Queuing Network Models of Packet Switching Networks Part 1: Open Networks *

J.W. Wong
Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada

and

S.S. Lam
Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712, U.S.A.

Received 20 February 1981; revised version received 28 December 1981

An overview of the application of product-form queuing network models to the performance analysis of store-and-forward packet-switching networks is presented. Multiple routing chains are used to model the different routing behaviors of data packets. Queuing networks with open chains are considered in this paper. Kleinrock's formula for the mean end-to-end delay of packets is first derived. The application of this delay formula to optimal channel capacity assignment and optimal routing is discussed. Analytic results for the mean and distribution of the end-to-end delay of each chain are presented. The issue of fairness among chains is also addressed.

The application of queuing networks with closed chains and population size constraints to the performance analysis of store-and-forward packet-switching networks is illustrated in a companion paper [1].

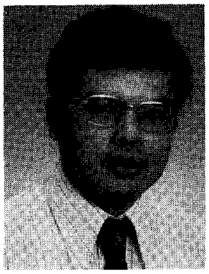
Keywords: Queuing Network Models, Packet Switching Networks, End-to-End Delay, Delay Distribution, Capacity Assignment, Routing, Fairness.

* The work of J.W. Wong was supported by the Natural Sciences and Engineering Research Council of Canada. The work of S.S. Lam was supported by National Science Foundation Grant No. ECS78-01803.

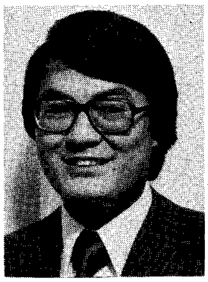
North-Holland Publishing Company
Performance Evaluation 2 (1982) 9–21

1. Introduction

A store-and-forward packet-switching network consists of a set of switching nodes interconnected by communication channels. Host computers and terminals constitute sources and sinks of data messages to be transported by the network (see Fig. 1). The basic unit of data transfer within the network



J.W. Wong received the B.S. in engineering, the M.S. and Ph.D. degrees in computer science from the University of California at Los Angeles in 1970, 1971, and 1975, respectively. He is currently an Associate Professor of Computer Science at the University of Waterloo, Waterloo, Ontario, Canada.



Simon S. Lam received the B.S.E.E. degree (with Distinction) from Washington State University, Pullman, Washington, in 1969, and the M.S. and Ph.D. degrees in engineering from the University of California at Los Angeles, in 1970 and 1974, respectively.

From 1972 to 1974, he was with the ARPA Network project at UCLA and did research on satellite packet communication. From 1974 to 1977, he was a research staff member with the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, where he worked on performance analysis problems of packet switching networks, SNA and satellite networks. Since September 1977, he has been with the University of Texas at Austin where he is now Associate Professor of Computer Sciences. His current research interests include modeling and analysis of computer systems, networks and communication protocols.

At the University of California at Los Angeles, he held a Phi Kappa Phi Fellowship from 1969 to 1970, and a Chancellor's Teaching Fellowship from 1969 to 1973. In 1975, he received the IEEE Leonard G. Abraham Award for the best paper in the field of Communications Systems. Simon is a member of Tau Beta Pi, Sigma Tau, Phi Kappa Phi, Pi Mu Epsilon and the Association for Computing Machinery, and a senior member of IEEE.

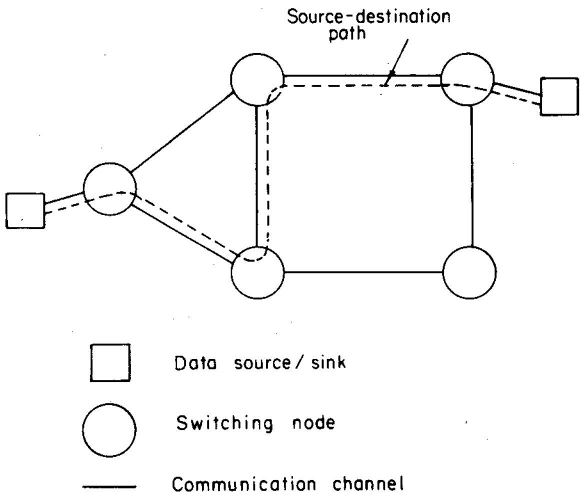


Fig. 1. Store-and-forward packet switching network.

is a packet. [1] Each packet traverses from its source node to its destination node through a series of nodes and communication channels along its path (route). Queues are formed for the communication channels inside the switching nodes. The progress of packets in the network is governed by certain communication protocols. The objective of this paper is to review recent efforts on the application of product-form open queuing networks [2–4] to model store-and-forward packet communication networks. The application of queuing networks with closed chains and population size constraints [2–5] is considered in a companion paper [1].

A store-and-forward network can be viewed as a collection of resources shared by data sources and sinks. There are three types of physical resources in the network: communication channels, packet buffers, and nodal processors. In modeling such a network, the nodal processors are often neglected because processor delays incurred by packets are typically substantially less than communication channel delays.

To transport a packet from one node to another in a store-and-forward network, the resources needed along the source–destination path are communication channels and one buffer in each node along the path. It is obvious that the set of communication channels and/or the set of nodal buffers can be preallocated. Preallocation is a 'safe' operational strategy. However, it is extremely wasteful because data sources are typically very bursty [6].

A store-and-forward protocol is a means for dynamically allocating network resources and thus sharing them statistically. In a store-and-forward network, a packet can progress from one node to the next along its route with the allocation of just a communication channel and two buffers (one at each side of the channel). If the packet is successfully received and accepted in the next node, a positive acknowledgement message will be returned to the previous node, either separately as a short packet or piggybacked in the header of a data packet traveling in the reverse direction. The packet buffer in the previous node can then be freed. If, however, no acknowledgement has been received at the end of a timeout period, the packet will be queued for retransmission.

Currently, there are two basic types of packet communication services: datagram and virtual channel [7]. We shall consider their differences from the modeling viewpoint only. In a datagram network, each data packet (datagram) traverses the network as an independent entity. In a virtual channel network, data packets belong to 'virtual channels' connecting data sources and sinks. The admittance of packets into a virtual channel is controlled. Also, packets in the same virtual channel are usually characterized by the same routing behavior.

