions of previously unsuccessful packets from local hosts/terminals and neighboring nodes, is an independent Poisson process. The arrival rate of such class  $r$  packets to node  $i$  is  $\gamma_i^{(r)}$  packets/s.

2) The central processor of node  $i$  is a FCFS server with a negative exponential service time distribution and a service rate of  $\mu_i$  packets/s. The  $(i, j)$  channel is a FCFS server with a negative exponential service time distribution and a service rate of  $\mu_{ij}$  packets/s. (Note that this is just a modified version of Kleinrock's independence assumption [3].)

3) The time duration that a packet spends in the ack (time-out) box of channel  $(i, j)$  is an independent random variable with a general probability distribution and a mean value of  $v_{ij}(t_{ij})$  seconds. (Note that each such box corresponds to a service facility with no queueing, i.e., infinitely many servers.)

4) The steady-state nodal blocking probabilities are assumed to be  $B_i$ ,  $1 \leq i \leq M$ . Let  $e_{ij}$  be the probability that a

packet transmitted in channel  $(i,j)$  has one or more bits in error. Assuming nodal blocking and channel errors to be independent events, the steady-state probability of success for a packet transmitted over channel  $(i,j)$  is given by

$$1 - B_{ij} = (1 - e_{ij})(1 - B_j), \quad 0 \leq i \leq M, \quad 1 \leq j \leq M + 1. \quad (3)$$

With probability  $(1 - B_{ij})(B_{ij})^n$ , a packet retransmits exactly  $n$  times over channel  $(i,j)$  before success.

In our model, the absence of an ack for a packet within a time-out interval is assumed to be equivalent to the event that the packet was unsuccessful. However, it is possible that the packet has been received correctly and accepted by, say, node  $j$ . Yet the returning ack has been delayed or lost. Under these circumstances, the packet is retransmitted and node  $j$  receives a duplicate packet. In the design of an actual network, this necessitates proper choice of the length of the time-out interval and use of some packet sequencing scheme for duplicate detection.

### THE ANALYSIS

In this section, analytic results are first presented for individual S/F nodes modeled as queueing networks. These single node results are then interfaced by imposing a continuity of flow constraint. Finally, results for the evaluation of some network performance measures are shown.

#### Distribution of Queue Lengths in a S/F Node

Focus upon a S/F node, say node  $i$ , with class  $r$  packets arriving at  $\gamma_i^{(r)}$  packets/s. Let  $E_i$  be the fraction of packets with detected errors which are discarded by the central processor. (See Fig. 2.) The combined arrival rate to the output channels of node  $i$  is thus  $\gamma_i^{(r)}(1 - E_i)$  packets/s when node  $i$  is not blocked. The traffic intensity at a service facility is defined to be the ratio of its arrival rate to its service rate. We define the following traffic intensities for service facilities within node  $i$  when it is not blocked. For  $r = 1, 2, \dots, R$  and  $j = 1, 2, \dots, M + 1$

$$\begin{aligned} a_{i0}^{(r)} &\triangleq \text{traffic intensity of class } r \text{ packets at the central} \\ &\quad \text{processor of node } i \\ &= \frac{\gamma_i^{(r)}}{\mu_i} \end{aligned}$$

$$\begin{aligned} a_{ij}^{(r)} &\triangleq \text{traffic intensity of class } r \text{ packets at channel } (i,j) \\ &= \frac{\gamma_i^{(r)} P_{ij}^{(r)} (1 - E_i)}{(1 - B_{ij}) \mu_{ij}} \end{aligned}$$

$$\begin{aligned} b_{ij}^{(r)} &\triangleq \text{traffic intensity of class } r \text{ packets at time-out} \\ &\quad \text{box } (i,j) \\ &= \frac{\gamma_i^{(r)} P_{ij}^{(r)} B_{ij} t_{ij} (1 - E_i)}{(1 - B_{ij})} \end{aligned}$$

$$\begin{aligned} c_{ij}^{(r)} &\triangleq \text{traffic intensity of class } r \text{ packets at ack box } (i,j) \\ &= \gamma_i^{(r)} P_{ij}^{(r)} v_{ij} (1 - E_i). \end{aligned}$$

Let

$$a_{ij} \triangleq \sum_{r=1}^R a_{ij}^{(r)}, \quad 0 \leq j \leq M + 1$$

$$b_{ij} \triangleq \sum_{r=1}^R b_{ij}^{(r)}, \quad 1 \leq j \leq M + 1$$

$$c_{ij} \triangleq \sum_{r=1}^R c_{ij}^{(r)}, \quad 1 \leq j \leq M + 1.$$

Also,

$$b_i \triangleq \sum_{j=1}^{M+1} b_{ij}$$

$$c_i \triangleq \sum_{j=1}^{M+1} c_{ij}.$$

We next define the following notations:

$q_{i0} \triangleq$  Number of packets in the central processor queue of node  $i$

$q_{ij} \triangleq$  Number of packets in queue for channel  $(i,j)$

$m_{ij} \triangleq$  Number of packets in time-out box  $(i,j)$

$l_{ij} \triangleq$  Number of packets in ack box  $(i,j)$ ,  $1 \leq j \leq M + 1$

$q \triangleq (q_{i0}, q_{i1}, \dots, q_{i,M+1})$

$m \triangleq (m_{i1}, m_{i2}, \dots, m_{i,M+1})$

$l \triangleq (l_{i1}, l_{i2}, \dots, l_{i,M+1})$

$q_i \triangleq$  Number of packets in all FCFS queues of node  $i$

$$= \sum_{j=0}^{M+1} q_{ij}$$

$m_i \triangleq$  Number of packets in all time-out boxes of node  $i$

$$= \sum_{j=1}^{M+1} m_{ij}$$

$l_i \triangleq$  Number of packets in all ack boxes of node  $i$

$$= \sum_{j=1}^{M+1} l_{ij}.$$

From queueing network theory [5], the stationary probability distribution of queue lengths in node  $i$  is given by the following product form solution:<sup>2</sup>

$$\begin{aligned} P(q, m, l) &= P(0) (a_{i0})^{q_{i0}} \prod_{j=1}^{M+1} \\ &\quad \cdot (a_{ij})^{q_{ij}} \frac{(b_{ij})^{m_{ij}}}{m_{ij}!} \frac{(c_{ij})^{l_{ij}}}{l_{ij}!} \end{aligned} \quad (4)$$

<sup>2</sup> Note that for the sake of clarity, the subscript  $i$  denoting node  $i$  is suppressed wherever there is no ambiguity. Also, the expression  $x^y$  is equal to one by definition when  $x = y = 0$ .

where  $P(0)$  is a normalization constant equal to the probability of the event that node  $i$  is empty. From the above equation we then have

$$\begin{aligned}
 P(q, k) &= P(0) \left[ \prod_{j=0}^{M+1} (a_{ij})^{q_{ij}} \right] \\
 &\quad \sum_{\substack{M+1 \\ \sum_{j=1} (m_{ij} + l_{ij}) = k}} \frac{(b_{ij})^{m_{ij}} (c_{ij})^{l_{ij}}}{m_{ij}! l_{ij}!} \\
 &= P(0) \left[ \prod_{j=0}^{M+1} (a_{ij})^{q_{ij}} \right] \frac{(b_i + c_i)^k}{k!} \quad (5)
 \end{aligned}$$

To evaluate the normalization constant  $P(0)$  for node  $i$ , we define the following:

$$g(n) \triangleq \begin{cases} 1 & n = 0 \\ \sum_{\substack{M+1 \\ \sum_{j=0} q_{ij} = n}} \prod_{j=0}^{M+1} (a_{ij})^{q_{ij}} & n \geq 1 \end{cases} \quad (6)$$

$$p(N, x) \triangleq \begin{cases} 1 & N = 0 \\ \sum_{n=0}^N g(n) \frac{x^{N-n}}{(N-n)!} & N \geq 1 \end{cases} \quad (7)$$

$$G(K) \triangleq \sum_{N=0}^K p(N, b_i + c_i) \quad (8)$$

$$P(N) \triangleq \text{Pr [exactly } N \text{ packets in node } i].$$

From (5)-(7), we have

$$P(N) = P(0) p(N, b_i + c_i). \quad (9)$$

Since node  $i$  has  $N_i$  buffers,  $P(N)$ ,  $N = 0, 1, \dots, N_i$  must sum to one; the normalization constant is thus given by

$$P(0) = \left[ \sum_{N=0}^{N_i} p(N, b_i + c_i) \right]^{-1} = \frac{1}{G(N_i)} \quad (10)$$

which may be obtained from evaluating (6)-(8). A recursive algorithm [13], [14] for the computation of  $g(n)$  in (6) is given in Appendix A.

The nodal blocking probability  $B_i$  is equal to the probability of the event that all  $N_i$  S/F buffers in node  $i$  are filled. Thus,

$$B_i = P(N_i). \quad (11)$$

The expected number of packets in node  $i$ , denoted by  $\bar{n}_i$ , is given by

$$\bar{n}_i = \sum_{N=1}^{N_i} NP(N). \quad (12)$$

### Marginal Distributions and Expected Queue Lengths

The marginal probability distribution of the queue size  $q_{ij}$  in node  $i$  is given by

$$\begin{aligned}
 \text{Pr } [q_{ij} \geq n] &= \sum_{\substack{M+1 \\ \sum_{l=0} q_{il} + k \leq N_i \\ \& \ q_{ij} \geq n}} P(q, k) \\
 &= (a_{ij})^n \frac{G(N_i - n)}{G(N_i)}. \quad (13)
 \end{aligned}$$

Thus,

$$\text{Pr } [q_{ij} = n] = \frac{(a_{ij})^n}{G(N_i)} [G(N_i - n) - a_{ij} G(N_i - n - 1)]. \quad (14)$$

The expected queue size is

$$\bar{q}_{ij} = \frac{1}{G(N_i)} \sum_{n=1}^{N_i} (a_{ij})^n G(N_i - n). \quad (15)$$

These last three equations correspond to similar results derived by Buzen for a closed-queueing network [13].

Next, the marginal probability distribution of the total number  $m_i$  of packets in the time-out boxes of node  $i$  is given by

$$\text{Pr } [m_i = m] = P(0) \frac{b_i^m}{m!} \sum_{N=0}^{N_i-m} p(N, c_i) \quad (16)$$

with expected value

$$\begin{aligned}
 \bar{m}_i &= P(0) \sum_{m=1}^{N_i} \frac{b_i^m}{(m-1)!} \sum_{N=0}^{N_i-m} p(N, c_i) \\
 &= P(0) b_i \sum_{l=0}^{N_i-1} \frac{b_i^l}{l!} \sum_{N=0}^{N_i-1-l} p(N, c_i) \\
 &= P(0) b_i \sum_{N=0}^{N_i-1} p(N, b_i + c_i) \\
 &= b_i \frac{G(N_i - 1)}{G(N_i)} \\
 &= b_i (1 - B_i). \quad (17)
 \end{aligned}$$

The second moment is given by

$$\bar{m}_i^2 = [G(N_i)]^{-1} [b_i^2 G(N_i - 2) + b_i G(N_i - 1)]. \quad (18)$$

Replacing  $b_i$  by the appropriate traffic intensity, (16)-(18) also apply to the total number  $l_i$  of packets in the ack boxes of node  $i$  as well as to the number of packets in specific boxes and belonging to specific classes.

### Network Interface of Single Node Results

To properly interface the single node results, we invoke the conservation of flows principle: for any service facility, total

flow in must be equal to total flow out. Let  $\lambda_i^{(r)}$  denote the throughput rate in packets per second of class  $r$  packets at node  $i$  for  $1 \leq r \leq R$ ,  $1 \leq i \leq M$ . We define the row vectors

$$\lambda^{(r)} \triangleq [\lambda_1^{(r)} \lambda_2^{(r)} \dots \lambda_M^{(r)}], \quad 1 \leq r \leq R$$

and the matrices

$$P^{(r)} \triangleq \begin{bmatrix} P_{11}^{(r)} & P_{12}^{(r)} & \dots & P_{1M}^{(r)} \\ P_{21}^{(r)} & P_{22}^{(r)} & & \\ \vdots & & \ddots & \\ P_{M1}^{(r)} & & & P_{MM}^{(r)} \end{bmatrix}, \quad 1 \leq r \leq R.$$

The network input traffic is specified by  $S^{(r)}$ ,  $1 \leq r \leq R$ . Under equilibrium conditions, we must have for all  $i, r$

$$\lambda_i^{(r)} = S_i^{(r)} + \sum_{j=1}^M \lambda_j^{(r)} P_{ji}^{(r)}$$

or

$$\lambda^{(r)} = S^{(r)} + \lambda^{(r)} P^{(r)} \quad 1 \leq r \leq R. \quad (19)$$

Hence

$$\lambda^{(r)} = S^{(r)} [I - P^{(r)}]^{-1} \quad 1 \leq r \leq R \quad (20)$$

where  $I$  is an  $M \times M$  identity matrix.

Because of retransmissions, the arrival rate  $\gamma_i^{(r)}$  of class  $r$  packets to node  $i$  must be greater than the corresponding throughput rate  $\lambda_i^{(r)}$ . The number of retransmissions over a channel is assumed to be geometrically distributed for each packet. Thus, we have

$$\begin{aligned} \gamma_i^{(r)} &= \frac{S_i^{(r)}}{1 - B_{0i}} + \sum_{j=1}^M \frac{\lambda_j^{(r)} P_{ji}^{(r)}}{1 - B_{ji}} \\ &= \left[ \frac{S_i^{(r)}}{1 - e_{0i}} + \sum_{j=1}^M \frac{\lambda_j^{(r)} P_{ji}^{(r)}}{1 - e_{ji}} \right] \frac{1}{1 - B_i}. \end{aligned} \quad (21)$$

The fraction of packets accepted by node  $i$  with no detected errors is

$$1 - E_i = \lambda_i^{(r)} / \left[ \frac{S_i^{(r)}}{1 - e_{0i}} + \sum_{j=1}^M \frac{\lambda_j^{(r)} P_{ji}^{(r)}}{1 - e_{ji}} \right]. \quad (22)$$

Substituting (21) into (11), we obtain a set of  $M$  nonlinear simultaneous equations which can be solved for the  $M$  nodal blocking probabilities,

$$B_i = f_i(B_1, B_2, \dots, B_M, \alpha), \quad 1 \leq i \leq M. \quad (23)$$

In the above equations, the network input level  $\alpha$  has been shown explicitly as a parameter. (Recall that the network input traffic pattern is assumed to be fixed.)

### Network Performance Measures

The buffer utilization factor of node  $i$  is defined to be

$$U_i \triangleq \frac{\bar{q}_i + \bar{m}_i + \bar{l}_i}{N_i} = \frac{\bar{n}_i}{N_i} \quad (24)$$

where  $\bar{n}_i$  is given by (12);  $\bar{q}_i$ ,  $\bar{m}_i$ , and  $\bar{l}_i$  denote the expected total number of packets in the queues, time-out boxes, and ack boxes, respectively, of node  $i$ . The average buffer utilization factor for the network is defined to be

$$U \triangleq \frac{\sum_{i=1}^M \bar{n}_i}{\sum_{i=1}^M N_i}. \quad (25)$$

Applying Little's formula [15], the expected nodal delay for node  $i$  is

$$D_i = \frac{\bar{q}_i + \bar{m}_i}{\lambda_i} = \frac{\bar{n}_i - \bar{l}_i}{\lambda_i} \quad (26)$$

where  $\lambda_i \triangleq \sum_{r=1}^R \lambda_i^{(r)}$  is the throughput rate of node  $i$ . Note that the time a packet spends in an ack box does not contribute toward its delay. Applying Little's formula again, the average network delay incurred by a packet (after it has been admitted into the network) is

$$\begin{aligned} D &= \frac{\sum_{i=1}^M (\bar{q}_i + \bar{m}_i) + \sum_{i=1}^M \sum_{j=1}^{M+1} \lambda_{ij} d_{ij}}{S} \\ &= \frac{\sum_{i=1}^M (\bar{n}_i - \bar{l}_i) + \sum_{i=1}^M \sum_{j=1}^{M+1} \lambda_{ij} d_{ij}}{S} \end{aligned} \quad (27)$$

where

$$\lambda_{ij} = \sum_{r=1}^R \lambda_i^{(r)} P_{ij}^{(r)}$$

$$d_{ij} \triangleq \text{Propagation delay in channel } (i, j)$$

and  $S$  given by (2) is the network throughput rate under equilibrium conditions.

The average network blocking probability for host/terminal inputs is defined to be

$$B_H \triangleq \frac{\sum_{i=1}^M S_i' B_i}{\sum_{i=1}^M S_i'} \quad (28)$$

where

$$S_i' \triangleq \frac{S_i}{(1 - e_{0i})(1 - B_i)}$$

and  $S_i \triangleq \sum_{r=1}^R S_i^{(r)}$  is the packet input rate into node  $i$  from its hosts and terminals. The average internode blocking probability for node to node transmissions is defined to be

$$B_N \triangleq \frac{\sum_{i=1}^M (\gamma_i - S_i') B_i}{\sum_{i=1}^M (\gamma_i - S_i')} \quad (29)$$

where  $\gamma_i \triangleq \sum_{r=1}^R \gamma_i^{(r)}$  is the arrival rate of all packets to node  $i$ .

