An Experimental Study of the Congestion Control of Packet Communication Networks

S.S. Lam and Y. L. Lien
University of Texas at Austin, USA

ABSTRACT

An experimental study was conducted using a network simulator to investigate the performance of store-and-forward packet communication networks as a function of: the network resource capacities (channels, buffers), the network load (number of virtual channels, virtual channel loads), protocols (flow control, congestion control, routing) and protocol parameters (virtual channel window sizes, input buffer limits). Some results from our study are shown. Network design strategies for the control of load fluctuations are proposed and discussed.

I. INTRODUCTION

The objective of a packet communication network is to reliably deliver packets from their sources to their destinations within acceptable time delays. Thus an important performance measure of a packet network is its throughput rate, in packets delivered per second.

Throughput is generated when individual packets progress through the network, following finite (preferably acyclic) paths. To generate a packet's worth of throughput, the network must first admit a packet and then allocate to it a set of resources consisting of the communication channels as well as one buffer at each node along its route from source to destination. (We shall not consider various types of logical resources which are also needed, such as control blocks, sequence numbers, etc.) A complete allocation of resources needed by a packet before its admittance and subsequent journey is deemed wasteful. Almost all packet networks employ a store-and-forward protocol, whereby a partial allocation of resources will enable packets to progress through the network; specifically, a packet residing in node i can proceed if it has acquired the resources of a buffer in node i, the channel (i,j) on its route, and a buffer in node j.

It is well-known that partial allocation of resources to concurrent "processes" may result in a circular-wait condition, under which none of the processes can satisfy its resource needs and progress [1]. The condition is known as a store-and-forward deadlock in packet networks [2]; the throughput rate of a deadlocked network is zero. Buffer allocation algorithms have been proposed [3,4] for the prevention of store-and-forward deadlocks. It has been proved that using these algorithms at least one packet in the network can satisfy its resource needs at any time. This ensures that the network throughput rate will never be zero.

The objective of this experimental study is to investigate conditions for packet networks to operate at a high throughput rate instead of just ensuring that it will not become zero. The impact of network protocols for flow control, congestion control, and routing on the network throughput rate is investigated. The network performance as a function of resource capacities (channels and buffers), network load (number of virtual channels, virtual channel load), and protocol parameters (window sizes for virtual channel flow control, input buffer limits for network congestion control) is investigated. Some of our results and conclusions are reported below. The reader is referred to [5] for further details and additional results.

II. FLOW AND CONGESTION CONTROL PROTOCOLS

The network congestion phenomenon can be described as follows. A packet in transit within a network is at any time either enabled (resource requirements for progress met) or blocked (resource requirements not met). The number $y(t)$ of enabled packets in the network at time t depends upon: nodal buffer capacities, the

number of packets in transit within the network, and the distribution of these packets over channel queues. The last two in turn depend upon the network load and the network's routing, and flow and congestion control protocols.

The network throughput rate is directly proportional to the expected number of enabled packets under steady-state conditions. The maximum possible value of y(t) is the number of communication channels in the network, assuming a nontrivial number of buffers at each node. On the other hand, there are two conditions under which y(t) takes on small values. First, there are few packets in the network due to a small network load. Second, there are many packets in the network competing for resources; however, their distribution over the channel queues are such that few packets can satisfy their resource requirements for progress (the network is congested!).

We shall consider networks that provide virtual channels between packet sources and sinks. The virtual channels are end-to-end flow controlled. A comprehensive survey of flow control protocols can be found in [6]. An important function of such end-to-end protocols is synchronization of the source input rate to the sink acceptance rate. All of them work by limiting the number of packets that a virtual channel can have in transit within the network; this number we shall refer to as the virtual channel window size.

A separate network congestion control protocol is often necessary for the network as a whole. Any network congestion control protocol must effectively reduce input into the network to alleviate temporary overloads on the network. We identify 3 basic types of congestion control protocols. The isarithmic principle proposed by Davis [7] and studied by Price [8] provides the above function by setting a limit on the number of packets permitted inside a network. The objective of limiting the admission of packets into a network can also be achieved by reducing the window sizes of virtual channels or by shutting down some virtual channels entirely.

The third type of congestion control protocols is called input buffer limits that is of special interest in this paper. Network nodes are required to differentiate between "input packets" generated by local sources and "transit packets" routed to them by other nodes. A limit is imposed upon the fraction of buffers in the node that input packets can occupy. This fraction is called the input buffer limit (IB limit or IBL) of the node. No limit is placed on the number of buffers that transit packets can occupy.

