Congestion Control of Store-and-Forward Networks by Input Buffer Limits—An Analysis

SIMON S. LAM, MEMBER, IEEE, AND MARTIN REISER

Abstract—**The use of input buffer limits for congestion control of store-and-forward networks is investigated. An analytic model is formulated. Based upon the analytic results, strategies are proposed for the design of input buffer limits to achieve the maximum network throughput as well as to provide a safety margin for uncertainties in traffic assumptions. A useful capacity law is discovered. Major conclusions drawn from the analysis are supported by simulation results for a four-node homogeneous network. These results indicate that input buffer limits which satisfy the capacity law are a simple and effective means of network congestion control. Further simulation studies are underway to investigate methods of implementation in a general network.**

1. INTRODUCTION

STORE-AND-FORWARD communication networks with no effective means of flow control have been shown to exhibit the throughput-load relationship illustrated in Figure 1 [1-4]. A characteristic, typical of many contention systems, is that as the offered load is increased from zero, the network throughput increases to a maximum and then turns down and decreases sharply to a low value (possibly zero). Physically, when a store-and-forward network is congested, some processes may be blocked, and data may be lost or held back due to a lack of resources [5]. In either case, work is not conserved; hence, the degradation in throughput.

Degradation in network throughput is often caused by deadlocks [1, 2, 5]. However, networks which are deadlock-free may still be degraded in the sense that the throughput, though nonzero, is relatively low [4]. Hence, control mechanisms are needed to prevent throughput degradation whether or not a network can be formally proved to be deadlock-free.

From now on, a network is said to be *congested* when it operates in the region of negative slope in Fig. 1.

Network versus end-to-end control

By network congestion control we mean any mechanism with the primary objective of preventing the network from operating in the congested region for any significant period of time. Typically, most networks are based upon the concept of logical channels (or connections, sessions etc.) and are end-to-end flow-controlled between pairs of sources and sinks. Examples of end-to-end controls are SNA pacing [6], RFNM in the ARPANET [5], and various window mechanisms [7, 8]. An important function of such end-to-end controls is synchronization of the source input rate to the sink acceptance rate. All of them work by limiting the number of messages (or packets) permitted in a logical channel. Suppose L_i is the maximum number of messages in logical channel i and the network has a total of K logical channels. The maximum number of messages permitted to enter the network is thus

$$N_{\max} = L_1 + L_2 + \cdots + L_K.$$

The fact that $N_{\max}$ is bounded does not imply that network congestion control is not necessary. In fact, one of the motivations for a store-and-forward network in the first place is that data traffic sources are typically bursty [9]. In other words, logical channels require actual transmission capacity only intermittently with a small duty cycle. If, for example, a network is operated such that $N_{\max}$ is at point B in Figure 1, it is obvious that network congestion control is not necessary. However, due to the bursty traffic, the average utilization of the network will be very low (such as at point A). It is therefore desirable for store-and-forward networks to operate on the principle of *overcommitment* such that $N_{\max}$ is far to the right (such as at point C) in Figure 1 and through averaging, the network utilization is at point B with a correspondingly high throughput. An immediate consequence is that network congestion control is now necessary to prevent the network operating point from going over the peak of the curve as a result of statistical fluctuations.

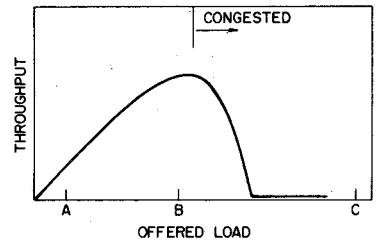


Fig. 1. Throughput versus offered load.

Network congestion control techniques

A network congestion control mechanism must be capable of: (1) detection of network congestion, and (2) shutting off input into the network according to some rule.

The isarithmic technique proposed by Davies [2] and studied by Price [3] does the above functions by limiting the number of packets permitted to enter the network. This is accomplished by circulating a fixed number of "containers"

Paper approved by the Editor for Computer Communication of the IEEE Communications Society for publication after presentation at the National Telecommunications Conference, Los Angeles, CA, December 5-7, 1977. Manuscript received August 8, 1977; revised August 29, 1978. The work of S. S. Lam was supported in part by the National Science Foundation under Grant ENG78-01803.

S. S. Lam was with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598. He is now with the Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712.

M. Reiser is with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

0090-6778/79/0100-0127$00.75

in the network. A packet can be sent through the network only if it can get hold of an empty container. A difficulty of this technique is finding a good adaptive scheme for distributing empty containers so as to maximize the network throughput and minimize delay.

A second technique, to be studied in this paper, attempts to control the network input rate by differentiating between input and transit traffic at each node and imposing a limit on the fraction of buffers in a node's buffer pool that input traffic can occupy. This fraction will be referred to as the *input buffer limit*. Note that transit traffic can occupy all buffers in the buffer pool. In times of extreme congestion, input traffic may be shut out by transit traffic but not vice versa; a desirable property. The advantage of discriminating against input traffic was first noticed by Price [3]. He observed that if one or two buffers are dedicated to transit traffic, the network throughput can be much improved. A similar idea was also suggested by Chou and Gerla [10]. This idea, however, is most clearly demonstrated and investigated in the GMD simulation studies [4, 11]. In addition, they have also shown that if the buffer pool is structured into nested subsets of buffers and messages are assigned to these subsets according to the number of hops they have covered, then it can be proved that store-and-forward deadlocks of the type described in [1, 5] can be avoided.

Summary of results

In the next section, a general analytic model is described for studying the use of input buffer limits for network congestion control. Next we present numerical results for a specific (homogeneous) network which illustrate the tradeoffs among offered load, buffer capacity and input buffer limit with regard to their impact on network throughput. Strategies are then proposed for selecting input buffer limits to achieve the maximum network throughput as well as to provide a safety margin for uncertainties and fluctuations in user traffic. A useful capacity law is discovered. Finally, simulation results are shown which support major conclusions drawn from the analytic results.

The model in this paper is different from previous analytic models such as the work of Pennotti and Schwartz [12] which is a model for end-to-end control and the work of Wong and Unsoy [13] which is a model for an isarithmic scheme with two levels of control.

2. ANALYTIC MODEL

Queuing network representation of a node

The queuing network representation of a store-and-forward node first introduced by Lam and Schweitzer [14, 15] is adopted with the addition of an input buffer limit for congestion control. A current limitation of queuing models is that no distinction can be made between messages and packets. The basic unit of data transfer and storage in this paper will be referred to as a message. Two classes of messages are distinguished: input and transit messages.

Referring to Figure 2, it is assumed that input messages are generated by locally attached sources at the rate of λ messages per second. Transit messages arrive from adjacent nodes at the rate of γ messages per second. The node has a pool of N_T buffers all of which may be occupied by transit messages. However, not more than $N_I (\leqslant N_T)$ buffers may be occupied by input messages. The ratio N_I/N_T is the input buffer limit of this node. Message arrivals (input or transit) may be lost due to a lack of buffers. We define

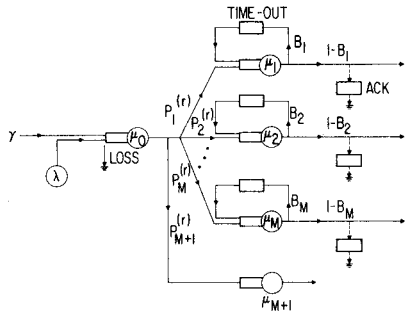


Fig. 2. Model of a store-and-forward node.

B_I = equilibrium loss probability of input messages

B_T = equilibrium loss probability of transit messages.

Since $N_I \leqslant N_T$, it follows that $B_I \geqslant B_T$.

A first-come-first-served (FCFS) server is used to represent the node processor which handles error checking, routing, etc. and operates at a rate of μ_0 messages per second. There are M output channels to adjacent nodes and a single channel (labeled $M+1$) to locally attached message sinks. The channels operate with rates of $\mu_1, \mu_2 \cdots, \mu_{M+1}$ messages per second and are also represented as FCFS servers. Whenever a message is transmitted to an adjacent node (say, over channel j), a copy of it is buffered. One of two things can then happen. With probability B_j, the transmitted message is lost in which case the buffered copy is put on the jth output channel queue for retransmission; this occurs after a time-out delay of t_j seconds on the average. With probability $1 - B_j$, the transmitted message is accepted by the adjacent node. Its positive acknowledgment return after a delay of v_j seconds on the average, at which time the node discards its copy of the message.

We shall assume that the local channel is reliable ($B_{M+1} = 0$) and also relatively fast (end-to-end control is not needed).

Input and transit messages are characterized by different routing probabilities. Input messages are routed to the jth channel with probability $P_j^{(1)}$ while transit messages are routed with probability $P_j^{(2)}$. We note that the model can be easily extended to more than two message classes with different routing probabilities and buffer limits, which can be used to represent the deadlock-avoidance scheme in [4, 11]. In practice, the number of classes is limited by the complexity of the numerical solution method used.