### COMPUTATIONAL ALGORITHMS

The analytic results presented so far depend upon solution of the set of nonlinear equations in (23) which we rewrite as

$$B = f(B, \alpha) \quad (30)$$

where  $B$  and  $f$  are  $M$ -dimensional column vectors. For a given  $\alpha$ , (30) can be solved through an iterative process in which an initial approximation  $B^0$  is used to generate a sequence of successive approximations  $B^1, B^2, \dots, B^k, \dots$  which converge to a solution. The Newton-Raphson method [16], [17] is used here such that

$$B^{k+1} = B^k - [I - \nabla f(B^k, \alpha)]^{-1} [B^k - f(B^k, \alpha)] \quad (31)$$

where  $\nabla f$ , the gradient of  $f$  with respect to  $B$ , is defined to be the following  $M \times M$  matrix:

$$\nabla f \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial B_1} & \frac{\partial f_1}{\partial B_2} & \dots & \frac{\partial f_1}{\partial B_M} \\ \frac{\partial f_2}{\partial B_1} & \frac{\partial f_2}{\partial B_2} & & \\ \vdots & & & \\ \frac{\partial f_M}{\partial B_1} & & & \frac{\partial f_M}{\partial B_M} \end{bmatrix} \quad (32)$$

For a stopping condition, we define the  $k$ th-iteration error estimate to be

$$\eta_k \triangleq \frac{\|f(B^k, \alpha) - B^k\|}{\|f(B^k, \alpha)\|}$$

where

$$\|x\| \triangleq \left( \sum_{i=1}^M x_i^2 \right)^{1/2}, \quad \text{for } x = [x_1 x_2 \dots x_M].$$

$B^k$  is accepted to be a solution of (30) if  $\eta_k$  is smaller than some prespecified convergence tolerance.

Given any reasonable network size  $M$  and nodal storage capacities  $N_i$ ,  $1 \leq i \leq M$ , a very efficient computational algorithm is needed for evaluating the gradient matrix  $\nabla f$ . In Appendix B, a solution for  $\nabla f$  is shown. Using this solution, the above computational procedure has been implemented in APL and currently runs on an IBM 370/168 system. For the 19 node network example to be described below, with  $B^0 = 0$  and a convergence tolerance of 0.0001, the above computational procedure usually converges within a few iterations. (However, if  $\alpha$  is large such that the network is near saturation, the above iterative procedure converges only if the initial approximation  $B^0$  is sufficiently close to the solution.)

For a given packet switching network, we would like to determine the S/F nodal storage capacities  $\{N_i\}$  such that 1) the sum  $\sum_{i=1}^M N_i$  is minimized, and 2) the resulting nodal blocking probabilities are each smaller than some prespecified value  $\epsilon$  ( $0 < \epsilon < 1$ ). We propose the algorithm below which generates a set of  $\{N_i\}$  by considering one S/F node at a time. We shall refer to it as the buffer capacity requirement algorithm (BCR). It can easily be shown that for a given  $\epsilon$ , the set of buffer capacities  $\{N_i\}$  determined by BCR satisfies condition 2) above. This buffer capacity assignment is also *balanced* in the sense that the resulting nodal blocking probabilities are approximately the same. Consequently, BCR is believed to be near optimal with respect to the criterion in 1) for a fixed value of  $B_H, B_N$  or  $B_{\max}$ .

#### Algorithm (BCR)

- Step 1: Let  $B_i = \epsilon$ ,  $1 \leq i \leq M$ .
- Step 2: For  $i = 1, 2, \dots, M$ , repeat the following steps.
- Step 3: Evaluate  $\gamma_i^{(r)}$  from (21) and the traffic intensities  $a_{ij}, b_i$ , and  $c_i$  for  $1 \leq r \leq R$  and  $0 \leq j \leq M + 1$ .
- Step 4:  $N_i \leftarrow 1$ .
- Step 5: Evaluate  $B_i$  from (6)-(11).
- Step 6: If  $B_i \leq \epsilon$  then  $N_i$  is the number of S/F buffers for node  $i$ . Otherwise,  $N_i \leftarrow N_i + 1$  and go to Step 5.

### A NETWORK EXAMPLE

In this section, numerical results for the performance of a packet switching network example are illustrated. The network to be considered consists of 19 S/F nodes<sup>3</sup> interconnected by 50, 19.2, and 9.6 kbits/s full duplex lines. (See Fig. 3). The routing algorithm is assumed to be fixed (nonbifurcated) and is completely specified by a  $19 \times 19$  fixed routing matrix<sup>4</sup>  $FRM = [FRM_{ij}]$ . Each element  $FRM_{ij}$  denotes the next node to forward a packet which is currently at node  $i$  and whose destination is node  $j$ . Since the routing of a packet at any intermediate node is based only upon its destination node, we distinguish  $R = 19$  classes of packets. The

<sup>3</sup>This 19 node network topology is the same as the one studied by Kleinrock [1].

<sup>4</sup>This routing matrix was generated by hand and attempts were made to balance the traffic within the network. The matrix is shown in [18].

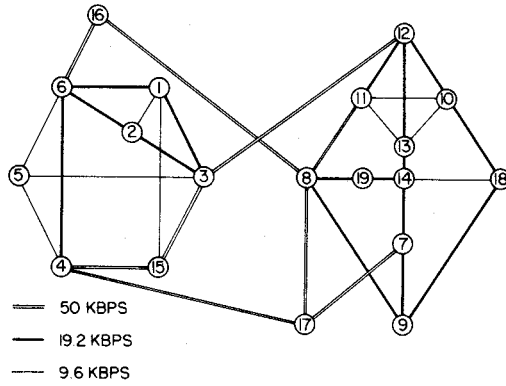


Fig. 3. The 19 node network.

transition probabilities  $P_{ij}^{(r)}$  used in our model can be determined from FRM as follows. For  $1 \leq r \leq 19$

$$P_{ij}^{(r)} = \begin{cases} 1, & \text{FRM}_{ir} = j \\ 0, & \text{otherwise} \end{cases} \quad 1 \leq i, j \leq 19$$

$$P_{i,20}^{(r)} = \begin{cases} 1, & i = r \\ 0, & \text{otherwise} \end{cases} \quad 1 \leq i \leq 19.$$

A uniform network input traffic pattern is assumed. ( $\sigma_i^{(r)} = 2.139$  packets/s for all distinct pairs of  $i$  and  $r$ . With infinite storage capacity, this network saturates as  $\alpha \rightarrow 1$ .)

