#/14?

FLOW CONTROL BASED ON LIMITING PERMIT GENERATION RATES

Leonard Kleinrock and Chong-Wei Tseng
Computer Science Department
University of California, Los Angeles

ABSTRACT

Two new static flow control schemes for computer communication networks are presented and analyzed. These schemes are based on mechanisms for limiting the generation rate of permits on logical channels between communicating users. Both open and closed queueing network models are used in analyzing the behavior of these systems. The schemes offer a new dimension in flow control. They are quite effective, easy to implement, simple to analyze and are capable of incorporating other common features of computer communication networks, such as piggy-back acknowledgements, time-outs, etc. The first scheme is bounded away from the "ideal" flow control behavior; the second scheme overcomes this shortcoming.

I. INTRODUCTION

It is commonly accepted that some form of flow control is required in any computer communication network (CCN). A CCN without suitable flow control is congestion prone and tends to be unstable and inefficient. That is, its throughput degrades dramatically when the offered load exceeds a critical threshold [Refs. 3, 5].

Various flow control schemes have been investigated and implemented in the past [Ref. 2]. They can be classified basically as schemes of "local flow control" or "global flow control." Examples of local flow control schemes are the "minimal allocation - maximal limit" scheme in the ARPANET [Ref. 9], the "square root scheme" [Ref. 4], the "sharing with minimal allocation" scheme [Ref. 6], the GMD "buffer class" scheme [Ref. 10], etc. These schemes are intended to prevent local store-and-forward buffer congestion (local throughput degradation) and direct (and indirect) store-and-forward deadlock. While possibly improving the local network performance, a local scheme cannot optimize the global network performance (high network utilization, high throughput, low delay, fairness of service, etc.). For this purpose, global flow control schemes are often used.

At first thought, one may think that a global flow control scheme should allocate network resources to users in a way such that, well before the network capacity is reached, users' inputs are throttled so that the network will not be overloaded. Indeed, this can be accomplished either by properly allocating a maximal number of permits or by assigning an appropriate window size to each group of users (the group can be as large as that containing all users or as small as that containing a single user). The idea is to keep the total number of packets simultaneously existing in the network below the network's capacity; that is, the network as a whole will not be overloaded. However, due to the *bursty* nature of computer traffic, a flow control scheme often *overallocates* the network resources to users for reasons of efficiency so that, on the average, the network utilization is kept high enough. When resources are overallocated, the network is liable to be overloaded from time to time. This (infrequent) overloading must be handled by some other mechanism such as a dynamic scheme or a local static flow control scheme as mentioned above.

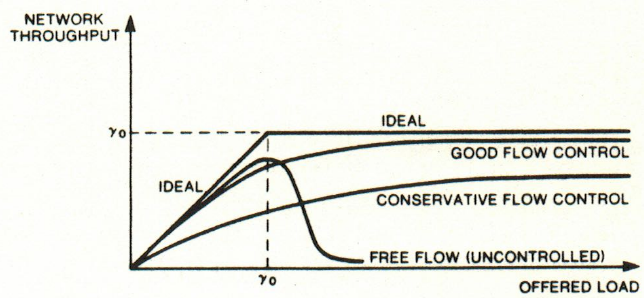

Fig. 1. The effectiveness of flow control schemes

A simple measure to test the effectiveness of a flow control scheme is discussed in [Ref. 8], and is shown in Fig. 1. We see that an ideal flow control scheme accepts (and delivers) the total input traffic offered as long as the traffic level is less than the network capacity, γ_0.

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense under contract MDA 703-77-C-0272.

Beyond an input traffic of γ_0, the throughput saturates at that level (γ_0). The closer to the ideal curve a flow control scheme can perform the better is that scheme.

In this paper we present a new class of flow control techniques. Two schemes are proposed and their models analyzed. Both are based on limiting the permit generation *rates* of logical channels. It is found that the second model gives a performance very close to the ideal. Both schemes are very easy to implement; more importantly they involve no computational complexity (one can calculate the entire network performance on the back of an envelope) and other network features such as "piggy-back" acknowledgements and "time-outs" can easily be incorporated in the models.

II. TWO FLOW CONTROL SCHEMES - MODEL AND ANALYSIS

Before we present our flow control procedures, let us make the following observation. Consider a queue of *permits* as shown in Figure 2. We assume that the permit interarrival time has a general distribution denoted by G_p. The permits are queued awaiting the arrival of messages, and (immediately) leave the queue only when they attach themselves to a message (one message per permit). If a message arrives and finds no permits in the queue, we assume the message is lost. Suppose that the message interarrival time has an exponential distribution denoted by M. The queue for permits can then be modeled as a G_p/M/1 queueing system [Ref. 7]. That is, the permits are "served" by the messages, and the service time is the time until the next message arrival.

With this observation, we can now present our flow control schemes.

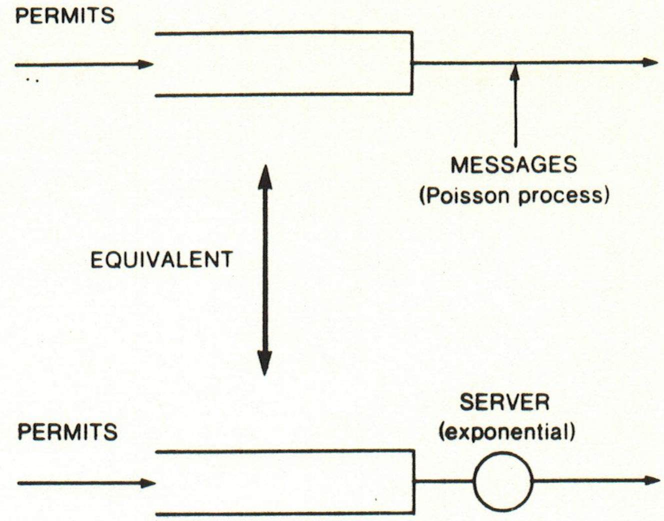

Fig. 2. Two mathematically equivalent systems

II.1 The Permit Circulation Model - Scheme 1

Consider the computer communication network consisting of a number of nodes inter-connected by communication channels as shown in Fig. 3.

II.1.1 Description of the Scheme

(1) There is an external source of permits which generates permits at a rate g. Assume that the generating process is Poisson. Further, assume these permits are distributed to permit queue i with a probability r_i. Permit queue i is associated with the physical channel i which connects node x to node y. The permits in queue i are to be used by messages which *originate* at node x and are *routed* through node y.

(2) There is also a sink which destroys permits.

(3) An externally arriving message at a node must secure a permit from the corresponding permit queue at that node before it can be launched into the network on the first leg of its journey. If the message finds no permit, it is immediately rejected and "lost". Note that externally arriving messages may not queue at their origin awaiting entry into the net. Once a message is accepted into the net it is stored-and-forwarded over some fixed route to its destination. We assume that messages arrive at logical channel r according to a Poisson process at rate γ_r (a logical channel is associated with each origin-destination pair).

(4) After launching a message into the network, a permit at queue i is then immediately passed to a neighbor queue j with probability r_{ij} (where queue j could be the sink).

(5) To prevent stalling of extra permits at queue i when the message traffic is low, we have a basic *circulation* rate μ_{i_0} to pump the permits out of queue i. That is, when the message rate is zero, the permit at the head of permit queue i will remain there for an exponentially distributed length of time with mean $1/\mu_{i_0}$, after which it will move to permit queue j with probability r_{ij}. This basic rate is added to the permit circulation rate due to message arrivals.

Before going any further, let us give an example to see how the control scheme can be modeled for a network.

Consider the 5-node network as shown in Figure 4(a) with the permit queue model shown in Figure 4(b). Note that we have ten physical communication channels (i.e., five full-duplex channels) and twenty logical (origin-destination) channels. Each physical channel corresponds to a permit queue. In Figure 4(b) we have a conceptual source and a conceptual sink. The permits generated at the source (with rate g) are distributed to queue i with probability r_i. Permits at queue i are removed (see the observation made at the beginning of this section) at a rate