The rate λ represents the magnitude of the offered load. Both finite and infinite offered loads are considered. The case of an infinite offered load is equivalent to having N_I input messages within the node at all times.

With assumptions of independence [16], Poisson arrivals and exponentially distributed message lengths, recent advances in the theory of queuing networks can be applied to the above model. With the constraint imposed by the input buffer limit,

the model is a special case of a wide class of queuing networks with population size constraints studied by Lam [17]. For the moment, let us assume that λ is finite and define the following traffic intensities for input messages:

$$a_{10} = \lambda/\mu_0$$

$$a_{1,M+1} = \lambda P_{M+1}^{(1)}/\mu_{M+1}$$

$$a_{1j} = \lambda P_j^{(1)}/((1-B_j)\mu_j)$$

$$b_{1j} = \lambda P_j^{(1)} B_j t_j/(1-B_j)$$

$$c_{1j} = \lambda P_j^{(1)} v_j \quad j = 1, 2, \cdots, M. \tag{1}$$

Similarly, traffic intensities for transit messages a_{20}, $a_{2,M+1}$, a_{2j}, b_{2j} and c_{2j} ($j = 1, 2, \cdots, M$) are defined as above with γ replacing λ and $P_j^{(2)}$ replacing $P_j^{(1)}$. Next, define

$$d_1 = \sum_{j=1}^{M} (b_{1j} + c_{1j})$$

$$d_2 = \sum_{j=1}^{M} (b_{2j} + c_{2j})$$

the notations

q_{1j} = number of input messages at the jth FCFS server
q_{2j} = number of transit messages at the jth FCFS server
k_1 = number of input messages waiting for acknowledgment or being timed out before retransmission
k_2 = number of transit messages waiting for acknowledgment or being timed out before retransmission

and the state vector

$$S = (q_{10}, \cdots, q_{1,M+1}; q_{20}, \cdots, q_{2,M+1}; k_1, k_2).$$

Applying the theorem in [17], the equilibrium probability density function of S has the product form

$$P(S) = C\left\{\prod_{j=0}^{M+1} (q_{1j}+q_{2j})!(a_{1j}{}^{q_{1j}}/q_{1j}!)(a_{2j}{}^{q_{2j}}/q_{2j}!)\right\}$$

$$\cdot (d_1{}^{k_1}/k_1!)(d_2{}^{k_2}/k_2!) = Cp(S)$$

where C is a normalization constant given by

$$C^{-1} = \sum_{x_1=0}^{N_I} \sum_{x_2=0}^{N_T-x_1} w(x_1, x_2) \tag{2}$$

with

$$w(x_1, x_2) = \sum_{S \in S(x_1,x_2)} p(S)$$

and

$$S(x_1, x_2) = \{S \mid k_1 + q_{10} + \cdots + q_{1,M+1} = x_1,$$

$$k_2 + q_{20} + \cdots + q_{2,M+1} = x_2\}.$$

The convolutional algorithm of Reiser and Kobayashi [18] can be employed to evaluate $w(x_1, x_2)$. We note that the form of the density function $P(S)$ is the same as that given by Baskett *et al.* [19].

The marginal equilibrium probability density function for the number of input messages in the node is

$$P_1(x_1) = C \sum_{x_2=0}^{N_T-x_1} w(x_1, x_2), \qquad x_1 = 0, 1, \cdots, N_I \tag{3}$$

and for the number of transit messages in the node is

$$P_2(x_2) = C \sum_{x_1=0}^{\min(N_I, N_T-x_2)} w(x_1, x_2), \qquad x_2 = 0, 1, \cdots, N_T. \tag{4}$$

The equilibrium loss probability for transit messages is

$$B_T = C \sum_{x_1=0}^{N_I} w(x_1, N_T - x_1) \tag{5}$$

and similarly for input messages

$$B_I = B_T + C \sum_{x_2=0}^{N_T-N_I-1} w(N_I, x_2). \tag{6}$$

For the case of an infinite offered load, the above results remain applicable if λ is replaced by an arbitrary constant in Eq. (1) and x_1 is fixed at the value of N_I in Eqs. (2)-(6).