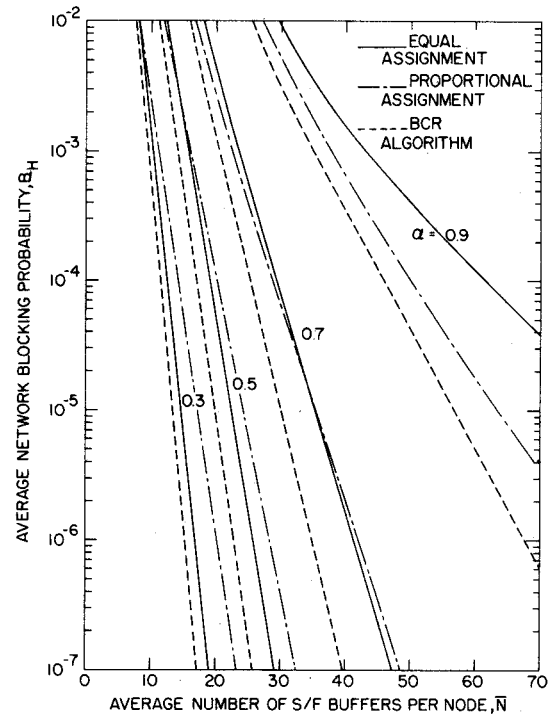
We also assume that the average processing time ( $1/\mu_i$ ) of a packet is 0.001 s for all central processors. The average packet length is 560 bits so that  $\mu_{ij} = C_{ij}/560$  packets/s where  $C_{ij}$  represents channel speed in bits/s. Corresponding to the channel speeds of 50, 19.2, and 9.6 kbits/s, the average ack delays are assumed to be 0.025, 0.065 and 0.130 s, respectively, and the average time-out intervals are assumed to be 0.125, 0.325, and 0.650 s, respectively. Local channels between S/F nodes and their sinks are assumed to be 100 kbits/s. The corresponding average ack and time-out delays are assumed to be zero. The packet error probability is assumed to be 0.001 for all channels. All channel propagation delays are assumed to be zero.

### Discussion of Results

Results illustrating the performance of the 19 node network are given in Figs. 4-8. In Fig. 4, a sample output of the APL program is shown for the network input level  $\alpha = 0.85$  and a fixed number ( $=30$ ) of S/F buffers at each of the 19 nodes. Note that the nodal blocking probabilities before and after the fourth iteration are equal (to three significant digits). Note also that a significant fraction of the average buffer utilization is due to packets (which have been successfully forwarded) waiting for the return of acks.

Three different buffer capacity assignment schemes have been considered. 1) Equal assignment (each node is given the same number of S/F buffers such as in a homogeneous network). 2) Proportional assignment (each node is given a number of S/F buffers proportional to its expected requirement computed under the assumption of infinite nodal

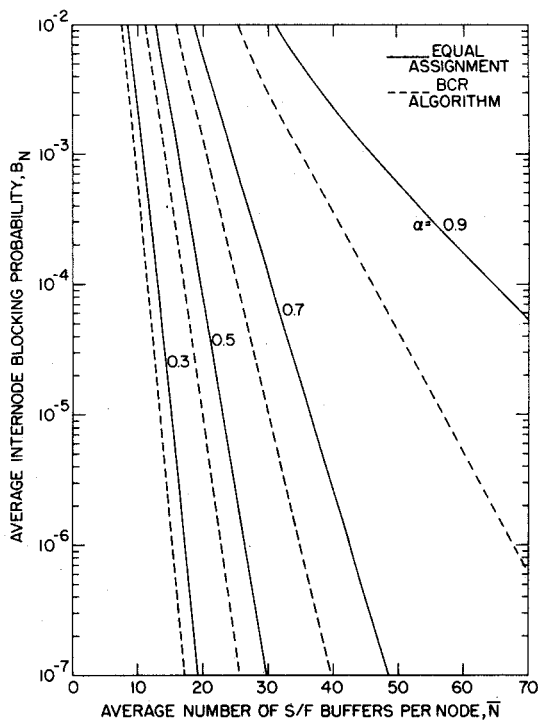
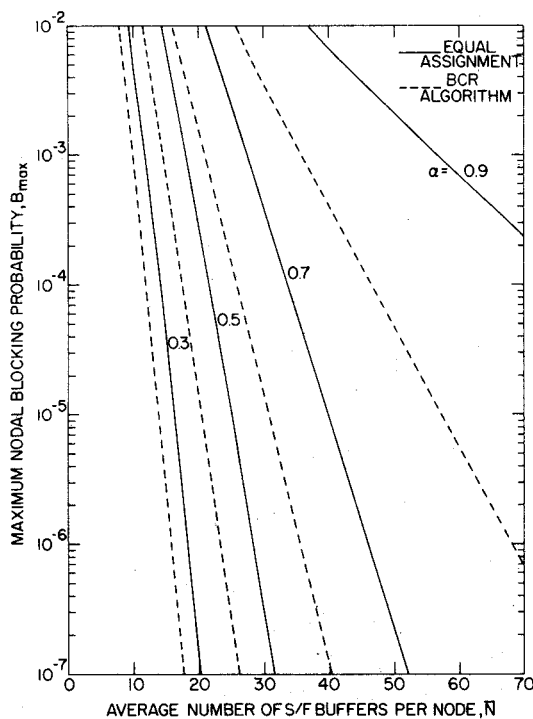
THE ERROR ESTIMATE IS 1 AFTER 1 ITERATION  
 THE ERROR ESTIMATE IS 0.03627 AFTER 2 ITERATIONS  
 THE ERROR ESTIMATE IS 0.0005538 AFTER 3 ITERATIONS  
 THE ERROR ESTIMATE IS 8.156E-6 AFTER 4 ITERATIONS  
 THE TOTAL NETWORK THROUGHPUT RATE = 621.7 PACKETS/SEC.  
 OR 3.481E5 BITS/SEC.  
 AVERAGE NETWORK DELAY = 0.1929 SECOND  
 AVERAGE NETWORK BUFFER UTILIZATION = 0.355  
 (0.2069 FOR QUEUEING, 0.003513 FOR TIME-OUT, 0.1446 FOR ACK)  
 AVERAGE NETWORK BLOCKING PROBABILITY = 0.002983  
 AVERAGE INTERNODE BLOCKING PROBABILITY = 0.004137  
 BLOCKING PROBABILITIES AT NODES 1-19:  
 2.345E-5 0.0001627 0.01159 0.006968 0.001308 0.00148 3.074E-5  
 0.009693 0.0004889 0.002065 0.001024 0.008298 0.009236 0.002985  
 8.249E-12 5.797E-8 8.884E-5 0.0001077 0.0008385  
 ARRIVAL RATES TO THE NODES:  
 80.06 74.62 167.5 150.3 67.41 140.3 107.4 196.6 92.85 94.82 111.1  
 163.3 99.17 107.7 87.34 112.8 154.7 81.89 76.49  
 THROUGHPUT RATES AT THE NODES:  
 79.98 74.53 165.4 149.1 67.26 140 107.2 194.5 92.71 94.52 110.9  
 161.8 98.16 107.2 87.25 112.7 154.5 81.8 76.35  
 AVERAGE DELAYS AT THE NODES:  
 0.04953 0.05586 0.06227 0.05927 0.1038 0.04975 0.03863 0.04862  
 0.06003 0.0792 0.05935 0.0604 0.0994 0.07501 0.01884 0.01917  
 0.02628 0.05709 0.07066  
 BUFFER UTILIZATIONS AT THE NODES:  
 0.2503 0.2373 0.552 0.4691 0.3825 0.4247 0.2561 0.5181 0.3156  
 0.427 0.3853 0.528 0.4948 0.4535 0.1194 0.1388 0.237 0.2818  
 0.2744  
 NODAL BLOCKING PROBABILITIES AFTER FINAL ITERATION:  
 2.345E-5 0.0001627 0.01159 0.006968 0.001308 0.00148 3.074E-5  
 0.009693 0.0004889 0.002065 0.001024 0.008298 0.009236 0.002985  
 8.25E-12 5.798E-8 8.884E-5 0.0001077 0.0008385

Fig. 4. Network performance at  $\alpha = 0.85$  and  $N_i = 30$  ( $1 \leq i \leq 19$ ).Fig. 5.  $B_H$  versus  $\bar{N}$ .

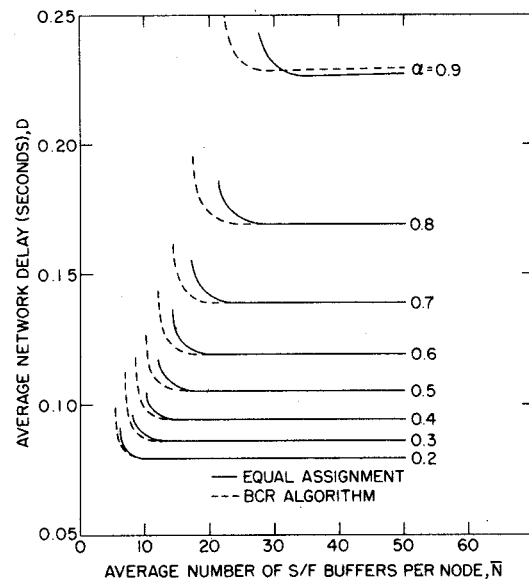
storage). 3) BCR. In Fig. 5,  $B_H$ , the average network blocking probability for host/terminal inputs, is shown versus  $\bar{N}$ , the average number of S/F buffers per node. In Fig. 6,  $B_N$ , the average internode blocking probability for node to node transmissions, versus  $\bar{N}$  is shown. In Fig. 7,

$$B_{\max} = \max_{1 \leq i \leq M} B_i$$

versus  $\bar{N}$  is shown. Note that for the same  $\bar{N}$  and  $\alpha$ ,  $B_N$  is slightly bigger than  $B_H$  in all cases. In general, the above blocking probabilities decrease as  $\bar{N}$  increases or  $\alpha$  decreases. For the same  $\alpha$  and  $\bar{N}$ , a network using BCR has significantly smaller nodal

Fig. 6.  $B_N$  versus  $\bar{N}$ .Fig. 7.  $B_{\max}$  versus  $\bar{N}$ .

blocking probabilities than one using equal assignment or proportional assignment. This is true since BCR gives rise to a relatively balanced set of  $\{B_i\}$ , which is evident from Figs. 5-7 in which  $B_H \cong B_N \cong B_{\max}$  for BCR. In Fig. 5, note that between equal assignment and proportional assignment, the former is better for small  $\alpha$  while the latter is better for large  $\alpha$ .

Fig. 8. Average network delay versus  $\bar{N}$ .

The average network delay  $D$  is shown versus  $\bar{N}$  for different  $\alpha$  in Fig. 8. (Delay values for proportional assignment are not shown. They lie between delay values for the other two schemes in all cases.) As  $\bar{N}$  increases,  $D$  decreases to a minimum point and then increases slightly before leveling off. Recall that  $D$  represents the average delay incurred by a packet after it has been admitted into the network. The average end-to-end delay incurred by a packet must include, in addition to  $D$ , the average admission delay due to nodal blocking.

