QUEUEING NETWORK MODELS

OF PACKET SWITCHING NETWORKS

S. S. Lam and J. W. Wong*

TR-167 March 1981

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

* J. W. Wong is with the Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada. His work 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. This paper is also available as Technical Report CS-81-06, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.

Table of Contents

Abstract

The application of product-form queueing networks to the performance analysis of store-and-forward packet-switching networks is considered. Multiple routing chains are used to model the different routing behaviors of packets. Queueing networks with open chains are first studied. The mean end-to-end delay over all chains is derived, and the application of this delay formula to optimal channel capacity assignment and optimal routing is discussed. Analytic results for the mean and distribution of end-to-end delay of each chain are presented. The issue of fairness among chains is also addressed.

Queueing networks with closed chains are studied next. The use of a closed chain to model a virtual channel with flow control window is illustrated. While analytic expressions for performance measures such as throughput and end-to-end delay are available, the computation of numerical results is very difficult. Two approximate methods which are computationally efficient are presented. Finally, queueing network models with population size constraints are considered, and the application of these models to buffer management in a switching node and permit-oriented congestion control is discussed.

Keywords: Queueing network models, packet switching networks, throughput, end-to-end delay, capacity assignment, routing, flow control, buffer management, permit-oriented congestion control, fairness.

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 is a packet*. Each packet traverses from its source node to its destination node through a series of communication channels along its path (or 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 queueing networks [1]-[4] to model store-and-forward packet communication networks.

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-

---

* 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.

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.

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 of a time-out period, the packet will be queued for retransmission.

Currently, there are two basic types of packet communication services: datagram and virtual channel [5]. 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 traffic 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. Measures 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 queueing network models have been successfully applied to the performance analysis of store-and-forward networks with some or all of the above features. To do so, several simplifying assumptions are necessary; they will be introduced in section 2. Queueing network models also have a number of limitations. One such limitation is that adaptive routing cannot be modeled. Analytic 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 queueing network models is affected by the presence of the 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 queueing 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 queueing network models to analyze a store-and-forward network was originally introduced by Kleinrock [6,7]:

> The Independence Assumption: "Each time a packet joins a queue in the network, its length is determined afresh from the probability density function
>
> $$b(x) = \mu \exp(-\mu x) \qquad x \geq 0$$
>
> where $1/\mu$ is the mean packet length (in number of bits)."

The above assumption removes the statistical dependence of the packet length at the various channels of a 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 [7,8]. Therefore, analytic results provided by queueing network models are merely approximations. However, these approximate results are generally deemed to be adaquate and valuable for the design and performance characterization of store-and-forward packet-switching networks [7].

We next define the class of queueing networks suitable for store-and-forward networks. (This class of network models is only a subset of network models that have a product-form solution [2]-[4].) The notation used throughout this paper is also in-

troduced.

Servers in the network model are indexed by i = 1,2,...,M. Server i works at a constant rate of C_i bits per second. 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 to different "routing chains"; these chains are indexed by k = 1,2,...,K. Specifically, the routing behavior of chain k is modeled by a first-order Markov chain with transition probabilities

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

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 appropriate definition of the transition probabilities [9].

Since the routing behavior of packets traveling between different source-destination node pairs are different, a queueing 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 source node of the chain according to a Poisson process with rate γ_k packets per second, where k = 1,2,...,K. Define

$$\gamma = \gamma_1 + \gamma_2 + \ldots + \gamma_K$$

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 queueing network model is determined from:

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

where

$$\delta_{ik} = \begin{cases} 1 & \text{if } i \text{ is source node of chain k} \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

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

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

The traffic intensity of chain k packets at server i is defined to be

$$\rho_{ik} = \frac{\lambda_{ik}}{\mu C_i} \tag{2.5}$$

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 queueing network be denoted by

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

where

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

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

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

and

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

A chain is said to be <u>open</u> 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 [2]:

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

where

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

and

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

The equilibrium probability of $\underline{n}$ also has a product form [2]:

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

where

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

A routing chain is said to be closed if the number of packets in the chain is fixed. Queueing networks with closed chains and networks with chain population size constraints are useful for modeling flow and congestion control in store-and-

forward networks. These models will be discussed in sections 4 and 5 respectively.

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

T = mean end-to-end delay over all packets transported by the network

γ^* = network throughput in packets per second

These two measures may be adequate by themselves or it may be necessary to characterize the performance of individual routing chains. Define

T_k = mean end-to-end delay of chain k packets transported by the network

γ_k^* = throughput of chain k in packets per second

We note that sometimes mean delays are not adequate for network design purposes and it is desirable to characterize the higher moments or percentiles of the delay random variables. Results on the distribution of end-to-end delay will be discussed in section 3.5.

If the network switching nodes have adequate buffers so that flow and congestion controls are not needed (a situation modeled by queueing networks with open chains), the chain throughput is the same as the chain arrival rates, i.e.,

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

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

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

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 number of trials needed for a packet to gain admittance to the network.

3. Open Queueing Network Models

We consider in this section the efficient utilization of communication channels in a packet network via routing assignments. Various performance criteria are addressed.

Queueing network models with open chains are employed. The number of buffers at each node is assumed to be very large (infinite) so that flow and congestion controls are not needed. The effect of packet transmission errors is also assumed to be negligible.

3.1 A Formula for Mean Network Transit Delay

The mean end-to-end delay T for an arbitrary packet transported by the network was first derived by Kleinrock [6,7]. It can be obtained as follows. By Little's formula [10], 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 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 eq.(2.11) gives rise to the following mean queue length:

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

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 eqs.(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 [11].

Recall that under our present assumption, $\gamma^* = \gamma$. Using T, as given by eq.(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 [7] for more details. However, eq.(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 eq.(2.1). A meaningful question to ask is: 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 [6,7] and to keep the problem simple, he made the following assumptions:

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

(b) the cost of channel capacity is a linear function 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:

$$\min_{\{C_i\}} \quad T = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{f_i}{C_i - f_i} \tag{3.7}$$

$$\text{subject to} \quad \sum_{i=1}^{M} d_i C_i \leq D$$

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 constrainted optimization problem can be converted to an unconstraint optimization problem by the Lagrange multiplier method which yields the following solution for optimal capacity assignment [6,7]:

$$C_i^* = f_i + \frac{D_e}{d_i} \frac{\sqrt{f_i d_i}}{\sum_{j=1}^{M} \sqrt{f_j d_j}} \tag{3.8}$$

where

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

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

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

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

$$\begin{aligned} &\underset{\{C_i\}}{\text{Min}} \quad \sum_{i=1}^{M} d_i C_i \\ &\text{subject to} \quad \frac{1}{\gamma} \sum_{i=1}^{M} \frac{f_i}{C_i - f_i} \leq T_{max} \end{aligned} \tag{3.11}$$

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

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

The minimum network cost for channels is then

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

The optimal capacity assignment given by eq.(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 discrete set of 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 [11]. 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 neighbours periodically. Considerable time delays, however, are needed for such information to propagate throughout the network [12].

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 does not necessarily

---

* The mean delay includes both the expected waiting time and packet transmission time. In practice, a fixed bias term is also included to reduce looping behavior.

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 [11,13].

Given the traffic requirements $\{\gamma_k\}$ and a specific routing assignment $\{p_{ij}^{(k)}\}$, the channel arrival rates $\{\lambda_i\}$ can be determined using eq.(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

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

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

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

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

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

$$\begin{aligned} \underline{f}' &= (1-\varepsilon)\underline{f} + \varepsilon\underline{v} \\ &= \underline{f} + \varepsilon(\underline{v}-\underline{f}) \qquad 0 \leq \varepsilon \leq 1 \end{aligned} \tag{3.14}$$

Note that the flow vector $\underline{v}$ can be chosen without satisfying v_i

$< C_i$ for any i. In this case, a sufficiently small ε should be used for $\underline{f}'$ to be feasible. Suppose $\varepsilon << 1$ so that the change in the flow vector (in the $\underline{v}$ direction) is small and is given by:

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

The resulting change in the mean network delay is

$$\begin{aligned}\Delta T(\underline{f}) &= \sum_{i=1}^{M} \frac{\partial T(\underline{f})}{\partial f_i} (v_i - f_i)\varepsilon \\ &= \varepsilon \sum_{i=1}^{M} L_i (v_i - f_i)\end{aligned} \tag{3.16}$$

where

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

Eq.(3.17) above requires that the function $T(\underline{f})$ be differentiable. If, moreover, the function $T(\underline{f})$ is also convex, then we know that a unique minimum exists among the set of feasible flow vectors (which is a convex set [11,13]). Thus, a necessary and sufficient condition for a feasible flow vector $\underline{f}$ to be optimal is

$$\Delta T(\underline{f}) \geq 0 \qquad \text{for any } \underline{v} \tag{3.18}$$

or

$$\sum_{i=1}^{M} L_i(v_i-f_i) \geq 0 \tag{3.19}$$

or

$$\min_{\underline{v}} \sum_{i=1}^{M} L_i v_i \geq \sum_{i=1}^{M} L_i f_i \tag{3.20}$$

If the condition in eq.(3.20) is not satisfied then $\underline{f}$ (and the corresponding routing assignment) is not optimal. Moreover, eq.(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 queueing network model, from eq.(3.5),

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

Hence

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

Given the traffic requirements $\{\gamma_k\}$, the optimal flow vector $\underline{f}$ (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 [11,13] 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 the excellent thesis of Gerla [11] for this and other network design problems.

3.4 Mean 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, as given by eq.(3.5), 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.

In sections 3.4 to 3.6, we shall consider only networks which employ path-oriented routing. This is also known as source routing. As the name suggests, 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 [14]

proposed for IBM's System Network Architecture [15]. 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 according to a probability distribution. For a virtual channel network with many virtual channels between each source-destination node pair, fixed or random routing can also be used 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 end-to-end delay for a class of routing algorithms. These results will be presented in Section 3.5.