Solution for a network of nodes

No exact analytic solution for queuing networks with blocking has yet been obtained. To get around this difficulty, we have adopted the following approach proposed in [14]. The overall problem (network of nodes) is decomposed into a set of analytically tractable problems (one for each node) through the assumption of equilibrium loss probabilities (B_I and B_T). With a general network of store-and-forward nodes, we first solve a number of queuing networks (one for each node). These single-node results are then interfaced by requiring that message flows within the store-and-forward network are conserved. This procedure gives rise to a set of nonlinear equations involving the equilibrium loss probabilities which can be solved numerically [14]. It was found that this decomposition approach gives reasonably good results when the loss probabilities are small ($\ll 1$) and is useful for determining nodal buffer requirements to achieve small loss probabilities and for comparing the relative merits of buffer capacity assignment strategies [14].

In this paper, since we are not interested in a specific network topology and its performance, we shall make the additional assumption that the network is homogeneous. In other words, B_j is equal to B_T of the node under consideration for every j. This approach may give rise to somewhat pessimistic results since it assumes that when a node is congested, its neighboring nodes are equally congested. (We emphasize that this assumption is not necessary for our analytic model above

and is made here to simplify our numerical calculations in Section 3 below.)

Under equilibrium conditions, the nodal throughput rate σ in messages per second is

$$\sigma = \mu_{M+1} P[q_{1,M+1} + q_{2,M+1} > 0]$$
$$= \lambda(1 - B_I) \qquad \text{if } \lambda \text{ is finite.}$$

The arrival rate γ of transit messages is given according to the following flow conservation relationship.

$$\gamma(1 - B_T) P_{M+1}^{(2)} + \sigma P_{M+1}^{(1)} = \sigma$$

so that

$$\gamma = \sigma n_h / (1 - B_T)$$
$$= \lambda n_h (1 - B_I)/(1 - B_T) \qquad \text{if } \lambda \text{ is finite}$$

where

$$n_h = (1 - P_{M+1}^{(1)})/P_{M+1}^{(2)}$$

is the average number of hops traversed by a message from source node to destination node in the assumed homogeneous network.

Finally, B_T and σ (or B_I if λ is finite) can be solved using a successive substitution method from the above equations together with Eqs. (1), (2), (5) and (6).

3. NUMERICAL RESULTS

In this section, we present some numerical results which illustrate the tradeoffs among offered load, buffer capacity and input buffer limit. Based upon these results, the design of input buffer limits is discussed in the next section. An example with the following parameters is considered.

$M = 3$ (number of output channels)

$\mu_0 = 500$ messages/second (nodal processor speed)

$\mu_1 = \mu_2 = \mu_3 = 9.6$ messages/second (channel speed)

$\mu_4 = 100$ messages/second (sink rate)

$t_1 = t_2 = t_3 = 0.6$ second (average time-out delay)

$v_1 = v_2 = v_3 = 0.12$ second (average acknowledgment delay)

$$P_1^{(r)} = 0.4(1 - P_4^{(r)})$$

$$P_2^{(r)} = P_3^{(r)} = 0.3(1 - P_4^{(r)}) \qquad r = 1, 2$$

$$P_4^{(1)} = 0.25$$

$$P_4^{(2)} = 0.6$$

From the above data, the average number of hops traversed by a message is

$$n_h = (1 - 0.25)/0.6 = 1.25$$

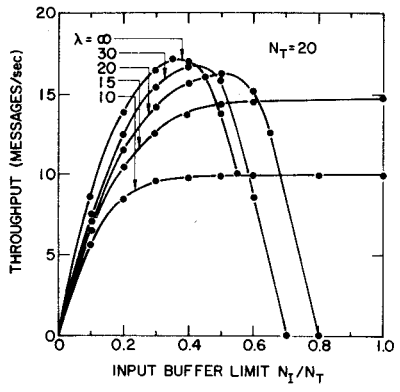


Fig. 3. Throughput versus input buffer limit ($N_T = 20$).

which is also the average ratio of transit messages to input messages at each node of the assumed homogeneous network.