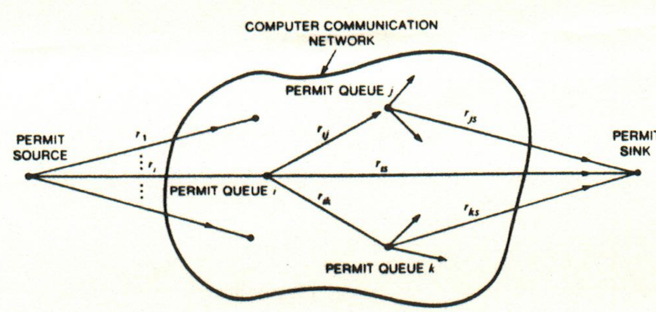

Fig. 3. The "currency circulation" model

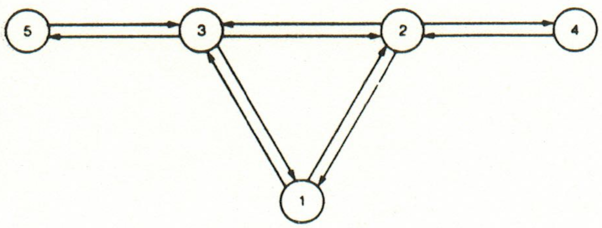

Fig. 4(a). A simple computer communication network example

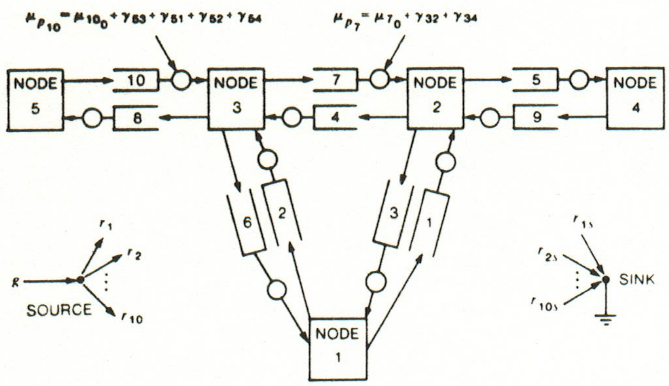

Fig. 4(b). The currency circulation model of the simple network in Figure 4(a)

$$\mu_{p_i} = \mu_{i_0} + \sum_{r \in F(i)} \gamma_r = \mu_{i_0} + \Gamma_i \tag{1}$$

where γ_r is the external message arrival rate of logical channel r, and $F(i)$ = {logical channel r whose messages are first routed through physical channel i}, and Γ_i is the sum of all γ_r's with each r belonging to $F(i)$.

After its removal from queue i, a permit enters queue j with probability r_{ij}. We shall continue this example later.

II.1.2 The Queueing Model for Permits

The external arrival rate of permits at queue i is $g_i = gr_i$, and, as described above, the departure rate of permits at queue i is given by Eq. (1).

The total arrival rate of permits to queue i (from outside the network and from other queues) is

$$\lambda_{p_i} = gr_i + \sum_{j=1}^{M} \lambda_{p_j} r_{ji} \tag{2}$$

where g is the total permit generation rate and M is the total number of permit queues.

Thus, we have a Jackson-type network of M queues for permits. The joint probability for the number of permits distributed at all of the queues is then

$$P_p[n_1, n_2, \ldots, n_M] = P_{p_1}(n_1) P_{p_2}(n_2) \cdots P_{p_M}(n_M) \tag{3}$$

where n_i denotes the number of permits in queue i and $P_{p_i}(n_i)$ is the probability that queue i has n_i permits; $P_{p_i}(n_i)$ is the solution to an M/M/1 queue with input rate λ_{p_i} as in Eq. (2) and output rate μ_{p_i} as in Eq. (1). i.e.,

$$P_{p_i}(n_i) = (1 - \rho_{p_i}) \rho_{p_i}^{n_i}$$

where $\rho_{p_i} = \lambda_{p_i} / \mu_{p_i}$

Observe that the permits do not necessarily enter or leave a queue according to a Poisson process.

II.1.3 The Queueing Model for Messages

Observe that each logical channel r is associated with a permit queue at its originating node (e.g., in Figure 4, logical channel 5-4 is associated with queue 10 at node 5).

Denote the throughput (message/sec) of logical channel r as S_{ir}. Then,

$$S_{ir} = [1 - P_{p_i}(0)]\gamma_r = \left[\frac{\lambda_{p_i}}{\mu_{p_i}}\right] \gamma_r = \frac{\lambda_{p_i}}{\mu_{i_0} + \Gamma_i} \gamma_r \tag{4}$$

where $r \in F(i)$, and $P_{p_i}(0)$ is the probability of no permit in permit queue i.

After entering the network, a message is routed through a set of nodes and physical channels to its destination. Upon arriving at a node, the message joins the message queue associated with the physical channel in its path, and waits there for transmission through the channel.

The state of the system for messages is indicated by using "customer classes" as in [Ref. 1] where each class corresponds to a logical channel. In particular consider the vector of column vectors

$$\left[\begin{pmatrix} K_{11} \\ K_{12} \\ \cdot \\ \cdot \\ \cdot \\ K_{1R} \end{pmatrix} \cdots \begin{pmatrix} K_{i1} \\ K_{i2} \\ \cdot \\ \cdot \\ \cdot \\ K_{iR} \end{pmatrix} \cdots \begin{pmatrix} K_{M1} \\ K_{M2} \\ \cdot \\ \cdot \\ \cdot \\ K_{MR} \end{pmatrix} \right]$$

where

(a) M is the total number of physical channels

(b) R is the total number of logical channels

(c) Column vector i denotes the state of message queue i associated with physical channel i

(d) K_{ir} $(r = 1, 2, \cdots, R)$ denotes the number of messages at queue i from logical channel r, regardless of their actual relative positions with respect to other messages in queue i.

If we assume that the messages enter the network from a Poisson process, then

$$P\left[\begin{pmatrix} K_{11} \\ \cdot \\ \cdot \\ \cdot \\ K_{1R} \end{pmatrix} \cdots \begin{pmatrix} K_{M1} \\ \cdot \\ \cdot \\ \cdot \\ K_{MR} \end{pmatrix} \right] = G \prod_{i=1}^{M} \frac{K_i!}{K_{i1}! \ldots K_{iR}!} \prod_{r=1}^{R} \left[\frac{\xi_{ir}}{\mu C_i}\right]^{K_{ir}} \tag{5}$$

where

$$K_i = \sum_{r=1}^{R} K_{ir} \tag{6}$$

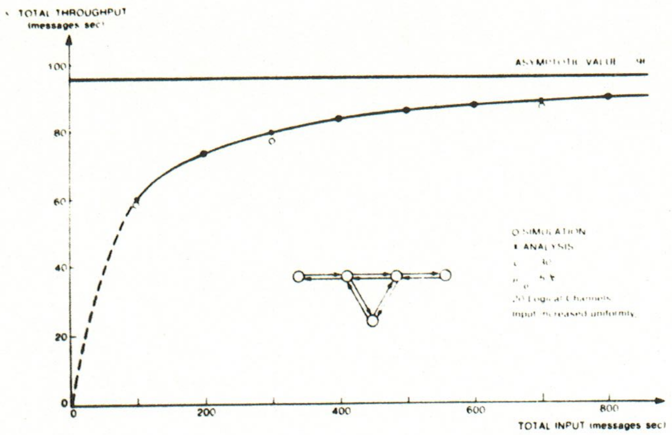

Fig. 5. Total throughput vs. total input

C_i is the bit service rate of channel i,

and

$1/\mu$ is the average message length which is assumed to have an exponential distribution,

$$\xi_{ir} = \begin{cases} 0, & \text{if messages of logical channel } r \\ & \text{are not routed through physical channel } i \\ S_{ir} & \text{otherwise} \end{cases}$$

Let $\rho_{ir} \triangleq \dfrac{\xi_{ir}}{\mu C_i}$ for $i = 1, 2, \ldots M$, $r = 1, 2, \ldots, R$,

and

$$\rho_i \triangleq \sum_{r=1}^{R} \rho_{ir}$$

Then the normalization constant G can be calculated from the following equation:

$$G = \prod_{i=1}^{M} (1-\rho_i) \tag{7}$$

We see that the marginal probability $P_i(K_i)$ which denotes the probability that the i^{th} physical channel has K_i messages is

$$P_i(K_i) = (1-\rho_i)\rho_i^{K_i} \tag{8}$$

Thus the system is completely solved. Moreover, from J. W. Wong's work, we may find $T_r^*(s)$, the Laplace transform of the density for the delay of class r messages with fixed routing [Ref. 12]

$$T_r^*(s) = \prod_{i \in I(r)} \frac{\mu C_i(1-\rho_i)}{s+\mu C_i(1-\rho_i)} \tag{9}$$

where $I(r)$ = {queue i | queue i is in the path of logical channel r}.

