AN EXPERIMENTAL STUDY OF THE CONGESTION CONTROL OF PACKET COMMUNICATION NETWORKS*

Simon S. Lam and Y. Luke Lien

TR - 142

March 1980

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

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

Abstract

An experimental study was conducted using a network simulator to investigate the performance of 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 size, input buffer limits). Performance characteristics are shown and the design of input buffer limits for network congestion control, virtual channel window size and nodal buffer capacity addressed. Network design strategies for the control of load fluctuations are proposed and discussed.

1. 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. Packets admitted into the network can be viewed as concurrent "processes" requiring different types of resources to progress. The packet buffers at a network node form one type of resource. There are different types of buffer resources corresponding to different nodes in the network. The set of communication channels (usually just one) that transport packets from one node to another node is another type of resource. There are different types of channel resources corresponding to different communication links.

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*. A complete allocation of resources needed by a packet before its admittance and subsequent journey is deemed wasteful. Almost all packet

---

*With various high-level network protocols (not considered herein) other types of resources are needed, such as, logical channels, sockets, control blocks, sequence numbers, etc. Nodal processors are not considered because nodal processing delays are typically much smaller than communication channel delays.

networks employ a store-and-forward protocol, whereby once a node has accepted a packet, it attempts to forward the packet to the succeeding node on the packet's route. It also buffers a copy of the packet. The buffer is released only after a positive acknowledgment has been received from the succeeding node indicating successful receipt and acceptance of the packet. With the store-and-forward protocol, 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. (We assume that packets follow fixed routes.)

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.

The experimental performance results illustrate principles for designing input buffer limits, virtual channel window sizes, and nodal buffer capacities. A summary of our conclusions on network design strategies follows:

(1) A congestion control protocol is not needed if a network has adequate resources (channel and buffer capacities) for its maximum possible load.

(2) If a network has adequate resources for its expected load but is subject to load fluctuations, then a congestion control protocol is a much less expensive solution than trying to provide adequate resources to handle the occasional short-term overloads.

(3) A congestion control protocol should be implemented with the objective of controlling temporary overloads only. Long-term increases in the network load can only be handled with more resources.

(4) Better routing that distributes traffic to reduce the variance of channel utilizations will enhance the effectiveness of input buffer limits for network congestion control.

The Network Congestion Phenomenon

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 queues for channels. The last two in turn depend upon the network load and the network's routing, and flow and congestion control protocols.

The maximum possible value of y(t) is the number of communication channels in the network (given a nontrivial number of buffers at each node). On the other hands, 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 <u>congested</u>!).

The network throughput rate is directly proportional to the expected number $\bar{y}$ of enabled packets under steady-state conditions*. Figure 1 is a qualitative picture of the network throughput rate, given that the network has n packets in transit, plotted as a function of n.

2. FLOW AND CONGESTION CONTROL PROTOCOLS

We shall consider networks that provide virtual channels between packet sources and sinks. The virtual channels are end-to-end flow controlled. Examples of end-to-end flow control protocols are SNA pacing [5,6], RFNM in the ARPANET [7], and various window mechanisms [8,9]. 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

*If the mean path length of packets is fixed and all channels have the same capacity c, then the network throughput rate is $(\bar{y}c)$/mean path length.

channel can have in transit within the network. Suppose L_i is the maximum number of packets for virtual channel i (called its window size) and the network has K virtual channels. The maximum number of packets permitted to enter the network is thus

$$n_{max} = L_1 + L_2 + \ldots + L_K.$$

The fact that n_{max} is bounded does not imply that a separate network congestion control protocol is not necessary. In fact, one of the motivations for packet networks in the first place is that data traffic sources are typically bursty [10]. In other words, virtual channels between source-sink pairs require network resources 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 a network congestion control protocol is not necessary. However, due to the bursty nature of resource demands from virtual channels, the average number of packets utilizing the network will be very low such as at point A. It is therefore desirable for packet 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; through averaging, the network utilization is at point B with a correspondingly high throughput rate. An immediate consequence is that a separate network congestion control protocol is now necessary to deal with temporary overloads (due to time and statistical fluctuations in network user demands).

Any network congestion control protocol must effectively reduce input into the network to alleviate temporary overloads on the network. We identify 3 types of congestion control protocols.

The isarithmic principle proposed by Davis [11] and studied by Price [12] provides the above function by setting a limit on the number of packets permitted inside a network. This is accomplished by circulating a fixed number of "containers" in the network. A packet can be sent through the network only if its source can get hold of an empty container. The effectiveness of the isarithmic principle is obvious from Figure 1. However, it is difficult to implement in practice because the number of enabled packets (hence the network throughput rate) depends upon the algorithm which individual nodes use to circulate and redistribute empty containers to places of need. Since individual nodes do not have fresh and accurate information about the rest of the network, the design of an effective algorithm is hard. For instance, since containers are not always at the places of need, should one apply the principle of overcommittment again and provide more containers than the value of B in Figure 1? Isarithmic protocols will not be considered further in this paper; however, the spirit of the isarithmic principle is evident in all congestion control protocols studied to date.

The objective of limiting the admission of packets into a network can be achieved by reducing the window sizes of virtual channels or by shutting down some virtual channels entirely. We have examined in our experiments, the effect of window sizes and number of virtual channels on the network throughput rate. Successful application of this type of congestion control protocols will depend upon the ability of individual nodes to detect network congestion conditions and subsequently make coordinated decisions to impose controls.

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 [12]. A similar idea was discussed by Chou and Gerla [13]. 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. Most of the performance characteristics reported below and network design strategies proposed are new. 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 effective demonstrated. On the other hand, the GMD study considered some elaborate protocols that we have omitted in our study; such as, the use of node-to-node windows, in addition to end-to-end windows for virtual channel flow control (the windows are dynamically controlled by nodes according to a heuristic algorithm); the use of different buffer limits for packets with different numbers of links traversed (the limits are dynamically controlled by nodes according to a heuristic algrithm.)

Our philosophy is different. Since individual nodes do not have fresh and accurate network status information, we prefer simplicity in network congestion protocols(requiring decision-making only infrequently).

In [14], the performance of packet networks employing IB limits for congestion control was analyzed by modeling them as an extended class of queueing networks [15]. Using the analytic model, the tradeoffs among network load, buffer capacity N_T and IB limits were investigated. 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 <u>IB capacity</u>. The explanation for the drastic degradation in the network throughpout 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 [14] 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).

A homogeneous network consisting of nodes with identical channel configurations and traffic demands was considered in [14]. It was found that for homogeneous networks the IB capacity (identical for all nodes) is equal to 1/(mean path length of packets). Our experimental results below illustrate IB capacities for general nonhomogeneous networks.

3. 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.)

Several network and traffic configurations have been simulated. They are described in detail below in conjunction with their experimental results. For instance, the most frequently used configuration has 7 nodes, 9 full-duplex links and 168 virtual channels. In some cases, up to 336 virtual channels were simulated. Most experiments were run for 150 seconds of simulated time, so that each virtual channel delivered about 150-300 packets of throughput during a simulation run (except for those cases in which the network is highly congested or deadlocked).

3.1 The Network Simulator

The class of networks that can be simulated is quite general and contains the following features:

(a) Network topology

An arbitrary topology of links and nodes can be specified.

(b) Traffic sources and sinks

In general each virtual channel has a traffic source described by a message interarrival time probability distribution and a message length probability distribution. Each message generated may be segmented into one or more packets. A packet is the basic unit of data transfer in the network. The length of a packet is explicitly modeled; the length of a packet remains fixed as it traverses through the network. Sinks are modeled by queues and absorption time probability distributions.

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. Note that a packet, at its destination node, is considered to be absorbed once it has been handed over to the sink interface protocol layer, although it may still be physically present in the node. 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.

(c) Routing

At present, fixed routing using a table look-up procedure is implemented.

(d) Queueing

Linked list facilities are provided in the simulator for queue management to simulate different scheduling disciplines. In the present study, all queues employ a FCFS discipline.

(e) Data link control

The data link control protocol in the simulated networks is similar to that of ARPANET [16]. Each communication channel is multiplexed into 8 logical channels. A packet must acquire a logical channel to be transmitted; following the transmission it must be positively acknowledged before the logical channel is released. With 8 logical channels a node can transmit up to 8 packets over a communication channel before receiving any positive acknowledgment. Packets are individually acknowledged. A positive acknowledgment may be piggybacked in a data packet or sent as a stand-alone short packet (assumed to be 1/10 of the length of a data packet). Packets are retransmitted if not acknowledged within a time-out period. 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.

(f) Buffer management and congestion control

Each node has a finite number of buffers. The buffers may be partitioned into classes to implement a deadlock-free buffer allocation algorithm [4]. In the present study, IB limits are the only mechanism simulated for network congestion control. Additional buffer classes were not simulated to reduce the simulation cost. (In practice, although they may not be needed for congestion control they may be desirable for avoiding deadlocks.)