An upper bound on the nodal throughput rate σ can be obtained by letting $N_T = N_I = \infty$ so that $B_I = B_T = 0$. In this case and under equilibrium conditions

$$\gamma = 1.25\sigma$$

and

$$\lambda = \sigma.$$

By inspection, we know that channel 1 is the bottleneck. The traffic intensity at channel 1 is equal to

$$(\lambda P_1^{(1)} + \gamma P_1^{(2)})/\mu_1 = (0.3\sigma + 0.16(1.25\sigma))/9.6$$

which must be less than one. Hence we obtain

$$\sigma < \sigma_{\max} = 19.2 \text{ messages/second.}$$

In Figure 3, we have plotted the throughput rate σ as a function of the input buffer limit (N_I/N_T) for $N_T = 20$. Note that for values of $\lambda < \sigma_{\max}$ (=19.2), σ increases monotonically as the input buffer limit is increased to one. For values of $\lambda > \sigma_{\max}$, σ increases to a maximum as the input buffer limit is increased, then turns down and decreases rapidly to zero. Note also that for a larger λ, the maximum occurs at a smaller input buffer limit. Similar results are shown in Figure 4 for $N_T = 40$. In this figure, the maximum achievable throughput is larger than that of Figure 3 for each λ. This observation is further demonstrated in Figure 5 for the case of $\lambda = \infty$. Also observe that as the total buffer pool (N_T) gets larger, the throughput curve has a wider region near the optimum, which is relatively flat. This is a desirable property which can be exploited in the design of input buffer limits. We shall return to this point later on.

In Figure 6, the throughput rate σ is shown as a function of the offered load λ. Notice that if the input buffer limit is properly designed, the throughput rate does not turn down as the offered load is increased to infinity.

In Figure 7, the equilibrium loss probabilities for input and transit messages are shown versus the input buffer limit for $N_T = 20$ and different values of λ. Note that for an input buf-

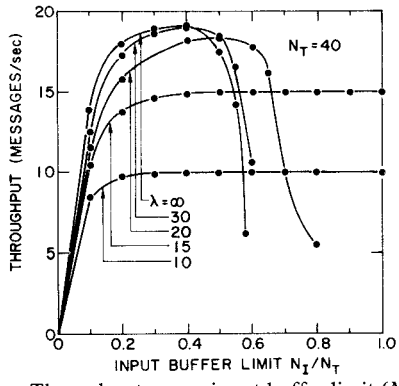


Fig. 4. Throughput versus input buffer limit (N_T = 40).

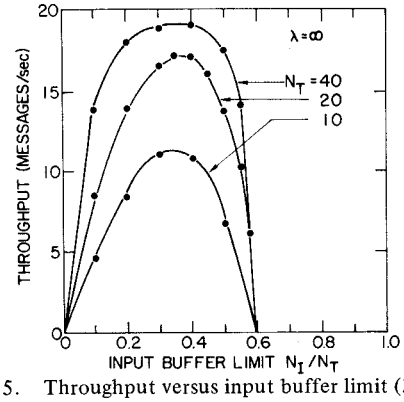


Fig. 5. Throughput versus input buffer limit (λ = ∞).

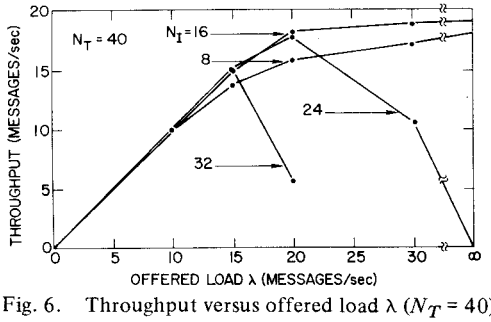


Fig. 6. Throughput versus offered load λ (N_T = 40).

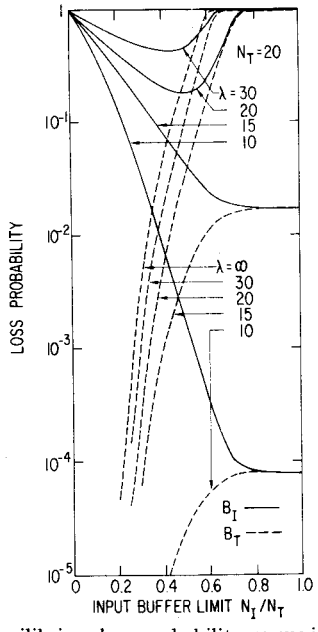


Fig. 7. Equilibrium loss probability versus input buffer limit (N_T = 20).

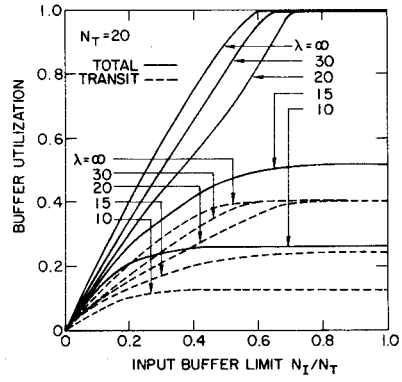


Fig. 8. Buffer utilization versus input buffer limit (N_T = 20).

fer limit corresponding to maximum throughput, B_I is typically more than an order of magnitude larger than B_T.