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.,

$$p_{ij}^{(k)} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are successive channels in } \pi_k \\ 0 & \text{otherwise} \end{cases} \tag{3.23}$$

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

$$\lambda_{ik} = \begin{cases} \gamma_k & \text{if } i \in \pi_k \\ 0 & \text{otherwise} \end{cases} \tag{3.24}$$

Since each channel in the network model is a FCFS server and all routing chains are assumed to be open, the equilibrium state probability has the following product form [2] (see eq.(2.7)):

$$P(S) = \prod_{i=1}^{M} (1-\rho_i) n_i! \prod_{k=1}^{K} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!} \tag{3.25}$$

where ρ_{ik} and ρ_i are given by eqs.(2.5) and (2.6) respectively. From eq.(3.25), we get the following expression for the marginal queue length distribution at channel i:

$$P(\underline{n}_i) = (1-\rho_i) n_i! \prod_{k=1}^{K} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!} \tag{3.26}$$

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

obtained from:

$$E[n_{ik}] = \sum_{j=0}^{\infty} j \sum_{\underline{n}_i \in R_j} P(\underline{n}_i) \tag{3.27}$$

where

$$R_j = \left\{ \underline{n}_i : n_{ik}=j \right\} \tag{3.28}$$

Substituting eq.(3.26) into eq.(3.27) and after some simplifications, we get:

$$E[n_{ik}] = \frac{\rho_{ik}}{1-\rho_i} \tag{3.29}$$

Applying Little's formula [10], the mean delay of chain k packets at channel i is

$$T_{ik} = E[n_{ik}]/\gamma_k = \frac{1}{\mu C_i(1-\rho_i)} \tag{3.30}$$

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:

$$T_k = \sum_{i \in \pi_k} \frac{1}{\mu C_i(1-\rho_i)} \tag{3.31}$$

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

obtained from:

$$T = \sum_{k=1}^{K} \frac{\gamma_k}{\gamma} T_k \tag{3.32}$$

and it can be verified that eqs.(3.4) and (3.32) are identical.

We now illustrate how the result in eq.(3.31) 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:

$$\gamma_k = \gamma_{s,d}\, \alpha_{s,d}^{(k)} \tag{3.33}$$

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

$$T_{s,d} = \sum_{k \in A_{s,d}} \alpha_{s,d}^{(k)}\, T_k \tag{3.34}$$

Similar results can also be obtained for a virtual channel network where fixed or random routing is used for each virtual channel.

3.5 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 [16,17]. Let $t_k(x)$ be the probability density function (pdf) of chain k end-to-end delay and $T_k^*(\xi)$ be its Laplace transform, i.e.,

$$T_k^*(\xi) = \int_0^\infty \exp(-\xi x) t_k(x) dx \tag{3.35}$$

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:

$$N_k(z) = \prod_{i \epsilon \pi_k} N_{ik}(z) \tag{3.36}$$

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:

$$N_{ik}(z) = \sum_{j=0}^{\infty} \sum_{\underline{n}_i \epsilon R_j} P(\underline{n}_i) z^j \tag{3.37}$$

where R_j is given by eq.(3.28). Substituting eq.(3.26) into eq.(3.37), and after some simplifications, we get

$$N_{ik}(z) = \frac{1-\rho_i}{1-\rho_i+\rho_{ik}(1-z)} \tag{3.38}$$

It then follows from eqs.(3.36) and (3.38) that

$$N_k(z) = \prod_{i \epsilon \pi_k} \frac{1-\rho_i}{1-\rho_i+\rho_{ik}(1-z)} \tag{3.39}$$

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 [18,19]:

$$N_k(z) = D_k(z) \tag{3.40}$$

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 a_k, the number of chain k arrivals during t_k. $D_k(z)$ is then the generating function of a_k. For chain k arrivals following a Poisson process, $D_k(z)$ is given by [18]

$$D_k(z) = T_k^*(\gamma_k - \gamma_k z) \tag{3.41}$$

provided that t_k and a_k are independent. Substituting ξ for $\gamma_k - \gamma_k z$, eq.(3.41) is reduced to:

$$T_k^*(\xi) = D_k(1-\xi/\gamma_k) \tag{3.42}$$

Finally, using eqs.(3.39) and (3.40) in eq.(3.42), we get

$$T_k^*(\xi) = \prod_{i \in \pi_k} \frac{\mu C_i (1-\rho_i)}{\xi + \mu C_i (1-\rho_i)} \tag{3.43}$$

Let $|\pi_k|$ be the number of channels in π_k. Eq.(3.43) 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 [16]:

$$\sigma_k^2 = \sum_{i \in \pi_k} \frac{1}{[\mu C_i (1-\rho_i)]^2} \tag{3.44}$$

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 [18].

It should be noted that eq.(3.41) (and therefore eq.(3.43)) is true only when t_k and a_k are independent. In a packet-switching network, these two random variables are not independent in general, as illustrated by the example shown in fig. 2 [17,45]. Suppose a "tagged" chain 1 packet arrives at channel i at time 0. In the time interval $(0,t_1)$, if a_1 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 is small, then most packets leaving channel is 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. Consequently, t_1 and a_1 are not independent. A similar argument also applies to t_2 and a_2.

From the above discussion, we observe that t_k is dependent on a_k 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 are such that no such overtaking is possible. We can therefore give the following sufficient condition for t_k and a_k to be independent [17]:

> Non-passing condition: "For each pair of channels i, j in π_k, packets arriving after any tagged chain k packets at channel i never overtake this tagged packet at channel j."

The Laplace transform of chain k end-to-end delay is given by eq.(3.43) if the above condition is satisfied.

Despite the fact that t_k and a_k are not independent in general, eq.(3.43) 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 non-passing 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 non-passing condition is not satisfied, simulation experi-

ments have shown that eq.(3.43) gives accurate approximations to the variance as well as 90-percentiles of end-to-end delay [17]. Furthermore, in some networks, the network topology and path assignments are such that the overtaking phenomenon shown in fig. 2 is not possible. Consequently, eq.(3.43) 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.

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 eq.(3.34), the pdf of (s,d) end-to-end delay can be obtained by inverting

$$T^*_{s,d}(\xi) = \sum_{k\in A_{s,d}} \alpha^{(k)}_{s,d} \prod_{i\in\pi_k} \frac{\mu C_i(1-\rho_i)}{\xi+\mu C_i(1-\rho_i)} \tag{3.45}$$

and the corresponding variance is

$$\sigma^2_{s,d} = \sum_{k\in A_{s,d}} \alpha^{(k)}_{s,d} \left[\sum_{i\in\pi_k} \frac{1}{[\mu C_i(1-\rho_i)]^2} + T_k^2 \right] - T^2_{s,d} \tag{3.46}$$

where T_k and $T_{s,d}$ are given by eqs.(3.31) and (3.34) respectively.

3.6 Fairness Among Chains

The results for end-to-end delay presented in Sections 3.4 and 3.5 are based on a FCFS discipline at each channel. One can easily observe from the result in eq.(3.31) 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. It is therefore unlikely for the T_k's to 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). 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 with network tariffs [20]. Some networks, e.g., Datapac [21], 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 strategy is then to make T_k the same for all k. In other networks, e.g., Telenet [22], a fixed tariff is applied regardless of location. 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 strategy mentioned above. One needs a parameterized queueing 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 [7]. Under this discipline, a chain k packet has priority level zero when it enters a node, and this priority level increases with rate β_k while waiting for service. The product form solution for open queueing network models does not apply when such a discipline is used. However, approximate analysis may be employed, and in [20], an approximate expression is obtained for T_k, the mean end-to-end delay of chain k. This expression is in terms of the β's.

Suppose it is desirable to have T_k the same for all k, the following fariness measure is used in [20] to determine the optimal setting of the β_k's.

$$F = \sum_{k=1}^{K} \frac{\gamma_k}{\gamma} (T_k - T)^2 \tag{3.47}$$

The optimal β_k's are obtained by minimizing F with the constraints that $\beta_k \geq 0$ for all k. The solution to this optimization problem for some example networks can be found in [20].

4. Queueing Network Models with Closed Chains

Most packet networks nowadays provide virtual channels between data sources and sinks. The virtual channels are end-to-end flow controlled. Examples of end-to-end flow controls are SNA pacing [23], RFNM in the ARPANET [24], and various window mechanisms [25,26]. An important function of end-to-end flow control protocols is the synchronization of the data source input rate to the data sink acceptance rate. All of them work by limiting the number of packets that a virtual channel can have in transit within the network. Hence, they also provide, to some extent, a congestion control capability for the network as a whole. However, when the number of virtual channels supported by the network is very large, a separate congestion control mechanism for the network is often necessary.

In this section, we shall consider the case that each flow-controlled virtual channel is modeled by a closed chain. The chain population size corresponds to the "window size" of the virtual channel. The effect of virtual channel window sizes on the throughput-delay characteristics of the network can be studied.

We shall continue to make the assumption that the number of buffers at each switching node is very large (infinite). But with flow control windows, new packet arrivals may sometimes be rejected because the window of a virtual channel is full. These packets are assumed to be lost. Thus the throughput γ_k^* of a virtual channel (modeled by routing chain k) is less than its exter-

nal arrival rate γ_k; the latter is often referred to as the offered load of the virtual channel. Note that in reality, packets that are rejected are not really lost as assumed by the model herein. These packets are merely delayed and stored external to the packet network and resubmitted later. The ratio γ_k/γ_k^* measures the mean number of trials for a packet to gain admittance into the network.