The advantage of favoring transit packets over input packets was observed by Price [8]. A similar idea was discussed by Chou and Gerla [9]. The use of IB limits was explored quite extensively in the GMD simulation study [4]. Our experimental study reported below covers different grounds from that of the GMD simulation study. Specifically, although the GMD study explored the use of IB limits for congestion control, it was not known how to design such limits nor were the conditions under which IB limits are effectively demonstrated.

In [10], the performance of packet networks employing IB limits for congestion control was analyzed by modeling them as an extended class of queueing networks [11]. It was found that when the network load is large, there is a critical value for the IB limits beyond which the throughput capacity of the network is seriously impaired. This critical value we shall refer to as the IB capacity. The explanation for the drastic degradation in the network throughput rate when IB limits exceed the IB capacity turns out to be an intuitive one. For each new packet that the network admits into an input buffer, additional buffers are needed elsewhere for the packet's subsequent journey to its destination. Therefore, there is a natural ratio of the number of input buffers to the total number of buffers in the network that serves as an upper bound for IB limits. Suppose IB limits are designed to be larger than the IB capacity. It will occur that (almost) all input buffers are filled by input packets; a likely occurrence when the network is temporarily heavily loaded. The network will subsequently not have enough buffers to satisfy the demands of the resulting transit packets. The number of packets which are enabled becomes very small and the network throughput rate decreases to a small value (or zero if the network is not deadlock-free).

A significant observation in [10] is that IB limits can be made much smaller than the IB capacity without sacrificing much network throughput (from the maximum throughput rate assuming infinite buffers).

III. THE EXPERIMENTAL STUDY

Our experiments were performed with a simulation program written in the Pascal language using the discrete-event simulation methodology. It currently runs on the University of Texas CDC Cyber 170/750 system. (An earlier version also exists and runs on a DEC-10 system.)

In the experiments for this particular study, the traffic source of each virtual channel is assumed to be a Poisson process with a rate λ (to be referred to as the virtual channel load). Each message generated consists of a single fixed length packet. Newly generated packets that cannot be admitted into the network are lost instantaneously. At a communication channel speed of, say 50 Kbps, both nodal processing times and sink absorption times of packets are negligible compared to channel delays. They were assumed to be zero in the present study. Fixed rout-

ing using a table look-up procedure is implemented. All queues employ a FCFS discipline.

The data link control protocol simulated is based upon that of ARPANET [12]. It is assumed that packet errors due to channel noise are negligible. Packets are not positively acknowledged solely because they have not been accepted due to buffer, flow or congestion control constraints. It is further assumed that positive acknowledgments are always accepted, even when the data packet containing the positive acknowledgment has been rejected.

Presently, end-to-end acknowledgments are not explicitly modeled so that when a packet is delivered to the sink of a virtual channel this is known to the source right away. End-to-end acknowledgments may be implemented in the simulator fairly easily but are deemed to add unnecessarily to the cost of simulation. The impact of an end-to-end acknowledgment delay is to make the effective window size somewhat smaller than what is specified presently.

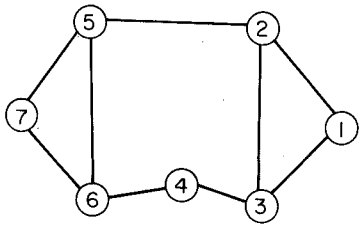


Fig. 1. An 7-node network.

The first network used in our study, shown in Figure 1, consists of 7 nodes and 9 full-duplex links. Between each source-sink pair of nodes, the first and second shortest routes between them are selected. (Routes of equal length are chosen randomly.) Altogether 84 different routes are used. When each route is used by k virtual channels, we shall say that the network load consists of 84 x k virtual channels. The number of virtual channels using each communication channel in the 7-node network varies from 7 to 15 (assuming one virtual channel per route).

The communication channel speed is assumed to be 50 packets per second; this corresponds to, for instance, a packet size of 1000 bits and a channel speed of 50 Kbps. The number N_T of store-and-forward buffers is the same for each node. The window size of a virtual channel is specified as an integer multiple of the virtual channel path length (in number of links). The motivation for this is to reduce the number of parameters that we need to consider. Its effect is to minimize the variation in the throughput rates of individual virtual channels.