Given a set of external traffice demands, the efficient utilization of a network's channel and buffer resources depends on the network's routing algorithm as well as its flow and congestion control techniques. Measurements of network performance typically include its throughput (in packets delivered per second) and some measure of the network transit delay. These performance measures may need to be characterized for all packets transported by the network or for individual classes of packets (e.g., packets between specific source–sink pairs).

Product-form queuing network models have been successfully applied to the performance analysis of store-and-forward network with some or all of the above features. To do so, several simplifying

[1] When a data message to be transported is longer than the size of a packet, it is segmented into several packets which need to be reassembled later to form the original message. The segmentation and reassembly functions may be either performed by the network nodes or by the data sources and sinks.

assumptions are necessary; they will be introduced in Section 2. Queuing network models also have a number of limitations. One such limitation is that adaptive routing [6] cannot be modeled. Analysis results are available for situations where a set of paths is provided between each source–destination node pair, and these paths may be either chosen deterministically or randomly, but not adaptively for packets.

The accuracy of queuing network models is affected by the presence of various communication protocols, which may impede the progress of packets through the network but which cannot be easily modeled (examples are segmentation and reassembly of messages, some of the data link control functions, etc.). Also, various network measurement and control traffic are often not accounted for in the models to be described below. Therefore queuing network models results should be viewed in most cases as a somewhat optimistic prediction of network performance.

2. Assumptions and definitions

The key assumption necessary for the application of queuing network models to analyze a store-and-forward network was originally introduced by Kleinrock [6,8].

The Independence Assumption. Each time a packet joins a queue in the network, its length is determined afresh from the probability density function

$$b(x) = \mu \exp(-\mu x), \quad x \geqslant 0,$$

where $1/\mu$ is the mean packet length (in number of bits).

The above assumption removes the statistical dependence of the transmission times of a packet at the various channels of its route. Without this assumption, the analysis of store-and-forward networks is not mathematically tractable.

In actual networks, packets usually have a maximum length. Also, measurement results indicate that packet lengths are not really exponentially distributed [6,9]. Therefore, analytic results provided by queuing network models are merely approximations. However, these approximate results are generally deemed to be adequate and valuable for the design and performance characterization of store-and-forward packet-switching networks [6].

We next define the class of queuing networks suitable for modeling store-and-forward networks. (This class of network models is only a subset of network models that have a product-form solution [2–5].) The notation to be used throughout this paper is also introduced.

Servers in the network model are indexed by $i = 1, 2, \ldots, M$. We shall only consider first-come first-served (FCFS) servers to model communication channels, and infinite-server (IS) servers to model random delays. Customers (i.e., packets) belong the different 'routing chains'; these chains are indexed by $k = 1, 2, \ldots, K$. Specifically, the routing behavior of chain k is modeled by a first-order Markov chain with transition probabilities

$$p_{ij}^{(k)} = \text{Prob}[\text{to server } j \,|\, \text{currently at server } i] \qquad i, j = 1, 2, \ldots, M. \quad (2.1)$$

We note that a first-order Markov chain is adequate for modeling both the case of a single route and the case of multiple routes between a given source and destination. Retransmission and rerouting due to random transmission errors can also be modeled by an appropriate definition of the transition probabilities [10].

Since the routing behavior of packets traveling between different source–destination node pairs are different, a queuing network model must specify at least one routing chain for each source–destination node pair. It is sometimes desirable to specify multiple chains between each source–destination node pair to correspond to virtual channels connecting several data sources in the same source node to several data sinks in the same destination node. For a datagram network, it is sufficient to define just one chain for each source–destination node pair.

It is assumed that chain k packets arrive to the first server of the chain according to a Poisson process with rate γ_k packets per second, where $k = 1, 2, \ldots, K$. Define

$$\gamma = \gamma_1 + \gamma_2 + \cdots + \gamma_K.$$

γ is the total external arrival rate to the network. Given γ_k and $p_{ij}^{(k)}$, the mean rate λ_{ik} of chain k packet arrivals to server i in the queuing network model is determined from:

$$\lambda_{ik} = \gamma_k \delta_{ik} + \sum_{j=1}^{M} \lambda_{jk} p_{ji}^{(k)}, \quad (2.2)$$

where

$$\delta_{ik} = \begin{cases} 1 & \text{if } i \text{ is the first server of chain } k, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The arrival rate of packets from all chains to server i is

$$\lambda_i = \sum_{k=1}^{K} \lambda_{ik}. \tag{2.4}$$

It is assumed that if server i is FCFS, it works at a constant rate of C_i bits per second. If server i is IS, it works at a rate of C_{ik} bits per second for chain k packets. The traffic intensity of chain k packets at server i is defined to be

$$\rho_{ik} = \begin{cases} \dfrac{\lambda_{ik}}{\mu C_i}, & \text{server } i \text{ is FCFS}, \\ \dfrac{\lambda_{ik}}{\mu C_{ik}}, & \text{server } i \text{ is IS}, \end{cases} \tag{2.5}$$

and the overall traffic intensity of server i is

$$\rho_i = \sum_{k=1}^{K} \rho_{ik}. \tag{2.6}$$

Let the state of the queuing network be denoted by

$$S = (\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_M),$$

where

$$\boldsymbol{n}_i = (n_{i1}, n_{i2}, \ldots, n_{iK}),$$

where n_{ik} is the total number of chain k packets at server i. Define

$$n_i = n_{i1} + n_{i2} + \cdots + n_{iK}$$

and

$$\boldsymbol{n} = (n_1, n_2, \ldots, n_M).$$

A chain is said to be *open* if it allows both external arrivals and departures to occur freely. As a result, the number of packets in a chain can range from 0 to ∞. For a network with open chains, the equilibrium probability of the network state S has the following-product form solution [3]:

$$P(S) = \prod_{i=1}^{M} \frac{p_i(\boldsymbol{n}_i)}{G_i}, \tag{2.7}$$

where

$$p_i(\boldsymbol{n}_i) = \begin{cases} n_i! \displaystyle\prod_{k=1}^{K} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!}, & \text{server } i \text{ is FCFS}, \\ \displaystyle\prod_{k=1}^{K} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!}, & \text{server } i \text{ is IS}, \end{cases} \tag{2.8}$$

and

$$G_i = \begin{cases} 1/(1-\rho_i), & \text{server } i \text{ if FCFS}, \\ \exp(\rho_i), & \text{server } i \text{ is IS}. \end{cases} \tag{2.9}$$

The equilibrium probability of $\boldsymbol{n}$ also has a product form [3]:

$$P(\boldsymbol{n}) = \prod_{i=1}^{M} \frac{p_i(n_i)}{G_i}, \tag{2.10}$$

where

$$p_i(n_i) = \begin{cases} \rho_i^{n_i}, & \text{server } i \text{ is FCFS}, \\ \dfrac{\rho_i^{n_i}}{n_i!}, & \text{server } i \text{ is IS}. \end{cases} \tag{2.11}$$

A routing chain is said to be *closed* if the number of packets in the chain is fixed. Queuing networks with closed chains as well as other forms of chain population size constraints are useful for modeling flow and congestion control in store-and-forward networks. These models are discussed in the companion paper [1].

Given a set of traffic demands modeled by the rates $\{\gamma_k,\ k = 1, 2, \ldots, K\}$, the basic performance measures of interest are the network throughput and mean end-to-end (or source-to-destination) delay. Define

γ^* = network throughput in packets per second,
T = mean end-to-end delay over all packets transported by the network.

These two measures may be adequate by themselves. In some cases, it may be necessary to characterize the performance of individual routing chains. We thus define

γ_k^* = throughput of chain k in packets per second,
T_k = mean end-to-end delay of chain k packets transported by the network.

If the network switching nodes have adequate buffers and channel capacities so that packets are never rejected (a situation modeled by queuing networks with open chains), the chain throughput is the same as the chain arrival rate, i.e.,

$$\gamma_k^* = \gamma_k, \quad k = 1, 2, \ldots, K. \tag{2.12}$$

Otherwise, some external arrivals are rejected due to buffer, flow or congestion control constraints and the throughput is smaller than the corresponding arrival rate, or

$$\gamma_k^* < \gamma_k, \quad k = 1, 2, \ldots, K. \tag{2.13}$$

The difference between γ_k and γ_k^* is the rate at which chain k arrivals are rejected. Since rejected packets are retransmitted later, the ratio γ_k/γ_k^* can be interpreted as the mean number of tries needed for a packet to gain admittance to the network. (See [1].)

The balance of this paper is organized as follows. An analytic formula for T, the mean end-to-end delay over all packets, is derived in Section 3. The application of this formula to channel capacity assignment and optimal routing is next discussed. In Section 4, the mean and distribution of the end-to-end delays of individual chains are considered. Finally, the issue of fairness among chains is discussed in Section 5.

3. Mean end-to-end delay over all chains

We consider in this section the efficient utilization of communication channels in a packet network via channel capacity assignment or routing assignment.

Queuing network models with open chains are employed. The number of buffers at each node is assumed to be very large (infinite). The effect of packet transmission errors is assumed to be negligible.

3.1. A formula for the mean end-to-end delay

The mean end-to-end delay T for an arbitrary packet transported by the network was first derived by Kleinrock [6,8]. It can be obtained as follows. By Little's formula [11], the mean number of packets in transit within the network is equal to γT. Let $E[n_i]$ be the mean number of packets at channel i. We have

$$\gamma T = \sum_{i=1}^{M} E[n_i]. \tag{3.1}$$

Since the communication channel delays typically dominate most other delays, we shall assume that the M servers are all communication channels modeled by FCFS queues. The marginal queue length distribution from (2.11) gives rise to the following mean queue lengths:

$$E[n_i] = \frac{\rho_i}{1-\rho_i}, \quad i = 1, 2, \ldots, M. \tag{3.2}$$

Thus

$$\gamma T = \sum_{i=1}^{M} \frac{\rho_i}{1-\rho_i}. \tag{3.3}$$

Since $\rho_i = \lambda_i/(\mu C_i)$ (see (2.4) to (2.6)), we have

$$T = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{\lambda_i}{\mu C_i - \lambda_i}, \tag{3.4}$$

which is sometimes written as

$$T = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{f_i}{C_i - f_i}, \tag{3.5}$$

where $f_i = \lambda_i/\mu$ (in bits per second) is called the channel i *flow* [12].

Recall that under our present assumption, $\gamma^* = \gamma$. Using T, as given by (3.5), as our performance measure, we shall consider next the problems of channel capacity assignment and optimal routing.

In practice, it may be advisable to refine the model considered above by including delays due to channel propagation times, nodal processing times, and any control message traffic. The reader is referred to [6] for more details. However, (3.5) is the basic formula used in the capacity assignment and optimal routing problems.

3.2. Capacity assignment

Suppose we are given the traffic requirement $\{\gamma_k, k = 1, 2, \ldots, K\}$. The network topology is fixed and routing has been specified in the form of (2.1). A meaningful question to ask is as follows: given a fixed budget for communication channels, how do we select the set of channel capacities $\{C_i, i = 1, 2, \ldots, M\}$? This problem was addressed by Kleinrock [6,8] and to keep the problem simple, he made the following assumptions:

(a) channel capacities can be selected from a continuum of values;

(b) the channel cost is a linear function of the channel capacity so that the network cost is

$$\sum_{i=1}^{M} d_i C_i + \text{a fixed cost}. \tag{3.6}$$

Let D be the available budget for channels after the fixed cost has been accounted for. One can then pose the following optimization problem:

$$\min_{\{C_i\}} T=\frac{1}{\gamma}\sum_{i=1}^{M}\frac{f_i}{C_i-f_i},$$

$$\text{subject to} \quad \sum_{i=1}^{M} d_iC_i \leqslant D. \tag{3.7}$$

Note that T is a convex function and the set of feasible channel capacities is a convex set. Hence, a unique optimal solution exists. The above constrained optimization problem can be converted to an unconstrained optimization problem by the Lagrange multiplier method which yields the following solution for optimal capacity assignment [6,8]:

$$C_i^* = f_i + \left(D_e\sqrt{f_id_i}\right)\Big/\left(d_i\sum_{j=1}^{M}\sqrt{f_jd_j}\right), \tag{3.8}$$

where

$$D_e = D - \sum_{j=1}^{M} f_jd_j > 0. \tag{3.9}$$

If $D_e \leqslant 0$, a feasible capacity assignment does not exist to achieve a finite T. The minimum mean delay corresponding to the above capacity assignment is:

$$T^* = \frac{1}{\gamma D_e}\left[\sum_{j=1}^{M}\sqrt{f_jd_j}\right]^2. \tag{3.10}$$