In Figure 8, buffer utilizations of transit and all messages are shown versus the input buffer limit for N_T = 20 and different values of λ.

4. DESIGN OF INPUT BUFFER LIMITS

From the above results, a rule of thumb for selecting the input buffer limit of a node is

$$(N_I/N_T) < \alpha_0 \tag{7}$$

where

α_0 = ratio of input message throughput to total message throughput of the node

$= \sigma/(\sigma + \gamma(1-B_T))$.

In the special case of a homogeneous network, α_0 is equal to $1/(1 + n_h)$. In the above example, this ratio is 0.44. Examining Figures 3-5, the throughput maximum corresponds to an input buffer limit of just under 0.44 for the heavily loaded cases ($\lambda > 20$) and for the lightly loaded cases ($\lambda \leqslant 20$) there is very little gain in throughput for choosing an input buffer limit larger than 0.44.

The rule of thumb in Eq. (7) can be interpreted as a *capacity law* that must be satisfied. Under the assumptions of a fixed network input traffic pattern and a fixed routing algorithm, the traffic ratio α_0 is uniquely determined for each node. An input buffer limit larger than α_0 is undesirable because, if in the event that the input buffer pools of all nodes in the network are completely filled with input messages, the network will not have enough buffers to accommodate the resulting transit messages (on the average)! In particular, this event has a high probability of occurrence when $\lambda > \sigma_{max}$; hence, the significant throughput degradation shown in Figures 3-5 when the capacity law is violated. It appears that this capacity law is a simple and robust condition that must be satisfied in much the same fashion as the $\rho < 1$ condition for a single server queue.

The input buffer limit of a node should be strictly less than α_0 to allow some margin for time and statistical fluctuations in user traffic as well as errors in our traffic estimates. We can envision two types of uncertainties: (1) in the offered load rate λ, and (2) in the estimated traffic ratio α_0.

The first type of uncertainty is not a big problem. Recall

from Figure 6 that as long as the capacity law is obeyed (N_I = 8 and 16), the throughput is unaffected even if $\lambda \to \infty$. If the capacity law is violated (N_I = 24, 32), any fluctuation in λ may cause unanticipated throughput degradation (although λ may be small in normal network operation).

The second type of uncertainty may be tackled in the following manner. Recall our earlier observation from Fig. 5 that as N_T gets large there is a relatively flat region surrounding the point of maximum throughput. As a result, the input buffer limit can be designed to be substantially smaller than α_0 without incurring much loss of throughput. For example, consider the N_T = 40 curve. The throughput is within 0.95 of the maximum for any input buffer limit between 0.22 and 0.46. Suppose the input buffer limit is designed to be 0.22. This implies that even if the actual traffic ratio turns out to be 0.22, instead of 0.44 that we assumed (an error of 100 percent), the above rule of thumb is still satisfied. Of course, if the actual traffic ratio turns out to be greater than 0.44, there will be some loss in throughput because the network is underutilized. As a result, we may want to select an input buffer limit somewhat larger than 0.22.

We note that $\alpha_0 = 1/(1 + n_h)$ is applicable only under the assumption of a homogeneous network. For a general nonhomogeneous network, α_0 has to be determined separately for each node by some other means. The capacity law in Eq. (7) remains a necessary (but not sufficient) condition for congestion control. Specific implementation algorithms based upon Eq. (7) are currently being investigated.

5. SIMULATION RESULTS

A four-node network with a completely connected topology was simulated. In the simulated network, messages flow from source to destination through end-to-end logical channels. A total of 44 logical channels are used, one for each of all 12 one-hop routes, all 24 two-hop routes and 8 of the three-hop routes. The simulation is different from the analytic model in several respects. First, the complete network of four nodes is simulated. Second, SDLC is simulated for data link control [6]. Third, all messages generated in the simulation have a fixed length (i.e., single packets).

The following parameters are assumed for each node:

$M = 3$

$\mu_0 = \infty$ messages/second (very fast nodal processor)

$\mu_1 = \mu_2 = \mu_3 = 1$ message/second

$\mu_4 = 11$ messages/second.

New messages are created for each logical channel according to a Poisson process.

For comparison, an equivalent analytic model of the simulated network was evaluated numerically. In the analytic model, we further assume

$t_1 = t_2 = t_3 = 5$ seconds

$v_1 = v_2 = v_3 = 1$ second.