## CONCLUSIONS

In this paper, an analytic model has been developed for a packet switching network in which each node has a finite pool of S/F buffers. Individual S/F nodes are modeled as queueing networks with different classes of packets. Both transmission control mechanisms of positive acknowledgment and time-out are included in this formulation. The single node queueing network results are interfaced by applying a continuity of flow constraint; an efficient computational procedure is given for iterative solution of the resulting set of nonlinear equations. A heuristic algorithm (BCR) is proposed for determining a balanced set of buffer requirements to achieve some small probability of nodal blocking. Finally, trading relations among network throughput, average delay, nodal blocking probabilities and S/F buffer requirements have been illustrated using a 19 node network example.

The model and analytic results developed in this paper may be used to supplement other network design and optimization techniques based upon the simpler models which assume infinite nodal storage [1]-[3]. Note, however, that we have been concerned mainly with S/F buffer requirements in a packet switching network. In an actual S/F node, additional storage capacity will be needed to satisfy other requirements, e.g., storage of the program code, storage of packets temporarily delayed by flow control mechanisms, provision of



buffers reserved for multipacket messages in-transit, etc. These other requirements must be taken into account in any global network design procedure. They constitute important areas for future research in packet switching networks.

#### APPENDIX A

The following recursive algorithm for evaluating the expression

$$g(n) = \sum_{j=0}^{M+1} \prod_{j=0}^{M+1} (a_j)^{q_j} \quad \sum_{j=0}^{M+1} q_j = n$$

has been proposed independently by Buzen [13], and Reiser and Kobayashi [14].

1) Define

$$\tilde{g}(m, k) \triangleq \sum_{j=0}^k \prod_{j=0}^k (a_j)^{q_j} \quad \sum_{j=0}^k q_j = m$$

$$\tilde{g}(m, 0) = a_0^m, \quad m = 0, 1, \dots, n$$

$$\tilde{g}(0, k) = 1, \quad k = 0, 1, \dots, M+1.$$

2) The iteration step is

$$\tilde{g}(m, k) = \tilde{g}(m, k-1) + a_k \tilde{g}(m-1, k) \quad m > 0 \quad k > 0.$$

3) Finally,

$$g(n) = \tilde{g}(n, M+1).$$

#### APPENDIX B

To evaluate  $\nabla f$ , we focus our attention upon node  $i$  and obtain solutions for  $\partial f_i / \partial B_k$ ,  $1 \leq k \leq M$ . As before, the subscript  $i$  denoting node  $i$  is suppressed wherever no ambiguity arises. From (9)–(11),

$$f_i = \frac{p(N_i, x)}{G(N_i)} = \frac{p(N_i, x)}{1 + \sum_{N=1}^{N_i} p(N, x)} \quad (B1)$$

where

$$x \triangleq b_i + c_i. \quad (B2)$$

Taking the partial derivative of (B1), with respect to  $B_k$ , we have

$$\frac{\partial f_i}{\partial B_k} = \frac{\partial p(N_i, x)}{\partial B_k} \frac{1}{G(N_i)} - \frac{p(N_i, x)}{[G(N_i)]^2} \left[ \sum_{N=1}^{N_i} \frac{\partial p(N, x)}{\partial B_k} \right].$$

Applying (9)–(11) and (23), the above equation may be re-

written as

$$\frac{\partial f_i}{\partial B_k} = P(0) \left[ \frac{\partial p(N_i, x)}{\partial B_k} - f_i \sum_{N=1}^{N_i} \frac{\partial p(N, x)}{\partial B_k} \right]. \quad (B3)$$

Taking the partial derivative of (7), with respect to  $B_k$ , we have

$$\begin{aligned} \frac{\partial p(N, x)}{\partial B_k} &= \sum_{n=1}^N \frac{\partial g(n)}{\partial B_k} \frac{x^{N-n}}{(N-n)!} \\ &+ \left[ \sum_{n=0}^{N-1} g(n) \frac{x^{N-n-1}}{(N-n-1)!} \right] \frac{\partial x}{\partial B_k}, \quad N \geq 1. \end{aligned} \quad (B4)$$

From (B2) and definitions of the traffic intensities at node  $i$ ,

$$\begin{aligned} x &= \sum_{j=1}^{M+1} (b_{ij} + c_{ij}) \\ &= \sum_{j=1}^{M+1} \sum_{r=1}^R \left[ \frac{\gamma_i^{(r)} P_{ij}^{(r)} B_{ij} t_{ij}}{(1 - B_{ij})} + \gamma_i^{(r)} P_{ij}^{(r)} v_{ij} \right] (1 - E_i) \end{aligned}$$

where from (3)

$$B_{ij} = 1 - (1 - e_{ij})(1 - B_j)$$

and thus for  $k \neq i$

$$\frac{\partial x}{\partial B_k} = \frac{\partial b_{ik}}{\partial B_k} = \frac{b_{ik}}{(1 - B_k)[1 - (1 - e_{ik})(1 - B_k)]}, \quad k \neq i. \quad (B5)$$

Applying (21) we have for  $k = i$

$$\frac{\partial x}{\partial B_k} = \frac{x}{1 - B_i}, \quad k = i. \quad (B6)$$

To evaluate the partial derivative of  $g(n)$ , with respect to  $B_k$ , we consider the following two cases.