The Effect of Increasing the Virtual Channel Load λ

For the moment, IB limits for network congestion control are not used. In Figure 2, the network throughput rate is plotted as a function of the number N_T of buffers at each node for λ = 1, 2, 10 packets per second. The network supports 84 x 2 virtual channels. Each virtual channel has a window size of 2 x path length.

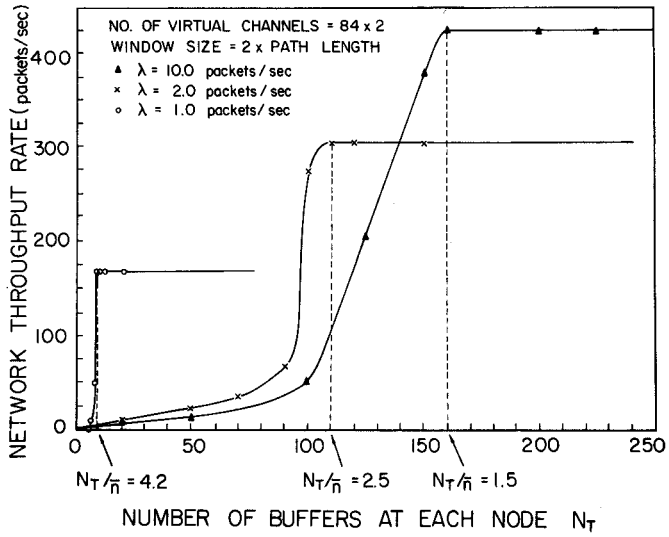


Fig. 2. Network throughput rate as a function of λ and N_T.

Note the drastic decrease in network throughput rate when N_T is less than a certain threshold value in each case. In other words, the network requires a minimum number of buffers before virtual channel windows could provide enough input control for the network to satisfy the resource needs of its admitted packets.

For λ = 1, 2 and 10 packets per second, the threshold values of N_T are 9, 110 and 160! Let $\bar{n}$ denote the average number of packets in a node, assuming that the network nodes have infinitely many buffers; $\bar{n}$ can be either calculated using a queueing network model or obtained from simulation. Now consider the ratio of the threshold values of N_T to $\bar{n}$ for each λ. That ratio is 4.2, 2.5 and 1.7 respectively for λ = 1, 2 and 10 (see Figure 2). Note that the ratio actually decreases as $\bar{n}$ increases.

λ is the rate at which packets are offered to a virtual channel. As λ becomes large, say λ = 2 packets per second, the number of packets that a virtual channel has in transit will be equal to the window size much of the time. As a result the rate at which packets are admitted by a virtual channel levels off very quickly as λ increases. Further increase in λ (say from 10 to ∞) will only have a marginal effect on the network loading; in this way, virtual channel windows provide an input control function for the network.

As expected the maximum throughput rate of each curve in Figure 2 increases as the

virtual channel load λ increases, assuming the provision of sufficient buffers. Given a modest supply of buffers (say 20 - 100), Figure 2 indicates that $\lambda = 1$ should be the expected load on the network in the long run, $\lambda = 2$ would be a moderate overload while $\lambda = 10$ would be a heavy overload on the network. With no other network congestion control protocol, to guard against a temporary overload of $\lambda = 2$, the network will require 110 buffers per node; to guard against a temporary overload of $\lambda = 10$, the network will require 160 buffers per node.

The Design of IB Limits for Congestion Control

In [10], it was discovered that for homogeneous networks consisting of nodes with the same channel configuration and traffic demands, the input buffer limit (IBL) of each node should be the same and satisfy

$$IBL < 1/\bar{H}$$

where $\bar{H}$ is the mean path length of packets (in number of nodes) in the network. The above design rule was found to work well by both analysis and simulation.

When we first turned our attention to designing IB limits for general nonhomogeneous networks, we treated the problem as a capacity assignment problem. We design IB limits for individual nodes to match their traffic demands. We investigated several such heuristic design algorithms and found that when networks with a small number of buffers were considered, none of these algorithms was robust (i.e., worked well for different network configurations, traffic patterns and nodal buffer capacities).

We subsequently discovered that despite the consideration of nonhomogeneous networks, a very robust IB limit design strategy is still uniform assignment: using the same IB limit for each node given by

$$IBL = \alpha/\bar{H} \qquad (1)$$

where α is a scaling factor less than 1 needed to account for the "traffic imbalance" in a nonhomogeneous network. In general, as to be shown below, the applicable α decreases as the network traffic imbalance increases (which will also be aggravated by an increase in the network load).

Design Strategies to Control Temporary Network Overloads

In Figure 3, we have plotted the same network throughput curves in Figure 2 together with new curves obtained using the same network loads but with the network employing IB limits for congestion control.

The experiments that we conducted lasted for about 150 seconds of simulated time each. For those cases in which the network throughput rate was not seriously degraded, each virtual channel transported close to 300 packets each (on the average).

Recall that with a modest supply of buffers (20 - 100), $\lambda = 1$ corresponds to the expected network load, $\lambda = 2$ is a moderate overload while $\lambda = 10$ is a heavy overload on the network. The largest applicable value of α in Eq.(1), for a simulation duration of 150 seconds, is 1 for $\lambda = 1$, 0.7 for $\lambda = 2$ and 0.4 for $\lambda = 10$. Note that as λ increases, α should be decreased.

Suppose N_T is 50. Without IB limits, a substantial increase in λ, to say $\lambda = 2$, will cause the network throughput rate to degrade badly. However with IB limits using $\alpha = 0.7$, the network can withstand an overload of $\lambda = 2$ packets per second for a least 150 seconds. If IB limits corresponding to $\alpha = 0.4$ are used, then the network can withstand an overload of $\lambda = 10$ packets per second for at least 150 seconds.

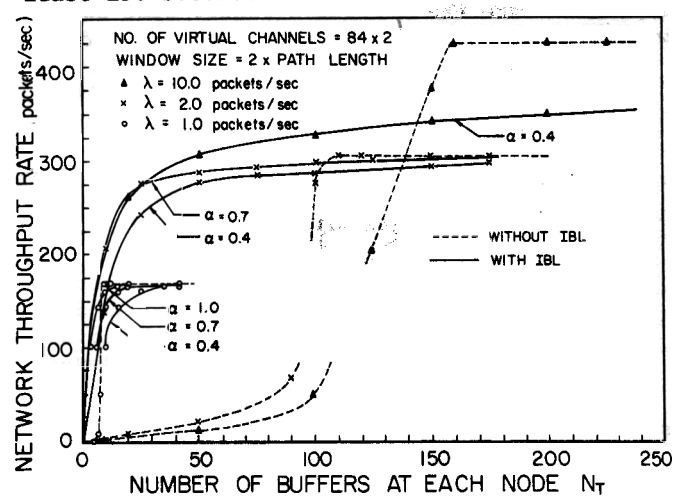


Fig. 3. Input buffer limits for overloads due to fluctuations in λ.

An important observation here is that IB limits provide protection against large fluctuations in the virtual channel load λ. This protection is obtained with little or no degradation in the network throughput performance when the network is not congested even though the IB limits are fixed assigned (non-adaptive).

While virtual channel windows provide some input control for the network when λ becomes large, they provide little control when the network overloading is from an increase in the number of virtual channels. An increase in the number of virtual channels will happen because in most packet networks, virtual channels are established by nodes without any central control. Figure 4 shows that without IB limits as the number of virtual channels increases from 84 to 168, the network buffer requirement increases very rapidly. Throughput curves for

virtual channel load λ increases, assuming the provision of sufficient buffers. Given a modest supply of buffers (say 20 - 100), Figure 2 indicates that $\lambda = 1$ should be the expected load on the network in the long run, $\lambda = 2$ would be a moderate overload while $\lambda = 10$ would be a heavy overload on the network. With no other network congestion control protocol, to guard against a temporary overload of $\lambda = 2$, the network will require 110 buffers per node; to guard against a temporary overload of $\lambda = 10$, the network will require 160 buffers per node.

The Design of IB Limits for Congestion Control

In [10], it was discovered that for homogeneous networks consisting of nodes with the same channel configuration and traffic demands, the input buffer limit (IBL) of each node should be the same and satisfy

$$IBL < 1/\bar{H}$$

where $\bar{H}$ is the mean path length of packets (in number of nodes) in the network. The above design rule was found to work well by both analysis and simulation.

When we first turned our attention to designing IB limits for general nonhomogeneous networks, we treated the problem as a capacity assignment problem. We design IB limits for individual nodes to match their traffic demands. We investigated several such heuristic design algorithms and found that when networks with a small number of buffers were considered, none of these algorithms was robust (i.e., worked well for different network configurations, traffic patterns and nodal buffer capacities).

We subsequently discovered that despite the consideration of nonhomogeneous networks, a very robust IB limit design strategy is still uniform assignment: using the same IB limit for each node given by

$$IBL = \alpha/\bar{H} \qquad (1)$$

where α is a scaling factor less than 1 needed to account for the "traffic imbalance" in a nonhomogeneous network. In general, as to be shown below, the applicable α decreases as the network traffic imbalance increases (which will also be aggravated by an increase in the network load).

Design Strategies to Control Temporary Network Overloads

In Figure 3, we have plotted the same network throughput curves in Figure 2 together with new curves obtained using the same network loads but with the network employing IB limits for congestion control.

The experiments that we conducted lasted for about 150 seconds of simulated time each. For those cases in which the network throughput rate was not seriously degraded, each virtual channel transported close to 300 pa kets each (on the average).

Recall that with a modest supply of bu fers (20 - 100), $\lambda = 1$ corresponds to the expected network load, $\lambda = 2$ is a moderate overload while $\lambda = 10$ is a heavy overload o the network. The largest applicable value α in Eq.(1), for a simulation duration of seconds, is 1 for $\lambda = 1$, 0.7 for $\lambda = 2$ and for $\lambda = 10$. Note that as λ increases, α sh be decreased.

Suppose N_T is 50. Without IB limits, substantial increase in λ, to say $\lambda = 2$, w cause the network throughput rate to degrad badly. However with IB limits using $\alpha = 0.$ the network can withstand an overload of $\lambda = 2$ packets per second for a least 150 se onds. If IB limits corresponding to $\alpha = 0.4$ are used, then the network can withstand an overload of $\lambda = 10$ packets per second for at least 150 seconds.

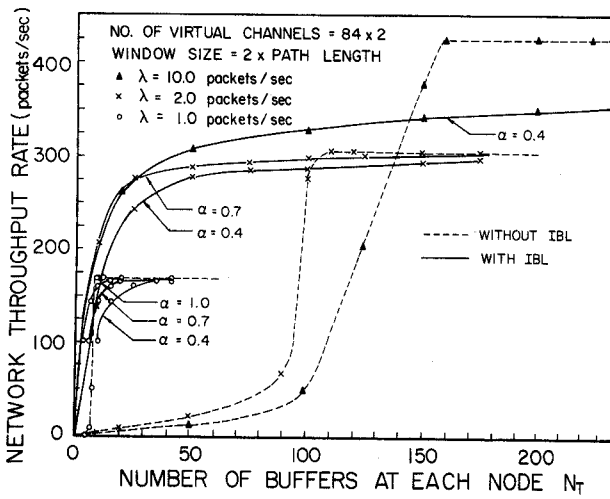


Fig. 3. Input buffer limits for overloads du to fluctuations in λ.

An important observation here is that IB limits provide protection against large fluct ations in the virtual channel load λ. This protection is obtained with little or no deg radation in the network throughput performanc when the network is not congested even though the IB limits are fixed assigned (non-adaptiv

While virtual channel windows provide so input control for the network when λ becomes large, they provide little control when the network overloading is from an increase in th number of virtual channels. An increase in t number of virtual channels will happen because in most packet networks, virtual channels are established by nodes without any central con- trol. Figure 4 shows that without IB limits as the number of virtual channels increases from 84 to 168, the network buffer requirement increases very rapidly. Throughput curves fo

for the same network loads are also shown when IB limits are used.

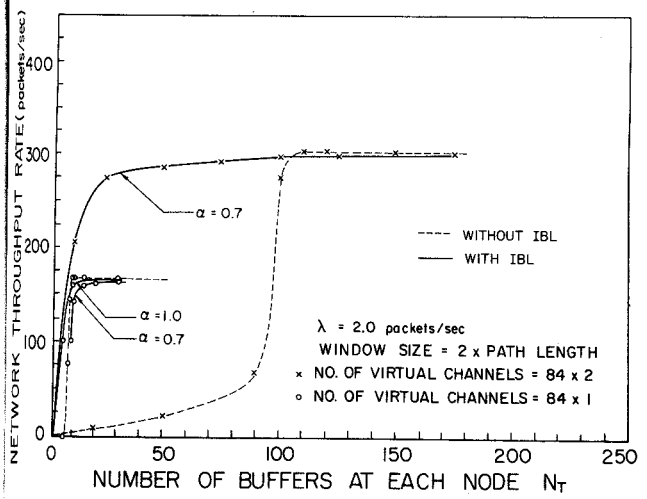


Fig. 4. Input buffer limits for overloads due to fluctuations in number of virtual channels.

Obviously one way to control network congestion and prevent throughput degradation is by holding down the number of virtual channels permitted in the network. This can be accomplished by requiring the establishment of a new virtual channel to be authorized by a central controller. An alternative is to provide a network congestion control protocol, such as IB limits.

Figure 4 shows that with N_T within the range of 20 to 100, input buffer limits using $\alpha = 0.7$ will enable the network to withstand a overload of 84 x 2 virtual channels for at least 150 seconds. This protection is achieved with a static assignment of input buffer limits. However, when the network is not congested (because of more buffers or a smaller load) there is little or no throughput degradation caused by the statically assigned input buffer limits.

Network loads corresponding to the use of different virtual channel window sizes are considered in Figure 5, both with and without the use of IB limits for congestion control. The largest applicable values of α that can be used (for a simulation duration of 150 seconds) are 1, 0.7 and 0.6 respectively for window sizes equal to 1x, 2x, and 3x path length.

We make two observations. First, Figure 5 shows that the strategy of reducing virtual channel window sizes when network congestion occurs will help; but Figure 5 also shows that the use of IB limits is more effective. Second, the window size of a virtual channel is typically negotiated between the source-destination pair of nodes and not subject to any form of central control. As a result of such distributed, possibly uncoordinated, decisions and because network users will demand large virtual channel window sizes to achieve their desired throughput rates, the overload condition of having a large number of virtual channels with large window sizes will occur. Figure 5 shows that a network overload due to all virtual channels having a window size equal to 3 times its path length, can be taken care of by installing input buffer limits using $\alpha = 0.6$. Note again that the network throughput degradation due to statically assigned IB limits with $\alpha = 0.6$ is quite small when the network is not congested.

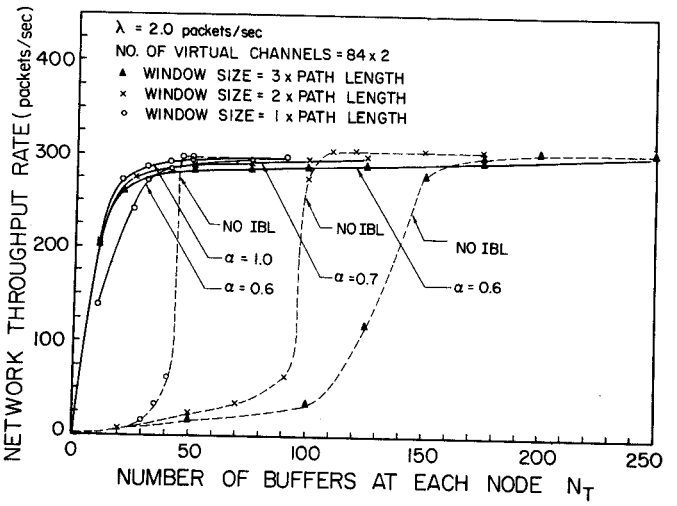


Fig. 9. Input buffer limits for overloads due to fluctuations in virtual channel window size.

IV. CONCLUSIONS

In general, the set of virtual channels constitutes the network load requiring the use of the network's channel and buffer resources. The routing, flow and congestion control protocols allocate and regulate such demands on the network.

Let us review the key variables affecting the performance of packet networks.

The rate λ models the load on a virtual channel that is a characteristic of the traffic source and is not subject to control. (The effect of λ was considered in Figure 2.)

The number and distribution of virtual channels are also not easily controlled for networks in which virtual channels are established and terminated by individual node pairs. If, however, a central controller is used to authorize the creation of new virtual channels, then overloads due to too many virtual channels can be prevented. (See Figure 4.)

Virtual channel windows sizes are useful for controlling the throughput rates of individual virtual channels. Figure 5 shows that a means of network congestion control is to adaptively reduce virtual channel window sizes.

The implementation of such a strategy requires either a central controller or a distributed algorithm that can effectively coordinate the actions of individual nodes. (Such an algorithm is not presently available. We encounter here the same difficulty present in the design of an effective algorithm for redistributing empty containers in an isarithmic protocol.)

We found that IB limits are effective for controlling short-term overloads on a network (due to time or statistical load fluctuations). We also found that the uniform assignment strategy of using the same IB limit at each node with

$$IBL = \alpha/\bar{H}$$

where $\bar{H}$ is the mean path length of packets and $\alpha < 1$, is an effective and robust method of network congestion control. Load fluctuations due to changes in the virtual channel loads, number of virtual channels, and virtual channel window sizes can be handled using IB limits designed with an appropriate choice of α. The value of α depends upon two considerations, namely, the severity and time duration of the overload being designed for. We found that networks using IB limits with $\alpha = 0.4$ could withstand very severe overloads for at least 150 seconds. (See Figures 3 - 5.)

If the network load has changed, it is desirable to improve the routing to reduce the variance in communication channel utilizations. This is the objective of optimal routing to minimize average network delay. We found that improved routes will also enhance the effectiveness of IB limits for network congestion control.

We found that IB limits are effective and "inexpensive" for controlling occasional short-term overloads on a network. However, if increases in the network load are on a long-term basis, then instead of relying on IB limits the network should be equipped with more resources (channels, buffers) to handle the larger load.

REFERENCES

[1] Coffman, E., M. Elphick and A. Shoshani, "System Deadlocks," Computing Surveys, June 1971.

[2] Kahn, R. and W. Crowther, "Flow Control in a Resource-Sharing Computer Network," IEEE Trans. on Commun., June 1972.

[3] Guenther, K. D., "Prevention of Buffer Deadlocks in Packet-Switching Networks," IFIP-IIASA Workshop on Data Communications, Luxenburg, Austria, 1975.

[4] Giessler, A., J. Haenle, A. Konig and E. Pade, "Free Buffer Allocation - An Investigation by Simulation," Computer Networks, 1978

[5] Lam, S. S. and Y. L. Lien, "An Experimental Study of the Congestion Control of Packet Communication Networks," Dept. of Comp. Sci., Univ. of Texas at Austin, Tech. Rep. TR-142, Mar. 1980.

[6] Gerla, M. and L. Kleinrock, "Flow Control: A Comparative Survey," IEEE Trans. on Commun., April 1980.

[7] Davis, D. W., "The Control of Congestion in Packet Switching Networks," IEEE Trans on Commun., June 1972.

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

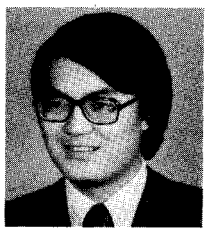
[9] Chou, W. and M. Gerla, "A Unified Flow and Congestion Control Model for Packet Networks," Proc. Third Int. Conf. on Computer Communication, Toronto, August 1976.

[10] Lam, S. S. and M. Reiser, "Congestion Control of Store-and-Forward Networks by Input Buffer Limits - An Analysis," IEEE Trans. on Commun., Jan. 1979.

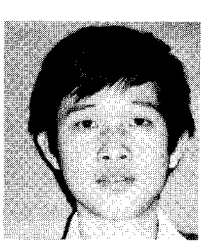
[11] Lam, S. S., "Queueing Networks with Population Size Constraints," IBM Journal of Res. and Develop., July 1977.

[12] McQuillan, J., W. Crowther, B. Cosell, D. Walden and F. Heart, "Improvements in the Design and Performance of the ARPA Network," AFIPS Conf. Proc., Dec. 1972.

Acknowledgment This work was supported by National Science Foundation Grant No. ENG78-01803.



Simon S. Lam is Associate Professor of Computer Sciences at the University of Texas at Austin. He received the BSEE degree from Washington State University in 1969 and the M.S. and Ph.D. degrees in engineering from UCLA in 1970 and 1974 respectively. From 1974 to 197 he was with the IBM T.J. Watson Research Center at Yorktown Heights, N.Y. His current research interests include computer-communication networ and protocols, and computer systems performance analysis.



Yeong-chang Luke Lien is a doctoral student in the University of Texas at Austin Department of Computer Sciences. He received the B.S. degree in physics from Fu-Jen University, Taiwan in 1973 and the M.S. degree in physics from the University of Kansas in 1977.