Thus, the average delay is

$$T_r = \sum_{i \in I(r)} \frac{1}{\mu C_i(1-\rho_i)} \tag{10}$$

II.1.4 Result of a Simple Example

We apply the above scheme to the network shown in Figure 4. The calculated throughput and the simulated throughput are both plotted in Figure 5, where we have (see Sec. II.1.1).

g = 30 (total generation rate of permits)

μ_{i_0} = 5, $\forall i$ (the basic circulation rate).

r_i = 1/10, $\forall i$ and

$$r_{ij} = \frac{1}{n_i}$$

where n_i is equal to one plus the number of queues adjacent to queue i; the extra "one" is for the sink. With these parameter values, we have

$$\sum_{i=1}^{10} \lambda_{p_i} = 96 \quad \text{(see Eq. (2))}.$$

We observe in Figure 5 that the asymptotic value of total throughput is also 96. We will explain why in the next section. Also note that we did not obtain results in the low input region since in that region permits enter a permit queue faster than they are removed (with the basic circulation rate μ_{i_0} = 5 fixed), i.e., we obtained an unstable system. This is one of the causes which motivated us to investigate the second scheme in Section 2.2.

II.1.5 Some Observations of the Model

(1) Upper limit of the throughput: Let S_j denote the total throughput of those logical *total* throughput of these logical channels belonging to F(j); then

$$S_j = [1-P_{p_j}(0)]\Gamma_j = \frac{\lambda_{p_j}}{\mu_{j_0}+\Gamma_j}\Gamma_j \tag{11}$$

As $\Gamma_j \rightarrow \infty$, $S_j \rightarrow \lambda_{p_j}$. This shows that our scheme does put a ceiling ($\sum_i \lambda_{p_i}$) on the acceptance rate of the total external input to protect the network from overloading. This also explains why the asymptotic value of the total throughput shown in Figure 5 is 96.

(2) Inefficiency of the scheme: In order to prevent permits from infinitely accumulating in a queue when the external traffic is near zero, we must let μ_{j_0} be greater than λ_{p_j} for all j.

From Eq. (11) we have $S_j \rightarrow 0$ as $\lambda_j \rightarrow 0$ and when $\Gamma_j = k\mu_{j_0}$, $k = 1, 2, \cdots$

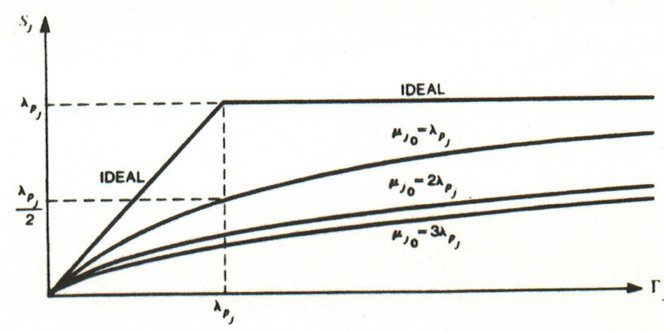

Fig. 6. Performance of currency circulation model

$$S_i \Big|_{\Gamma_i = k\mu_{j0}} = \frac{\lambda_{p_i}}{(k+1)\mu_{j0}} k\mu_{j0} = \frac{k}{k+1} \lambda_{p_i}$$

Figure 6 shows a family of performance curves for this control scheme.

We see that the best performance we can obtain is the uppermost curve under the ideal one. This is apparently not satisfactory (it is too conservative). In the next section we will attack this weak point and obtain a nearly perfect static control scheme.

II.2 The Permit Killer Model - Scheme 2

The control scheme described in Section II.1 has some weak points. First, the performance curve is far below the ideal one (see Figure 6). Second, we do not know how to specify the transition matrix for permits optimally. In this section we look into a more interesting scheme which produces much more desirable results.

II.2.1 The Model Description

This scheme works similar to the one in the last section except that:

(1) After launching a message into the network, a permit is *destroyed immediately* instead of being circulated to other permit queues.

(2) Permits for logical channel r are generated (from a Poisson process) at rate g_r and are queued in permit queue q_r.

(3) We have a finite buffer size N_r for each permit queue q_r. When the buffer is full, the newly generated permits are destroyed thus controlling the unlimited growth of permit queues. A permit queue is shown in Figure 7(a).

II.2.2 Analysis of the Scheme

The model for permits: Each permit queue q_r (for logical channel r) can be described by the finite Markov chain in Figure 7(b).

Let $P_{k_r} = P[k_r$ permits in $q_r]$. We then have

$$P_{k_r} = \left(\frac{g_r}{\gamma_r}\right)^{k_r} P_{o_r} \quad \text{for } k_r = 0, 1, 2, \cdots, N_r \qquad (12)$$

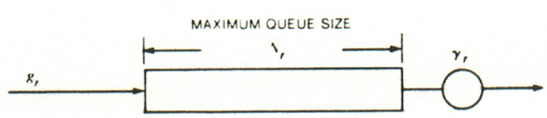

Fig. 7(a). Permit queue r for logical channel r

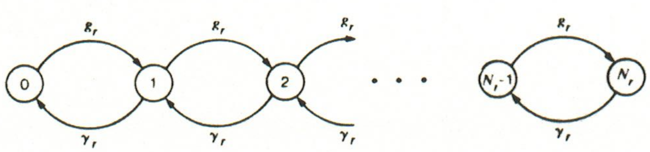

Fig. 7(b). The Markov chain describing the permit queue

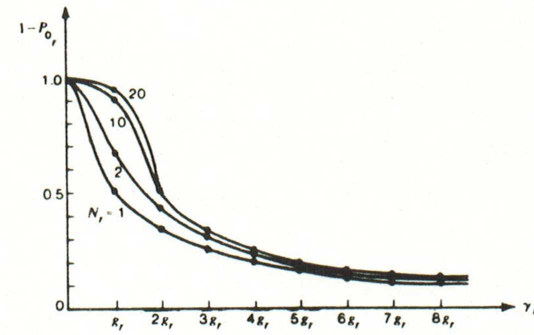

Fig. 8. The probability of a permit queue

where

$$P_{o_r} = \frac{1 - (g_r/\gamma_r)}{1 - (g_r/\gamma_r)^{N_r+1}} \qquad (13)$$

In Figure 8, we plot $1 - P_{o_r}$ versus γ_r. The total throughput for logical channel r is

$$S_{ir} = \left(1 - P_{o_r}\right) \gamma_r = \frac{\rho_r - \rho_r^{N_r+1}}{1 - \rho_r^{N_r+1}} \gamma_r \qquad (14)$$

where $\rho_r \triangleq g_r/\gamma_r$

For large N_r,

as $\gamma_r \to \infty$, $S_{ir} \to g_r$; as $\gamma_r \to 0$, $S_{ir} \to 0$; and

$$\text{as } \gamma_r \to g_r, \; S_{ir} \to \frac{N_r}{N_r + 1} g_r \qquad (15)$$

II.2.3 Result of the Simple Example

In Figure 9, we show the result of applying the scheme to our example network (Figure 4) with $g_r = 5$. The plot is for total throughput versus input rate per logical channel.

We see that we can bring our performance curve as close to the ideal curve as we desire by letting N_r be sufficiently large. This is an extremely desirable property. The computer communication network can now be viewed as a network of queues with external inputs S_{ir}'s and with some routing scheme (fixed for our case). The delay can easily be obtained through the application

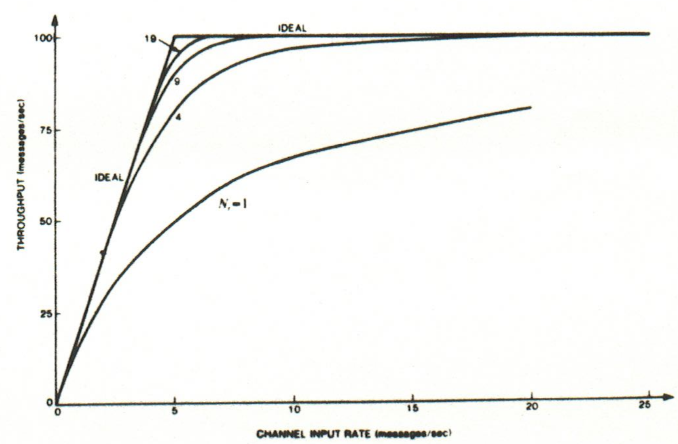

Fig. 9. The performance curves of Scheme 2

of J. Wong's delay equations. The network model is thus completely solved. Observe again that S_{ir} is not a Poisson process. But it becomes Poisson asymptotically when $N_r \to \infty$.

III. CONCLUSION

We have presented two flow control schemes, each based on limiting the permit generation rates for logical channels.

The advantages of these kinds of flow control schemes are:

(1) They require no end-to-end acknowledgements, in contrast to window schemes. This may save considerable processing capacity as well as transmission bandwidth. (Furthermore, our schemes become particularly suitable if, for some security reason, the destination node is not allowed to know the identity of the source, and hence no ACK can be routed back.)

(2) The killer scheme can easily be adjusted to the ideal flow control scheme as closely as we desire. With most other flow control schemes, we usually have no idea how to achieve such behavior.

(3) Our schemes are easy to implement. No complicated software or algorithms are involved.

(4) Most importantly, these procedures are easily evaluated analytically. They can save considerable effort in the network development phase. Also, the computational simplicity of our schemes have allowed us to easily incorporate many other aspects of network protocols (such as piggy-back ACK's, and time-outs) into our model [Ref. 11].

REFERENCES

[1] Baskett, F., K. Chandy, R. Muntz, and F. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *Journal of the ACM,* Vol. 22, No. 2, April 1975, pp. 248-260.

[2] Gerla, M. and L. Kleinrock, "Flow Control: A Comparative Survey," *IEEE Transactions on Communications,* April 1980, pp. 553-574.

[3] Giessler, A., J. Haaenle, A. Koenig and E. Pade, "Free Buffer Allocation-An Investigation by Simulation," *Computer Networks,* Vol. 2, July 1978, pp. 191-208.

[4] Irland, M., "Buffer Management in a Packet Switch," *IEEE Transactions on Communications,* Vol. COM-26, March 1978, pp. 328-337.

[5] Kahn, R. and W. Crowther, "Flow Control in a Resource-Sharing Computer Network," *IEEE Transactions on Communications,* Vol. COM-20, June 1972, pp. 539-546.

[6] Kamoun, F. and L. Kleinrock, "Analysis of Shared Storage in a Computer Network Environment: A General Case," *IEEE Transactions on Communications,* Vol. COM-28, July 1980.

[7] Kleinrock, L., *Queueing Systems, Vol. I: Theory,* Wiley Interscience, New York, 1975.

[8] Kleinrock, L., "On Flow Control in Computer Networks," *Conference Record, International Conference on Communications,* Toronto, June 1978, pp. 27.2.1-27.2.5.

[9] McQuillan, J. M., W. Crowther, B. Cosell, D. Walden and F. Heart, "Improvements in the Design and Performance of the ARPA Network," *AFIPS Conference Proceedings,* Vol. 41, FJCC, Anaheim, California, December 1972, pp. 741-754

[10] Raubold, E. and J. Haenle, "A Method of Deadlock-Free Resource Allocation and Flow Control in Packet Networks," *Proceedings of the Third International Conference on Computer Communication,* Toronto, August 1976,

[11] Tseng, C., "Flow Control in Store-and-Forward Computer Communications Networks," Ph.D. Dissertation to be published, Computer Science Department, University of California, Los Angeles, 1980.

[12] Wong, J., "Distribution of End-to-End Delay in Message-Switched Networks," *Computer Networks,* Vol. 2, February 1978, pp. 44-49.

Leonard Kleinrock received his B.E.E. degree from City College of New York and the M.S.E.E. and Ph.D.E.E. degrees from Massachusetts Institute of Technology in 1959 and 1963, respectively. In 1963, he joined the faculty of the School of Engineering and Applied Science, University of California, Los Angeles, where he is now Professor of Computer Science. His research spans the fields of computer networks, computer systems modeling and analysis, queueing theory, and resource sharing and allocation in general. He is the author of three major books in the field of computer networks: *Communication Nets: Stochastic Message Flow and Delay, Queueing Systems, Vol. I: Theory* and *Queueing Systems, Vol. II: Computer Applications.* He has published over 100 articles and contributed to several books. He was recently elected to the National Academy of Engineering, is a Guggenheim Fellow, an IEEE Fellow, and has received various outstanding teacher and best paper awards, including the 1976 Lanchester prize for the outstanding paper in operations research, and the ICC '78 prize-winning paper award.

Chong-Wei Tseng was born in 1945. He received his B.S. in Nuclear Engineering from National Tsing-hua University in Taiwain in 1968. He earned his M.S. in Electrical Engineering and Computer Science at the University of California, Berkeley, in 1974 and his Ph.D. in Computer Science at UCLA in 1980. He is currently on the technical staff in charge on flow control and congestion control of the CSN project with Transaction Technology Inc. He was the winner of the best paper award in the 1973 Asilomar Conference on Circuits and Systems. His interest is in the modeling, analysis and design of computer communications networks.