The dual of the optimization problem in (3.7) can also be formulated to minimize the network cost subject to a mean delay constraint as follows:

$$\min_{\{C_i\}} \sum_{i=1}^{M} d_iC_i,$$

$$\text{subject to} \quad \frac{1}{\gamma}\sum_{i=1}^{M}\frac{f_i}{C_i-f_i} \leqslant T_{\max}. \tag{3.11}$$

Again, applying the Lagrange multiplier method, the optimal capacity assignment is

$$C_i^* = f_i + \frac{\sum_{j=1}^{M}\sqrt{f_jd_j}}{\gamma T_{\max}}\sqrt{\frac{f_i}{d_i}}. \tag{3.12}$$

The minimum network cost for channels is then

$$D^* = \sum_{j=1}^{M} f_jd_j + \frac{1}{\gamma T_{\max}}\left[\sum_{j=1}^{M}\sqrt{f_jd_j}\right]^2. \tag{3.13}$$

The optimal capacity assignment given by (3.8) or (3.12) provides a network designer with some initial guidance for selecting channel capacities. In reality, channel capacities are limited to a set of discrete capacity values that are available from common-carriers. Also, as a result of economy of scale, the cost-function $d_i(C_i)$ of a channel with capacity C_i should be a concave function in C_i instead of a linear function assumed above. To incorporate the above considerations into the capacity assignment problem, one must then resort to numerical solution techniques [12]. Finally, we also note that most communication links available from the common-carriers are full-duplex with the same capacity for each of the channels in opposite directions. The usual assumption that enables us to apply the above optimal capacity assignment result is to consider a symmetric network traffic pattern. If, however, the network traffic pattern is highly asymmetric, then one must again resort to a numerical solution technique to account for this additional constraint.

3.3. Optimal routing

In the capacity assignment problem above, the routing was assumed to be pre-specified. Suppose the channel capacities have already been selected. We now consider the problem of assigning routes to satisfy a set of traffic requirements $\{\gamma_k, k=1,2,\ldots,K\}$ so that some performance criterion is optimized. This is known as the optimal routing problem.

Operationally, optimal routing is difficult to achieve. Due to the geographical distribution of network nodes, fresh information on the global status of a network is generally not available. In the ARPANET, for example, each node exchanges status information with its neighbors periodically. Considerable time delays, however, are needed for such information to propagate throughout the network [13].

With most performance objectives, optimal routing can be formulated as a shortest path problem with an appropriate distance metric for the communication channels within a path. In the ARPANET, the distance metric is simply the (estimated) mean delay of a communication channel. Each packet, regardless of its origin, is routed to an outgoing channel along the path with the shortest (estimated) mean delay to the packet's destination node. Note that this particular routing strategy minimizes the (estimated) mean delay of each individual packet. It has been shown that such an individual optimization strategy does not necessarily lead to a globally optimized situation; specifically, the mean transit delay T for all packets transported by the network is not optimized. In order to optimize T, the following result was obtained [12,14].

Given the traffic requirements $\{\gamma_k\}$ and a specific routing assignment $\{p_{ij}^{(k)}\}$, the channel arrival rates $\{\lambda_i\}$ can be determined using (2.2) and (2.4). Recall that the flow in channel i is $f_i = \lambda_i/\mu$ in bits per second. Denote the set of channel flows by the flow vector

$$\boldsymbol{f} = (f_1, f_2, \ldots, f_M).$$

A flow vector $\boldsymbol{f}$ is said to be feasible if

$$0 \leqslant f_i < C_i \quad \text{for } i = 1, 2, \ldots, M$$

and if it is the result of a routing assignment which satisfies the traffic requirements.

Necessary and sufficient conditions for $\boldsymbol{f}$ (hence, indirectly for the routing assignment) to minimize T are obtained as follows. Let $T(\boldsymbol{f})$ be the mean network delay corresponding to the feasible flow vector $\boldsymbol{f}$. Let $\boldsymbol{v}$ be another flow vector. Given $\boldsymbol{f}$, a feasible flow vector $\boldsymbol{f}'$ near $\boldsymbol{f}$ can be represented as a convex combination of $\boldsymbol{f}$ and $\boldsymbol{v}$, i.e.,

$$\boldsymbol{f}' = (1-\varepsilon)\boldsymbol{f} + \varepsilon\boldsymbol{v} = \boldsymbol{f} + \varepsilon(\boldsymbol{v} - \boldsymbol{f}), \quad 0 \leqslant \varepsilon \leqslant 1. \tag{3.14}$$

Note that the flow vector $\boldsymbol{v}$ can be chosen without satisfying $v_i < C_i$ for all i. In this case, a sufficiently small ε should be used in order for $\boldsymbol{f}'$ to be feasible. Suppose $\varepsilon \ll 1$ so that the change in the flow vector (in the $\boldsymbol{v}$ direction) is small and is given by:

$$\Delta \boldsymbol{f} = \varepsilon(\boldsymbol{v} - \boldsymbol{f}). \tag{3.15}$$

The resulting change in the mean network delay is

$$\Delta T(\boldsymbol{f}) = \sum_{i=1}^{M} \frac{\partial T(\boldsymbol{f})}{\partial f_i}(v_i - f_i)\varepsilon = \varepsilon \sum_{i=1}^{M} L_i(v_i - f_i), \tag{3.16}$$

where

$$L_i = \frac{\partial T(\boldsymbol{f})}{\partial f_i}. \tag{3.17}$$

Eq. (3.17) above requires that the function $T(\boldsymbol{f})$ be differentiable. If, moreover, the function $T(\boldsymbol{f})$ is also convex, then we know that a unique minimum exists among the set of feasible flow vectors (which is a convex set [12,14]). Thus, a necessary and sufficient condition for a feasible flow vector $\boldsymbol{f}$ to be optimal is

$$\Delta T(\boldsymbol{f}) \geqslant 0 \quad \text{for any } \boldsymbol{v}, \tag{3.18}$$

or

$$\sum_{i=1}^{M} L_i(v_i - f_i) \geqslant 0 \quad \text{for any } \boldsymbol{v}, \tag{3.19}$$

or

$$\min_{\boldsymbol{v}} \sum_{i=1}^{M} L_i v_i \geqslant \sum_{i=1}^{M} L_i f_i. \tag{3.20}$$

If the condition in (3.20) is not satisfied, then $\boldsymbol{f}$ (and the corresponding routing assignment) is not optimal. Moreover, (3.16) indicates that if a 'small' amount of source–destination traffic is to be rerouted (or if some new traffic is to be added to the network) then that traffic should follow a shortest path from its source node to its destination node using $\{L_i\}$ as the distance metric to minimize its impact on the mean network delay. Recall that with the open queuing network model, from (3.5),

$$T(\boldsymbol{f}) = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{f_i}{C_i - f_i}. \tag{3.21}$$

Hence

$$L_i = \frac{\partial T(\boldsymbol{f})}{\partial f_i} = \frac{1}{\gamma} \frac{C_i}{(C_i - f_i)^2}. \tag{3.22}$$

Given the traffic requirements $\{\gamma_k\}$, the optimal flow vector $\boldsymbol{f}$ (and hence, route assignments) can be determined by a downhill search technique using any feasible flow vector as a starting point. The reader is referred to [12,14] for details.