Alternatively, the following interpretation of lost arrivals is also appropriate. Whenever the window of a virtual channel is full, the data source is notified and prevented from submitting any packet to the network; the arrival rate of packets to the virtual channel is thus zero. When the virtual channel window is open again, the data source is notified, and packets will then arrive at rate γ_k. With either interpretation, the above model assumptions enable us to focus our attention upon the network behavior only and ignore the behavior of queues external to the network.

4.1 The Product-Form Solution of Closed Queueing Networks

A closed chain has a fixed number of circulating customers. However, it is sometimes physically meaningful to think of a closed chain as an open chain with the following two mechanisms in place at all times [1,4]:

(a) a loss mechanism whereby an external arrival is rejected and lost foreover;

(b) a trigger mechanism whereby a departure from the network triggers an instantaneous injection of a new customer into

the same chain as the departed customer (from an infinite source of customers).

For a closed chain (say k), its packet arrival rate to server i in the queueing network can only be determined to within a multiplicative constant (called the scaling factor of the chain) from

$$\lambda_{ik} = \sum_{j=1}^{M} \lambda_{jk} p_{ji}^{(k)} \qquad i = 1, 2, \ldots, M \tag{4.1}$$

This is due to the fact that the matrix of transition probabilities of chain k is a stochastic matrix.

Consider a network consisting of closed chains only. Let N_k be the population size of chain k and define:

$$\underline{N} = (N_1, N_2, \ldots, N_K)$$

The equilibirum network state probability has the following product form [2]:

$$P(S) = \frac{1}{G(\underline{N})} \prod_{i=1}^{M} p_i(\underline{n}_i) \tag{4.2}$$

where $p_i(n_i)$ is given by eq.(2.8) and $G(\underline{N})$ is the normalization constant and is by definition

$$G(\underline{N}) = \sum_{S \in V} \prod_{i=1}^{M} p_i(\underline{n}_i) \tag{4.3}$$

where

$$V = \left\{ S: \sum_{i=1}^{M} \underline{n}_i = \underline{N} \right\}$$

We mentioned earlier that the λ_{ik}'s can only be determined to within a multiplicative constant. This multiplicative constant is absorbed into $G(\underline{N})$ and will be taken care of in the normalization process.

Similar to eq.(2.10), we also have:

$$P(\underline{n}) = \frac{1}{G(\underline{N})} \prod_{i=1}^{M} p_i(n_i) \tag{4.4}$$

where $p_i(n_i)$ is given by eq.(2.11).

Note that the normalization constant $G(\underline{N})$ is the sum of an extremely large number of product terms when K is large and also when the chain population sizes $\{N_k\}$ are large. There are two difficult problems in the evaluation of $G(\underline{N})$. First, depending on the scaling factors selected in the determination of $\{\lambda_{ik}\}$ in eq.(4.1), $G(\underline{N})$ may become very large (causing a floating point overflow) or very small (causing a floating point underflow). This problem has recently been successfully solved by the discovery of a simple dynamic scaling technique [27]. The second problem is the extremely large computational time and space requirements to evaluate $G(\underline{N})$ even for moderate values of K and $\{N_k\}$. For example, if the convolution algorithm is used [3,28], an array of G values indexed from $\underline{0}$ to $\underline{N}$ is necessary; and the

storage requirement for the array alone is proportional to the product $N_1N_2...N_K$. The time requirement of the convolution algorithm is also very large with an operation count of $O(MKN_1N_2...N_K)$.

Below we first illustrate how one can model a virtual channel by a closed chain, and then discuss two approximate approaches that avoid the large space-time computational requirements when the number of virtual channels is large. The first approach is to focus upon a single closed chain and replace all other closed chains by an equivalent open chain [29]. The second is a heuristic solution technique [30] that is a natural extension of the mean value analysis (MVA) algorithm of Reiser and Lavenberg [31].

4.2 Modeling a Virtual Channel with a Closed Chain

Fig. 3 depicts a model of a single virtual channel between a packet source and a packet sink. The flow control window of the virtual channel is modeled by a closed chain with a fixed number of customers equal to the window size W. An IS server is inserted and joins the packet sink to the packet source to "close" the chain. It models a random delay corresponding to the time for an acknowledgement to be returned from sink to source. The service time at the source (or sink) represents the time delay to generate (or absorb) a data packet. Note that the source queue is empty when the number of unacknowledged packets is W. This models the behavior that the source is not authorized to

generate its next data packet until an acknowledgement is received. Note also that the source server is an exponential server. This accurately models the assumption that new packets are generated from a Poisson source.

We shall not consider the explicit modeling of the acknowledgement (ACK) traffic for two reasons. First, most of the time, ACK messages are piggybacked on regular data packets; stand-alone ACK packets are typically very short and transmitted with priority. Thus the impact of ACK traffic on the delay performance of data traffic is minimal. (In [30], Reiser suggested that ACK traffic may be accounted for by reducing the effective channel capacities by an amount equal to the throughput of ACK packets.) Secondly, from a modeling viewpoint, the distinction of ACK packets as a separate class of packets with a smaller mean packet length than the data packets and possibly priority queueing service cannot be accommodated by product-from queueing networks. With the above considerations, the model illustrated above is deemed to be an adequate representation of a virtual channel that is also simple to use.

4.3 Analysis of a Single Virtual Channel

As mentioned in section 4.1, one approach to avoid the large computational requirements of closed networks is to approximate a number of closed chains by an equivalent open chain. We thus have a queueing network model where some routing chains are open while the others are closed. This is referred to as a mixed

network model [2]. In this section, we first present analytic results for mixed networks and then illustrate how these results can be applied to a model which focuses on a single virtual channel.

The equilibrium state probability of a mixed network also has the product form solution [2], i.e.,

$$P(S) = \frac{1}{G} \prod_{i=1}^{M} P_i(\underline{n}_i) \tag{4.5}$$

The normalization constant G is now obtained by summing over the open and closed chains. G can be written as [3]:

$$G = G^O \, G^C(\underline{N}) \tag{4.6}$$

Let $\rho_i^o = \sum_{\text{all open chain } k} \rho_{ik}$ and $\rho_i^c = \sum_{\text{all closed chain } k} \rho_{ik}$, $i = 1,2,\ldots,M$.

G^O is given by:

$$G^O = \prod_{i=1}^{M} G_i^o \tag{4.7}$$

where

$$G_i^o = \begin{cases} 1/(1-\rho_i^o) & \text{server } i \text{ is FCFS} \\ \exp(\rho_i^o) & \text{server } i \text{ is IS} \end{cases} \tag{4.8}$$

$G^C(\underline{N})$ is the normalization constant of a network model with

closed chains only and with the following modification to the traffic intensities of closed chains:

$$\rho'_{ik} = \begin{cases} \rho_{ik}/(1-\rho_i^o) & \text{server } i \text{ is FCFS} \\ \rho_{ik} & \text{server } i \text{ is IS} \end{cases} \qquad (4.9)$$

We now apply the results to the single virtual chain model shown in fig. 4. This model is based on the abstraction of all other network traffic into a single open chain, an approach used by Pennotti and Schwartz [29]. There are two routing chains: chain 1 is closed and models the virtual channel, and chain 2 is the open chain mentioned above. For simplicity, the time for the sink to absorb a data packet and the acknowledgement delay are both assumed to be zero (these details can be included without much difficulty). The service time at the source server is assumed to be exponentially distributed with mean $1/\gamma_1$. As discussed in section 4.2, this assumption models the behavior that new packets are generated from a Poisson source. Without loss of generality, packets belonging to the open chain (chain 2) are assumed to depart from the network after receiving service from one channel.

For chain 1, a solution to eq.(4.1) is $\lambda_{i1} = 1$ for all i. The equilibrium state probability is then given by:

$$P(S) = \frac{1}{G} \prod_{i=1}^{M} n_i! \prod_{k=1}^{2} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!} \qquad (4.10)$$

where

$$\rho_{i1} = \begin{cases} 1/\gamma_1 & i = 1 \\ 1/(\mu C_i) & i > 1 \end{cases} \tag{4.11}$$

and

$$\rho_{i2} = \begin{cases} 0 & i = 1 \\ \lambda_{i2}/(\mu C_i) & i > 1 \end{cases} \tag{4.12}$$

Applying eq.(4.6) and recognizing that $n_{12} = 0$ (chain 2 packets never visit the source server), G is given by:

$$G = \left[\prod_{i=2}^{M} (1-\rho_{i2}) \right] G^c(W) \tag{4.13}$$

where $G^c(W)$ is the normalization constant for a closed network with chain 1 only and with traffic intensity $\rho'_{i1} = \rho_{i1}/(1-\rho_{i2})$.

From the equilibrium state probability, the marginal queue length distribution of chain 1 can be shown to be given by [3,29]:

$$P(\underline{n}_1) = G^c(W) \prod_{i=1}^{M} \left[\frac{\rho_{i1}}{1-\rho_{i2}} \right]^{n_{i1}} \tag{4.14}$$

This is identical to a network model with a single closed chain and with the service rate of channel i reduced to $\mu C_i(1-\rho_{i2})$. Finally, the throughput seen by the virtual channel is given by:

$$\gamma_1^* = \text{Prob[source server is busy]}\ \mu C_1$$

Using Buzen's result [28], γ_1^* can be conveniently expressed as:

$$\gamma_1^* = \frac{G^c(W-1)}{G^c(W)} \tag{4.15}$$

The mean number of chain 1 packets at channel i can be obtained from:

$$E[n_{i1}] = \sum_{\underline{n}_1 \in V} n_{i1} P(\underline{n}_1) \tag{4.16}$$

where

$$V = \left\{ \underline{n}_1 : \sum_{j=1}^{M} n_{j1} = W \right\}$$

The mean number of chain 2 (open chain) packets at channel i has the following simple form [29]:

$$E[n_{i2}] = \frac{\lambda_{i2}}{\mu C_i - \lambda_{i2}} (1+E[n_{i1}]) \tag{4.17}$$

Note that eq.(4.17) can be interpreted as the mean queue length with chain 2 only multiplied by $1 + E[n_{i1}]$, a factor due to the presence of chain 1 packets.