(g) End-to-end flow control

Virtual channels are end-to-end flow controlled using windows. The window size of a virtual channel is the number of packets that it can have in transit within the network, and is specified separately for each virtual channel. 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.)

3.2 Performance Characteristics and Network Design Strategies

The first network used in our study, shown in Figure 2, 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. Table 1 shows the number of virtual channels using each communication channel in the 7-node network; the number varies from 7 to 15 (assuming one virtual channel per route).

The virtual channel load λ takes on values of 1, 2 or 10 packets per second.

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-foreward 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. We have considered window sizes of 1, 2 or 3 x path length. (To give an idea of the actual window sizes used, the mean virtual channel path length in the networks simulated is typically between 2 and 3 links.)

We next examine the results of a series of experiments to investigate contributions to the network load by the virtual channel load λ, the number of virtual channels, and the virtual channel window size, as well as their impact on network performance. For the moment, IB limits for network congestion control are not used.

The effect of increasing the virtual channel load λ

In Figure 3, 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.

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 us look at these buffer requirements in a slightly different perspective. 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 3). Note that the ratio actually decreases as $\bar{n}$ increases. A possible explanation is that the decrease is a consequence of the variance reduction effect of the law of large numbers.

λ is the rate at which packets are offered to a virtual channel. As λ becomes large, say $\lambda = 2$ 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 3 increases as the virtual channel load λ increases, assuming the provision of sufficient buffers. Given a modest supply of buffers (say 20-100), Figure 3 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 effect of increasing the number of virtual channels

In Figure 4, the network throughput rate is plotted as a function of N_T for a network supporting 84 x 1, 84 x 2 and 84 x 4 virtual channels. The offered load to each virtual channel is $\lambda = 2$ packets per second. Each virtual channel has a window size equal to 2 x path length.

Increasing the number of virtual channels increases the number of packets in transit within the network. The network requires a minimum number of buffers at each node to meet the resource needs of the admitted packets. Corresponding to the network load of 84 x 1, 84 x 2, and 84 x 4 virtual channels, the threshold values of N_T are 9, 110 and 300 respectively; the respective ratios of $N_T/\bar{n}$ are 4.3, 2.5 and 1.7. As expected, the maximum throughput rate of each curve in Figure 4 increases as the network load increases, assuming the provision of sufficient buffers.

With a modest supply of buffers (say 20-100 per node), Figure 4 indicates that 84 x 1 virtual channels correspond to the expected network load while 84 x 2 virtual channels would be a moderate overload and 84 x 4 virtual channels would be a heavy overload on the network. Note that the buffer requirement here for the case of 84 x 4 virtual channels is much more severe than the case of $\lambda = 10$ in Figure 3. The explanation is as follows. While virtual channel windows provide some form of 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. When network transit delays become large because of too many packets in the network, the virtual channel acceptance rate does decrease somewhat (in accordance with Little's formula [17]). However it is an indirect means of control and Figure 4 indicates that it is not very effective.

In real networks, the number of virtual channels fluctuates in time depending upon network user demands. The number of possible virtual channels can be very large. For example, a single X.25 packet network interface [18] can potentially activate up to 4096 virtual channels!

Figure 4 shows that with no other network congestion control protocol, to guard against a temporary network overload due to an increase in the number of virtual channels the network will require a tremendous amount of additional buffers.

<u>The effect of increasing the virtual channel window size</u>

In Figure 5, the network throughput rate is plotted as a function of N_T for virtual channels with window sizes equal to 1 x, 2 x and 3 x path length. The network supports 84 x 2 virtual channels. The virtual channel load is $\lambda = 2$ packets per second.

Note that increasing the virtual channel window sizes increases the number of packets in transit within the network and the network resource requirements. Corresponding to window sizes of 1 x, 2 x and 3 x path length, the theshold values of N_T are 45, 110 and 175. It is interesting to note that the threshold ratios of $N_T/\bar{n}$ remain constant at 2.5 for all three cases. This may be due to the fact that the maximum throughput rate (assuming the provision of sufficient buffers) is almost the same for all three curves.

That the maximum network throughput rate does not change as the window sizes are increased is expected because the network load (λ and the number K of virtual channels) is the same for all three cases. If the network has sufficient buffers and if virtual channel window sizes are not too small relative to network transit delays, then almost all packets offered to the network are accepted and transported. The network throughput rate is thus close to $K\lambda$ in all three cases.

Figure 5 shows that window sizes equal to 1 x path length give rise to as much network throughput as the other 2 cases of larger window sizes. But the threshold value of N_T required is the smallest. This observation suggests that we should employ as small a virtual channel window size as possible. We should, however, keep in mind that from the point of view of individual virtual channels, each virtual channel must have a window size big enough to achieve its desired throughput rate. From Little's formula, we know that if λ_{out} is the desired virtual channel throughput rate, and T is the mean network transit time, then the window size for the virtual channel should be at least λ_{out} T.

In the above 3 sets of experiments, no explicit network congestion control protocol was used, although the virtual channel windows did provide some amount of input control for the network. We consider next the use of IB limits for congestion control and to examine the resulting network performance characteristics. Various network design strategies for congestion control are discussed.

The design of IB limits for congestion control

In [14], 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

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

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. By considering a packet network as a queueing network and virtual channels as "closed subchains", various network statistics (virtual channel throughput rates, mean queue lengths, etc.) can be calculated under the assumption of infinite nodal buffer capacities [19]. We then attempted to invent heuristic algorithms for designing IB limits for individual nodes to match their traffic demands.

We investigated several such heuristic 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

$$IBL = \alpha/\bar{H} \tag{1}$$

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).

Given values of IBL and N_T, the maximum number N_I of buffers in a node that input packets can occupy is determined from

$$N_I = \lfloor IBL \cdot N_T \rfloor \tag{2}$$

where $\lfloor x \rfloor$ is the largest integer smaller than or equal to x.

In Figure 6 the impact of uniformly assigned IB limits on the network throughput rate performance is shown. The network has 84 x 2 virtual channels, $\lambda = 2$ packets per second, and virtual channel window sizes of 2 x path length. This network load was considered to be a moderate overload if individual nodes have 20-100 buffers each. The mean path length $\bar{\bar{H}}$ of packets in this network was found to be 3.3 nodes under the assumption of infinite buffers. Figure 6 shows that IB limits with a properly designed α can provide a significant improvement in the network throughput rate despite the overload.

The experiments that we conducted for Figure 6 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).

Figure 6 shows that the network using IB limits with $\alpha = 0.7$ can withstand the moderate overload for at least 150 seconds. (The exact duration depends upon the specific value of N_T.) If the network overload is infrequent and is not expected to persist for more than 150 seconds, we see that input buffer limits using $\alpha = 0.7$ will provide the best network throughput performance. If, however, a larger network overload or a longer overload duration is expected then a smaller value of α may have to be used. We found that although the case of IB limits using $\alpha = 0.8$ is shown in Figure 6 to result in degraded network throughput performance at $N_T = 50$, it was originally found to be adequate up to a simulated time of 75 seconds.

Design strategies to control temporary network overloads

Let us reconsider the three different network loads first illustrated in Figure 3 for a network with no explicit congestion control protocol. In Figure 7, we have plotted the same network throughput curves together with new curves obtained using the same network loads but with the network employing IB limits for congestion control.

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, 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 $\alpha = 0.7$, then the network can withstand an overload of $\lambda = 2$ packets per second for at 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.

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).

A network overload condition may also result from an increase in the number of virtual channels. This will happen because in most packet networks, virtual channels are established by nodes without any central control. Figure 4 shows that as the number of virtual channels increases, the network buffer requirement increases very rapidly. Throughput curves for the network loads of 84 x 1 and 84 x 2 virtual channels in Figure 4 are reproduced in Figure 8 together with the throughput curves corresponding to the same network loads but with the addition of IB limits in the network.

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 8 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. 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.

The network loads considered in Figure 5 for different virtual channel window sizes are reconsidered in Figure 9, 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 1 x, 2 x, and 3 x path length.

We make two observations. First, Figure 9 shows that the strategy of reducing virtual channel window sizes when network congestion occurs will help; but Figure 9 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 9 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.

The effect of improving routing

The design of IB limits using the uniform assignment strategy requires $\alpha < 1$ in Equation (1) for general nonhomogeneous network with an "unbalanced" distribution of traffic over its nodes. The extent of the traffic imbalance in a network is magnified when the network load increases. The experimental results presented in Figures 6-9 show that the applicable value of α for IB limits should be smaller if the network overload (being designed for) is more severe or prolonged.

The traffic imbalance in a network can be alleviated with improved routing. It was found in [14] that for homogeneous networks, $\alpha = 1$ can be used in Equation (1) for the design of IB limits.

To further confirm this observation, we considered again the above 7-node network with 84 x 2 virtual channels, $\lambda = 2$ packets per second and window sizes equal to 2 x path length. Previously the 84 routes that we used were made up of the two shortest paths between each source-destination node pair, where ties in the selection of routes are broken by random selection. For our experiment here, we selected a somewhat different set of 84 routes. Again, shortest paths are selected but when there is a tie between routes having the same path length, we tried to select the route that provides a more balanced utilization of communication channels. Table 2 shows the number of virtual channels using each communication channel in the network. The traffic distribution in Table 2 is somewhat more balanced than the traffic distribution in Table 1.

The throughput rate versus N_T curves for the network using the original set of routes and the improved set of routes, both with and without the use of IB limits for congestion control are shown in Figure 10. Observe from Figure 10 that with improved routing, the throughput performance is better for the network without the use of IB limits. If IB limits are used, the network throughput performance is also improved and the maximum applicable α is 1.0 (for 150 seconds of simulated time) instead of 0.7 before the routes were improved!

A different network configuration

To illustrate that the performance characteristics and design strategies discussed above are not unique to the network and traffic configurations of Figure 3 and Table 1, similar experiments were repeated for an 8-node network, shown in Figure 11, with 11 full-duplex links. As before, the two shortest routes (ties broken by random selection) between each source-destination pair of nodes are used. There are 112 distinct routes altogether. Two cases have been considered corresponding to each route used by 1 virtual channel and 2 virtual channels. The number of virtual channels supported by a communication channel ranges from 1 to 16 (assuming 1 virtual channel per route) as shown in Table 3. The network throughput rate versus N_T performance is shown in Figure 12 for two network loads corresponding to 112 virtual channels and 112 x 2 virtual channels. The virtual channel load is λ = 2 packets per second. Window sizes are equal to 2 x path length. The same conclusions that we arrived at for the 7-node network can be drawn here. Figure 12 illustrates that the network with a modest supply of buffers at individual nodes (20 - 100 buffers per node) and employing IB limits with $\alpha = 0.4$, can withstand a temporary overload of a two-fold increase in the number of virtual channels.

Similarly Figure 13 shows the network throughput rate versus N_T for the network load of λ = 2 packets per second and 112 x 2 virtual channels. The two cases of window sizes equal to 1 x path length and 2 x paths length are examined. Conclusions similar to those drawn from Figure 9 for the 7-node network can be made here. In addition, it seems that because the 8-node network operates at a higher throughput rate than the 7-node network, the degradation in throughput due to the use of IB limits is somewhat larger than that in Figure 9.

4. DISCUSSIONS

Other network performance measures

The agregate network throughput rate has been our sole measure of network performance. Two other useful performance measures are:

1) the distribution of virtual channel throughput rates, and
2) the average delay of packets admitted into the network.

Although we have not shown them, both measures (and various others) are available from our simulator.

We know from Little's formula that if (ideally) all communication channel delays are the same, then the maximum throughput rate of a virtual channel is directly proportional to the ratio of (window size)/(path length).

From our experimental results, we found that even if communication channel delays vary, window sizes are still a fairly effective means of controlling virtual channel throughput rates. For example, in the experiments described above we attempted to provide the same level of throughput for all virtual channels by using the same ratio of (window size)/(path length) for each virtual channel. The networks considered have widely varying channel utilizations. Two cases were examined: (1) a network with adequate buffers and no IB limits, and (2) a network with a small N_T and IB limits. In the first case, when the network was under an average load (such as $\lambda = 1$ in Fig. 3), the coefficient of variation of the virtual channel throughput rates was always less than 0.1. The coefficient of variation increased to 0.1 - 0.15 for a moderately heavy load (such as $\lambda = 2$ in Fig. 3), and to 0.6 - 0.7 for an extremely heavy load (such as $\lambda = 10$ in Fig. 3). In the second case, we observed that the coefficient of variation was slightly larger (0.1 to 0.2) than that of case 1 when the network was under its expected load or a moderate overload. On the other hand, the coefficient of variation was smaller (0.2 to 0.5) when the network was subject to a heavy overload.

Another network performance measure that we have not explicitly shown is the average network delay for packets admitted into the network. The average network delay should be interpreted differently for two different network operating conditions. First, if the network has sufficient channel and buffer resources for its load (i.e. N_T within the regions of high throughput rate in Figures 3-5), then the average network delay increases as the network throughput rate increases, such as predicted by queueing theory under the assumption of infinite buffer capacity [17].

Second, if the network does not have sufficient buffer capacities for its load, then the following behavior was observed. When N_T is decreased, the network throughput rate decreases due to more and more packets being rejected by the network. The average network delay for those packets admitted into the network actually becomes smaller!

Now suppose again that the network is heavily loaded and we want to compare different congestion control protocols at a fixed value of N_T. We claim that the network throughput rate is the most important performance measure. The measures of average network delay and packet rejection probability are both closely related to the network throughput rate and can be inferred from it. Under steady-state conditions, a low network throughput rate implies that the network rejection probability is high and vice versa. We also conjecture that a high network throughput rate implies that the average network delay of packets admitted into the network is small. We have not rigorously proved this. (It is, however, interesting to note that for a slotted ALOHA channel which has the same load-dependent congestion behavior shown in Figure 1, it is proved that a control algorithm that maximizes the channel throughput rate also minimizes the average packet delay [20].)

Network design strategies

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 Fig. 3.)

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 Fig. 4.)

Virtual channel windows sizes are useful for controlling the throughput rates of individual virtual channels. Fig. 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 as we did 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

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

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 Figs. 6-9.)

If the network load has changed, it is desirable to improve the routing to reduce the variance in communication channel utilizations. (This is the same objective as that of optimal routing to minimize average network delay [17].) We found that improved routes will enhance the effectiveness of IB limits for network congestion control. (See Fig. 10.)

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., Vol. COM-20, 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, Vol. 2, 1978.

[5] Deaton, G. and D. Franse, "A computer network flow control study," Proc. Fourth ICCC, Kgoto, Japan, Sept. 1978.

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

[7] 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.

[8] 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.

[9] Cerf, V. and R. Kahn, "A protocol for packet network intercommunication," IEEE Trans. on Commun., Vol. COM-22, May 1974.

[10] Lam, S. S., "A new measure for characterizing data traffic," IEEE Trans. on Commun., Vol. COM-26, Jan. 1978.

[11] Davis, D. W., "The control of congestion in packet switching networks," IEEE Trans. on Commun., Vol. COM-20, June 1972.

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

[13] Chou, W. and M. Gerla, "A unified flow and congestion control model for packet networks," Proc. Third Internationa. Conf. on Computer Communication, Toronto, August 1976.

[14] 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.

[15] Lam, S. S., "Queueing networks with population size constraints," IBM Journal of Res. and Develop., Vol. 21, July 1977.

[16] McQuillan, J., W. Crowther, B. Cosell, D. Walden and F. Heart, "Improvements in the design and performance of the ARPA network," AFIPS Conf. Proc., Fall Joint Computer Conference, Dec. 1972.

[17] Kleinrock, L., Queueing Systems Vol. 1: Theory (1975), Vol. 2: Computer Applications (1976), Wiley-Interscience, New York.

[18] Rybczynski, A., "A new communication protocol for accessing data networks -- The international packet-mode interface," National Computer Conference, AFIPS Conf. Proc., Vol. 45, 1976.

[19] Reiser, M., "A queueing network analysis of computer communication networks with window flow control," IEEE Trans. on Commun., Aug. 1979.

[20] Lam, S. S. and L. Kleinrock, "Packet switching in a multiaccess Broadcast channel: Dynamic control procedures," IEEE Trans. on Commun. Sept. 1975.

List of Figures

List of Tables

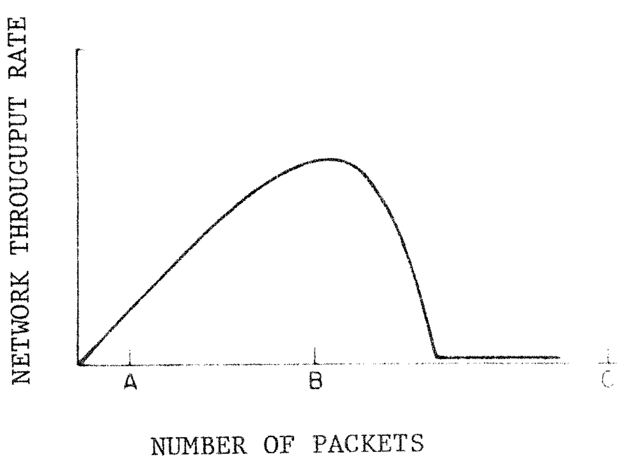

Fig. 1. Network throughput rate versus
number of packets in transit.

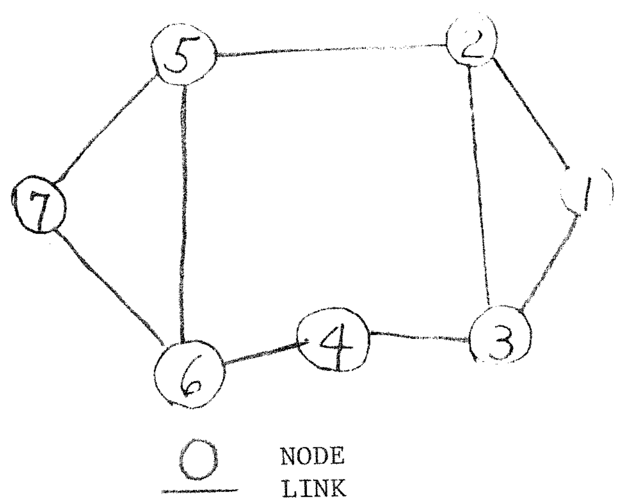

Fig. 2. An 7-node network.

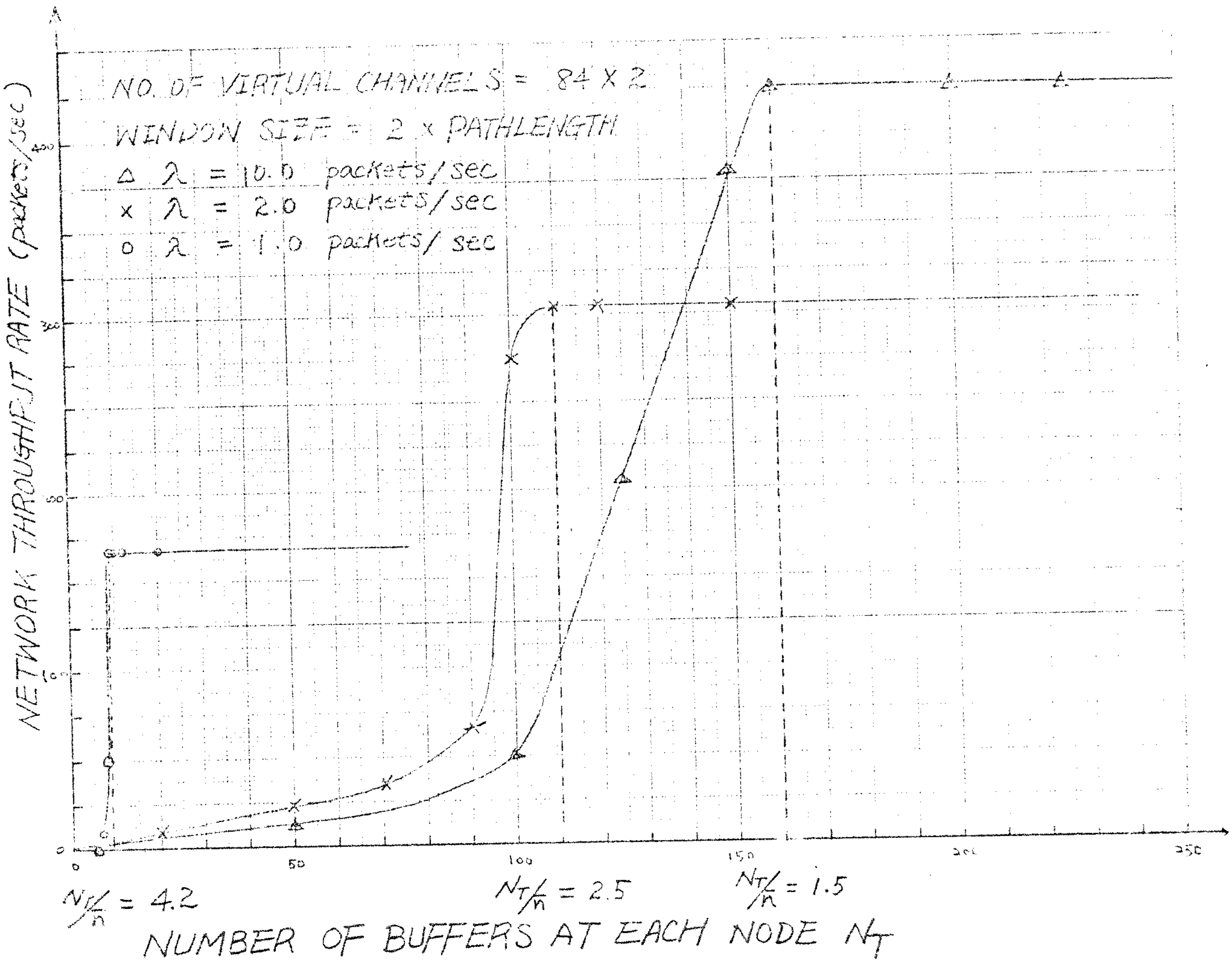

Fig. 3. Network throughput rate as a function of λ and N_T.

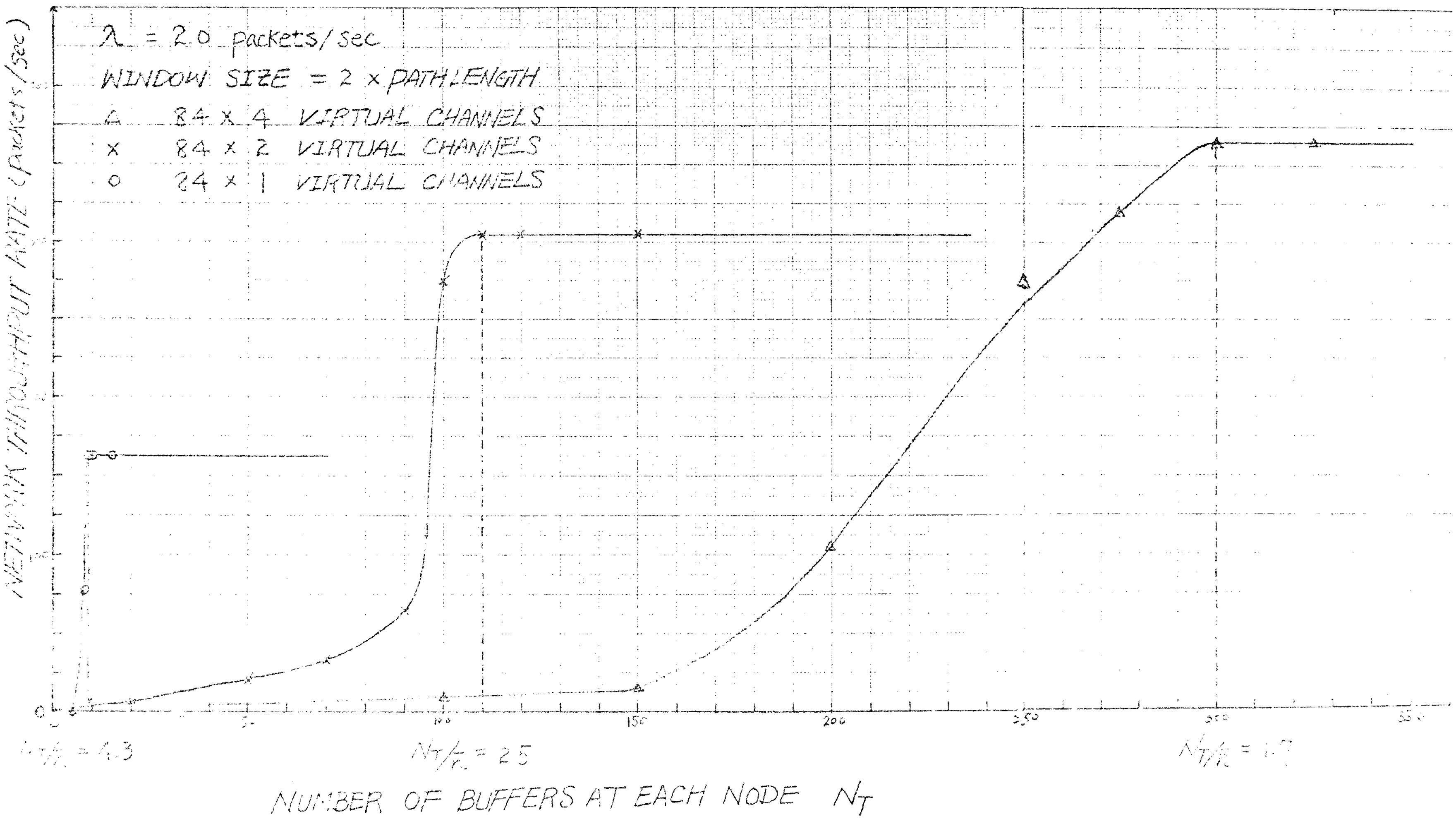

Fig. 4. Network throughput rate as a function of number of virtual channels and N_T.

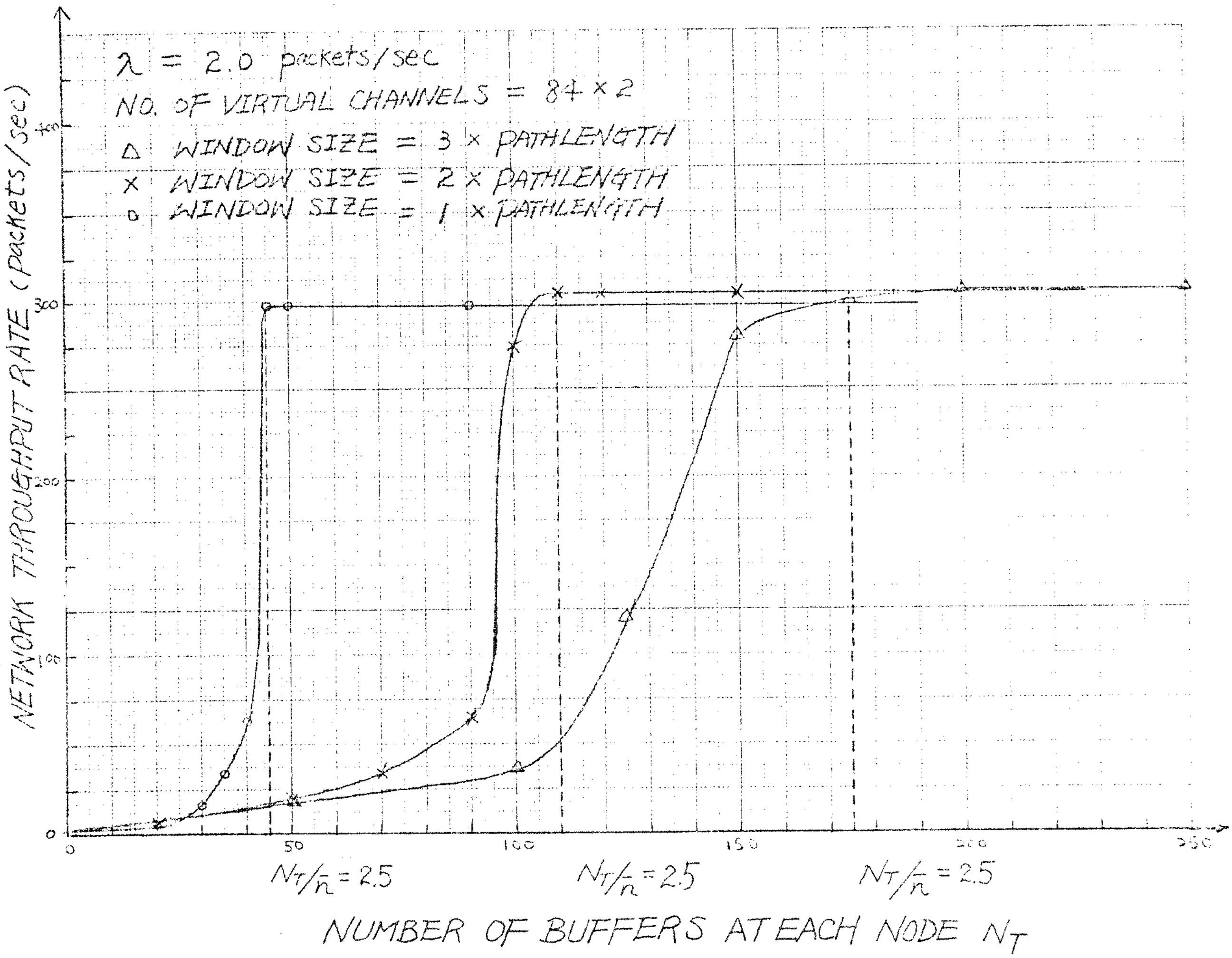

Fig. 5. Network throughput rate as a function of virtual channel window size and N_T.

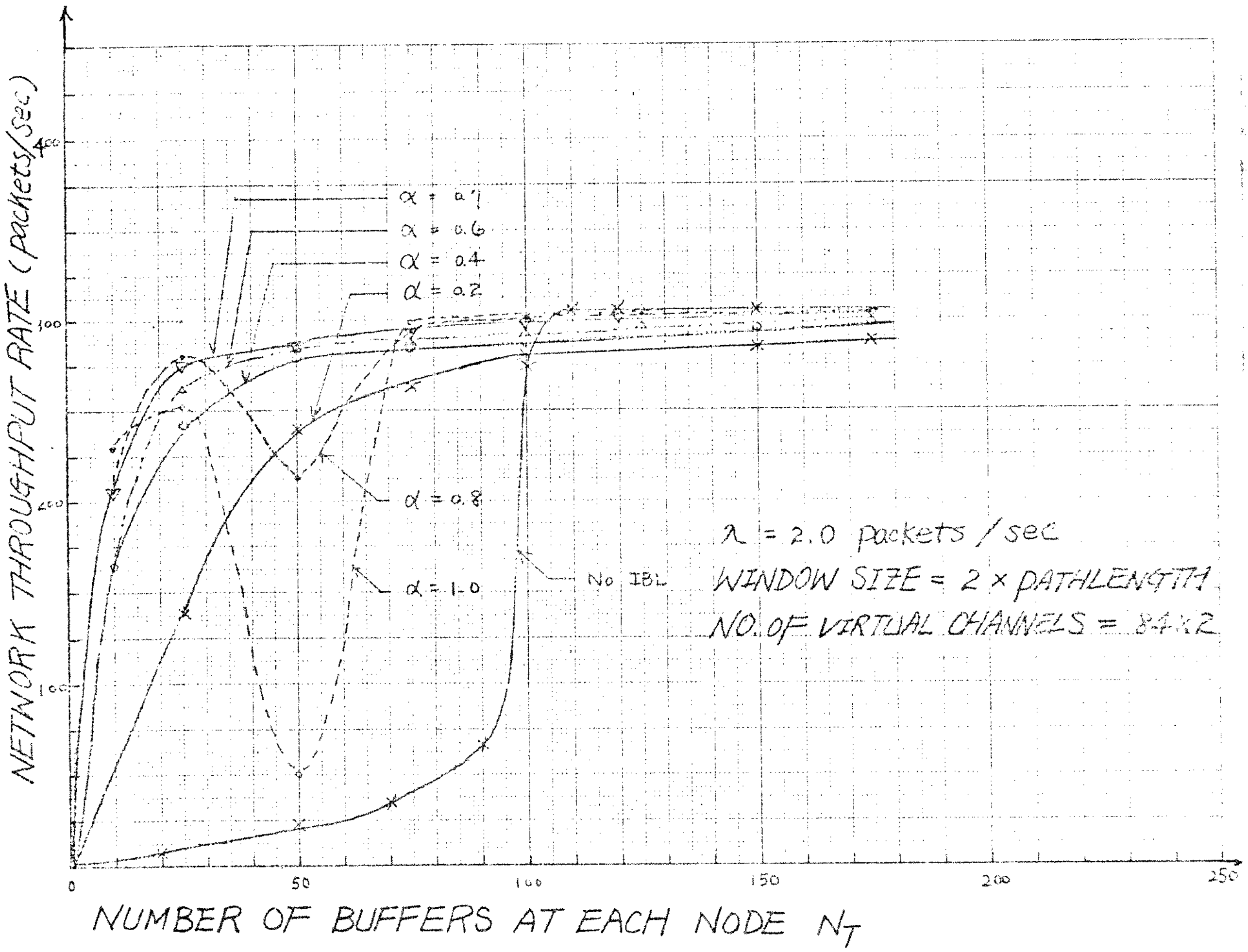

Fig. 6. The effectiveness of uniformly assigned input buffer limits.

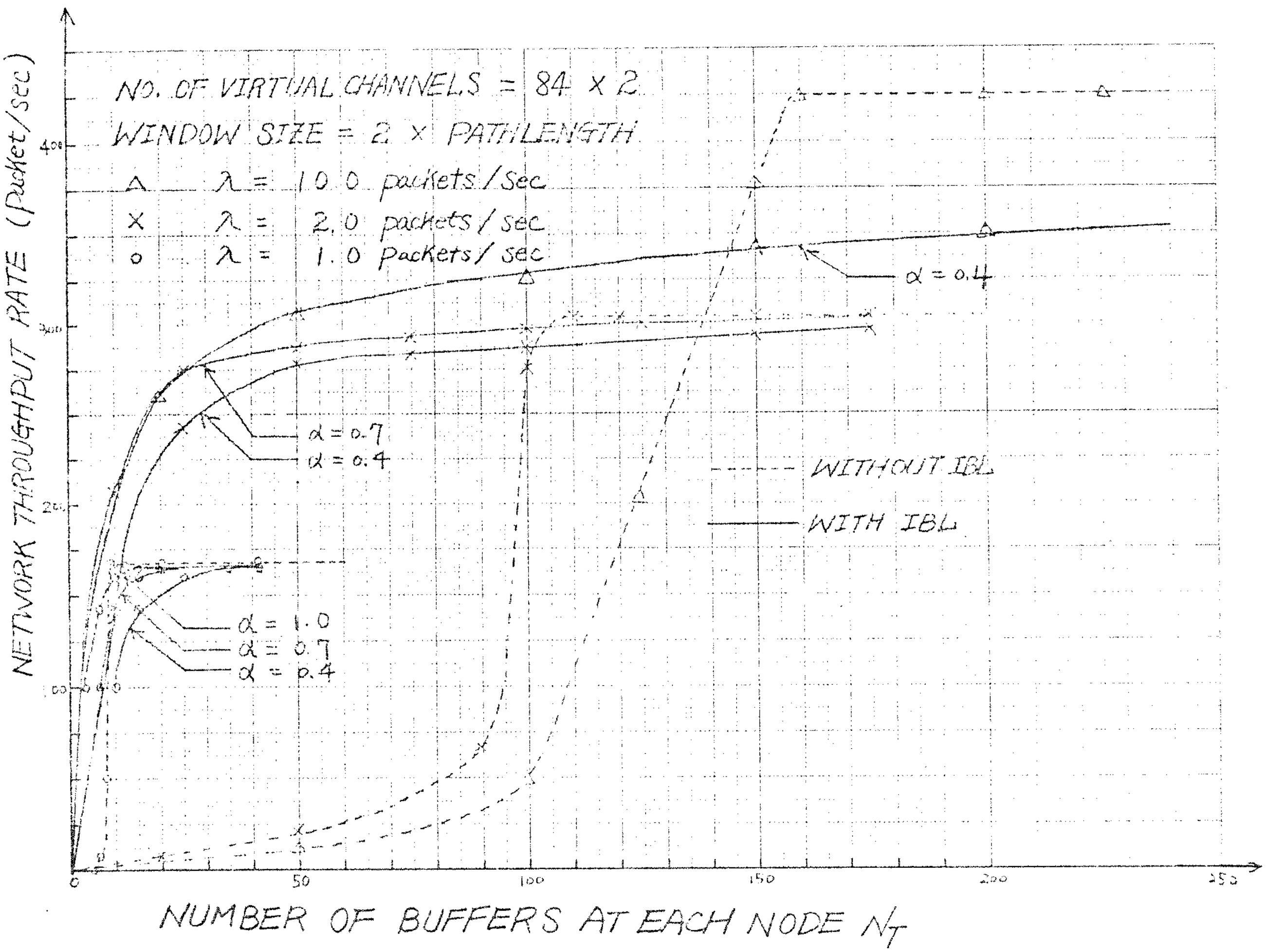

Fig. 7. Input buffer limits for overloads due to fluctuations in λ.

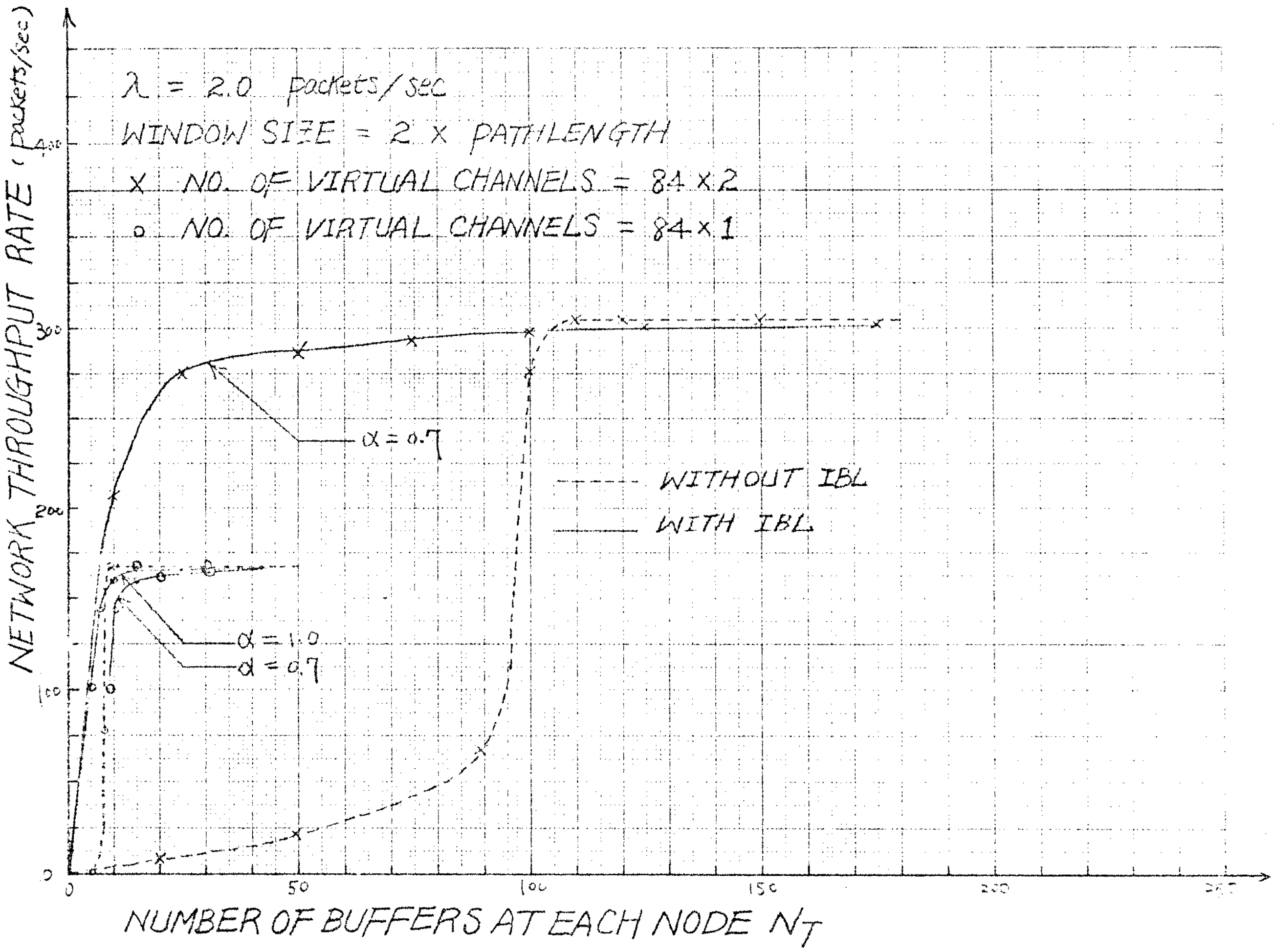

Fig. 8. Input buffer limits for overloads due to fluctuations in number of virtual channels.

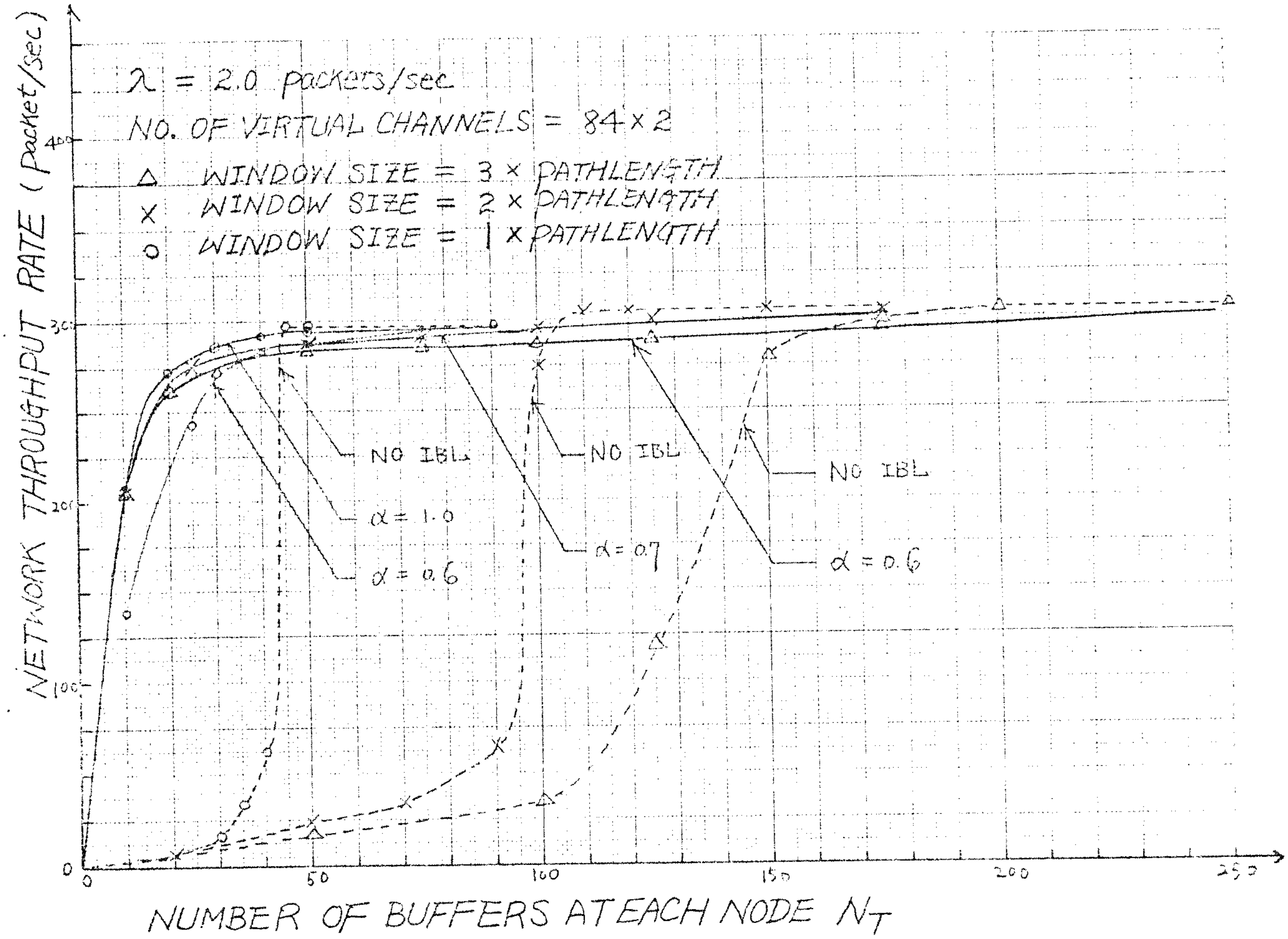

Fig. 9. Input buffer limits for overloads due to fluctuations in virtual channel window sizes.

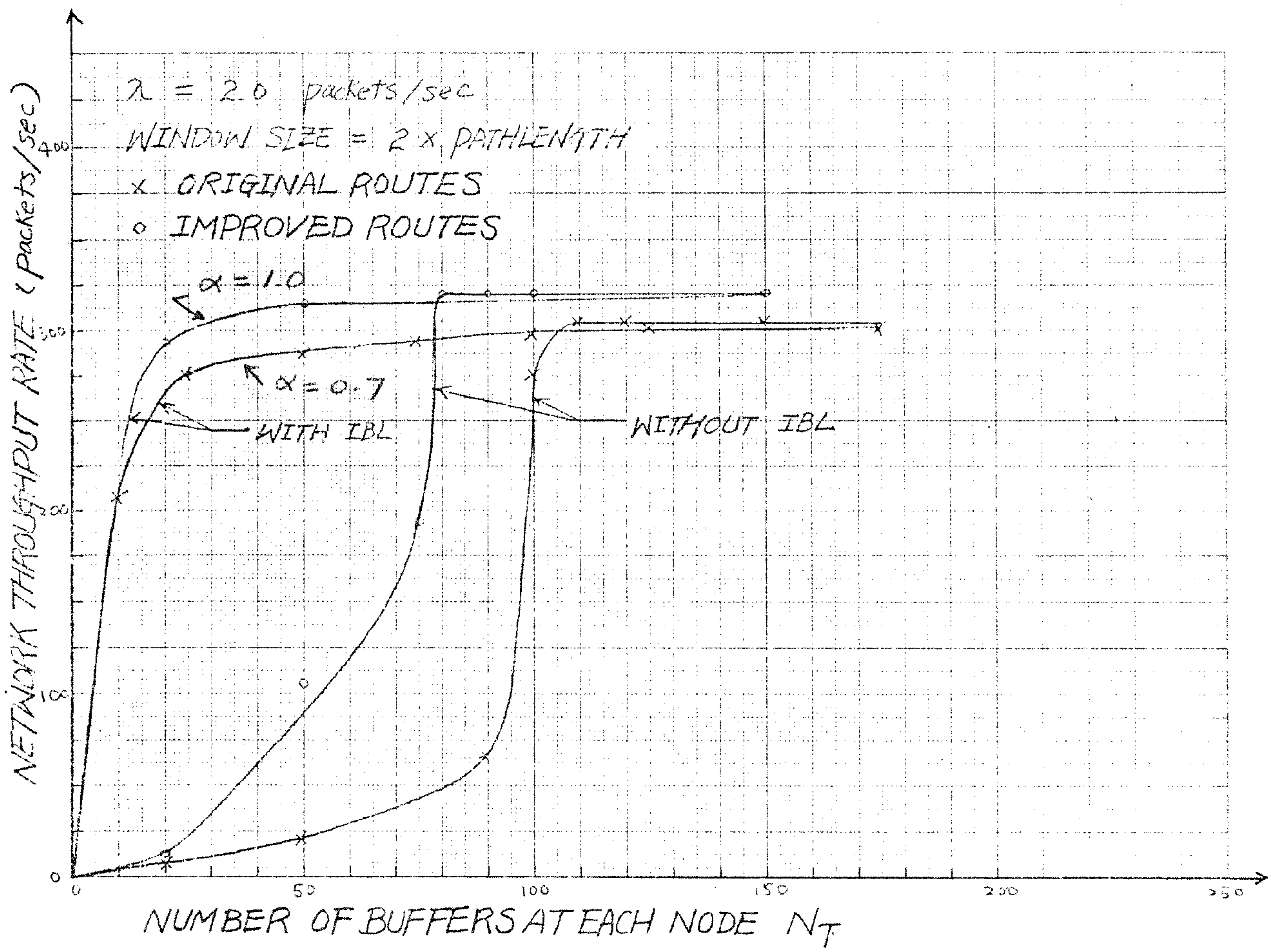

Fig. 10. The effect of improved routes on network performance.

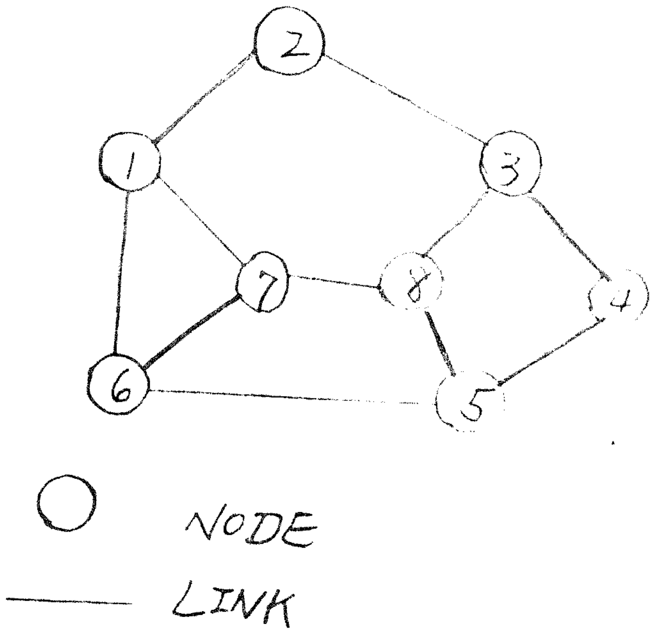

Fig. 11. An 8-node network.

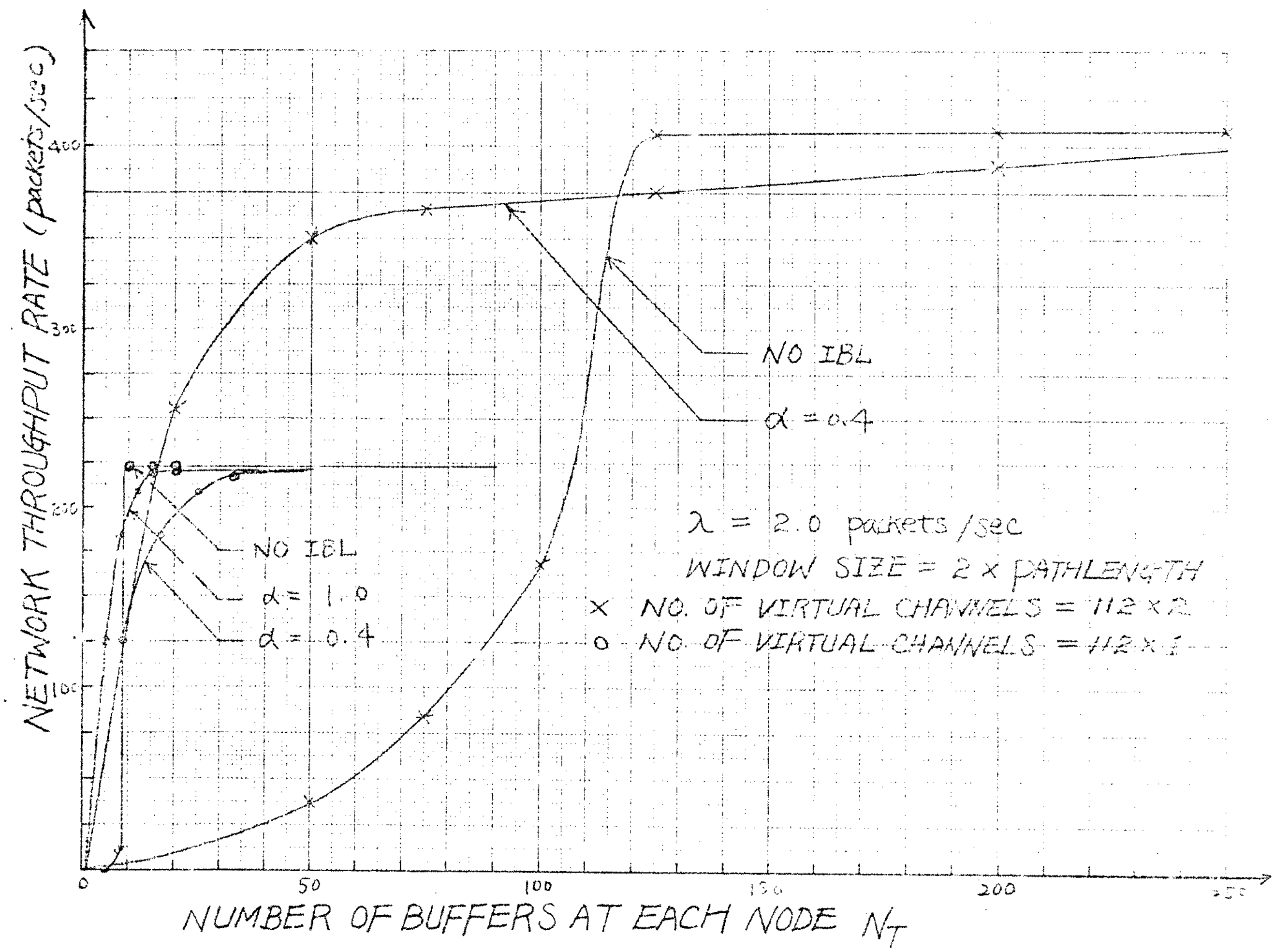

Fig. 12. Network throughput rate as a function of number of virtual channels and N_T (8-node network).

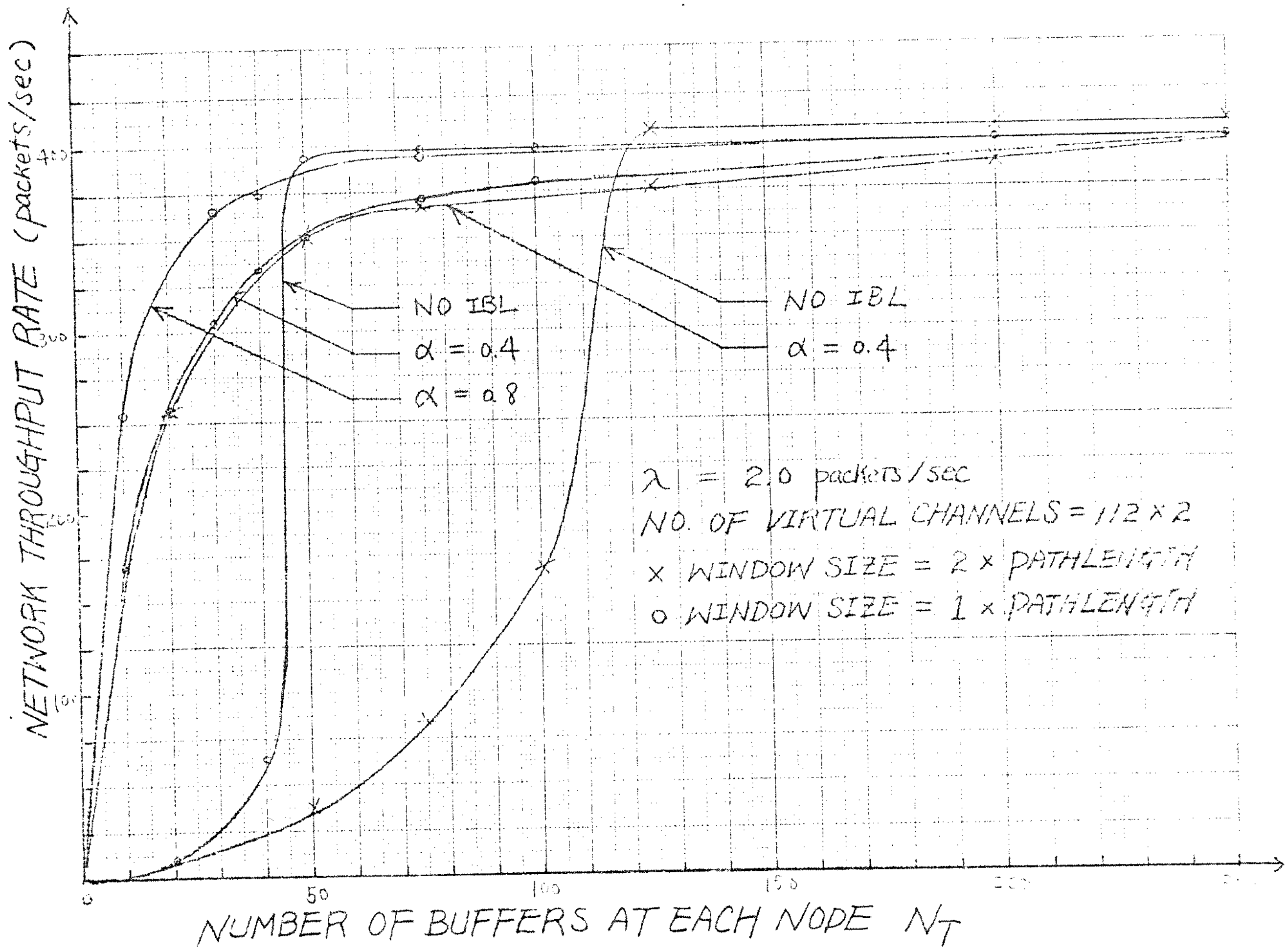

Fig. 13. Network throughput rate as a function of virtual channel window size and N_T (8-node network).

COMMUNICATION CHANNEL	NO. OF VIRTUAL CHANNELS	COMMUNICATION CHANNEL	NO. OF VIRTUAL CHANNELS
(1,2)	7	(4,6)	13
(1,3)	7	(5,2)	15
(2,1)	7	(5,6)	14
(2,3)	14	(5,7)	7
(2,5)	15	(6,4)	13
(3,1)	7	(6,5)	14
(3,2)	14	(6,7)	7
(3,4)	13	(7,5)	7
(4,3)	13	(7,6)	7

Table 1. Utilization of communication channels in 7-node network.

COMMUNICATION CHANNEL	NO. OF VIRTUAL CHANNELS	COMMUNICATION CHANNEL	NO. OF VIRTUAL CHANNELS
(1,2)	8	(4,6)	13
(1,3)	8	(5,2)	13
(2,1)	8	(5,6)	13
(2,3)	13	(5,7)	8
(2,5)	13	(6,4)	13
(3,1)	8	(6,5)	13
(3,2)	13	(6,7)	8
(3,4)	13	(7,5)	8
(4,3)	13	(7,6)	8

Table 2. Utilization of communication channels in 7-node network with improved routes.

COMMUNICATION CHANNEL	NO. OF VIRTUAL CHANNELS	COMMUNICATION CHANNEL	NO. OF VIRTUAL CHANNELS
(1,2)	11	(5,6)	11
(1,6)	8	(5,8)	14
(1,7)	13	(6,1)	8
(2,1)	11	(6,5)	11
(2,3)	11	(6,7)	11
(3,2)	11	(7,1)	13
(3,4)	11	(7,6)	11
(3,8)	14	(7,8)	16
(4,3)	11	(8,3)	14
(4,5)	11	(8,5)	14
(5,4)	11	(8,7)	16

Table 3. Utilization of communication channels in 8-node network.