Finally, we comment that in practice, instead of doing capacity assignment and optimal routing as separate problems, it is desirable to do both optimizations together. The resulting problem is much more difficult than each of the above, and one must resort to heuristic search techniques for optimal solutions. The reader should consult Gerla's thesis [12] for this and other network design problems.

4. End-to-end delay for each routing chain

Our discussion so far has been based on the mean end-to-end delay over all packets transported by the network. The key result, given by (3.4), provides a gross characterization of network delay. It is useful in the formulation of various optimization problems for network design. For various reasons, one might be interested in a more detailed characterization of network delay than the mean over all packets. For example, users sending packets from node A to node B will be interested in the end-to-end delay from A to B.

4.1. Mean end-to-end delay

We shall consider only networks which employ path-oriented routing. When multiple paths exist between a given source–destination node pair, the source node selects the complete path for each packet to follow in order to reach the destination. A notable example of path-oriented routing is the explicit path routing technique of Jueneman and Kerr [15] proposed for IBM's System Network Architecture [16]. The ARPANET, on the other hand, is a notable exception where routing decisions are made by intermediate store-and-forward nodes. Path-oriented routing has the advantages that (a) routing decisions are decentralized, (b) packets are guaranteed to arrive in FCFS order along each path, (c) loops can be avoided, and (d) the impact of bad decisions made by a source node is limited. A simple example of path-oriented routing is fixed routing where there is a unique path for each source–destination node pair. Another example is 'random routing' where one or more paths are set up for each source–destination node pair and the path of each packet is selected independently by the source node according to a probability distribution. In a virtual channel network, several virtual channels may be present between each source–destination node pair; either fixed or random routing may be employed for each virtual channel.

For path-oriented routing, each path can conveniently be modeled by a routing chain. In this section, we consider the mean end-to-end delay of each routing chain in the network. The results can then be used to obtain the mean end-to-end delay for any given source–destination node pair (or any virtual channel) which employs multiple paths. It is also possible to get the probability density function of the end-to-end delay for a class of routing algorithms. These results will be presented in Section 4.2.

Let π_k be the path (or ordered set of channels) over which chain k packets are routed. The transition probabilities of chain k take on values of 0 and 1 only, i.e.,

$$p_{ij}^{(k)} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are successive channels in } \pi_k, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

With this definition for $p_{ij}^{(k)}$, it is easy to verify that the solution to (2.2) is

$$\lambda_{ik} = \begin{cases} \gamma_k & \text{if } i \in \pi_k, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

Since each channel in the network model is a FCFS server and all routing chains are assumed to be open, the equilibrium states probability has the following product form [3] (see (2.7)):

$$P(S) = \prod_{i=1}^{M} (1-\rho_i) n_i! \prod_{k=1}^{K} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!}, \tag{4.3}$$

where ρ_{ik} and ρ_i are given by (2.5) and (2.6) respectively. From (4.3), we get the following expression for the marginal queue length distribution at channel i:

$$P(\boldsymbol{n}_i) = (1-\rho_i) n_i! \prod_{k=1}^{K} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!}. \tag{4.4}$$

The mean number of chain k packets at channel i can then be obtained from:

$$E[n_{ik}] = \sum_{j=0}^{\infty} j \sum_{\boldsymbol{n}_i \in R_j} P(\boldsymbol{n}_i), \tag{4.5}$$

where

$$R_j = \{\boldsymbol{n}_i : n_{ik} = j\}. \tag{4.6}$$

Substituting (4.4) into (4.5) and after some simplications, we get:

$$E[n_{ik}] = \frac{\rho_{ik}}{1-\rho_i}. \tag{4.7}$$

Applying Little's formula [11], the mean delay of chain k packets at channel i is

$$T_{ik} = E[n_{ik}]/\gamma_k = \frac{1}{\mu C_i (1-\rho_i)}. \tag{4.8}$$

It is of interest to note that the mean delay at channel i is determined by the total utilization of channel i, and is the same for all chains that are routed through this channel. Finally, the mean end-to-end delay of chain k is:

$$T_k = \sum_{i \in \pi_k} \frac{1}{\mu C_i(1-\rho_i)}. \tag{4.9}$$

As a remark, the mean end-to-end delay over all packets can be obtained from:

$$T = \sum_{k=1}^{K} \frac{\gamma_k}{\gamma} T_k \tag{4.10}$$

and it can be verified that (3.4) and (4.10) are identical.

We now illustrate how the result in (4.9) can be used to obtain the mean end-to-end delay for each source–destination node pair. For convenience, we refer to packets sent from source node s to destination node d as (s,d) packets. Let $\gamma_{s,d}$ be the mean arrival rate of (s,d) packets and $A_{s,d}$ be the set of routing chains for these packets. Also let $\alpha_{s,d}^{(k)}$ be the probability that a (s,d) packet is sent along the path corresponding to chain k. $\alpha_{s,d}^{(k)}$ is zero if chain $k \notin A_{s,d}$; and for the case of fixed routing, there is only one chain in each $A_{s,d}$ and the $\alpha_{s,d}^{(k)}$ for this chain is one. From the above definitions, it is easy to see that the mean arrival rate of chain k is given by:

$$\gamma_k = \gamma_{s,d}\alpha_{s,d}^{(k)} \tag{4.11}$$

and the mean end-to-end delay of (s,d) packets is:

$$T_{s,d} = \sum_{k \in A_{s,d}} \alpha_{s,d}^{(k)} T_k. \tag{4.12}$$

Similar results can also be obtained for a virtual channel network (with no flow control) where fixed or random routing is used for each virtual channel.