4.4 A Heuristic Solution Technique based upon the MVA Algorithm

The time and space complexity of the MVA algorithm [31] is on the same order as the convolution algorithm. However, it has an intuitively appealing extension to an efficient heuristic

solution technique that is shown to be asymptotically valid as the chain population sizes become infinite [31].

Let Q(k) be the set of servers visited by chain k and C(i) be the set of chains that visit server i. Also define the following notation for a closed queueing network with population vector $\underline{N}$.

$T_{ik}(\underline{N})$ -- mean delay of a chain k packet at server i

$q_{ik}(\underline{N})$ -- mean number of chain k packets at server i

$\gamma_k^*(\underline{N})$ -- throughput of chain k in packets per second (this throughput is measured either at the source node or the sink node)

We first introduce the MVA algorithm which is based upon the following recursive equation:

$$T_{ik}(\underline{N}) = \begin{cases} \tau_{ik}[1+\sum_{c\in C(i)} q_{ic}(\underline{N}-\underline{1}_k)] & \text{server i is FCFS} \\ \tau_{ik} & \text{server i is IS} \end{cases} \qquad (4.18)$$

where τ_{ik} is the mean service time of server i. It is equal to $1/\mu C_i$ for any k if server i is a communication channel (FCFS server). $\underline{1}_k$ is a K-element vector with the k-th component equal to one and all others equal to zero. $q_{ic}(N-\underline{1}_k)$ is zero if N_k in $\underline{N}$ is zero.

Using Little's formula first for chain k, and then for chain k at server i, the following are also derived for the MVA

algorithm [31]:

$$\gamma_k^*(\underline{N}) = \frac{N_k}{\sum_{i \in Q(k)} T_{ik}(\underline{N})} \tag{4.19}$$

and

$$q_{ik}(\underline{N}) = \gamma_k^*(\underline{N}) T_{ik}(\underline{N}) \tag{4.20}$$

Starting with $q_{ik}(\underline{0})$ for $i = 1,2,\ldots,M$ and $k = 1,2,\ldots,K$, $T_{ik}(\underline{N})$, $\gamma_k^*(\underline{N})$ and $q_{ik}(\underline{N})$ can be solved recursively using eqs.(4.18) to (4.20).

A heuristic technique was proposed by Reiser [30] to solve the above set of equations iteratively. Suppose we can calculate the difference

$$\begin{aligned} \varepsilon_{ic}(k-) &= q_{ic}(\underline{N}) - q_{ic}(\underline{N}-\underline{1}_k) \\ &= f(i,c,k,\underline{N}) \end{aligned} \tag{4.21}$$

with some function f. Then the MVA equations above can be written as:

$$T_{ik} = \begin{cases} \tau_{ik}\{1 + \sum_{c \in C(i)} [q_{ic} - \varepsilon_{ic}(k-)]\} & \text{server } i \text{ is FCFS} \\ \tau_{ik} & \text{server } i \text{ is IS} \end{cases} \tag{4.22}$$

$$\gamma_k^* = \frac{N_k}{\sum_{i \in Q(k)} T_{ik}} \tag{4.23}$$

and

$$q_{ik} = \gamma_k^* T_{ik} \tag{4.24}$$

where the argument $\underline{N}$ has been omitted from T_{ik}, γ_k^*, and q_{ik}.

Eqs.(4.21) to (4.24) form a set of nonlinear simultaneous equations. A simple method for solving them is by a successive substitution technique starting with an initial set of mean queue lengths $\{q_{ik}\}$ and iterate sequentially through the four equations until convergence is observed.

The space requirement is now substantially less since we only need to keep a single set of values for T_{ik}, γ_k^*, q_{ik} and $\varepsilon_{ic}(k-)$. The time requirement is also significantly reduced with the following heuristic method for evaluating $\varepsilon_{ic}(k-)$, i = 1,2,...,M, c = 1,2,...,K, in eq.(4.21) [30].

(a) Assuming that the chain with one less packet is affected the most, use the estimate

$$\varepsilon_{ic}(k-) = 0 \qquad \text{for any } c \neq k \tag{4.25}$$

(b) $\varepsilon_{ik}(k-)$ is estimated by a single chain network with suitably redefined parameters as follows. The mean service time for all FCFS servers in the single chain network is

$$\tau_i \leftarrow \frac{\tau_{ik}}{1 - \sum_{\substack{c \in C(i) \\ c \neq k}} \gamma_c^* \tau_{ic}} \tag{4.26}$$

where $\{\gamma_c^*\}$ is the set of chain throughputs at the current iteration step. Eq.(4.26) suggests that a chain k packet sees only a fraction of the channel capacity and it is consistent with the interference effect of the open chain traffic on closed chains at a FCFS queue considered in the last section. Let $q_i(N_k)$ denote the mean queue size of server i in the single-chain network. The following estimate is used in conjunction with those in eq.(4.25) for the heuristic procedure

$$\varepsilon_{ik}(k-) = q_i(N_k) - q_i(N_k-1) \tag{4.27}$$

The complexity per iteration step in the heuristic soltuion is of the order $KM(N_1+N_2+\ldots+N_K)$ which is affordable even for large population sizes provided that convergence is achievable within a small number of iterations. It was observed empirically by Reiser [30] that the above iterative procedure converges rapidly from any initial values for $\{q_{ik}\}$ and $\{\gamma_k^*\}$. The only requirements to be satisfied by the initial condition are

$$\sum_{i\in Q(k)} q_{ik} = N_k \qquad \text{all } k \tag{4.28}$$

and

$$q_{ik} \geq 0 \qquad \text{all } i,k \tag{4.29}$$

It was also argued that the iterative procedure is asymptotically valid as population sizes becomes infinite. This heuristic techniques has also been generalized to queueing network models that do not have the product-form solution [30].

5. Queueing Network Models with Population Size Constraints

In this section, we shall consider the modeling of congestion control and buffer management strategies in packet networks. To do so, we need results for the class of queueing networks with population size constraints.

5.1 The Model

Routing chains in queueing network models considered prior to now are either open or closed. Recall our discussion in section 4.1 that a closed chain can be viewed as an open chain with the loss and trigger mechanisms in place at all times. Suppose the loss and trigger mechanisms are invoked or revoked as a function of the network's population vector $\underline{N} = (N_1, N_2, \ldots, N_K)$. Such queueing networks are said to have population size constraints. Given the loss and trigger mechanisms as functions of $\underline{N}$, let V be the set of feasible network population vectors. A sufficient condition for the equilibrium network state probability P(S) to have a product-from solution is [4]:

> "For any chain k and population vector $\underline{N}$ and $\underline{N} + \underline{1}_k$ in V, the loss mechanism is invoked for a chain k external arrival in any network state with population vector $\underline{N}$ if and only if the trigger mechanism is invoked for a chain k external departure in any network state with population vector $\underline{N} + \underline{1}_k$."

This is equivalent to the condition that feasible transitions between population vectors in V are paired.

By permitting V to be a singleton set as well as an infinite

set, both networks of closed chains and networks of open chains are included here as special cases. It is shown in [27] that the improper equilibrium probability of $\underline{N}$ is equal to the normalization constant $G(\underline{N})$ of an equivalent closed network with population vector $\underline{N}$ and chain arrival rates to individual servers given by eq.(4.1) for closed chains and eq.(2.2) for chains that permit external arrivals. For constant external chain arrival rates, a queueing network model with population size constraints has the product form solution given by eq.(4.2) with the following expression for the normalization constant:

$$G = \sum_{\underline{N} \in V} G(\underline{N}) \tag{5.1}$$

In the next three sections, we analyse strategies for buffer management and congestion control using queueing network models with population size constraints.

5.2 Finite-Buffer Single Node Model

When a packet is in transit in a packet switching network, it occupies buffer space in the intermediate store-and-forward nodes. When nodes with finite buffers are considered, the allocation of buffers to the various chains will affect the node's performance (and hence, the network's performance).

To model a packet switching network with finite buffer space, one must consider the situation when a packet routed to a node finds no available buffer in that node. The usual

protocol is to require the sending node to keep the packet until an acknowledgement is returned from the receiving node. If the acknowledgement is not received within a time-out interval, the packet is retransmitted. Buffer space in the sending node is therefore occupied by this packet until the acknowledgement is received.

Exact analytic results for a network model with finite buffers at each node is not presently available. Models for a single switching node, however, have been successfully analyzed [32]-[36]. In this section, the general buffer allocation scheme of Kamoun and Kleinrock [33] is discussed. Their model does not include any form of acknowledgement. When a packet is transmitted on an outgoing channel, the buffer space it occupies is assumed to be released immediately. A single node model with acknowledgement and time-out [9,35] will be discussed in section 5.4. An approximation technique to analyze a network model with finite buffers will also be described in that section.

The single node model analysed by Kamoun and Kleinrock [33] is shown in fig. 5. It is a queueing network model with M servers, one for each outgoing channel. There are M routing chains: packets routed to the same outgoing channel are in the same chain. The arrival process of chain i packets to the node is assumed to be Poisson with rate λ_i, i = 1,2,...,M. The population size constraints for the various routing chains are determined by the buffer management scheme used.

Packets from the various chains are sharing a total of B

buffers. Depending on the buffer allocation scheme, an arriving packet may be rejected from entering the node. This packet is assumed to be lost and will never return. In a network with finite buffers, rejected packets are not really lost, but are retransmitted later; hence, the arrival process to each node is not likely to be Poisson. The Poisson arrival assumption mentioned above is therefore only an approximation.

