Performance of a Broadcast Packet Switch

RICHARD G. BUBENIK AND JONATHAN S. TURNER, MEMBER, IEEE

Abstract—**This paper reports the results of a simulation study undertaken to evaluate a high performance packet switching fabric supporting point-to-point and multipoint communications. This switching fabric contains several components each based on conventional binary routing networks. The most novel element is the *copy network* which performs the packet replication needed for multipoint connections. We present results characterizing the performance of the copy network, in particular, quantifying its dependence on *fanout* and the location of active sources. We also evaluate several architectural alternatives for conventional binary routing networks. For example, we quantify the performance gains obtainable by using *cut-through switching* in the context of binary routing networks with small buffers. One surprising result is that networks constructed from nodes with more than two input and output ports can perform less well than those constructed from binary nodes. We quantify and explain this result, showing that it is a consequence of a subtle effect of the FIFO queueing discipline used in the nodes. We also show that substantially better performance can be obtained by relaxing the strict FIFO discipline.**

I. INTRODUCTION

IN [17], Turner describes a packet switched communications system, which supports multipoint connections in addition to conventional point-to-point connections. The principal component of this system is a packet switch called a *switch module* which replicates and distributes multipoint packets and routes point-to-point packets. This paper reports on a simulation study of the switch module undertaken to characterize its performance. Some of these results were reported previously in Bubenik's masters thesis [1].

The switch module described in [17] uses a buffered binary routing network as one of its primary components. This network is one of a class that includes the banyan, delta, shuffle-exchange, and omega networks among others. When these networks are operated in a buffered mode, the distinctions between them become unimportant and we will therefore refer to them all simply as binary routing networks. The performance of such networks has been studied extensively. See, for example, [2]–[4], [6]–[8], [11], [13]. Our work differs from previous studies in several respects. The most important is that we evaluate the performance of a *copy network*; this is a binary routing network modified to perform the packet replication required for multipoint connections, rather than packet routing. This is, to our knowledge, the first study of a network of this sort. Our work also differs from most previous studies in that we consider binary routing networks with a limited amount of buffering (a few packets worth) at each node, an explicit flow control mechanism between nodes and *cut-through switching* [10]. In cut-through switching, a packet need not be fully buffered at every stage of the network, but may proceed directly to the next stage, once the header has been decoded. This yields significant performance gains. We also consider several architectural tradeoffs in order to assess their effect on system performance.

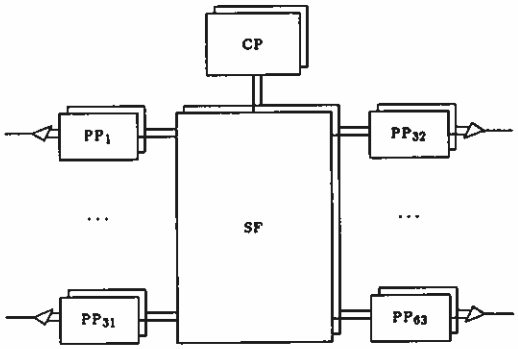

Fig. 1. Switch module.

We note that most analytical studies of systems like the ones we consider here make simplifying assumptions that reduce their usefulness. The work of Jenq [8] is of the most direct relevance, but his analysis applies only to "single-buffered" networks without cut-through. While we have used Jenq's work to validate our simulation results in those cases where the two are comparable, these cases are of only limited practical interest.

Section II briefly describes the architecture and operation of a single switch module. Section III gives the results of simulations of the routing and distribution networks, including results on several architectural variations of the basic design. Section IV gives results for the copy network under a variety of conditions and in Section V we offer our conclusions and outline directions for future research.

II. SYSTEM DESCRIPTION

Reference [17] describes a broadcast packet switching system comprising a number of components called switch modules or SM's. A network made up of these switch modules can support multipoint connections suitable for applications such as entertainment video distribution, LAN interconnection, and voice/video teleconferencing. In this paper, we study the performance of a single SM using a special-purpose simulation program designed for this purpose. The simulator was written in the C++ programming language [14].

We briefly review the operation of the switch module here. The reader is referred to [17] for further details. The overall structure is shown in Fig. 1. The SM terminates up to 63 fiber optic links (FOL). Data are carried on the FOL's in the form of large fixed-length packets of approximately 5000 bits each. The *packet processors* (PP) perform link-level protocol functions, including the determination of how each packet is routed. The switching fabric (SF) is a parallel packet switching interconnection network that provides routing of point-to-point packets plus replication and distribution of multipoint packets. The *connection processor* (CP) is responsible for establishing connections, including both point-to-point and multipoint connections. It will not be of concern to us here.

Paper approved by the Editor for Communication Switching of the IEEE Communications Society. Manuscript received August 25, 1986; revised October 30, 1987 and April 1, 1988. This work was supported under grants from Bell Communications Research, Bell Northern Research, Italtel, SIT, NEC, and the National Science Foundation DCI 8600947. This paper was presented in part at ICC '87, Seattle, WA, June 1987. This work was performed at Washington University, St. Louis, MO.

The authors are with the Department of Computer Science, Washington University, St Louis, MO 63103.

IEEE Log Number 8824904.

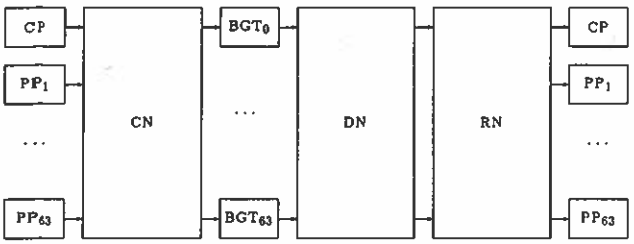

Fig. 2. Switch fabric.

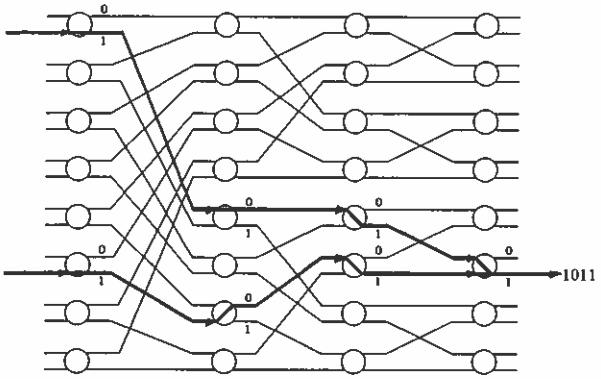

Fig. 3. 16 × 16 binary routing network.

When a packet enters the SM, it is reformated by the addition of several new fields, the routing field being the only one that concerns us here. The routing field (RF) contains information needed to process a packet within the switch fabric. In the case of point-to-point packets, it includes an outgoing link number which is used to route the packet through the switch fabric. In the case of multipoint packets, it includes a *fanout* field which specifies the number of outgoing links that must receive copies of the packets.

A block diagram of the switch fabric (SF) is given in Fig. 2. It contains four major components, a *copy network*, a set of *broadcast and group translators*, a *distribution network*, and a *routing network*. The SF runs at a clock rate of 25 Mbits/s and has eight-bit side internal data paths. This gives an effective bit rate of 200 Mbits/s on the internal data paths, or two times the speed of the FOL's. An occupancy of 80 percent on the FOL's translates to a 40 percent occupancy on the internal data paths, which keeps contention and delay low.

When a multipoint packet having k destinations passes through the copy network (CN), it is replicated so that k copies of that packet emerge from the CN. Point-to-point packets pass through the CN without change. The principal function of the broadcast and group translators (BGT) is to assign an outgoing link number to the multiple copies of multipoint packets. As the BGT's effect on performance is completely deterministic, we will ignore them here. The reader is referred to [17] for details.

The distribution and routing networks (DN, RN) move the packets to the proper outgoing PP. The RN is a 64 × 64 binary routing network. Its structure is illustrated in Fig. 3 which shows a 16 × 16 version. The key property of such networks is that they are *bit addressable*—that is, the route a packet takes through the network is determined by successive bits of its destination address. The figure shows paths from two different input ports to output port 1011. Note that at the first stage the packet is routed out the lower port of the node (corresponding to the first one bit of the destination address), at the second stage it is routed out the upper port (corresponding to the zero bit), and in the third and fourth stages it is routed out the lower ports. The self-routing property is shared by a variety of different although similar interconnection

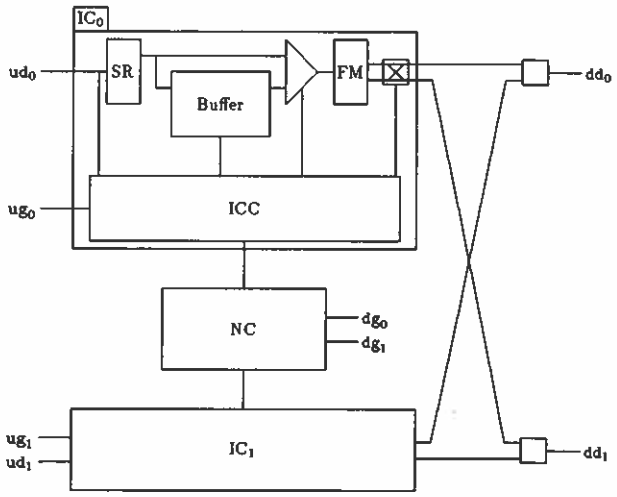

Fig. 4. Switch node.

patterns, including the so-called banyan, delta, shuffle-exchange, and omega networks [5].

Buffers are provided at the inputs of each node. Each buffer holds one or more complete packets. The simulation results given below evaluate several possible buffer sizes. A packet may pass through a node without being buffered at all if the desired output port is available when the packet first arrives. Indeed, in a lightly loaded network, a packet can pass through the RN with no buffering at all. In this case it encounters a delay of just a few clock times in each node.

The data paths between nodes are eight bits wide. In addition, there is a single upstream control lead used to implement a simple flow control mechanism. This prevents loss of packets due to buffer overflows within the fabric. The entire network is operated synchronously, both on a bit basis and a packet basis—that is, all packets entering a given stage do so during the same clock cycle.

Fig. 4 is a more detailed picture of a single switch node. Data enters the node on the upstream data lines (ud_0, ud_1) and leaves on the downstream data lines (dd_0, dd_1). The *node control* (NC), at the center of the figure monitors the availability of the downstream neighbors via the downstream grant lines (dg_0, dg_1) and arbitrates access to the outgoing links. The two *input controllers* (IC) receive packets from their upstream neighbors, request the appropriate output link (or links) from the node control, buffer packets if necessary and apply upstream flow control using the upstream grant lines (ug_1, ug_2) as needed to prevent buffer overflow. Nodes can also be constructed with more than two input and output ports. In the next section, we compare the performance of networks constructed from two port nodes and four port nodes.

The entire switch fabric operates synchronously using a fixed length packet cycle. Each packet cycle can be viewed as having two phases. During the first phase of the cycle, grant signals propagate from the outgoing packet processors back through to the front of the RN, DN, and CN. The outgoing packet processors always accept new packets (if they do not have room in the outgoing buffer, packets are lost). Each node in the switch fabric examines its downstream grant lines and the state of its buffers, then generates the appropriate upstream grant signal. An IC in a node can assert its upstream grant line if it has at least one empty buffer slot. If its buffer is occupied, it attempts to reserve one or more output ports for use during the next cycle (the requested port or ports is determined by the first buffered packet). If the node control grants the reservation request, the IC is assured of being able to move one packet forward during the next cycle and will therefore assert its upstream grant line. Hence, the only case in which the

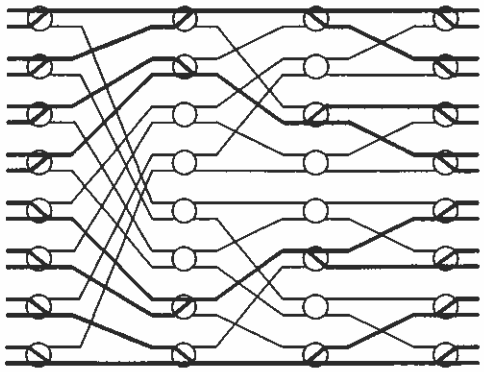

Fig. 5. Congestion in binary routing networks.

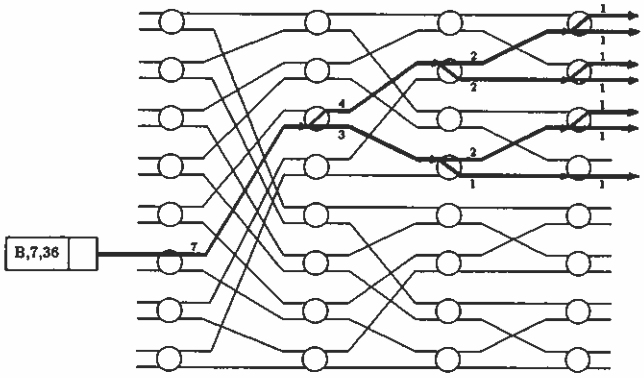

Fig. 6. 16 × 16 copy network.

upstream grant line is not asserted is when an IC has full input buffers and has its reservation request denied. After the grant signals ripple back through to the inputs of the switch fabric, packets flow forward. This is the second phase of the packet cycle. It is possible for the two phases to overlap in time, so the effective length of a packet cycle is just the time required to transmit a packet across one of the internal data paths.

One problem with binary routing networks is that they can become congested in the presence of certain traffic patterns. This is illustrated in Fig. 5, which shows a traffic pattern corresponding to several *communities of interest*. In this pattern, all traffic entering the first four inputs is destined for the first four outputs, all traffic entering the second group of four inputs is destined for the second group of four outputs, and so forth. Note that with this pattern, only one fourth of the links joining the second and third stages are carrying traffic. Thus, if the inputs are heavily loaded the internal links will be hopelessly overloaded and traffic will back up. In a 64 × 64 network, there are six stages and the links between the third and fourth stages can in the worst case by carrying all the traffic on just eight of the 64 links. The purpose of the DN is to eliminate this problem by evenly distributing packets it receives across all its outputs. It has the same internal structure as the RN, but its nodes ignore the destination addresses on packets and route them alternately to each of their output ports. This strategy is modified if one or both ports is unavailable. In this case, the first port to become available is used. This approach is designed to break up any communities of interest and make the combination of the DN and RN robust in the face of pathological traffic patterns. This configuration is evaluated in the next section.

The structure of the copy network (CN) is the same as that of the RN and DN. The CN's function is to make copies of multipoint packets as they pass through, as illustrated in Fig. 6. The packet entering at left has a fanout of 7 meaning that it will be sent to seven different output links. At the first stage, the packet is routed out the upper port. This is an arbitrary decision—the lower link could have been used at this point. At the second stage, the packet is sent out on both outgoing links and the fanout fields in the outgoing packets are modified. The upper packet generates four copies and the lower one three. In general, a node in the copy network will replicate a packet of its current fanout value exceeds one-half the number of CN output ports reachable from that node. The fanout values are split as evenly as possible, with an arbitrary decision being made as to which port gets the "bigger half" in the case of an odd fanout value. When a multipoint packet is not replicated, it is routed arbitrarily to one of the node's two output ports. Point-to-point packets are routed through the CN arbitrarily, taking the "path of least resistance."

A simulation model was written that includes components for the incoming packet processor, copy network, distribution network, and routing network. The packet processor was modeled simply as an infinite queue. Each of the networks in the switch fabric was modeled explicitly. The simulation mimics the synchronous nature of the actual system. During each packet cycle, grants are propagated from outputs to inputs, new packets are generated by packet sources feeding into the input buffers and packets are moved forward through the input buffers and switch fabric. Each packet source has a parameter specifying the probability that a packet is generated during a particular cycle. Each cycle is treated independently and at most one packet enters a particular input buffer during a packet cycle. The results described in the next section cover several different configurations in addition to the one just described. In particular, in most of the runs not all the components described above are included. Also several variations on the basic design described above are explored. Details of these variations will be given as the results are presented. We note that the simulation results described below are independent of most of the specific choices of system parameters mentioned above. In particular, they are independent of the choice of packet length and relative speed of the switch fabric data paths and fiber optic links.

III. Evaluation of the Routing and Distribution Networks

This section describes a basic set of simulations, selected from an extensive simulation study described fully in [1]. We focus here on the routing and distribution networks. Results for the copy network appear in the following section. Unless otherwise stated, the quoted results are for a switch fabric with 64 input and output ports and two port switch nodes.

The first set of results, shown in Fig. 7, is for a configuration consisting of a set of input buffers feeding directly into a routing network. The CN and DN have been omitted to permit a detailed examination of the RN. This delay plot (and all subsequent ones) gives delay in packet times where a packet time is just the time required to transmit a packet across one of the internal data paths. Assuming 5000 bit packets and a 200 Mbits/s data rate on the internal data paths, one packet time is 25 μs. Two sets of delay curves are given. The curves that flatten out at large offered loads give the delay through the RN alone, while the other curves give the total delay through the input buffers and the RN. Delays are measured from the time the front end of a packet enters a component to the time the front end leaves the component. Hence, they neglect the time needed to buffer the packet as it exits the switch fabric (that time is one packet time). Notice also that the simulation ignores details such as the pipeline delay experienced by a packet as it passes through a node, since this is a constant which is completely determined by details of the node design. These simulations were run under the *uniform traffic assumption*—that is, packets generated at the sources were assigned random destination addresses with each destination address equally likely.

In these plots, offered load is expressed as a fraction of the

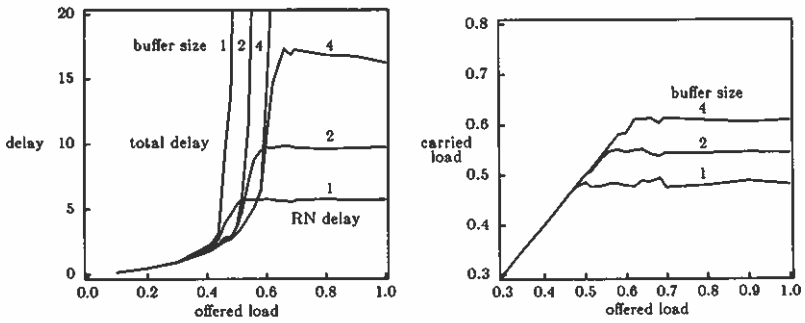

Fig. 7. Delay and throughput curves for routing network.

dn-perf

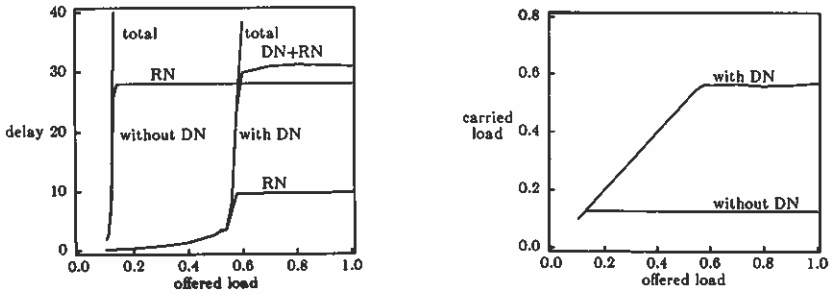

Fig. 8. Performance in the presence of communities of interest.

capacity of the switch fabric under ideal conditions. Under such ideal conditions, the switch fabric is capable of passing a packet out of every output port during every packet cycle. What has been plotted is the offered load measured at the packet sources. Carried load is the fraction of the ideal capacity that the system actually delivers and is measured at the outputs of the switch fabric.

The results in Fig. 7 give the delay for nodes with 1, 2, and 4 buffer slots per input port. As one would expect, increasing the number of buffers increases the throughput of the network and reduces total delay. The carried load curves show clearly the effect of node buffer size on network throughput. As mentioned earlier, the internal data paths operate at twice the rate of the external links meaning that the offered loads within the switch fabric cannot exceed 0.5. If the external links carry a load of 0.8, the internal loading is 0.4. The carried load plot shows that even with a buffer size of one, the RN has sufficient throughput to carry this load. On the other hand, the delay plot shows that the safety margin is unacceptably small with this buffer size. A buffer size of two is probably required to provide an adequate margin against occasional traffic surges. It is worth noting that at low loads, most of the delay is incurred within the RN while at high loads the RN delay becomes constant while the input buffer delay becomes very large. This is a pattern that will be reappear in later results. Note that at an offered load of 0.4, the total delay is approximately two packet times, which translates to about 50 μs, assuming 5000 bit packets and a 200 Mbits/s data rate.

The next set of results demonstrates the effectiveness of the distribution network in breaking up pathological traffic patterns. See Fig. 8. The curves labeled "without DN" show the performance of the RN when subjected to a traffic pattern calculated to cause extensive congestion. In particular, traffic was constrained to eight "communities of interest" similar to those shown in Fig. 5. This pattern forces all the traffic though just eight of the 64 links in the middle stage of the RN. As expected, this leads to a maximum throughput of about 0.125. The curves labeled "with DN" show the performance when the DN is inserted in front of the RN and the same traffic pattern is applied. Note that the throughput in this case matches that of the RN-only configuration with uniform traffic. Also note that at loads of less than 0.5, the DN adds very little delay. Apparently, at these loads packets encounter little or no "back pressure" from the RN and simply flow through the DN taking the path of least resistance.

A. Effect of Cut-Through

Perhaps the principal difference between the routing network described here and buffered binary routing networks discussed elsewhere is the use of *cut-through operation* in the switch nodes. In most previous work, packets are buffered completely at each node in the switch fabric before being advanced to the next stage. We relax this restriction, allowing a packet to proceed immediately to the outgoing link whenever possible. Cut-through switching has also been employed in the fast packet switch described in [12], [15], [18]. The concept of cut-through switching was first discussed in print in [10], although in a different context. Their analysis of the performance of cut-through switching shows significant advantages, but does not apply to the specifics of our situation. To quantify the advantage of cut-through switching we performed a simulation of the routing network in which cut-through was not used. This simulation is for a 64 port routing network with two slot buffers at the node inputs. The resulting comparison appears in Fig. 9.

Notice that at low loads, the total delay without cut-through is at least six packet times, corresponding to the enforced buffering at each stage of the simulated network. When cut-through is used, the total delay is less than a single packet time. At an offered load of 0.4, the total delay without cut-through is about eight packet times, while with cut-through, it is about two. We also note that the use of cut-through increases the throughput of the switch fabric from 0.5 to 0.55. The delay advantage is significant in the target application as it translates to a delay of 50 μs versus 200 μs. The advantage is even more striking than this comparison reveals, since (as we will see later) most of the delay in the switch fabric comprising the CN, DN, and RN occurs in the RN when cut-through is

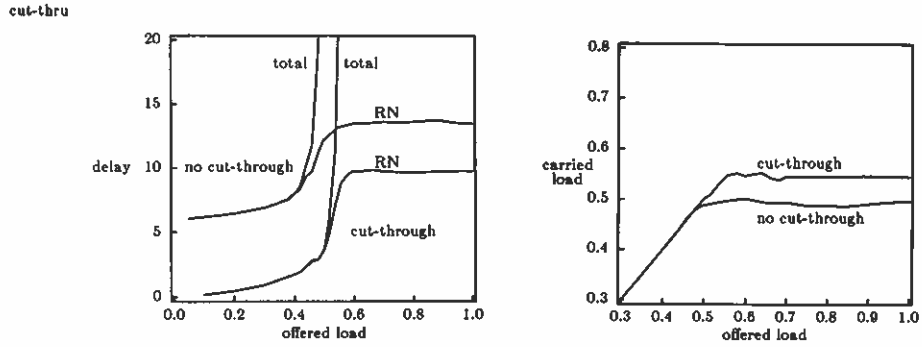

Fig. 9. Effect of cut-through.

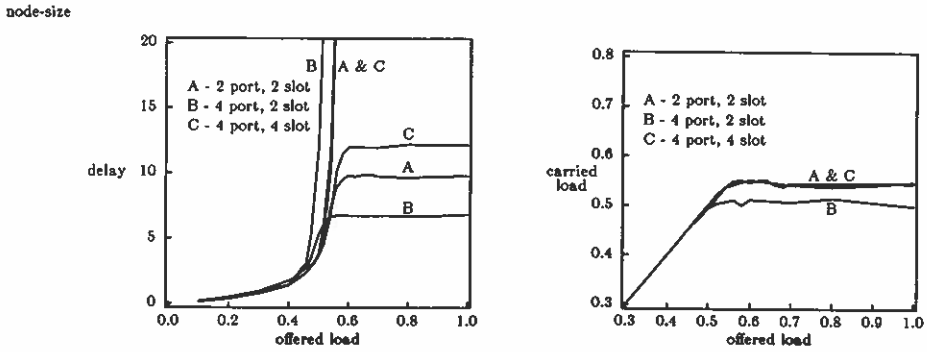

Fig. 10. Effect of node size.

used. Without cut-through the delay would be at least 18 packet times, whereas with cut-through, it is less than three packet times.

Given the advantages obtained with cut-through switching, it is perhaps surprising that it is not universally used. To learn the reason it is not, one must look more closely at specific applications of buffered binary routing networks. In most cases where they have been proposed, the intended application is an interconnection network in a parallel computing system. Typically, these applications involve small packets in which the addressing information may be a large part of the total packet; the advantages of cut-through switching are significantly reduced in such applications and the extra control complexity required of the nodes may not be justified. On the other hand, in communications systems applications where the addressing information is typically a small part of the packet, cut-through is clearly the right choice.

B. Effect of Node Size

We now consider the effect of varying the number of node input and output ports on routing network performance. In particular, we compare the performance of a routing network with 64 input and output ports comprising two port nodes to one comprising four port nodes. Four port nodes appear to offer several advantages. First, they can implement a given size network with half the number of stages required when using two port nodes. This makes them topologically richer and since the links between stages are the places where contention occurs, there is less opportunity for contention with four port nodes. This leads one to expect networks constructed with four port nodes to have larger throughput than networks using two port nodes. One also expects smaller delays, since the number of places a packet can be buffered is smaller. Finally, because there are fewer stages, fewer buffers are needed (assuming the number of buffer slots per node input port is the same in both cases). Hence, larger nodes offer advantages in system complexity as well as performance. The intuitive case in favor of larger nodes is compelling. There is also analytical support for the contention that larger nodes perform better than smaller ones. In [13], Patel analyzes the performance of unbuffered binary routing networks and shows that larger nodes offer substantial performance advantages.

We examined three different configurations. Network A is a 64 port routing network with two port nodes and two buffer slots per node, network B is a 64 port routing network with four port nodes and two buffer slots per node, network C is a 64 port routing network with four port nodes and four buffer slots per node. In light of the arguments given above, the results shown in Fig. 10 are surprising. Notice that network A outperforms B—it offers higher throughput and smaller total delay. One possible explanation for this behavior is that network B has half the total number of buffer slots that network A has and incurs a performance penalty as a result. Network C however, has the same number of buffer slots as A and while it performs better than B, it is still no better than A.

This apparent anomaly is a subtle consequence of the strict FIFO queueing discipline used in the node buffers. If the output port required by the first packet in a node buffer is unavailable, it and all other packets in that buffer must wait. This is true even if the subsequent packets in the buffer require output ports that are available. This queueing discipline limits the way in which packets can proceed through the network. Moreover, it has a more limiting effect on networks constructed from four port nodes than it does on networks constructed from two port nodes. It is this effect that leads to the apparent anomaly.[1]

C. Effect of Bypass Queueing Discipline

In light of the results just discussed, an obvious next step is to relax the FIFO queueing discipline in the node buffers. We now consider a *bypass queueing discipline* in which a packet

[1] When this effect was first observed, the only likely explanation appeared to be a faulty simulation. It was only after spending several days in a fruitless hunt for simulator bugs that we came to understand the real nature of this effect.

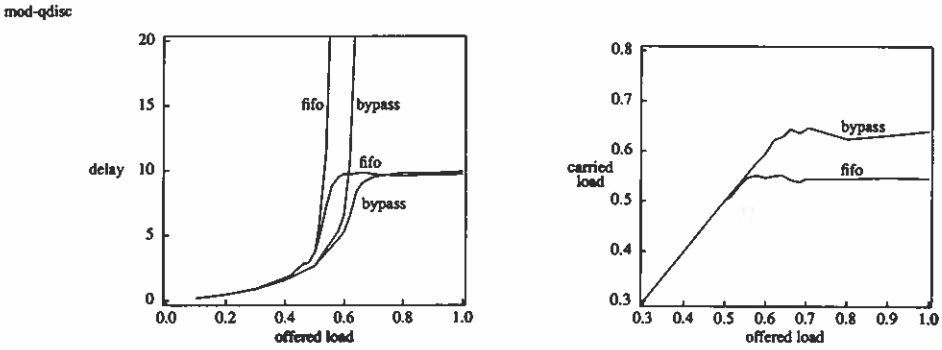

Fig. 11. Effect of bypass queueing discipline.

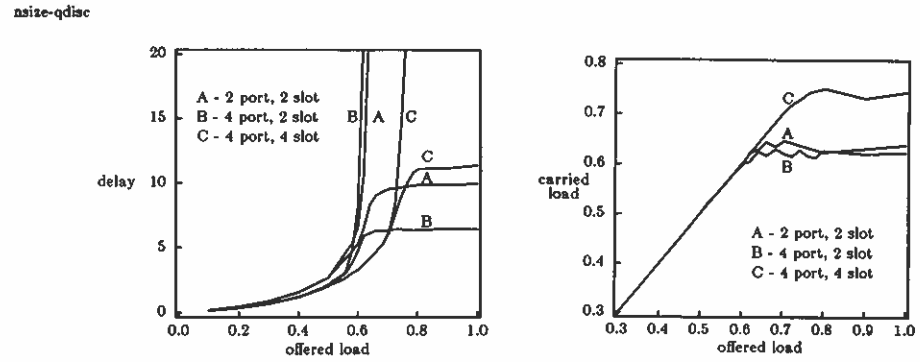

Fig. 12. Effect of node size with bypass queueing discipline.

other than the first one is allowed to proceed if all its predecessors in the queue are blocked, but it is not. Fig. 11 shows the effect of bypass queueing on a routing network with two port nodes and two buffer slots per node. Note that the throughput of the network increases from about 0.55 to 0.63, a substantial improvement.

We now return to our comparison of two port and four port nodes. Fig. 12 is similar to the earlier comparison, but here all networks use bypass queueing. Here, we observe the expected advantage for four port nodes. In particular, network C achieves a throughput of about 0.75 versus 0.63 for networks *A* and *B*. This plot indicates a choice for a network designer. He may use four port nodes to reduce the total number of buffer slots in the network while maintaining the same performance level, or he may use them to increase throughput while holding the number of buffer slots constant. Of course, this tradeoff assumes the bypass queueing discipline, which may be more difficult to implement than the original FIFO discipline. If the original discipline is used, the designer is better off with two port nodes.

We note that the nodes used in the networks described here perform all their buffering on the input side. Another alternative is to buffer at the outputs. In our case, buffering at the input was chosen because it appears simpler to implement. We have not compared the performance of the two alternatives.

IV. Performance of the Copy Network

We now examine the performance of the copy network (CN). Recall, that the function of the CN is to replicate packets associated with multipoint connections. To our knowledge, ours is the first performance study of networks of this type.

The results in this section are for a 64 port CN with two port nodes, each having two buffer slots per input. The CN is preceded by a set of input buffers. Nothing is connected to the output side, meaning that there is no "back-pressure" at the last stage of switching. This configuration allows us to focus on properties determined entirely by the CN. In a later section, we look at the CN performance in the context of a fully configured switch fabric.

In our first set of results, packets arrive at all inputs in the CN at the same average rate. The fanout assigned to each packet is selected from a truncated geometric distribution with parameter p. More specifically, the probability that the fanout is k is given by

$$\Pr(\text{fanout}=k)=\begin{cases}p(1-p)^{k-1} & 1\le k<N\\ (1-p)^{N-1} & k=N\end{cases}$$

where $0 \le p \le 1$ and N is the number of ports in the network ($N = 64$ for the results given here). The average fanout obtained with this distribution is given by

$$E(\text{fanout})=\begin{cases}(1/p)[1-(1-p)^N] & 0<p\le 1\\ N & p=0\end{cases}$$

Since, each packet entering the CN may give rise to multiple output packets, the offered load for the CN is defined as the product of the input load and the average fanout. Given these definitions, we can now consider the first set of simulation results shown in Fig. 13. The delay plot shows CN delay and total delay for several values of average fanout. The carried load curves are also given for several values of the average fanout. When the average fanout is one, the CN carries the load with no contention and no delay. To understand this, one must realize that to achieve an average fanout of one, the fanout of *all* packets must be one. Hence, no packet replication occurs and there is no source of contention. When the average fanout becomes two the situation changes dramatically. First, the maximum carried load drops to about 0.865 and there is a corresponding increase in delay. This drop in performance is

cn-perf

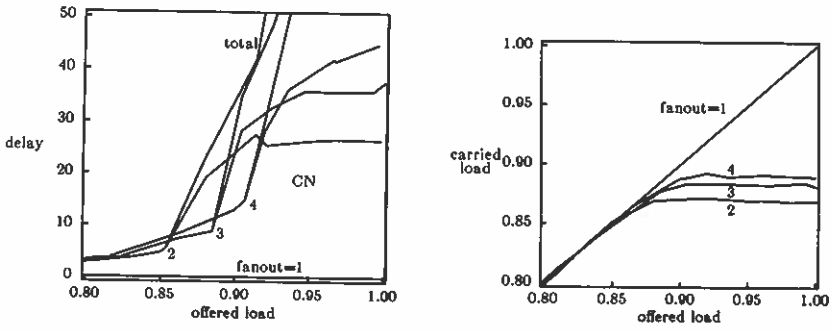

Fig. 13. CN delay and load with random fanouts.

cn-thru1

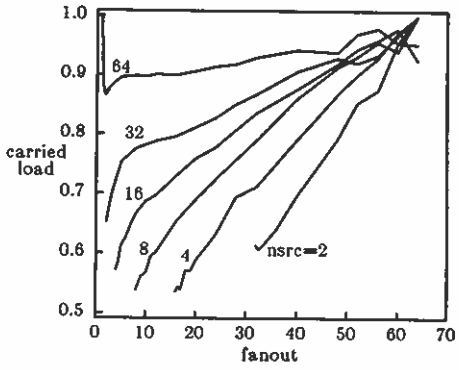

Fig. 14. CN throughput with random fanouts.

cn-thru2

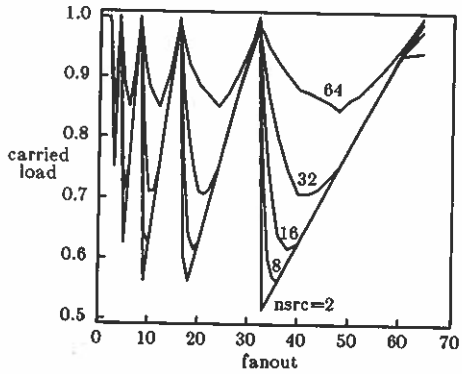

Fig. 15. CN throughput for fixed fanouts.

the expected consequence of contention caused by packet replication. The next effect one notices is that as the fanout continues to increase, the performance improves. While this may appear counter-intuitive, it can be explained by observing that for a given value of offered load, a larger fanout means a smaller packet arrival rate (since offered load is the product of input load and fanout). Thus, when the average fanout doubles from two to four, the packet arrival rate at the CN input ports is halved. This effect tends to reduce contention and counteracts the contention-increasing effect of larger fanout.

The last set of results indicates a dependence of performance on the fanout of the sources. We next consider the effect on performance of limiting the number of active CN input ports. In the next set of results, packets were generated only on input ports 0 to *nsrc* − 1 where *nsrc* is a parameter that was varied. This choice of input ports leads to maximum contention in the early stages of the network. The results are shown in Fig. 14. In this plot, fanout is given on the horizontal axis and carried load on the vertical axis. The offered load in all cases is one, so this plot shows the maximum throughput provided by the network under a wide range of different conditions. The best performance is obtained when all inputs are active. In this case, the minimum throughput occurs with a fanout of about 2. When the number of sources is small, there is a lot of contention in the early stages leading to sharply lower throughputs at small fanouts. We note again that as fanouts increase, throughput climbs rapidly due to the lower arrival rates at the input ports which reduce contention. Note that in all cases, the throughput exceeds 0.5.

The results given above are for randomly generated fanouts. Our next set focuses more closely on the effect of different fanouts. In this set, the fanouts were fixed at a constant for each simulation run, rather than being selected at random. The results are shown in Fig. 15. In this plot, offered load is again held at 1, so we see the maximum throughput obtained for various values of fanout and *nsrc*. The difference between this and the previous plot is striking. For each value of *nsrc*, the throughput takes on values close to one whenever the fanout is a power of two. Between powers of two, the throughput drops sharply achieving its minimum value when the number of sources is small. This plot shows more clearly the effect of different fanout values since in each simulation run all packets had the same fanout and consequently were replicated in the same stages of the copy network. This contrasts to the previous set where in each run there was a mixture of different fanout values. The excellent performance at fanouts equal to powers of two is an expected effect as is the sharp deterioration just above powers of two. This deterioration is explained by noting that when the fanout passes a power of two, copying begins one stage earlier in the network than previously, while the packet arrival rate is about the same. This leads to additional contention and the observed performance deterioration. As the fanout continues to increase, the arrival rate decreases (in order to maintain the same offered load), reducing contention in the early stages and improving performance.

We conclude this section with a review of a few analytical results obtained in [1] and discussed further in [20]. We view the traffic imposed on an N port copy network as coming from a collection of sources $S_1, \cdots, S_k$. Each source S_i is described by an ordered triple (P_i, B_i, F_i) where P_i is the input port at which packets from S_i arrive $(0 \leq P_i \leq N)$, B_i is the average arrival rate of packets from $S_i (0 \leq B_i \leq 1)$ and F_i is the required fanout $(1 \leq F_i < N)$. A *traffic configuration* (or simply configuration) is a set of sources. In a *legal configuration* the sum of arrival rates of sources associated with any

particular input port must be at most 1. That is, for $i \in [0, N-1]$

$$\sum_{j,P_j=i} B_j \leq 1.$$

Also, the sum of the products of arrival rate and fanout must be no more than N.

$$\sum_{j=0}^{k} B_j F_j \leq N.$$

Configurations that violate these restrictions exceed the capacities of either the input ports or the output ports of the CN and hence its performance under such conditions is immaterial.

We label the links in the CN with ordered pairs (i, j) where i is the stage number of the link and j is its index within the stage. Stage 0 links are those feeding the nodes in the first stage, stage 1 links connect the nodes in the first and second stages, and so forth. Links are numbered within stages from top to bottom at their left ends, starting from 0. Hence, the first link leaving the top node in the first stage is link (1, 0).

Let $S = (P, B, F)$ be a source. We define the average load induced by S on a link (i, j) in the copy network by

$$\lambda_{ij}(P, B, F) = \begin{cases} 0 & \text{if there is no path from link } (0, P) \text{ to link } (i, j) \\ 2^{-i}B & \text{if there is a path and } 0 \leq i \leq n - \lceil \log_2 F \rceil \\ 2^{-(n-\lceil \log_2 F \rceil)}B & \text{if there is a path and } i \geq n - \lceil \log_2 F \rceil \end{cases}$$

This definition simply reflects the behavior of the CN with respect to a particular source. When a packet from source S enters the CN it follows a random path until it reaches stage $f = \lceil \log_2 F \rceil$. Stage f is the first at which replication takes place. For a set of sources S, we define

$$\lambda_{ij}(S) = \sum_{(P,B,F) \in S} \lambda_{ij}(P, B, F).$$

The following theorems are proved in [1].

Theorem 1: Let f be an integer in $[0, \log_2 N]$ and let $S = (S_1, \cdots, S_k)$ be any legal configuration in which $F_i = 2^f$ for $1 \leq i \leq k$. Then, for any link (i, j), $\lambda_{ij}(S) \leq 1$.

Theorem 1 tells us that when all the sources have fanouts equal to the same power of two, there is no legal configuration that can induce an overload on any internal link. If we drop the restriction to powers of two we get the following.

Theorem 2: Let F be an integer in $[1, N]$ and let $S = (S_1, \cdots, S_k)$ be any legal configuration in which $F_i = F$ for $1 \leq i \leq k$. Then, for any link (i, j), $\lambda_{ij}(S) \leq 2$.

So, if the fanouts are all equal, the CN can handle any legal configuration in which the arrival rates on all input ports are ≤ 0.5 without inducing an overload on any internal link. We get a similar result ($\lambda_{ij} \leq 2$) if we insist on fanouts equal to powers of two but drop the restriction that all fanouts be equal. Finally, if we drop all restrictions we obtain the following.

Theorem 3: Let $S = (S_1, \cdots, S_k)$ be any legal configuration. Then, for any link (i, j), $\lambda_{ij}(S) \leq 3$.

This implies that the CN can handle any legal configuration in which the arrival rates on all input ports are $\leq 1/3$ without inducing an overload on any internal link. In each of the above theorems, the bound on λ_{ij} cannot be improved. In Fig. 16 we show a legal traffic configuration for a six-stage CN that induces a load exceeding 2.5 on a link in the center stage. This example generalizes to larger copy networks and induces loads approaching 3 as the network size increases. Fig. 17 shows the performance curves for a simulation of a six-stage CN with this traffic pattern. Note that the maximum throughput for this traffic pattern is about 0.35 and as the offered load continues to increase the throughput drops to about 0.22. At high offered loads multipoint packets must wait a long time in a node where they must replicate because it is rare that both output ports become available simultaneously and if only one is available there is usually a point-to-point packet in the other node buffer that takes it. This suggests that better performance might be obtained at high loads by allowing multipoint packets to be copied serially rather than in parallel. This change would not affect Theorem 3, but could prevent the throughput from dropping at high loads. On the other hand, the implementation of nodes that operate in this way appears substantially more complicated.

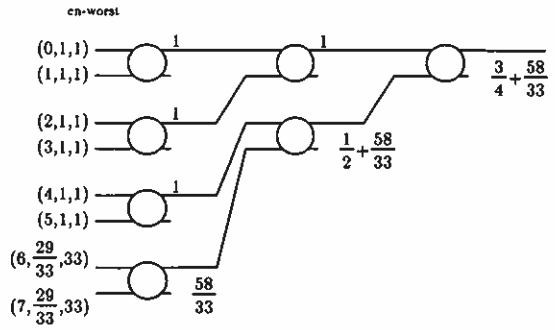

Fig. 16. Pathological traffic pattern for CN.

V. Closing Remarks

The results presented above allow us to select from among several alternative designs for the system described in [17]. In our recommended configuration, we use two slot buffers at the inputs to each node; we also use cut-through switching and bypass queueing. We use four port nodes in the routing and distribution networks, but two port nodes in the copy network. The performance of this configuration is given in Fig. 18. This is for an input traffic pattern with eight active sources and fanouts selected from a truncated geometric distribution with a mean of eight. This traffic pattern provides a fairly demanding test of the copy network, but is not worst case. Note that at the nominal operating point of 0.4, the delay is just over two packet times (50 μs). Also note that at this load, the delay is split fairly evenly between the CN and the RN and the DN contributes very little delay. Our results give us confidence that this configuration will perform well across a wide range of operating conditions. While there exist pathological traffic patterns that can exceed the capacity of the system, they are sufficiently contrived that it appears unlikely that they would pose an actual threat to a real system.

The simulations results allow us to draw several conclusions concerning the performance implications of alternative node designs. The first is that cut-through switching offers significant performance advantages. In applications where the header is short relative to the packet length, cut-through switching is clearly the method of choice. Without cut-through, the *minimum delay* in the recommended configuration would be twelve packet times (300 μs). Second, two port nodes are the best choice if queueing is performed at the input side of the switch nodes and a strict FIFO queueing discipline is used. The bypass queueing discipline improves performance significantly, especially for larger switch nodes; when bypass queueing is used, large nodes are preferred.

Our results show that the performance of the copy network

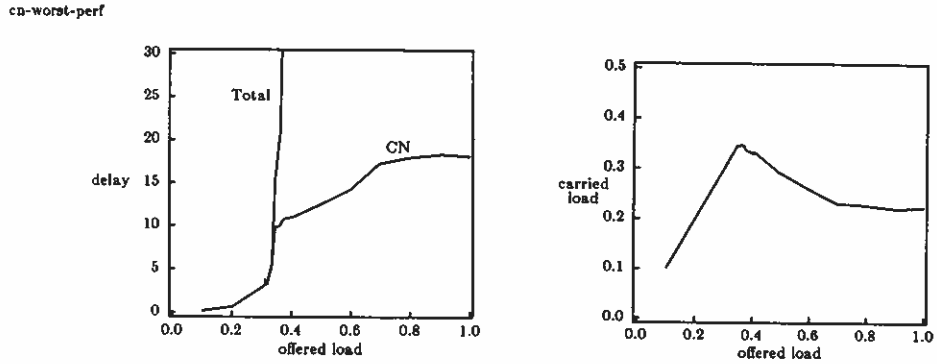

Fig. 17. CN performance for pathological traffic pattern.

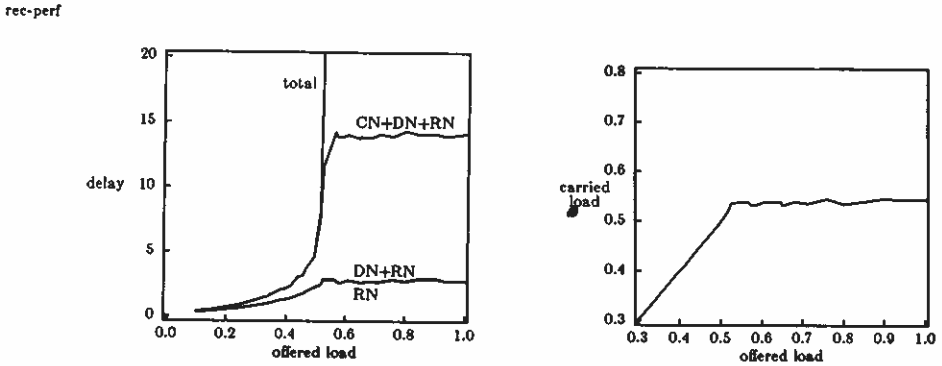

Fig. 18. Performance of recommended configuration.

is sensitive to the fanout of the incident traffic and also to input ports on which the traffic arrives. There are pathological traffic patterns that can induce a load of 3 on some of the copy network's internal links, but this is as bad as things can get. To make the copy network robust in the fact of arbitrary traffic patterns, there are at least two approaches. The first is to operate the copy network so that the load at the input ports is no more than one third (this may require higher speed operation). The second is to place a second distribution network in front of the copy network. In this configuration, the maximum load obtainable on an internal copy network link is one.

There are many interesting questions that remain to be explored. One area for further research is to seek analytical models of the kinds of networks studied here. An analysis of binary routing networks with cut-through switching and small buffers would be of particular interest. Another direction for additional work is the exploration of other architectural variations. In particular, our use of a separate distribution network is clearly just one of many ways to avoid congestion in binary routing networks. A second approach is described in [19] and many others are possible. A comparison of buffering strategies in the switch nodes would also be worthwhile, we have limited ourselves to buffering at the inputs, but clearly output buffering merits consideration as well. Reference [9] does consider queueing strategies in switching fabrics (in a slightly different context from ours) and shows a clear performance advantage for output buffering over input buffering. Their work does not give any insight as to how output buffering compares to input buffering with bypass queueing. We believe input buffering with bypass queueing to be equivalent to shared buffering and expect it be slightly superior to output buffering, but cannot justify that conjecture at this time.

When it comes to the copy network there remains a wealth of open problems. The most important perhaps is whether or not there exist implementations using binary nodes and $\log_2 N$ stages for which the maximum load that can be induced on an internal link is smaller than three. There are several questions that concern packet replication strategies. 1) Is it best to divide the fanouts evenly as we have done here or to divide them unevenly? 2) Should priority be given to packets that require replication? 3) Should sequential copying be allowed at a node, rather than requiring that both copies proceed in parallel? Finally, we note that the issue of copy networks with nodes having more than two ports remains largely unexplored.

References

[1] R. G. Bubenik, "Performance evaluation of a broadcast packet switch," M.S. thesis, Comput. Sci. Dep., Washington Univ., Aug. 1985.

[2] D. M. Dias and J. R. Jump, "Packet switching interconnection networks for modular systems," *Computer*, vol. 14, no. 12, pp. 43-53, Dec. 1981.

[3] D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," *IEEE Trans. Comput.*, vol. C-30, pp. 273-282, Apr. 1981.

[4] D. M. Dias and Manoj Kumar, "Packet switching in N log N multistage networks," in *Proc. IEEE GLOBECOM '84*, Dec. 1984, pp. 114-120.

[5] T.-Y. Feng, "A survey of interconnection networks," *Computer*, vol. 14, no. 12, pp. 12-30, Dec. 1983.

[6] M. A. Franklin, "VLSI performance comparison of banyan and crossbar communications networks," *IEEE Trans. Comput.*, vol. C-30, pp. 283-290, Apr. 1981.

[7] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. 6th Annu. Symp. Comput. Architect.*, Apr. 1979, pp. 182-187.

[8] Y. C. Jenq, "Performance analysis of a packet switch based on a single-buffered banyan network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, no. 6, pp. 1014-1021, Dec. 1983.

[9] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input vs. output queueing in a space division packet switch," in *Proc. IEEE GLOBECOM '86*, Dec. 1986, pp. 659-665.

[10] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communication switching technique," *Comput. Networks,* vol. 3, pp. 267–286, 1979.
[11] C. P. Kruskal and M. Snir, "The performance of multistage interconnection networks for multiprocessors," *IEEE Trans. Comput.,* vol. C-32, pp. 1091–1098, Dec. 1983.
[12] J. K. Kulzer and W. A. Montgomery, "Statistical switching architectures for future services," presented at Proc. