OPTIMAL ROUTING IN NETWORKS WITH FLOW-CONTROLLED VIRTUAL CHANNELS*

Simon S. Lam

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

and

Y. Luke Lien

IBM T. J. Watson Research Center
Yorktown Heights, New York 10598

Abstract

Packet switching networks with flow-controlled virtual channels are naturally modeled as queueing networks with closed chains. Available network design and analysis techniques, however, are mostly based upon an open-chain queueing network model. In this paper, we first examine the traffic conditions under which an open-chain model accurately predicts the mean end-to-end delays of a closed-chain model having the same chain throughputs. We next consider the problem of optimally routing a small amount of incremental traffic corresponding to the addition of a new virtual channel (with a window size of one) to a network. We model the new virtual channel as a closed chain. Existing flows in the network are modeled as open chains. An optimal routing algorithm is then presented. The algorithm solves a constrained optimization problem that is a compromise between problems of unconstrained individual-optimization and unconstrained network-optimization.

1. INTRODUCTION

The early store-and-forward packet switching networks are mostly datagram networks. In these networks, each packet carries its own source-destination addresses. It is treated as an independent entity with regard to its acceptance into the network and subsequent movement through the network. The current generation of packet switching networks, however, are mostly virtual channel networks [ROBE 78]. In these networks, packets are associated with logical source-destination connections called virtual (or logical) channels. Each packet is identified by its virtual channel ID. Among other attributes, virtual channels are individually end-to-end flow-controlled. Examples of flow controls are SNA pacing [IBM 75], RFNM in ARPANET [OPDE 74] and various window mechanisms [POUZ 73, CERF 74]. All of them work by limiting the number of packets that a virtual channel can have in transit within the network. (This number will be referred to as the virtual channel window size.) An important function of end-to-end flow controls is the synchronization of the data source input rate to the data sink acceptance rate. They also provide, to some extent, a form of congestion control capability for the network.

We will not dwell upon the details and relative merits of datagram and virtual channel networks [ROBE 78]. Our main interest here is on models for network performance analysis and design. Datagram networks are naturally modeled as open-chain queueing networks given the independence assumption of Kleinrock [KLEI 64]. Such a model forms the basis of extensive studies on the design and analysis of store-and-forward packet switching networks [KLEI 64, KLEI 76, GERL 77, SCHW 77, GALL 77].

Packet switching networks with flow-controlled virtual channels, on the other hand, are naturally modeled as queueing networks with closed routing chains [BASK 75]; each closed chain represents a flow-controlled virtual channel and the chain population size is equal to the virtual channel window size [REIS 79, LAM 82]. In practice, virtual channel networks are becoming the dominant form of networks. However, available tools for network analysis and design are still mainly based upon open-chain queueing network models. A serious drawback of closed-chain models is the large computational time and space needed to calculate network performance measures (chain throughputs and mean end-to-end delays). Some progress has been made recently to reduce these computational requirements and the solution of networks with many virtual channels is feasible [LAM 81a, LAM 81b]. Nevertheless, the computational requriements remain substantially more than those of open-chain models.

We propose an approach to incorporate the behavior of flow-controlled virtual channels in network design and optimization tools using a combination of both closed-chain and open-chain models. A closed-chain model is first solved to provide chain throughputs and mean end-to-end delays. Given chain throughputs, an open-chain model is then employed for a sequence of intermediate optimization steps (such as, for example, the routing of incremental flows to be considered later on in this paper). To avoid the accumulation of errors, the closed-chain model is applied at

* This work was supported by National Science Foundation Grant No. ECS78-01803.

various checkpoints of the optimization procedure to re-calculate chain throughputs and mean delays.

Two related problems are investigated in this paper. First, we examine the traffic conditions under which an open-chain model accurately predicts the mean end-to-end delays of a closed-chain model having the same chain throughputs. We found that in general the approximation is fairly accurate when communication channels in the network are not heavily utilized. Second, we consider the problem of optimally routing a small amount of incremental traffic corresponding to the addition of a new virtual channel (with a window size of one) to a network. We model the new virtual channel as a closed chain. Existing flows, on the other hand, are modeled as open chains. An optimal routing algorithm to solve a constrained optimization problem is presented.

Our optimization problem is similar to the classical flow deviation problem [FRAT 73, GERL 73, KLEI 76, GALL 77] in that the objective is to minimize the impact of a small amount of incremental flow on the mean transit delay of all packets in the network. But unlike flow deviation the incremental flow in our model is not infinitesimal, and our optimization is constrained by a bound on the mean end-to-end delay of the new virtual channel.

The balance of this paper is organized as follows. In Section 2, open-chain and closed-chain queueing network models are described. The accuracy of approximating a closed-chain model by an open-chain model is then examined. In Section 3, the optimal routing problem is formulated. In Section 4, an optimal routing algorithm is presented.

2. OPEN-CHAIN AND CLOSED-CHAIN MODELS

In a packet switching network, communication channels and nodal processors can be modeled as FIFO queues with exponentially distributed service times given the independence assumption of Kleinrock [KLEI 64]. We shall assume that the packet switching network has sufficient buffers so that blocking due to buffer overflow has negligible probability. (The problem of buffer requirements and loss probabilities have been considered in [LAM 76].)

Suppose that there are K uni-directional virtual channels between pairs of nodes. Packets in the same virtual channel follow a fixed route (which may be chosen probabilistically from a finite set of routes between source and sink).

Open-chain model

A model in which each virtual channel is represented as an open chain assumes that the external packet arrivals to the source node of a virtual channel constitute a Poisson process at a known constant rate. (An open-chain model also assumes that the number of packets belonging to a virtual channel that are travelling within the network is not limited.) Let the rate of all packet arrivals to server m be λ_m packets/second. The work rate of server m is C_m bits/second and the average length of a packet is $1/\mu$ bits. The traffic intensity of server m is defined to be $\rho_m = \lambda_m/(\mu C_m)$. Given that $\rho_m < 1$ for all m, the throughput of each virtual channel is the same as its external input rate and the mean end-to-end delay of its packets is the sum of mean delays of the servers along its route. The mean delay of server m is given by the M/M/1 mean delay formula and is equal to $1/(\mu C_m - \lambda_m)$. Thus, both the throughputs and mean end-to-end delays of virtual channels can be obtained very easily for an open-chain model.

Closed-chain model

The flow-control window size of a virtual channel limits the maximum number of packets that it can have in transit within the communication network at the same time. Let there be K virtual channels and N_k denote the window size of virtual channel k, for $k = 1,2,\ldots,K$. We model the handling of external arrivals before they are admitted into the network by an additional FIFO server that works at a rate of γ_k packets/second. (See Figure 1.) If the number of packets in transit within a virtual channel is equal to its window size, then the source server is "blocked." A blocked source server is later unblocked when an end-to-end acknowledgment returns from the sink indicating receipt of a packet. It is assumed that the queue of external arrivals waiting to enter the network is never empty. Thus, the actual input rate of virtual channel k is determined by γ_k and the fraction of time its source server is unblocked.

The blocking behavior is naturally modeled in a queueing network by a closed chain with a fixed number of circulating customers. Each customer corresponds to an "access token." Initially, N_k tokens are placed at the source server of virtual channel k. Each packet admitted into the network carries a token with it. When there is no more token at the source server, it is blocked. A packet arriving at the sink node of the virtual channel releases its token which is then carried back by the end-to-end acknowledgment to the source server to be reused again. Thus, the N_k circulating tokens of a virtual channel correspond to N_k circulating customers of a closed chain.

We model the delay incurred by the return of an end-to-end acknowledgment from the sink to the source by an infinite-server (IS) service center [REIS 79, LAM 82]; the distribution of such random delays may be different for different virtual channels. It is not really important to model the route of the acknowledgments explicitly because these acknowledgements typically either are piggybacked in data packets, or if sent separately, are very short. Thus, they consume relatively small amounts of buffer and channel resources in the network, which may be accounted for separately in a straightforward manner.

To solve for the performance measures (virtual channel throughputs and mean end-to-end delays) of a closed-chain model, the computational time and space requirements of both the (sequential) convolution algorithm [REIS 75] and the MVA algorithm [REIS 80] grow exponentially with K; specifically, they are proportional to $\prod_{k=1}^{K} (N_k+1)$. These requirements are thus beyond the limits of present computers when network models with 10 or more virtual channels are considered.

The tree convolution algorithm, developed by these authors [LAM 81a, LAM 81b], is intended for the solution of networks in which chains do not visit all servers in the network. In models of communication networks and distributed systems, it is often true that chains visit only a small fraction of all queues in the network (sparseness property). Furthermore, chains are often clustered in certain parts of the network and their routes are constrained by the network topology (locality property). By making use of the routing information of chains, the time and space requirements of the tree convolution algorithm can be made substantially less than those of the (sequential) convolution and MVA algorithms. The number of closed chains that can be handled varies depending upon the extent of sparseness and locality present in their routes. We have solved numerically many network examples with 32-50 routing chains. In some extreme cases, the solution of networks with up to 100 routing chains has been found to be possible [LIEN 81].

Network design using both open-chain and closed-chain models

The large computational requirements of closed-chain models make them unattractive for use in network design procedures. Both Pennotti and Schwartz [PENN 75] and Gerla and Nilsson [GERL 80] suggested the use of open chains to approximately model flow-controlled virtual channels. The difficulty encountered is that the throughputs of the flow-controlled virtual channels needed as input parameters of an open-chain model are not known. We propose to use a combination of both open-chain and closed-chain models in design procedures for networks with flow-controlled virtual channels. A closed-chain model is first employed and chain throughputs and mean delays are computed exactly using the tree convolution algorithm. An open-chain model with the same chain throughputs is then employed for a sequence of intermediate optimization steps in the network design procedure (e.g., the routing of incremental flows to be studied below). To avoid the accumulation of errors, the closed-chain model is employed at various checkpoints of the design procedure to recompute chain throughputs and mean delays.

Traffic conditions under which an open-chain model is applicable

With the tree convolution algorithm we can solve models with many closed chains. Suppose that the throughputs of flow-controlled virtual channels in a network are first computed using a closed-chain model and an open-chain model with the same throughputs is specified. The mean end-to-end delays predicted by the open-chain model can then be compared with those given by the closed-chain model. The results of such a comparative study for a network example with 64 communication channels and 32 virtual channels are next presented.

In the network example, the source server work-rate is assumed to be $\gamma = 1$ packet/second for all chains. The service rate of each communication channel is assumed to be $\mu C = 10$ packets/second. The mean end-to-end acknowledgment delay for virtual channel k is assumed to be $h_k/\mu C$, where h_k is the number of communication channels in the route of virtual channel k. The virtual channel window size is $N_k = 3$ for all k. The average utilization of the 64 communication channels is 0.185, with a maximum utilization of 0.281 and a minimum utilization of 0.082 and a standard deviation of 0.066. Table 1 shows the throughputs, mean delays and delay estimates of the 32 virtual channels. The mean delays are given by the closed-chain model. The delay estimates are mean delays given by the open-chain model. The percentage errors in the delay estimates are quite small in this case.

We next proceed to investigate the effect of varying the relative source and channel speeds, γ and μC. We vary γ from 10 to 0.5 while keeping μC and all the other parameters constant. The results are shown in Table 2. Note that the channel utilizations are highest at $\gamma = 10$ and lowest at $\gamma = 0.5$. The accuracy of the open-chain model is very poor for $\gamma = 10$ (same value as μC) and improves as γ decreases.

Observe that when $\gamma << \mu C$, the source server is the "bottleneck" in each routing chain; thus, it behaves like a Poisson source at rate γ much of the time (i.e., like an open chain). However, when γ approached μC in magnitude, the open-chain model gave very large errors. This behavior may be explained as follows. When γ is almost the same as μC, the bottleneck in each routing chain is at one of the communication channels within the network. Note that when the utilization of an M/M/1 queue is high, its delay distribution has a long tail, which gives rise to a poor estimate of the delay in a closed-chain model where the queue lengths are bounded.

The effect of varying the virtual channel window size was also investigated. Window sizes of 2, 3 and 4 were considered. It was found that as the window size was increased, the accuracy of the open-chain model improved, despite increases in the channel utilizations.

We also considered the effect of routing. In general, we found that the accuracy of the open-chain model suffers from the presence of highly utilized servers within the packet switching network, either due to poor routing or due to a high-level of input traffic (large γ).

It was also found that in almost all cases considered, the delay estimates of the open-chain model were larger than the mean delays of the closed-chain model. There are two possible reasons for the overestimates. First, the delay estimates are obtained from M/M/1 delay distributions that have long tails. Second, the mean-value analysis shows that the mean delay encountered by a closed-chain customer is determined by the mean queue lengths of a network with that customer removed [REIS 80]. The open-chain model as described above does not account for this behavior. A consequence of the overestimation of delays is that the impact of bottlenecks on chain delays in an open-chain model is exaggerated compared to that in a closed-chain model. This means that if an open-chain model is used for the routing of incremental flows (see the following section), bottlenecks will be avoided more "rigorously" than if a closed-chain model is employed.

3. OPTIMAL ROUTING OF INCREMENTAL FLOWS

We consider the problem of introducing a small amount of incremental flow from a source node to a destination node into a network with existing flows. Several optimal routing problems may be formulated depending upon the nature of the incremental flow and the optimization objective. We next review the underlying objectives of ARPANET routing and flow deviation in this context. A new optimal routing problem is then formulated.

The objective of the ARPANET routing algorithm [MCQU 78] is to minimize the (estimated) delay of an individual packet from its source node to its destination node. Let t_m be the (estimated) delay of communication channel m. The optimal route is given by the path Q for which $\sum_{m \in Q} t_m$ is minimized over all paths from the given source node to the given destination node. In other words the (estimated) communication channel delays constitute the distance metric for shortest path routing. We observe that in ARPANET routing, the incremental flow is an individual packet and the individual-optimization objective is pursued.

It has been observed by several authors [AGNE 76, GALL 77] that routing algorithms with the objective of individual-optimization do not necessarily lead to network-optimization, i.e., minimizing the mean delay of all packets in the network. The flow deviation method [FRAT 73, GERL 73, KLEI 76] considers an incremental flow that is infinitesimal relative to existing flows in the network. The network-optimization objective is pursued; specifically, the route for the incremental flow is chosen to minimize the (infinitesimal) increase ΔT in the mean network transit delay T of all packets. Let f_m denote the flow in communication channel m (in bits per second) and d_m denote the value of the partial derivative of T with respect to f_m evaluated at the existing flow value. It was shown that the optimal route for the incremental flow is given by the shortest path using $\{d_m\}$ as the distance metric.

We next pose a similar problem for networks with flow-controlled virtual channels. The incremental flow corresponds to the addition of a new virtual channel with a window size of one (not necessarily an infinitesimal amount of flow). The network-optimization objective is first considered.

One method to evaluate ΔT is to calculate T using the tree convolution algorithm for the network both with and without the additional virtual channel (given a specific route for it). However, to determine the optimal route with this approach would require numerous applications of the tree convolution algorithm and would be very expensive in terms of computation time.

We shall adopt the solution approach proposed in Section 2. A closed-chain model is initially used to calculate the throughputs of the existing flow-controlled virtual channels. These are then modeled as open chains. The new virtual channel to be added is modeled as a closed chain.

Let the aggregate arrival rate of the existing traffic in the network to communication channel m be denoted by λ_m packets/second. The service rate of channel m is μC_m packets/second where $1/\mu$ is the average length of a packet in bits and C_m is the channel speed in bits/second. Define $\rho_m = \lambda_m/(\mu C_m)$. The total throughput rate at which open chain packets leave (or enter) the network is γ_o packets/second. The closed chain representing the virtual channel being added has a population size of one (i.e. window size is one), a source server work-rate of γ packets/second and a mean end-to-end acknowledgment delay of τ seconds. The source and sink nodes of the virtual channel are known but its route is to be determined.

Let Q denote the set of communication channels constituting a route chosen for the new virtual channel. From the arrival theorem [SEVC 79, LAVE 80], the mean delay encountered by the new virtual channel's packet at channel $m \in Q$ is $1/(\mu C_m - \lambda_m)$. The mean network transit delay of the new virtual channel is

$$T_c = \sum_{m \in Q} \frac{1}{\mu C_m - \lambda_m} \tag{1}$$

Applying Little's formula [LITT 61], the throughput rate of the new virtual channel is

$$\gamma_c = \frac{1}{(1/\gamma) + \tau + T_c} \tag{2}$$

Let T_o be the mean network transit delay and $\bar{n}_o$ be the mean number of packets in the network before the addition of the new virtual channel. The increase in the mean delay due to the new virtual channel is

$$\Delta T = \frac{\sum_{m \in Q} \Delta\bar{n}_{m,o} + \bar{n}_o + \gamma_c T_c}{\gamma_o + \gamma_c} - T_o \tag{3}$$

where $\Delta\bar{n}_{m,o}$ is the increase in the mean queue length of the open chains at channel m due to the new virtual channel, and is given by

$$\Delta\bar{n}_{m,o} = \frac{\lambda_m}{\mu C_m - \lambda_m} \bar{n}_{m,c} \tag{4}$$

where $\bar{n}_{m,c}$ is the mean number of new packets (belonging to the added virtual channel) at channel m. A derivation of (4) is given in [PENN 75]. It may also be obtained by differentiating the moment generating function of the product-form solution for networks with both open and closed chains [REIS 75, LAM 82]. An application of Little's formula yields

$$\bar{n}_{m,c} = \gamma_c/(\mu C_m - \lambda_m). \tag{5}$$

Finally, we have

$$\Delta T = \frac{\sum_{m \in Q} \frac{\lambda_m \gamma_c}{(\mu C_m - \lambda_m)^2} + \gamma_c T_c - \gamma_c T_o}{\gamma_o + \gamma_c}$$

$$= \frac{\sum_{m \in Q} \left[\frac{\lambda_m}{(\mu C_m - \lambda_m)^2} + \frac{1}{\mu C_m - \lambda_m} \right] - T_o}{(\gamma_o/\gamma_c) + 1}$$

$$= \frac{\sum_{m\varepsilon Q} \frac{\mu C_m}{(\mu C_m - \lambda_m)^2} - T_o}{\gamma_o(\frac{1}{\gamma} + \tau + \sum_{m\varepsilon Q} \frac{1}{\mu C_m - \lambda_m}) + 1} \quad (6)$$

To minimize ΔT, the route should be chosen to try to minimize the numerator and to maximize the denominator if possible. Minimizing the numerator implies the choice of a shortest path from source to destination using $\mu C_m/(\mu C_m - \lambda_m)2$ as the distance metric. Note that this is essentially the same as the distance metric of

$$\frac{C_m}{(C_m - f_m)^2} \qquad \text{where } f_m = \lambda_m/\mu \quad (7)$$

given by the flow deviation method for an open-chain model. This similarity is interesting since the derivation of (6) above and the derivation of (7) in the flow deviation method are based upon different models.

Maximizing the denominator, on the other hand, implies that the longest path should be chosen with $1/(\mu C_m - \lambda_m)$ as the distance metric. Note that the incremental flow γ_c is much smaller than the existing network throughput γ_o in the objective function ΔT in (6). To minimize ΔT, a route may possibly be selected with a very long delay for the incremental flow. Since we are considering an amount of incremental traffic that is not infinitesimal, it makes sense to impose a maximum delay bound τ_{max} on the mean delay of the new virtual channel. (Most likely, the user requesting for the new virtual channel will want his mean delay to be bounded.) Hence, we formulate the following constrained optimization problem:

$$\underset{Q}{\text{Min}}\ \Delta T \qquad \text{subject to } \sum_{m\varepsilon Q} \frac{1}{\mu C_m - \lambda_m} < \tau_{max}. \quad (8)$$

A dual of the above problem is

$$\underset{Q}{\text{Min}} \sum_{m\varepsilon Q} \frac{1}{\mu C_m - \lambda_m} \qquad \text{subject to } \Delta T < \Delta_{max} \quad (9)$$

where Δ_{max} is a bound on ΔT. An algorithm to solve the problem in (8) is presented in Section 4.

We can interpret the constrained problem in (8) or (9) as a compromise between the objectives of individual-optimization and network-optimization. Note that the individual-optimization objective of ARPANET routing does not consider the impact of the incremental flow on the network. On the other hand, the network-optimization objective of flow deviation ignores the performance of the incremental flow (since it is assumed to be infinitesimal). The constrained problems in (8) and (9) take into account both considerations.

Let us reexamine the mean end-to-end acknowledgment delay which has been assumed to be a constant τ. This corresponds to the assumption that virtual channels (in opposite directions) between any two nodes employ routes that are chosen independently. A different but equally plausible assumption that one can make is that flows in the network are symmetric and virtual channels between any two nodes employ the same route (traversed in opposite directions). In this case, τ in Eqs. (2) and (6) should be changed to T_c.

4. AN OPTIMAL ROUTING ALGORITHM

Our algorithm to solve the constrained problem in (8) is based upon a branch-and-bound technique.

Consider the packet switching network as a directed graph described by (V,E) where V is a set of vertices (network nodes) and E is a set of directed arcs (communication channels). A path in the network is a sequence of <u>distinct</u> nodes $Q = v_0, v_1, \ldots, v_n$ such that (v_i, v_{i+1}) is an arc in E for $i = 0,1,\ldots,n-1$. We shall only consider acyclic paths.

Let v_s be the source node and v_d be the destination node of the virtual channel for which a path (or route) is desired. We shall consider only those paths that originate at v_s and either end at v_d or do not contain v_d. A path is said to be <u>complete</u> if it ends at v_d. A path is said to be <u>incomplete</u> if it does not contain v_d.

Each complete path Q from v_s to v_d is associated with two measures:

(i) its cost COST(Q) given by Eq. (6), and
(ii) its mean end-to-end delay DELAY(Q) given by

$$\text{DELAY}(Q) = \sum_{m\varepsilon Q} \frac{1}{\mu C_m - \lambda_m} \quad (10)$$

Each incomplete path Q is also associated with two measures:

(i) its estimated cost ECOST(Q) given by

$$\text{ECOST}(Q) = \frac{\sum_{m\varepsilon Q} \frac{\mu C_m}{(\mu C_m - \lambda_m)^2} - T_o}{\gamma_o\ (\frac{1}{\gamma} + \tau + \tau_{max}) + 1} \quad (11)$$

(ii) its mean end-to-end delay DELAY(Q) given by Eq. (10).

If the network is assumed to have symmetric flows and symmetric routes so that τ is replaced by T_c in (2) and (6), then τ should be replaced by τ_{max} in (11).

The following data structures are used in the algorithm below:

C_PATHS the set of complete paths constructed

I_PATHS the set of incomplete paths constructed

R_NODES the set of nodes that have not been visited by an incomplete path.

Given a path $Q = v_0, v_1, \ldots, v_n$, it is said to be <u>extendable</u> to v_{n+1} to form a new path $v_0, v_1, \ldots, v_n, v_{n+1}$ if v_{n+1} is not already in the path, a communication channel exists from v_n to v_{n+1}, and DELAY of the extended path is less than τ_{max}.

Algorithm {to find an optimal path}

begin

use a shortest path algorithm to find a minimum delay path Q' from v_s to v_d;

if DELAY(Q') $\geq \tau_{max}$ then

quit {comment: no feasible solution exists};

initialize C_PATHS to be the empty set, I_PATHS to contain the path consisting of v_s only, and R_NODES to be V - $\{v_s\}$;

repeat

Consider all paths that can be formed by extending a path in I_PATHS to a node in R_NODES and select Q in I_PATHS that is extendable to v in R_NODES to form Q_{new} such that ECOST(Q_{new}) is minimized;

if Q_{new} is incomplete then

begin

add Q_{new} to I_PATHS;

delete v from R_NODES

end

else {comment: Q_{new} is complete and v is v_d}

begin

add Q_{new} to C_PATHS;

label Q in I_PATHS to be nonextendable to v_d

end

until one of the following conditions is true or both are true:

1. Q_{new} is complete and COST of Q_{new} is less than COST of any path in C_PATHS and is less than ECOST of any extendable path in I_PATHS {comment: Q_{new} is the optimal solution}
2. no more path in I_PATHS is extendable to a node in R_NODES {comment: the path in C_PATHS with minimum COST is the optimal solution}

end

Theorem. If a feasible solution exists, the algorithm terminates with an optimal path Q* for the problem in (8).

We provide only a proof outline of the above theorem. If a feasible solution exists, termination of the algorithm is due to the assumption of a finite graph with a finite number of acyclic paths from v_s to v_d. The optimality of the path Q* is guaranteed by the termination conditions of the algorithm. It is sufficient to show that Q* is better than all complete paths that may be extended from the paths in I_PATHS. Consider Q in I_PATHS. Suppose that it is extendable by channel f to a new path Q'. Consider two cases. First, Q' is a complete path. We have

$$\mathrm{COST}(Q') = \frac{\sum_{m\varepsilon Q} \frac{\mu C_m}{(\mu C_m - \lambda_m)^2} + \frac{\mu C_f}{(\mu C_f - \lambda_f)^2} - T_o}{\gamma_o(\frac{1}{\gamma} + \tau + \sum_{m\varepsilon Q'} \frac{1}{\mu C_m - \lambda_m}) + 1}$$

$$> \frac{\sum_{m\varepsilon Q} \frac{\mu C_m}{(\mu C_m - \lambda_m)^2} - T_o}{\gamma_o(\frac{1}{\gamma} + \tau + \tau_{max}) + 1} = \mathrm{ECOST}(Q) > \mathrm{COST}(Q^*)$$

The second case is that Q' is incomplete. We then have

$$\mathrm{ECOST}(Q') = \frac{\sum_{m\varepsilon Q} \frac{\mu C_m}{(\mu C_m - \lambda_m)^2} + \frac{\mu C_f}{(\mu C_f - \lambda_f)^2} - T_o}{\gamma_o(\frac{1}{\gamma} + \tau + \tau_{max}) + 1}$$

$$> \mathrm{ECOST}(Q) > \mathrm{COST}(Q^*)$$

In this case, Q' is added to I_PATHS. Extend incomplete paths in I_PATHS to nodes in R_NODES until all paths in I_PATHS become nonextendable. The proof is completed by applying induction.

5. CONCLUSIONS

Both open-chain queueing networks and closed-chain queueing networks have been employed in the past to model packet switching networks with flow-controlled virtual channels. A closed-chain model is the more natural of the two. Despite some recent advances in computational techniques (such as the tree convolution algorithm), the computational requirements of a closed-chain model are still too large to be used in network design procedures. An open-chain model, on the other hand, encounters the difficulty that the throughputs of flow-controlled virtual channels, needed as input parameters for the model, are not known.

We examined the traffic conditions under which an open-chain model accurately predicts the mean end-to-end delays of a closed-chain model. We then proposed an approach to employ both closed-chain and open-chain models in network design procedures. A closed-chain model is used to compute virtual channel throughputs. An open-chain model is used for intermediate optimization steps.

The problems of optimally routing incremental flows were explored. We observed that the underlying objectives of ARPANET routing and flow deviation correspond to unconstrained individual-optimization and network-optimization problems (respectively). We formulated a constrained optimal routing problem that is a compromise between the two objectives. An algorithm that finds an optimal solution to the problem has been presented.

REFERENCES

AGNE 76 Agnew, C. E., "On Quadratic Adaptive Routing Algorithms," Comm. ACM 19 (1976) pp. 18-22.

BASK 75 Baskett, F., K. M. Chandy, R. R. Muntz and F. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," JACM, April 1975.

CERF 74 Cerf, V. and R. Kahn, "A Protocol for Packet Network Intercommunication," IEEE Trans. on Communications, COM-22 (1974), 637-648.

FRAT 73 Fratta, L., M. Gerla and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-forward Network Design," Networks 3 (1973), 97-133.

GALL 77 Gallager, R., "An Optimal Routing Algorithm Using Distributed Computation," IEEE Trans. on Commun., Vol. COM-25, Jan. 1977, pp. 73-85.

GERL 73 Gerla, M., "The Design of Store-and-forward Networks for Computer Communications," Ph.D. dissertation, Department of Computer Scienes, UCLA (Jan. 1973).

GERL 77 Gerla, M. and L. Kleinrock, "On the Topological Design of Distributed Computer Networks," IEEE Trans. on Communications, COM-25 (1977), pp. 48-60.

GERL 80 Gerla, M. and P. O. Nilsson, "Routing and Flow Control Interplay in Computer Networks," Proc. Fifth ICCC, Atlanta, Oct. 1980.

IBM 75 IBM Corp., Systems Network Architecture General Information, GA27-3102-0 (Jan. 1975).

KLEI 64 Kleinrock, L., Communication Nets--Stochastic Message Flow and Delays, (McGraw-Hill, New York, 1964).

KLEI 76 Kleinrock, L., Queueing Systems, Volume 2: Computer Applications (Wiley-Interscience, New York, 1976).

LAM 76 Lam, S. S., "Store-and-forward Buffer Requirements in a Packet Switching Network," IEEE Trans. on Communications COM-24 (1976) pp. 394-403.

LAM 81a Lam, S. S. and Y. L. Lien, "A Tree Convolution Algorithm for the Solution of Queueing Networks," Dept. of Computer Sciences, University of Texas at Austin, Technical Report TR-165, January 1981.

LAM 81b Lam, S. S. and Y. L. Lien, "Modeling and Analysis of Flow Controlled Packet Switching Networks," Proc. 7th Data Communications Symposium, Mexico City, Oct. 1981.

LAM 82 Lam, S. S. and J. W. Wong, "Queueing Network Models of Packet Switching Networks, Part 2: Networks with Population Size Constraints," Performance Evaluation, Vol. 2, No. 2, 1982 (to appear).

LAVE 80 Lavenberg, S. S. and M. Reiser, "Stationary State Probabilities of Arrival Instants for Closed Queueing Networks with Multiple Types of Customers," J. Applied Probability, Dec. 1980.

LIEN 81 Lien, Y. L., "Modeling and Analysis of Flow-controlled Computer Communication Networks," Ph.D. Thesis, Dept. of Computer Sciences, University of Texas at Austin, Dec. 1981.

LITT 61 Little, J. D. C., "A Proof of the Queueing Formula: $L = \lambda W$," Operations Research 9 (1961) pp. 383-387.

MCQU 78 McQuillan, J. M., G. Falk and I. Richer, "A Review of the Development and Performance of the ARPANET Routing Algorithm," IEEE Trans. on Commun. COM-26 (1978), pp. 1802-1811.

OPDE 74 Opderbeck, H. and L. Kleinrock, "The Influence of Control Procedures on the Performance of Packet-switched Networks," Proc. National Telecommunications Conference, San Diego (Dec. 1974).

PENN 75 Pennotti, M. and M. Schwartz, "Congestion Control in Store and forward Tandem Links," IEEE Trans. on Communications COM-23 (1975), pp. 1434-1443.

POUZ 73 Pouzin, L., "Presentation and Major Design Aspects of the CYCLADES Computer Network," Proc. Third Data Communications Symposium, St. Petersburg, Florida (Nov. 1973).

REIS 75 Reiser, M. and H. Kobayashi, "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms," IBM Journal of Research and Development 19 (1975).

REIS 79 Reiser, M., "A Queueing Network Analysis of Computer Communication Networks with Window Flow Control," IEEE Trans. on Communications COM-27 (1979) pp. 1199-1209.

REIS 80 Reiser, M. and S. Lavenberg, "Mean Value Analysis of Closed Multi-chain Queueing Networks," JACM 27 (1980) pp. 313-322.

ROBE 78 Roberts, L. G., "The Evolution of Packet Switching," Proc. IEEE, Vol. 66, Nov. 1978.

SCHW 77 Schwartz, M., Computer-Communication Network Design and Analysis, Prentice-Hall, 1977.

SEVC 79 Sevcik, K. C. and I. Mitrani, "The Distribution of Queueing Network States at Input and Output Instants," Proc. 4th Intl. Symp. on Modeling and Performance Evaluation of Computer Systems, Vienna, Austria, 1979.

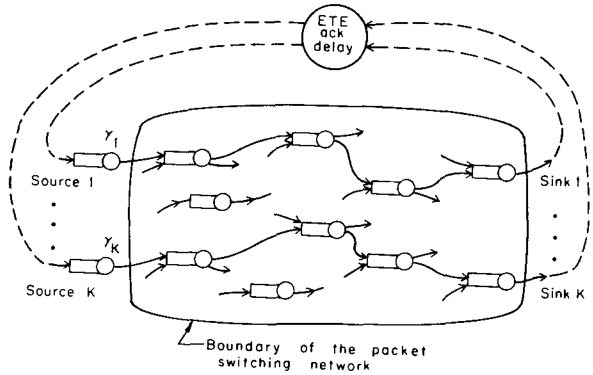

Fig. 1. An illustration of the closed-chain queueing network model.

chain	throughput rate	mean delay	delay estimate	% error
1	9.20e-01	6.02e-01	6.15e-01	2.28e+00
2	9.20e-01	6.02e-01	6.15e-01	2.28e+00
3	9.17e-01	6.25e-01	6.41e-01	2.49e+00
4	9.17e-01	6.25e-01	6.41e-01	2.49e+00
5	9.20e-01	6.03e-01	6.17e-01	2.27e+00
6	9.20e-01	6.03e-01	6.17e-01	2.27e+00
7	9.47e-01	4.96e-01	5.07e-01	2.11e+00
8	9.47e-01	4.96e-01	5.07e-01	2.11e+00
9	9.41e-01	5.36e-01	5.49e-01	2.57e+00
10	9.41e-01	5.36e-01	5.49e-01	2.57e+00
11	9.45e-01	5.06e-01	5.18e-01	2.29e+00
12	9.45e-01	5.06e-01	5.18e-01	2.29e+00
13	8.18e-01	9.91e-01	1.02e+00	2.85e+00
14	8.18e-01	9.91e-01	1.02e+00	2.85e+00
15	9.86e-01	2.73e-01	2.78e-01	1.74e+00
16	9.86e-01	2.73e-01	2.78e-01	1.74e+00
17	9.88e-01	2.44e-01	2.47e-01	1.21e+00
18	9.88e-01	2.44e-01	2.47e-01	1.21e+00
19	9.89e-01	2.33e-01	2.35e-01	1.01e+00
20	9.89e-01	2.33e-01	2.35e-01	1.01e+00
21	9.89e-01	2.32e-01	2.35e-01	1.03e+00
22	9.89e-01	2.32e-01	2.35e-01	1.03e+00
23	9.97e-01	1.23e-01	1.24e-01	7.36e-01
24	9.97e-01	1.23e-01	1.24e-01	7.37e-01
25	9.17e-01	6.25e-01	6.40e-01	2.50e+00
26	9.17e-01	6.25e-01	6.40e-01	2.50e+00
27	9.70e-01	3.79e-01	3.86e-01	1.84e+00
28	9.70e-01	3.79e-01	3.86e-01	1.84e+00
29	8.56e-01	8.46e-01	8.68e-01	2.60e+00
30	8.56e-01	8.46e-01	8.68e-01	2.60e+00
31	8.84e-01	7.57e-01	7.78e-01	2.73e+00
32	8.84e-01	7.57e-01	7.78e-01	2.73e+00

Errors in delay estimates

Average : 2.02e+00

Variance : 4.32e-01

Standard Deviation : 6.57e-01

Table 1. Mean delays and delay estimates for the network example.

Case of	utilizations of communication channels				% errors in delay estimates			
	mean	max.	min.	st. dev.	mean	max.	min.	st. dev.
$\gamma = 10$	0.466	0.808	0.134	0.188	40.3	127	15.3	26.3
$\gamma = 2$	0.299	0.469	0.112	0.107	7.34	9.86	4.74	1.37
$\gamma = 1$	0.185	0.281	0.082	0.066	2.02	2.85	0.74	0.66
$\gamma = 2/3$	0.130	0.195	0.061	0.047	0.80	1.39	0.22	0.34
$\gamma = 1/2$	0.100	0.148	0.048	0.036	0.40	0.77	0.09	0.19

Table 2. Channel utilizations and errors in delay estimates for different values of γ.

Case of	utilizations of communication channels				% errors in delay estimates			
	mean	max.	min.	st. dev.	mean	max.	min.	st. dev.
window size = 2	0.159	0.248	0.064	0.057	4.02	4.73	2.94	0.46
window size = 3	0.185	0.281	0.082	0.066	2.02	2.85	0.74	0.66
window size = 4	0.197	0.294	0.092	0.070	0.88	1.72	0.18	0.45

Table 3. Channel utilizations and errors in delay estimates for different window sizes.