4.2. Distribution of end-to-end delay

In this section, we consider the distribution of end-to-end delay given path-oriented routing. This is a detailed characterization of end-to-end delay and the results are useful for calculating statistics such as variance and 90-percentile.

Our discussion is based on the work reported in [17,18]. Let $t_k(x)$ be the probability density function (pdf) of the end-to-end delay of chain k and $T_k^*(\xi)$ be its Laplace transform, i.e.,

$$T_k^*(\xi) = \int_0^\infty \exp(-\xi x) t_k(x)\, dx. \tag{4.13}$$

Let $N_k(z)$ be the generating function of the number of chain k packets in the network. As a result of the product form solution, $N_k(z)$ can be written as:

$$N_k(z) = \prod_{i \in \pi_k} N_{ik}(z), \tag{4.14}$$

where $N_{ik}(z)$ is the generating function of the number of chain k packets at channel i. $N_{ik}(z)$ is by definition, given by:

$$N_{ik}(z) = \sum_{j=0}^{\infty} \sum_{\boldsymbol{n}_i \in R_j} P(\boldsymbol{n}_i) z^j \tag{4.15}$$

where R_j is given by (4.6). Substituting (4.4) into (4.15), and after some simplifications, we get

$$N_{ik}(z) = \frac{1-\rho_i}{1-\rho_i+\rho_{ik}(1-z)}. \tag{4.16}$$

It then follows from (4.14) and (4.15) that

$$N_k(z) = \prod_{i \in \pi_k} \frac{1-\rho_i}{1-\rho_i+\rho_{ik}(1-z)}. \tag{4.17}$$

Since the arrival process of chain k packets is Poisson and the number of chain k packets in the network changes by unit steps, we also have [19,20]:

$$N_k(z) = D_k(z) \tag{4.18}$$

where $D_k(z)$ is the generating function of the number of chain k packets left behind in the network by a chain k departure.

Consider an arbitrary 'tagged' chain k packet. Let its end-to-end delay be t_k (Laplace transform of pdf is $T_k^*(\xi)$). With path-oriented routing and FCFS discipline at each channel, the number of chain k packets left behind when the tagged packet departs is equal to the number of chain k arrivals during t_k.

Suppose the tagged packet enters the network at time 0. Let $a_k(\tau)$ be the number of chain k arrivals in $[0,\tau]$. $D_k(z)$ is then the generating function of $a_k(t_k)$. For chain k arrivals following a Poisson process, $D_k(z)$ is given by [19]:

$$D_k(z) = T_k^*(\gamma_k - \gamma_k z) \tag{4.19}$$

provided that t_k and $a_k(\tau)$ are independent for all τ. Substituting ξ for $\gamma_k - \gamma_k z$, (4.19) is reduced to:

$$T_k^*(\xi) = D_k(1-\xi/\gamma_k). \tag{4.20}$$

Finally, using (4.17) and (4.18) in (4.19), we get

$$T_k^*(\xi) = \prod_{i \in \pi_k} \frac{\mu C_i(1-\rho_i)}{\xi + \mu C_i(1-\rho_i)}. \tag{4.21}$$

Let $|\pi_k|$ be the number of channels in π_k. Eq. (4.21) indicates that the end-to-end delay of chain k is given by the sum of $|\pi_k|$ independent random variables; the i-th random variable is exponentially distributed with mean $1/[\mu C_i(1-\rho_i)]$. This observation allows us to write the following expression for the variance of chain k end-to-end delay [17]:

$$\sigma_k^2 = \sum_{i \in \pi_k} \frac{1}{[\mu C_i(1-\rho_i)]^2}. \tag{4.22}$$

To obtain statistics such as the 90-percentile of end-to-end delay, one must first obtain $t_k(x)$ by inverting $T_k^*(\xi)$. This can easily be done by the technique of partial fraction expansion [19].

It should be noted that (4.19) (and therefore (4.21)) is true only when t_k and $a_k(\tau)$ are independent for all τ. In an arbitrary network, these two random variables are not independent in general, as illustrated by the example shown in Fig. 2 [18,24]. Suppose a tagged chain 1 packet arrives at channel i at time 0, and is somewhere in the network (except at channel j) at time τ. If $a_1(\tau)$ is large, then most packets leaving channel i (after the tagged packet) are from chain 1 and the tagged packet is expected to find a small number of chain 2 packets when it arrives at channel j. On the other hand, if $a_1(\tau)$ is small, then most packets leaving channel i are from chain 2 and the tagged packet is expected to find a large number of chain 2 packets at channel j. The delay experienced by the tagged packet at channel j is therefore affected by $a_1(\tau)$. Consequently, t_1 and $a_1(\tau)$ are not independent. A similar argument also applies to t_2 and $a_2(\tau)$.

From the above discussion, we observed that t_k is dependent on $a_k(\tau)$ whenever it is possible for packets (belonging to other chains) arriving after a tagged chain k packet at one channel to overtake this tagged packet at another channel. This dependency would not be present if the paths in the network were such that no such overtaking is possible. We can therefore give the following sufficient condition for t_k and $a_k(\tau)$ to be independent for all τ [18].

Nonpassing Condition. For each pair of channels i, j in π_k, packet arriving after any tagged chain k packet at channel i never overtake this tagged packet at channel j.

The Laplace transform of chain k end-to-end delay is given by (4.21) if the above condition is satisfied.

Despite the fact that t_k and $a_k(\tau)$ are not independent in general, (4.21) is very useful in practice for characterizing in detail the end-to-end delays of routing chains. For a given network model, it is likely that the nonpassing condition is satisfied for a large fraction of routing chains. The result is then applicable to each of these chains. For those chains where the nonpassing condition is not satisfied, simulation experiments have shown that (4.21) gives accurate approximations to the variance as well as 90-percentiles of end-to-end delay [18]. Furthermore, in some networks, the network topology and path assignments are such that the overtaking phenomenon shown in Fig. 2 is not possible. Consequently, (4.21) is applicable to all chains in the network. Obvious examples of such networks are tree networks and ring networks. Another example is the class of networks where the routing algorithm does not use any path with more than three channels. One such network is the example shown in Fig. 1 under shortest path routing.

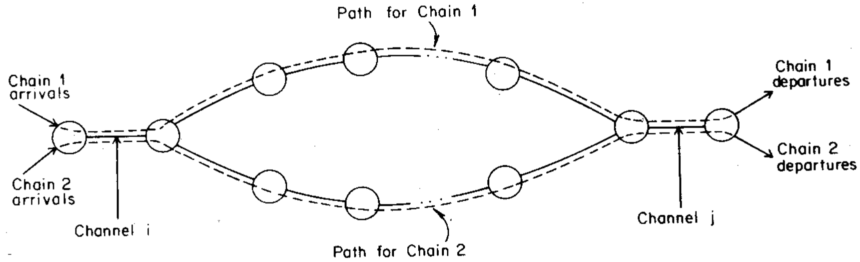


Fig. 2. Dependency between t_k and $a_k(\tau)$.

As a final remark, one can also obtain results for the pdf of end-to-end delay for a given source–destination node pair (or a given virtual channel). Following the developments which lead to $T_{s,d}$ in (4.12), the pdf of (s,d) end-to-end delay can be obtained by inverting

$$T^*_{s,d}(\xi) = \sum_{k \in A_{s,d}} \alpha^{(k)}_{s,d} \prod_{i \in \pi_k} \frac{\mu C_i(1-\rho_i)}{\xi + \mu C_i(1-\rho_i)} \tag{4.23}$$

and the corresponding variance is

$$\sigma^2_{s,d} = \sum_{k \in A_{s,d}} \alpha^{(k)}_{s,d} \times \left[\sum_{i \in \pi_k} \frac{1}{[\mu C_i(1-\rho_i)]^2} + T_k^2 \right] - T^2_{s,d} \tag{4.24}$$

where T_k and $T_{s,d}$ are given by (4.9) and (4.12) respectively.

5. Fairness among chains

The analysis of end-to-end delay presented in Section 4 is based on a FCFS discipline at each channel. One can easily observe from the result in (4.9) that the mean end-to-end delay of chain k is affected by the number of channels in π_k, and the utilization of these channels. As a result, there is a disparity of end-to-end delays experienced by different chains. A natural question to ask is whether a network with FCFS discipline at each channel is fair or not.

One approach to study the fairness of a network is to relate the mean end-to-end delay to network tariffs [21]. Some networks, e.g., Datapac [22], have an uneven tariff structure. For these networks, one can argue that subscribers who are paying more due to their physical locations should not be penalized with a longer end-to-end delay. A reasonable (or 'fair') strategy is then to make T_k the same for all k. In other networks, e.g., Telenet [23], a fixed tariff is applied regardless of location; and a reasonable strategy is to have T_k proportional to the number of channels in π_k.

It is easy to observe that FCFS may not be flexible enough to implement either one of the strategies mentioned above. One needs a parameterized queuing discipline which enables the chain delays to be adjustable by changing the parameter values. An example of such a discipline is Kleinrock's time-dependent priority discipline [6]. Under this discipline, a parameter β_k is associated with routing chain k, $k = 1, 2, \ldots, K$. Suppose a chain k packet arrives at channel i at time t_0, its instantaneous priority at time t $(t > t_0)$ is given by $(t - t_0)\beta_k$. When the channel is ready for the next packet, the instantaneous priorities of all packets in queue are evaluated, and the one with the highest value is served. This discipline thus favors chains with large values of β_k, and also those packets that have been waiting in queue for a long time.

The product-form solution for open queuing network models does not apply when Kleinrock's time-dependent priority discipline is used. However, the following assumption may be employed to get approximate results for the chain delays [21].

Poisson Assumption. For all i and k, the arrival process of chain k packets to channel i is Poisson regardless of the channel queuing discipline used.

Suppose the routing chains are ordered such that: $0 \leq \beta_1 \leq \beta_2 \leq \cdots \leq \beta_K$.

The mean delay of chain k packets at channel i (denoted by $T_{ik}(\text{td})$ for Kleinrock's time-dependent discipline) can be expressed as:

$$T_{ik}(\text{td}) = W_{ik} + \frac{1}{\mu C_i} \tag{5.1}$$

where W_{ik} is the mean waiting time in queue and is given recursively by [6]

$$W_{ik} = \left(\frac{\rho_i}{\mu C_i(1-\rho_i)} - \sum_{j=1}^{k-1} \rho_{ij} W_{ij} \left(1 - \frac{\beta_j}{\beta_k} \right) \right) \times \left(1 - \sum_{j=k+1}^{K} \rho_{ij} \left(1 - \frac{\beta_k}{\beta_j} \right) \right)^{-1} \tag{5.2}$$

where ρ_{ij} and ρ_i are given by (2.5) and (2.6) respectively. The mean end-to-end delay under Kleinrock's time-dependent discipline can then be obtained from:

$$T_k(\text{td}) = \sum_{i \in \pi_k} T_{ik}(\text{td}). \tag{5.3}$$

It is easy to see from (5.1) and (5.3) that the chain delays can be adjusted by manipulating the β_k's. Such manipulation, however, may affect the mean end-to-end delay over all packets; and adjusting the chain delays may become undesirable if the mean end-to-end delay is subtantially increased. In what follows, we shall first present a conservation theorem [21] which deals with the effect of channel scheduling on mean end-to-end delay, and then illustrate how this theorem and Kleinrock's time-dependent discipline can be used to improve the fairness of a network.

5.1. Conservation Theorem

The effect of channel scheduling on mean end-to-end delay can be characterized by the following theorem [21].

Conservation Theorem. For our open network model with the Poisson assumption, the mean end-to-end delay is identical for all work-conserving, non-preemptive disciplines, and is given by (3.4), i.e.,

$$T = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{\lambda_i}{\mu C_i - \lambda_i}.$$

This theorem indicates that the sensitivity of mean end-to-end delay to channel scheduling is determined by the accuracy of the Poisson assumption. Simulation results reported in [21] have shown that the Poisson assumption is accurate for Kleinrock's time-dependent discipline, and that adjusting the chain delays by manipulating the β_k's would not have a significant effect on the mean end-to-end delay.

5.2. Improvement of fairness

We have mentioned earlier that a network can be considered as being 'fair' if the T_k's bear some desired relationship with respect to each other (e.g., T_k is the same for all k, or T_k is proportional to the number of channels in π_k). In general, suppose the objective is to have T_k proportional to α_k, where α_k is a parameter indicating the importance of chain k (chain k is more important than chain m if $\alpha_k < \alpha_m$). This objective implies that ideally, we would like to have $T_k = \alpha_k \cdot A$ for all k, where A is a constant. Under this ideal condition, the mean end-to-end delay over all chains is given by:

$$T = \sum_{k=1}^{K} \frac{\gamma_k}{\gamma} \alpha_k A. \qquad (5.4)$$

Eq. (5.4) and the Conservation Theorem allow us to obtain the following expression for A:

$$A = \left(\sum_{i=1}^{M} \frac{\lambda_i}{\mu C_i - \lambda_i} \right) \Big/ \left(\sum_{k=1}^{K} \gamma_k \alpha_k \right). \qquad (5.5)$$

It follows that the ideal (or 'target') T_k can be determined uniquely from:

$$T_k\,(\text{target}) = \alpha_k \left(\sum_{i=1}^{M} \frac{\lambda_i}{\mu C_i - \lambda_i} \right) \Big/ \left(\sum_{m=1}^{K} \gamma_m \alpha_m \right). \qquad (5.6)$$

The original objective of having T_k proportional to α_k is thus equivalent to that of having the chain delay equal to the target delay for all chains. Due to the disparity in channel utilizations and in the number of channels visited by the various chains, this objective may not be achievable. As an example, consider a routing chain (say chain k) which visits only one channel. One might encounter a situation where the utilization of this channel is so low that T_k(td) is smaller than T_k (target) even if chain k is given the lowest priority (by setting β_k to zero). Under such situations, one would be interested in the deviation of the chain delays from the target delays. This deviation is used in [21] to define the following measure of fairness:

$$F = \frac{1}{T^2} \sum_{k=1}^{K} \frac{\gamma_k}{\gamma} \left(T_k(\text{td}) - T_k\,(\text{target}) \right)^2. \qquad (5.7)$$

F has the same form as the squared coefficient of variation. In general, a smaller F means a fairer network, and a network is said to be ideally fair if $F = 0$.

We note from (5.1) and (5.3) that the chain delays under Kleinrock's time-dependent discipline are functions of the β_k's. Since the target delays can be uniquely determined, F is also a function of the β_k's. One can therefore formulate the following optimization problem to determine the best setting of the β_k's so that F is minimized:

Given:	topology, $\{\gamma_k\}$, routing algorithm.
Minimize:	F.
Constraints:	$\beta_k \geqslant 0$, $k = 1, 2, \ldots, K$.

The solution to this optimization problem for some example networks can be found in [21].

6. Concluding remarks

In this paper, we have discussed in detail the application of product-form open queuing networks to the performance analysis of store-and-forward packet-switching networks. The topics considered include channel capacity assignment, optimal routing, distribution of end-to-end delay, and fairness among routing chains. In a companion paper [1], the application of queuing networks with closed chains and other forms of population size constraints to the performance analysis of networks with window flow control, buffer management schemes, and permit-oriented network congestion control is discussed.

References

[1] S.S. Lam and J.W. Wong, Queuing network models of packet switching networks, Part 2: Networks with population size constraints, Performance Evaluation 2 (3) (1982) to appear.

[2] J.R. Jackson, Jobshop-like queuing systems, Management Sci. 10 (1963) 131–142.

[3] F. Baskett, K.M. Chandy, R.R. Muntz and F. Palacios, Open, closed, and mixed networks of queues with different classes of customers, J. ACM 22 (1975) 248–260.

[4] M. Reiser and H. Kobayashi, Queuing networks with multiple closed chains: Theory and computational algorithms, IBM J. Res. Develop. 19 (1975).

[5] S.S. Lam, Queuing networks with population size constraints, IBM J. Res. Develop. 21 (1977) 370–378.

[6] L. Kleinrock, Queuing Systems, Vol. 2: Computer Applications (Wiley-Interscience, New York, 1976).

[7] J.M. McQuillan and V. Cerf, A Practical View of Computer Communications Protocols (IEEE Press, 1978).

[8] L. Kleinrock, Communication Nets – Stochastic Message Flow and Delays (McGraw-Hill, New York, 1964).

[9] L. Kleinrock and W.E. Naylor, On measured behavior of the ARPA network, AFIPS Conf. Proc., National Computer Conference 43 (1974) pp. 767–780.

[10] S.S. Lam, Store-and-forward buffer requirements in a packet switching network, IEEE Trans. Comm. COM-24 (1976) 394–403.

[11] J.D.C. Little, A proof of the queuing formula: $L = \lambda W$, Oper. Res. 9 (1961) 383–387.

[12] M. Gerla, The design of store-and-forward networks for computer communications, Ph.D. dissertation, Department of Computer Science, UCLA (1973).

[13] J.M. McQuillan, I. Richer and E.C. Rosen, The new routing algorithm for the ARPANET, IEEE Trans. Comm. COM-28 (1980) 711–719.

[14] L. Fratta, M. Gerla and L. Kleinrock, The flow deviation method: An approach to store-and-forward network design, Networks 3 (1973) 97–133.

[15] R.R. Jueneman and G.S. Kerr, Explicit path routing in communications networks, Proc. Internat. Conf. on Computer Communication (1976) pp. 340–342.

[16] V. Ahuja, Routing and flow control in systems network architecture, IBM Systems J. 18 (1979) 298–314.

[17] J.W. Wong, Distribution of end-to-end delay in message-switched networks, Comput. Networks 2 (1978) 44–49.

[18] J.W. Wong, Distribution of end-to-end delay in packet switching networks – Exact and approximate results, CCNG Rept., University of Waterloo, to appear.

[19] L. Kleinrock, Queuing Systems, Vol. 1: Theory (Wiley-Interscience, New York, 1975).

[20] R. Cooper, Introduction to Queuing Theory (MacMillan, New York, 1972).

[21] J.W. Wong, J.P. Sauve and J.A. Field, A study of fairness in packet-switching networks, IEEE Trans. Comm. COM-30 (1982) 346–353.

[22] The Bell Telophone Company of Canada, General Tariff CTC (TP), No. 6716, Item 4200, Datapac Service, Effective June 15, 1977.

[23] Logica Ltd., Packet Switching Report (London, 1978) p. 152.

[24] R. Foley, Virginia Polytechnic Institute and State University, private communication.