The general scheme for sharing buffers [33] can be specified by the following rules:

(a) the number of buffers dedicated to chain i is b_i ($b_i \geq 0$).

(b) the maximum number of buffers that chain i packets can occupy is B_i ($B_i \leq B$).

We must have $b_i \leq B_i$ for all i, $\sum_{i=1}^{M} b_i \leq B$, and $\sum_{i=1}^{M} B_i \geq B$.

It is often desirable to use an over-commitment strategy such that $\sum_{i=1}^{M} B_i > B$.

Let a state of the single node model be

$$\underline{m} = (m_1, m_2, \ldots, m_M)$$

where m_i is the number of chain i packets in the node*.

The general buffer alocation scheme can be modeled by a queueing network model with population size constraints. The set

---

* In a queueing network model, n_{ik} is the number of chain k packets at channel i. Since the single node model has the simple behavior that chain i packets only visit channel i, $n_{ik} = 0$ for $k \neq i$. n_{ii} is therefore the only non-zero element in the vector $\underline{n}_i$. We use m_i instead of $(0,\ldots,n_{ii},\ldots 0)$ for convenience.

of feasible states can be written as:

$$V = \left\{ \underline{m}: m_i \le B_i, \text{ all } i, \text{ and } \sum_{i=1}^{M} \max(0, m_i - b_i) \le B - \sum_{i=1}^{M} b_i \right\} \qquad (5.2)$$

Applying the product form solution, the equilibrium state probability is given by:

$$P(\underline{m}) = \frac{1}{G} \prod_{i=1}^{M} \rho_i^{m_i} \qquad (5.3)$$

where $\rho_i = \lambda_i/(\mu C_i)$ and

$$G = \sum_{\underline{m} \in V} \prod_{i=1}^{M} \rho_i^{m_i} \qquad (5.4)$$

Some special cases of the general scheme have been explicitly studied by Kamoun and Kleinrock [33]. They are:

(a) Complete partitioning ($b_i > 0$ and $\sum_{i=1}^{M} b_i = B$) -- the B buffers are partitioned into M groups, the i-th group (with size b_i) is allocated to chain i. There is no sharing of buffers among chains. Hence, B_i is equal to b_i, and we also have $\sum_{i=1}^{M} B_i = B$.

(b) Complete sharing ($b_i = 0$ and $B_i = B$) -- The B buffers are completely shared by the M chains. Buffers are allocated on a FCFS basis.

(c) Sharing with maximum allocation ($b_i = 0$ and $B_i < B$) -- the

B buffers are shared by all chains with the restriction that the number of buffers occupied by chain i cannot exceed B_i.

(d) Sharing with minimum allocations ($b_i > 0$ and $B_i = B$) -- chain i packets are guaranteed at least b_i buffers, and the remaining $B - \sum_{i=1}^{M} b_i$ buffers are shared by all chains.

We now illustrate how one can obtain analytic expressions for performance measures such as blocking probability, throughput and mean delay. Let $V_i \subset V$ be the set of states in which a chain i arrival is rejected. For the general buffer allocation scheme, V_i is given by:

$$V_i = \left\{ \underline{m}: m_i = B_i \quad \text{or} \quad \left(m_i \geq b_i \text{ and } \sum_{j=1}^{M} \max(0, m_j - b_j) = B - \sum_{j=1}^{M} b_j\right) \right\} \tag{5.5}$$

Let PB_i be the blocking probability of chain i packets.

$$PB_i = \sum_{\underline{m} \in V_i} P(\underline{m}) \tag{5.6}$$

The throughput of chain i packets is then given by:

$$\gamma_i^* = \lambda_i (1 - PB_i) \tag{5.7}$$

To get the mean delay experienced by chain i packets, we must first get the mean number of chain i packets in the node.

This can be obtained from:

$$E[m_i] = \sum_{\underline{m} \in V} m_i P(\underline{m}) \tag{5.8}$$

Little's formula [10] is then used to get the following expression for the mean delay of chain i:

$$T_i = E[m_i]/\gamma_i^* \tag{5.9}$$

It should be noted that the mean delay in eq.(5.9) is for packets which are accepted into the node. It does not include those that are rejected.

Eqs.(5.6) to (5.9) are expressed in terms of a summation over a set of network states. Very often, they can be simplified and expressed in terms of the normalization constant G. The reader is referred to reference [33] for more details. Numerical examples showing the relative merits of the four special cases are also provided in [33]. The general conclusion is that the best scheme and the best setting of parameters (b_i and B_i) depend upon the values of the ρ_i's.

5.3 Permit-Oriented Congestion Control

A packet switching network can be viewed as a set of resources shared by a population of users. Examples of such resources are channels and buffers. If the resources are not managed properly, an increased demand from a single user or a

group of users may cause degradation in network performance. This degradation is usually in the form of reduced throughput [37,38]. When this happens, the network is said to be in a state of congestion. The objective of congestion control is then to prevent the network from going into the congestion state. Congestion control schemes usually involve some form of restriction on the amount of network resources allocated to each external user. The window flow control mechanism discussed in section 4.1 is an example of congestion control because it places a limit on the number of packets belonging to a virtual channel in the network. However, if the number of virtual channels is large, the combined load can become excessive, and some additional congestion control may be required.

A basic congestion control technique is to apply control at the point of entry to the network. An example of such a technique is the isarithmic control scheme suggested by Davies [39]. This scheme places a limit on the total number of packets in the network, no discrimination is made on the basis of routing chains. It can be implemented by circulating a number of "permits" in the network, and requiring a packet to secure a permit before it can be admitted into the network. Another example is end-to-end window control which places a limit on the number of packets belonging to each source-destination node pair.

We observe that there are similarities between permit-oriented congestion control in a network and the buffer allocation problem in a switching node. The permits are analogous to

the buffers shared by the routing chains. Hence, the general scheme (based upon the concepts of maximum and minimum allocations) discussed in section 5.2 can be extended to permit-oriented congestion control for the whole network. In this section, we present the work reported in [40] which consider the case of maximum permit allocation only.

The routing chains in the network are assume to form disjoint groups, and the number of permits used by each group cannot exceed some pre-specified maximum. The notion of group allows us the flexibility of imposing control on selected sets of routing chains. The resulting congestion control scheme is essentially a two-level isarithmic control [40]. At level 1, a limit is placed on the total number of packets in the network; and at level 2, separate limits are placed on each group. Let L be the total number of permits, D be the number of groups, and L_u be the limit for group u, u = 1,2,...,D. The general scheme can be implemented by two types of permits. Type 1 consists of L permits corresponding to level 1 control. Type 2 permits are distinguishable by group number, and the number of permits for group u is L_u. A packet must acquire both type of permits before it can be admitted into the network.

It should be noted that the two types of permits are not always necessary. For example, the complete sharing scheme (i.e., $L_u = L$ for all u) is the same as Davies isarithmic control [39], and it can be implemented by the L type 1 permits only. Also, in the complete partitioning scheme (i.e., $L_1+L_2+...+L_D = L$),

type 1 permits are not required.

The implementation of permit-oriented congestion control is substantially more complicated than schemes for buffer allocation. A packet acquires one or more permits when it enters the network and releases its permit(s) when it reaches its destination node. The distribution of free permits is an important implementation issue. One would like to minimize the probability that when a permit is needed at a particular node, all the free permits are somewhere else in the network. Davies [39] suggested that each node may keep up to a maximum number of free permits, and extra permits are sent to randomly-selected neighbors.

We now illustrate how one can use a queueing network model with population size constraints to study permit-oriented congestion control. For reasons of mathematical tractability, we assume that the buffer space at each node is unlimited. We also assume that there is no delay in circulating the permits through the network, and the permits are allocated to packets from outside the network on a FCFS basis. In reality, a packet's entry to the network may be delayed because the free permits may be in other parts of the network. The last assumption therefore results in optimistic estimates of network performance.

Recall that the state of the network model is given by $S = (\underline{n}_1, \underline{n}_2, \ldots, \underline{n}_M)$ where $\underline{n}_i = (n_{i1}, n_{i2}, \ldots, n_{iK})$; n_{ik} is the number of chain k packets at channel i. The K routing chains are partitioned into D groups, and a packet is said to belong to group u (denoted by Γ_u) if its routing chain is in group u. For a

state S of the network, let $|S|_u$ be the number of group u packets and $|S|$ be the total number of packets in the network. A group u packet arriving from outside the network is assumed to be lost if $|S| = L$ or $|S|_u = L_u$. With this assumption, the set of feasible states is given by:

$$V = \left\{ S: \sum_{i=1}^{M} \sum_{k=1}^{K} n_{ik} \leq L \text{ and } \sum_{i=1}^{M} \sum_{k \in \Gamma_u} n_{ik} \leq L_u, \text{ all } u \right\} \tag{5.10}$$

and the equilibrium state probability is:

$$P(S) = \frac{1}{G} \prod_{i=1}^{M} n_i! \prod_{k=1}^{K} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!} \tag{5.11}$$

where

$$G = \sum_{S \in V} \prod_{i=1}^{M} n_i! \prod_{k=1}^{K} \frac{\rho_{ik}^{n_{ik}}}{n_{ik}!} \tag{5.12}$$

To obtain expressions for network throughput of each group, it is convenient to define a less detailed state description

$$S' = (\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_M)$$

where

$$\underline{y}_i = (m_{i1}, m_{i2}, \ldots, m_{iD})$$

where m_{iu} is the number of group u packets at channel i.