Case 1— $k \neq i$ : Taking the partial derivative of (6), with respect to  $B_k$ , we have

$$\frac{\partial g(n)}{\partial B_k} = \left[ \sum_{j=0}^{M+1} \frac{q_{ik}}{a_{ik}} \prod_{j=0}^{M+1} (a_{ij})^{q_{ij}} \frac{\partial a_{ik}}{\partial B_k} \right], \quad n \geq 1$$

where

$$a_{ik} \triangleq \sum_{r=1}^R \frac{\gamma_i^{(r)} P_{ik}^{(r)} (1 - E_i)}{(1 - e_{ik})(1 - B_k) \mu_{ik}}$$

and

$$\frac{\partial a_{ik}}{\partial B_k} = \frac{a_{ik}}{1 - B_k}.$$

Thus,

$$\begin{aligned}
\frac{\partial g(n)}{\partial B_k} &= \frac{1}{1 - B_k} \sum_{\substack{M+1 \\ \sum_{ij=n}}} q_{ik} \prod_{j=0}^{M+1} (a_{ij})^{q_{ij}} \\
&= \frac{1}{1 - B_k} \sum_{l=1}^n \left[ \sum_{\substack{M+1 \\ \sum_{ij=n-l} \\ j \neq k}} \prod_{j=0}^{M+1} (a_{ij})^{q_{ij}} \right] l(a_{ik})^l \\
&= \frac{1}{1 - B_k} \sum_{l=1}^n [g(n) - a_{ik}g(n-1)] l(a_{ik})^l. \quad (B7)
\end{aligned}$$

Case  $2-k = i$ : We assumed that  $P_{ii}^{(r)} = 0$  for all  $i$  and  $r$ . Then, from definitions of the traffic intensities  $a_{ij}$ ,  $0 \leq j \leq M+1$ , at node  $i$  and (6), we have

$$g(n) = \frac{\tilde{g}(n)}{(1 - B_i)^n}$$

where

$$\tilde{g}(n) = g(n)|_{B_i=0}.$$

Thus, we have

$$\frac{\partial g(n)}{\partial B_k} = \frac{n}{(1 - B_i)} g(n), \quad k = i. \quad (B8)$$

Finally,  $\partial f_i / \partial B_k$  can be evaluated by applying (B3)–(B8).

### ACKNOWLEDGMENT

The author would like to thank H. Kobayashi, P. J. Schweitzer, D. T. Tang, and P. S. Wolfe of the Thomas J. Watson Research Center for helpful discussions on various aspects of this work. He is also grateful to an anonymous referee for a number of useful comments and suggestions given to the original manuscript.

### REFERENCES

- [1] L. Kleinrock, "Analytic and Simulation Methods in Computer Network Design," *Spring Joint Computer Conference, AFIPS Conference Proceedings*, 1970, Vol. 36, pp. 569–579.
- [2] M. Gerla and L. Kleinrock, "The Design of Packet Switching Networks for Computer Communications—a Survey," to be published.
- [3] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*. McGraw Hill, New York, 1964, out of print; reprinted by Dover Publications, New York, 1972.
- [4] J. R. Jackson, "Jobshop-Like Queueing Systems," *Management Science*, Vol. 10, October 1963, pp. 131–142.
- [5] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," *Journal of the ACM*, Vol. 22, No. 2, April 1975, pp. 248–260.

- [6] H. Frank, R. E. Kahn and L. Kleinrock, "Computer Communication Network Design—Experience with Theory and Practice," *Spring Joint Computer Conference, AFIPS Conference Proceedings*, 1972, Vol. 40, pp. 255–270.
- [7] W. W. Chu, "A Study of Asynchronous Time Division Multiplexing for Time-Sharing Computer Systems," *Fall Joint Computer Conference, AFIPS Conference Proceedings*, 1969, Vol. 35, pp. 669–678.
- [8] J. F. Ziegler, "Nodal Blocking in Large Networks," Ph.D. Dissertation, School of Engineering and Applied Science, University of California, Los Angeles, 1971.
- [9] F. Closs, "Packet Arrival and Buffer Statistics in a Packet Switching Node," IBM Research Laboratory, Zurich, Switzerland, Research Report RZ594, September 1973.
- [10] P. J. Schweitzer and S. S. Lam "Buffer Overflow in a Store-and-Forward Network Node," IBM T. J. Watson Research Center, Yorktown Heights, New York, Research Report RC5759, Dec. 1975.
- [11] F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The Interface Message Processor for the ARPA Computer Network," *Spring Joint Comput. Conf., AFIPS Conference Proceedings*, 1970, Vol. 36, pp. 551–567.
- [12] J. M. McQuillan, W. R. Crowther, B. P. Cosell, D. C. Walden, and F. E. Heart, "Improvements in the Design and Performance of the ARPA Network," *Fall Joint Computer Conference, AFIPS Conference Proceedings*, 1972, Vol. 39, pp. 741–754.
- [13] J. P. Buzen, "Computational Algorithms for Closed Queueing Networks with Exponential Servers," *Communications of the ACM*, Vol. 16, September 1973, pp. 527–531.
- [14] M. Reiser and H. Kobayashi, "Numerical Solution of Semi-closed Exponential Server Queueing Networks," *Proceedings of the 7th Asilomar Conference on Circuits, Systems and Computers*, November 1973, pp. 308–312.
- [15] J. D. C. Little, "A Proof of the Queueing Formula:  $L = \lambda W$ ," *Operations Research*, Vol. 9, 1961, pp. 383–387.
- [16] F. B. Hildebrand, *Introduction to Numerical Analysis*, McGraw-Hill, New York, 1956.
- [17] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Non-linear Equations in Several Variables*, Academic Press, New York, 1970.
- [18] S. S. Lam, "Store-and-Forward Buffer Requirements in a Packet Switching Network," IBM T. J. Watson Research Center, Yorktown Heights, New York, Research Report RC5432, May 1975.



Simon S. Lam (S'69–M'74) was born in Macao, on July 31, 1947. He received the B.S.E.E. degree in electrical engineering from Washington State University, Pullman, WA, in 1969, and the M.S. and Ph.D. degrees in engineering from the University of California, Los Angeles, in 1970 and 1974, respectively.

At the University of California, Los Angeles, he held a Phi Kappa Phi Fellowship from 1969 to 1970, and a Chancellor's Teaching Fellowship from 1969 to 1973. He also participated in the ARPA Network project at UCLA as a postgraduate research engineer from 1972 to 1974 and did research on satellite packet communication. Since June 1974 he has been a research staff member with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. His current research interests include computer-communication networks, satellite data networks, and queueing theory.

Dr. Lam is a member of Tau Beta Pi, Sigma Tau, Phi Kappa Phi, Pi Mu Epsilon, and the Association for Computing Machinery.