Routing probabilities for input and transit messages in the analytic model are calculated from the one-hop, two-hop and three-hop routes used in the simulated network. For the network considered, they are the same at each of the four nodes and are given by

$$P_1^{(1)} = P_3^{(1)} = 4/11, P_2^{(1)} = 3/11, P_4^{(1)} = 0;$$

$$P_1^{(2)} = P_3^{(2)} = 1/7, P_2^{(2)} = 4/21, P_4^{(2)} = 11/21.$$

The ratio α_0 is the same at each node and is calculated to be

$$\alpha_0 = 11/32 = 0.344.$$

An upper bound on the nodal throughput rate (obtained as in Section 3 above) is

$$\sigma < \sigma_{\max} = 11/7 = 1.57 \text{ messages/second.}$$

The throughput per node is plotted versus input buffer limit in Fig. 9. Consider the two cases for which both analytic and simulation results are shown: (1) $\lambda = 2.2$, $N_T = 30$, and (2) $\lambda = 1.1$, $N_T = 50$. We note the following discrepancies.

First, the part of the throughput curve (given by analysis) with negative slope is not realizable in simulation. For example, with $\lambda = 2.2, N_T = 30$ and an input buffer limit of 0.4, simulation shows that the network throughput degrades very rapidly to zero as the network enters into a store-and-forward deadlock. This behavior should be expected from the analytic results. Since the loss probabilities under these conditions are very high (0.1 – 1.0), a deadlock will thus occur very rapidly in the absence of any deadlock-avoidance scheme. The assumption of equilibrium also breaks down.

Second, throughput values given by the analytic model are in most cases pessimistic compared to corresponding simulation values. This can be explained by the differences between the simulated network and the equivalent analytic model. In particular, while constant message length is assumed in simulation, exponentially distributed message length is assumed in the analysis. (Queuing systems characterized by a larger variance typically have worse performance.) Also, as we discussed earlier in Section 2, the single node approximation of a homogeneous network in the analysis would give rise to somewhat pessimistic results.*

On the other hand, our simulation results support conclusions drawn from our analytic results. First, the capacity law for selecting the input buffer limit of a node using the ratio α_0 as an upper bound is valid. Second, as predicted by the analysis, the throughput curve does become more "square-shaped" as N_T is increased thus offering a larger safety margin for error. (See Fig. 9.) Third, when the offered load λ (=1.1) is less than $\sigma_{\max}$ (=1.57) the throughput curve does not turn down. There is very little gain in throughput for selecting an

* A second simulation model was later developed with (i) a data link control protocol using selective retransmission, and (ii) exponentially distributed message length (but without Kleinrock's independence assumption [16]). In this case, the simulated throughput results were found to be slightly smaller than the analytic results. The discrepancy in each instance was about 5% or less.

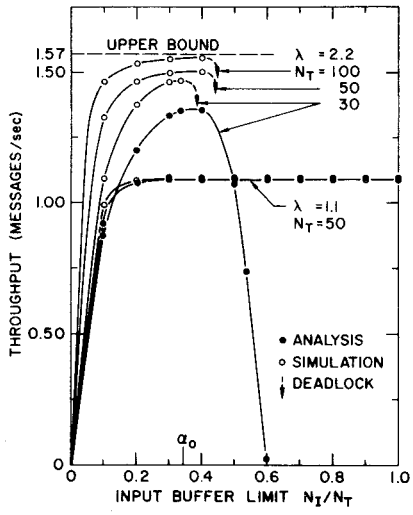


Fig. 9. Simulation results.

input buffer limit larger than α_0. Moreover, if the capacity law is satisfied, fluctuations in λ (from 1.1 to 2.2 in Fig. 9) do not give rise to throughput degradation.

6. CONCLUSIONS

We have studied input buffer limits for congestion control of store-and-forward networks. A general analytic model has been formulated. Numerical results for a specific example (under the homogeneous network assumption) suggest strategies for the design of input buffer limits to achieve the maximum network throughput as well as to provide a safety margin for uncertainties in traffic assumptions. Conclusions drawn from the analytic results are supported by simulation results for a four-node homogeneous network. Specifically, a useful capacity law was found. These results indicate that input buffer limits which satisfy the capacity law are a simple and effective means of network congestion control. Further simulation studies are underway to substantiate this conclusion in general and to investigate various implementation algorithms. Techniques for both static and dynamic determination of the traffic ratio α_0 in a nonhomogeneous network and time-varying traffic environment are being studied. Finally, we must keep in mind that input buffer limits are being considered as a solution to the network-wide congestion problem only. End-to-end controls are still necessary for purposes of source-sink speed synchronization, data integrity, etc. Also, although the probability of store-and-forward deadlocks can be significantly reduced as a result of properly designed input buffer limits, deadlock-free operation is not guaranteed.