The set of feasible states is now:

$$V' = \left\{ S': \sum_{i=1}^{M} \sum_{u=1}^{D} m_{iu} \le L \text{ and } \sum_{i=1}^{M} m_{iu} \le L_u, \text{ all } u \right\} \tag{5.13}$$

By adding all state probabilities P(S) such that $\sum_{k \in \Gamma_u} n_{ik} = m_{iu}$ for $i = 1,2,\ldots,M$ and $u = 1,2,\ldots,D$, we get [40]:

$$P(S') = \frac{1}{G'} \prod_{i=1}^{M} m_i! \prod_{u=1}^{D} \frac{\phi_{iu}^{m_{iu}}}{m_{iu}!} \tag{5.14}$$

where $m_i = \sum_{u=1}^{D} m_{iu}$, $\phi_{iu} = \sum_{k \in \Gamma_u} \rho_{ik}$, and

$$G' = \sum_{S' \in V'} \prod_{i=1}^{M} m_i! \prod_{u=1}^{D} \frac{\phi_{iu}^{m_{iu}}}{m_{iu}!} \tag{5.15}$$

To obtain results for network throughput of each group, we follow the developments which lead to eq.(5.7) in section 5.2. The blocking probability of group u packets is:

$$PB_u = \sum_{S' \in V'_u} P(S') \tag{5.16}$$

where

$$V'_u = \left\{ S': \sum_{i=1}^{M} \sum_{u=1}^{D} m_{iu} = L \text{ or } \sum_{i=1}^{M} m_{iu} = L_u \right\} \tag{5.17}$$

The throughput of group u is then given by:

$$\gamma_u^* = \sum_{k \in \Gamma_u} \gamma_k (1-PB_u) \tag{5.18}$$

One problem with the analytic results in permit-oriented congestion control is that the amount of computation to get γ_u^* grows with $MDL_1L_2...L_D$. When D is large, numerical results are very difficult to obtain. One alternative is to use Reiser's heuristics [30] to obtain approximate results (see Section 4.4). The applicability of Reiser's technique is limited to the special case of the complete partitioning scheme only [41]. End-to-end congestion control is an example of such a scheme.

5.4 Finite-Buffer Network Model

The peak throughput of a network is attained when all its communication channels are transmitting packets (assuming that the mean packet length and the mean path lengths of packets are constant). The network throughput over a period of time may be less than the peak value because of (i) the lack of input traffic, or (ii) the constraints that force communication channels to be "nonproductive" part of the time. In sections 5.2 and 5.3 above, we addressed the throughput degradation behavior arising from interference between different streams of traffic. In this section, we consider the throughput degradaion in network throughput due to insufficient buffers at switching nodes.

The objective of the queueing networks to be introduced is

to determine nodal buffer requirements to maintain a high level of network throughput. The performance of the "input buffer limit" strategy is also studied.

To get around the difficulty of modeling blocking in a queueing network, the following approximate solution technique was proposed in [9]. The overall problem (network of switching nodes) is first decomposed into a set of analytically tractable problems, i.e., a queueing network model for each switching node. The single-node results are then "interfaced" by requiring that the various packet flows within the network are conserved.

The queueing network model of a switching node shown in fig. 6 was proposed by Lam and Schweitzer [9,35]. It was also employed later by Lam and Reiser [36]. FCFS servers are used to model the nodal processor, the communication channels and the sink for packets destined for this node. IS servers are used to model acknowledgement delays and time-out delays.

Two types of packets are distinguished*: transit packets forwarded by adjacent nodes and new input packets generated locally. They are represented by two routing chains with external arrival rates γ_1 and γ_2 respectively. Suppose the node has N_T buffers. Transit packets are rejected only when all buffers are occupied. However, when there are N_I input packets in the node, any newly arrived input packet is rejected. We have $N_I \leq N_T$. The ratio N_I/N_T is said to be the input buffer limit.

---

* Only two types of packets are considered to reduce the computational requirements of the model.

For each routing chain, routing transition probabilities from the nodal processor to one of the communication channels or the sink are determined by the routing assignment of the network. The routing transition probability from a communication channel to either the ACK or time-out IS server depends upon the rejection probability PB_j for transit packets of the adjacent node (say node j). It a packet is accepted by the adjacent node, it joins the ACK server and subsequently leaves the current node. If a packet is rejected by the adjacent node, no ACK will be returned; conceptually, the packet joins the time-out server and subsequently rejoins the channel queue. Note that such retransmissions need to be accounted for in determining the arrival rate γ_2 of transit packets offered to each switching node.

Suppose the set of nodal blocking probabilities $\{PB_j\}$ for transit packets is known for each communication channel in fig. 6 (for simplicity, we shall assume no random transmission errors without loss of generality). Then the node can be modeled by a product form queueing network with two routing chains and the population size constraints

$$0 \leq N_1 + N_2 \leq N_T$$

and

$$0 \leq N_1 \leq N_I$$

Let $p_m(N_1,N_2)$ be the equilibrium probability of having N_1 input packets and N_2 transit packets in node m. The equilibrium

blocking probability for transit packets at this node is

$$PB_m = \sum_{N_1=0}^{N_I} P_m(N_1, N_T - N_1) \qquad (5.19)$$

Since the set $\{P_m(N_1,N_2)\}$ also depends on $\{PB_j\}$, we have a set of non-linear equations which can be solved numerically for $\{PB_j\}$.

A numerical solution technique based on the Newton-Raphson method was developed [9] for the special case of $N_I = N_T$ at each node (i.e., no input buffer limit control). It was found that the model is accurate when switching nodes have adequate buffers (for given external input rates) so that $\{PB_j\}$ takes on small values. The model is thus useful for predicting buffer requirements to achieve small nodal blocking probabilities.

The above model was also employed in [36] to study the design of input buffer limits that can effectively prevent throughput degradation due to insufficient buffers when the network is under a heavy external load. Both the analytic results in [36] and a subsequent simulation study [42] showed that input buffer limits can be designed to provide a very effective congestion control mechanism for temporary network overloads. For a detailed treatment of input buffer limits as a congestion control mechanism, see [36,42]. A slightly different input buffer limit strategy was later considered by Saad and Schwartz [43] and independently by Kamoun, et.al. [44].

6. Concluding Remarks

In this paper, we have discussed in detail the application of product-form queueing networks to the performance analysis of store-and-forward packet-switching networks. The topics considered include optimal capacity assignment, optimal routing, distribution of end-to-end delay, fairness among routing chains, virtual channel with window flow control, buffer management in a switching node, and permit-oriented congestion control. Other survey papers on related subjects are also available in the literature. Examples of such papers are the modeling of computer communication networks [46], topological design of store-and-forward networks [47], and a comparative survey of flow control [48].

References

[1] J.R. Jackson, Jobshop-like Queueing Systems, Management Science 10 (1963) 131-142.

[2] F. Baskett, K.M. Chandy, R.R. Muntz and F. Palacios, Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, JACM 22 (1975) 248-260.

[3] M. Reiser and H. Kobayashi, Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms, IBM Journal of Research and Development 19 (1975).

[4] S.S. Lam, Queueing Networks with Population Size Constraints, IBM Journal of Research and Development 21 (1977) 370-378.

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

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

[7] L. Kleinrock, Queueing Systems, Volume 2: Computer Applications (Wiley-Interscience, New York, 1976).

[8] L. Kleinrock and W.E. Naylor, On Measured Behavior of the ARPA Network, AFIPS Conference Proc., National Computer Conference 43 (May 1974) 767-780.

[9] S.S. Lam, Store-and-Forward Buffer Requirements in a Packet Switching Network, IEEE Trans. on Communications COM-24 (1976) 394-403.

[10] J.C. Little, A Proof of the Queueing Formula: $L = \lambda W$, Operations Research 9 (1961) 383-387.

[11] M. Gerla, The Design of Store-and-Forward Networks for Computer Communications, Ph.D dissertation, Department of Computer Science, UCLA (Jan. 1973).

[12] J.M. McQuillan, I. Richer and E.C. Rosen, The New Routing Algorithm for the ARPANET, IEEE Trans. on Communications COM-28 (1980), 711-719.

[13] L. Fratta, M. Gerla and L. Kleinrock, The Flow Deviation Method: An Approach to Store-and-Forward Network Design, Networks 3 (1973) 97-133.

[14] R.R. Jueneman and G.S. Kerr, Explicit Path Routing in Communications Networks, Proc. International Conference on Computer Communication, Toronto (Aug. 1976) 340-342.

[15] V. Ahuja, Routing and Flow Control in Systems Network Architecture, IBM Systems Journal 18 (1979) 298-314.

[16] J.W. Wong, Distribution of End-to-End Delay in Message-Switched Networks, Computer Networks 2 (1978) 44-49.

[17] J.W. Wong, Distribution of End-to-End Delay in Packet Switching Networks -- Exact and Approximate Results, CCNG Report, University of Waterloo, to appear.

[18] L. Kleinrock, Queueing Systems, Volume 1: Theory (Wiley-Interscience, New York, 1975).

[19] R. Cooper, Introduction to Queueing Theory (MacMillan, New York, 1972).

[20] J.W. Wong, J.P. Sauve and J.A. Field, A Study of Fairness in Packet-Switching Networks, CCNG Report E-90, University of Waterloo (June 1980).

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

[22] Logica Ltd., Packet Switching Report, London (Sept. 1978) 152.

[23] IBM Corp., Systems Network Architecture General Information, GA27-3102-0 (Jan. 1975).

[24] H. Opderbeck and L. Kleinrock, The Influence of Control Procedures on the Performance of Packet-Switched Networks, Proc. National Telecommunications Conference, San Diego (Dec. 1974)

[25] L. Pouzin, Presentation and Major Design Aspects of the CYCLADES Computer Network, Proc. Third Data Communications Symposium, St. Petersburg, Florida (Nov. 1973)

[26] V. Cerf and R. Kahn, A Protocol for Packet Network Intercommunication, IEEE Trans. on Communications, COM-22 (1974) 637-648.

[27] S.S. Lam, Dynamic Scaling and Growth Behavior of Queueing Network Normalization Constants, Technical Report TR-148, Department of Computer Science, University of Texas at Austin (June 1980), to appear in JACM.

[28] J.P. Buzen, Computational Algorithms for Closed Queueing Networks with Exponential Servers, CACM 16 (1973) 527-531.

[29] M. Pennotti and M. Schwartz, Congestion Control in Store and Forward Tandem Links, IEEE Trans. on Communications COM-23 (1975) 1434-1443.

[30] M. Reiser, A Queueing Network Analysis of Computer Communication Networks with Window Flow Control, IEEE Trans. on Communications COM-27 (1979) 1199-1209.

[31] M. Reiser and S. Lavenberg, Mean Value Analysis of Closed Multichain Queueing Networks, JACM 27 (1980) 313-322.

[32] M. Irland, Buffer Management in a Packet Switch, IEEE Trans. on Communications COM-26 (1978) 328-337.

[33] F. Kamoun and L. Kleinrock, Analysis of Shared Finite Storage in a Computer Network Environment under General Traffic Conditions, IEEE Trans. on Communications, COM-28 (1980) 992-1003.

[34] M.A. Rich and M. Schwartz, Buffer Sharing in Computer-Communication Network Nodes, IEEE Trans. on Communications, COM-25 (1977) 958-970.

[35] P.J. Schweitzer and S.S. Lam, Buffer Overflow in a Store-and-Forward Network Node, IBM Research Report RC5759 (Dec. 1975).

[36] S.S. Lam and M. Reiser, Congestion Control of Store-and-Forward Networks by Input Buffer Limits, IEEE Trans. on Communications COM-27 (1979), 127-134.

[37] W.L. Price, Data Network Simulation Experiments at the National Physical Laboratory, Computer Networks 1 (1977)

[38] A. Giessler, J. Haenle, A. Koenig and E. Pade, Packet Networks with Deadlock-Free Buffer Allocation -- An Investigation by Simulation, Computer Networks 2 (1978) 191-208.

[39] D. Davies, The Control of Congestion in Packet Switching Networks, IEEE Trans. on Communications COM-23 (1975) 546-550.

[40] J.W. Wong and M.S. Unsoy, Analysis of Flow Control in Switched Data Networks, Proc. IFIP Congress 77, Toronto (Aug. 1977) 315-320.

[41] M.S. Unsoy, Credit-Oriented Congestion Control in Two-Level Hierarchical Packet Switching Networks, Ph.D Thesis, University of Waterloo, 1980.

[42] S.S. Lam and Y.L Lien, An Experimental Study of the Congestion Control of Packet Communication Networks, Proc. Fifth International Conference on Computer Communication, Atlanta (Oct. 1980) 791-796.

[43] S. Saad and M. Schwartz, Input Buffer Limiting Mechanisms for Congestion Control, ICC'80 Conference Record, Seattle (June 1980) 23.1.1-23.1.5.

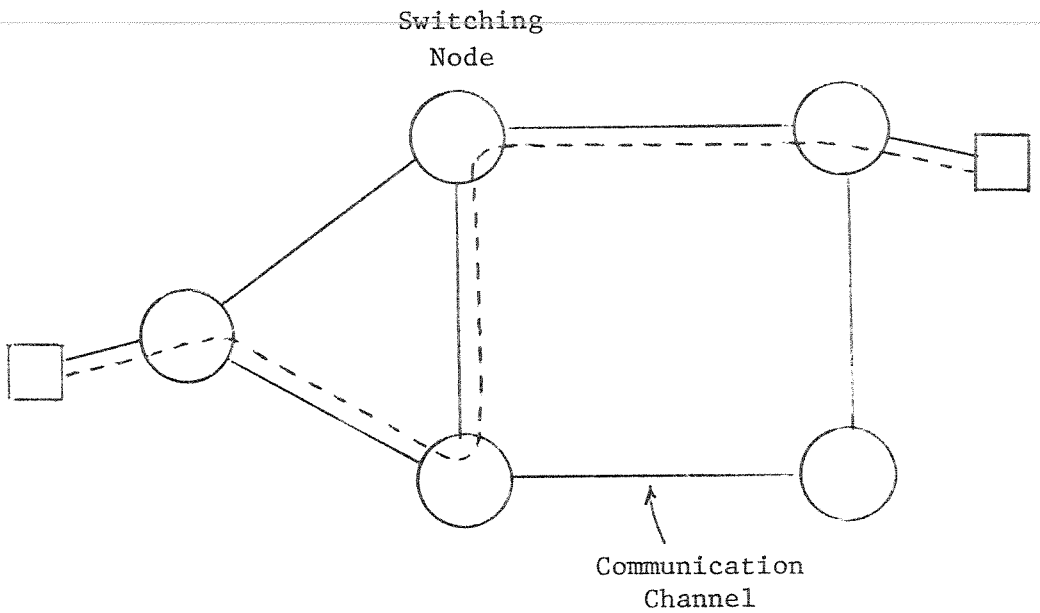
[44] F. Kamoun, A. Belguith and J.L. Grange, Congestion Control with A Buffer Management Strategy Based on Traffic Priorities, Proc. Fifth International Conference on Computer Communication, Atlanta (Oct. 1980) 845-850.

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

[46] J.W. Wong, Queueing Network Modeling of Computer Communication Networks, ACM Computing Surveys 10 (1978) 343-352.

[47] M. Gerla and L. Kleinrock, On the Topological Design of Distributed Computer Networks, IEEE Trans. on Communications COM-25 (1977) 48-60.

[48] M. Gerla and L. Kleinrock, Flow Control: A Comparative Survey, IEEE Trans. on Communications COM-28 (1980) 553-574.



data source/sink

source-destination path

Figure 1. Store-and-Forward Packet-Switching Network

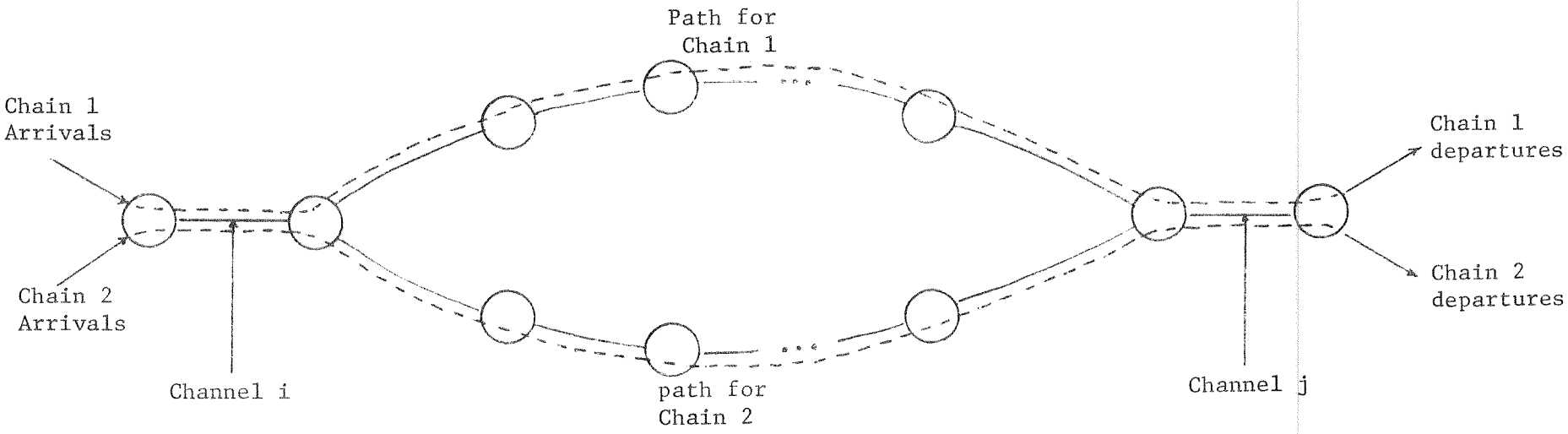


Figure 2. Dependency between t_k and a_k

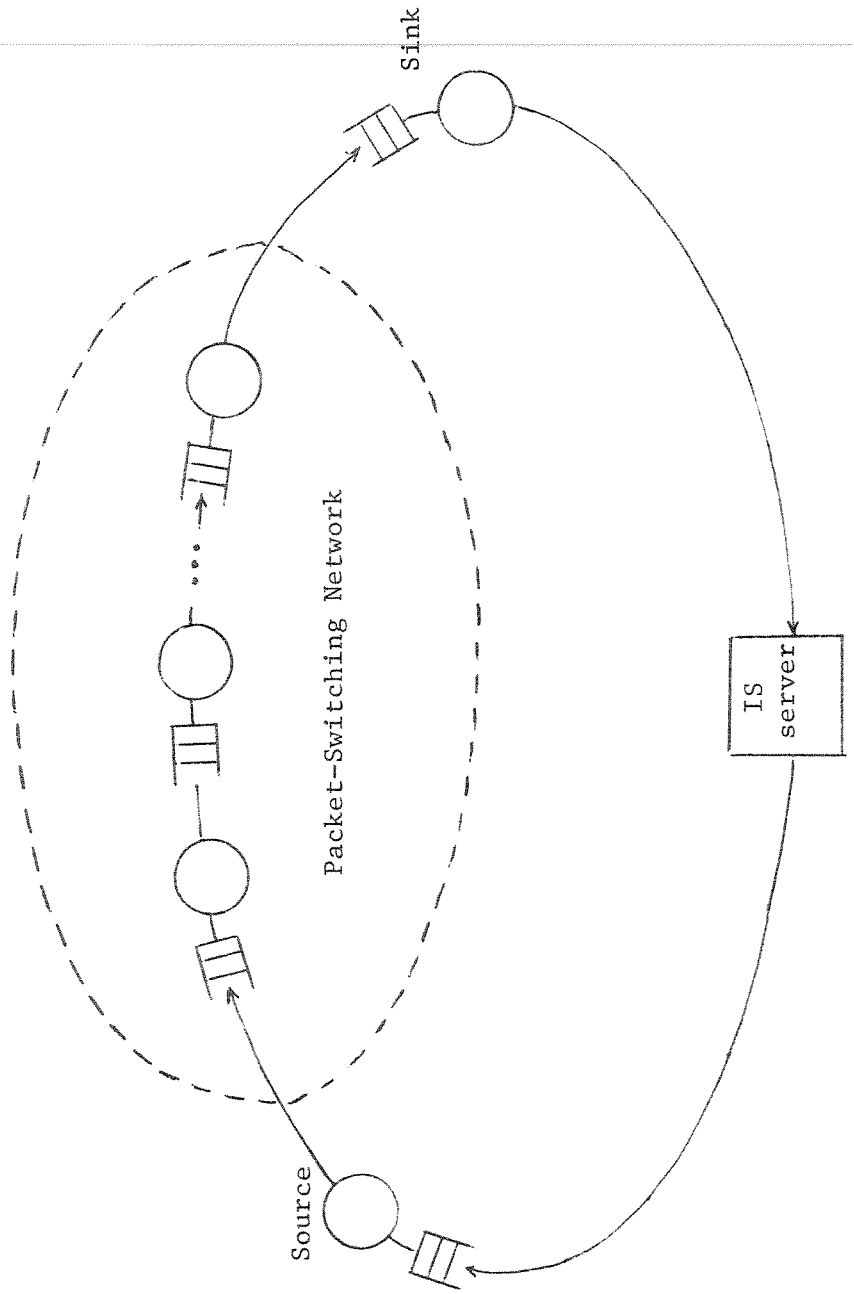
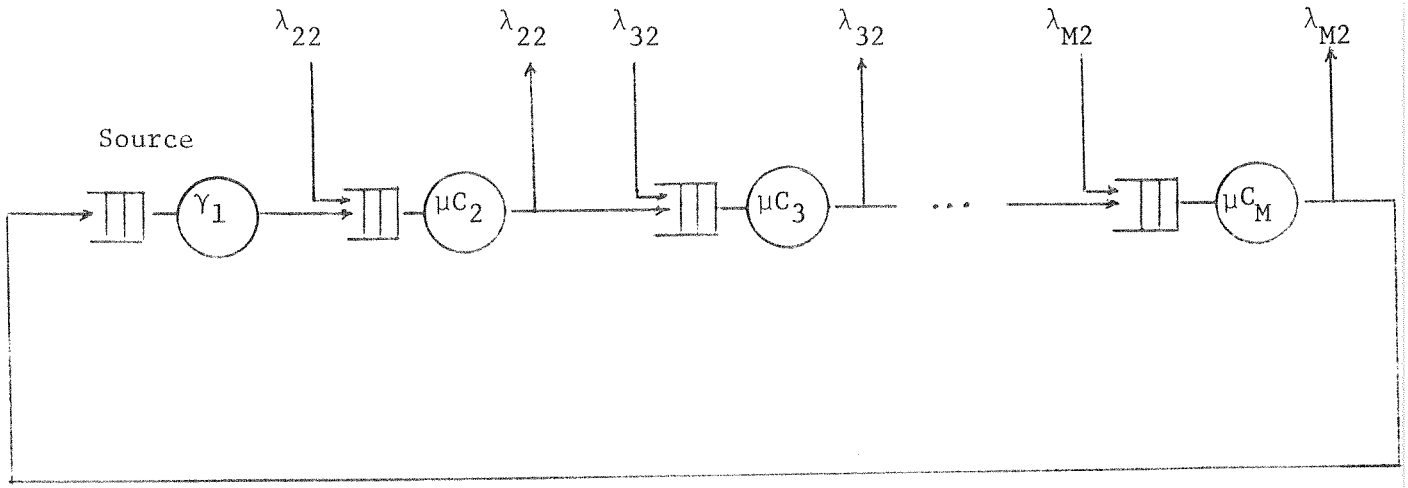


Figure 3. Modeling a Virtual Channel with a Closed Chain



W cycling Chain 1 packets

Figure 4. Model of a Single Virtual Channel

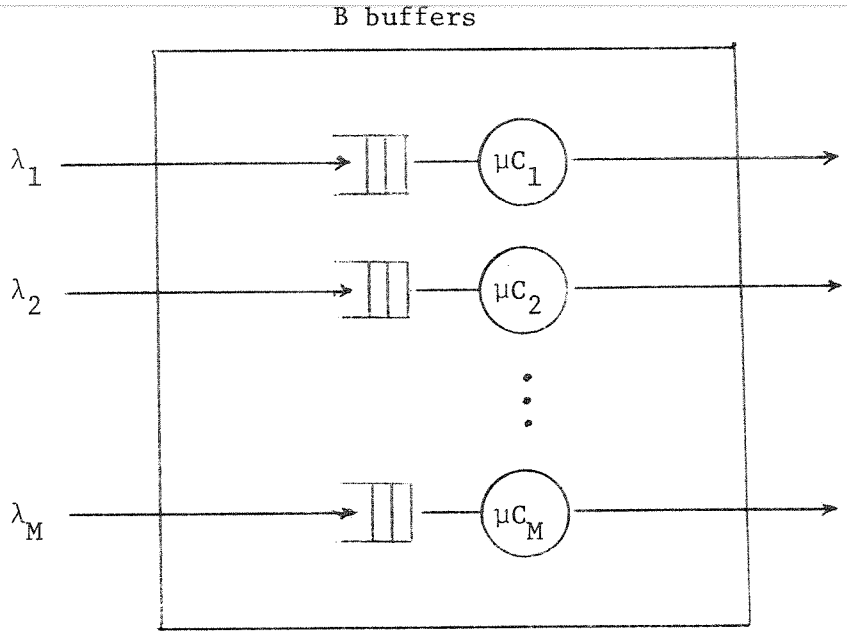


Figure 5. Single Node Model with Finite Buffers

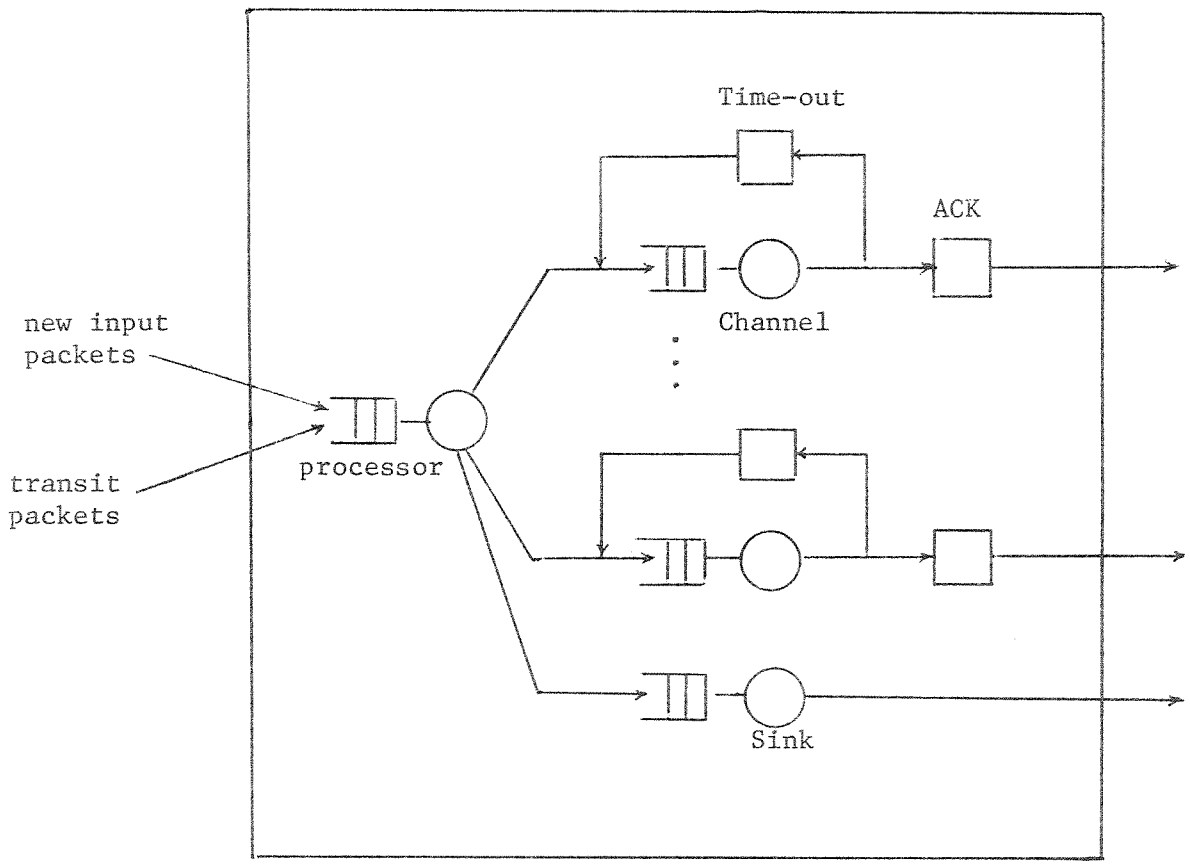


Figure 6. Single Node Model with ACK and Time-out