ACKNOWLEDGMENT

The authors thank the anonymous referees for their constructive comments.

REFERENCES

[1] Kahn, R. and W. Crowther, "Flow control in a Resource-Sharing Computer Network," *IEEE Trans. Commun.*, Vol. COM-20, June 1972.

[2] Davis, D. W., "The Control of Congestion in Packet Switching Networks," *IEEE Trans. Commun.*, Vol. COM-20, June 1972.

[3] Price, W. L., "Data Network Simulation Experiments at the National Physical Laboratory 1968-1976," *Computer Networks*, Vol. 1, 1977.

[4] Giessler, A., J. Haenle, A. Koenig and E. Pade, "Packet Networks with Deadlock-Free Buffer Allocation–An Investigation by Simulation," GMD report, Darmstadt, Germany, 1976.

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

[6] IBM Corp., *Systems Network Architecture General Information*, GA27-3102-0, January 1975.

[7] Pouzin, L., "Presentation and Major Design Aspects of the CYCLADES Computer Network," *Data Networks: Analysis and Design*, Third Data Communications Symposium, St. Petersburg, Florida, Nov. 1973.

[8] Cerf, V. and R. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Trans. Commun.*, Vol. COM-22, May 1974.

[9] Lam, S. S., "A New Measure for Characterizing Data Traffic," *IEEE Trans. Commun.*, Vol. COM-26, Jan. 1978.

[10] Chou, W. and M. Gerla, "A Unified Flow and Congestion Control Model for Packet Networks," *Proc. Third International Conf. on Computer Communication*, Toronto, August 1976.

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

[12] Pennotti, M. C. and M. Schwartz, "Congestion Control in Store and Forward Tandem Links," *IEEE Trans. Commun.*, Vol. COM-23, Dec. 1975.

[13] Wong, J. and M. Unsoy, "Analysis of Flow Control in Switched Data Networks," *Proc. IFIP Congress*, Toronto, August 1977.

[14] Lam, S. S., "Store-and-Forward Buffer Requirements in a Packet Switching Network," *IEEE Trans. Commun.*, Vol. COM-24, April 1976.

[15] Schweitzer, P. J. and S. S. Lam, "Buffer Overflow in a Store-and-Forward Network Node," *IBM Journal of Res. and Develop.*, Vol. 20, Nov. 1976.

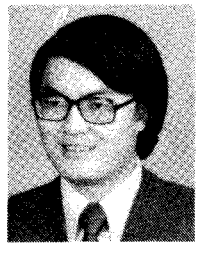
[16] Kleinrock, L., *Communication Nets: Stochastic Message Flow and Delay*, McGraw Hill, New York 1964.

[17] Lam, S. S., "Queueing Networks with Population Size Constraints," *IBM Journal of Res. and Develop.*, Vol. 21, July 1977.

[18] Reiser, M. and H. Kobayashi, "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms," *IBM Journal of Res. and Develop.*, Vol. 19, May 1975.

[19] Baskett, F., K. M. Chandy, R. R. Muntz and F. G. Palacios, "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," *J. ACM*, Vol. 22, April 1975.

★



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

From 1972 to 1974, he was a Postgraduate Research Engineer with the ARPA Network project at UCLA and did research on satellite packet communication. From 1974 to 1977, he was a research staff member with the IBM Thomas J. Watson Research Center, Yorktown Heights, New York where he worked on the performance analysis of various aspects of packet switching networks, SNA and satellite net-

works, as well as queuing network theory. Since September 1977, he has been an Assistant Professor of Computer Sciences at the University of Texas at Austin. His current research interests include computer systems modeling and analysis, computer-communication networks and packet broadcasting networks.

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

Martin Reiser was born in Zurich, Switzerland, in 1943. He received the M.S. and Ph.D. degrees from the ETH (Swiss Federal Institute of Technology). He joined the Research Division of IBM in 1968, where he held various staff and managerial positions in the areas of large scale numerical computation, performance evaluation of computer systems and computer communications. He spent six years in the United States and has held the position of Manager of Communications in the IBM Zurich Research Laboratory since early 1979.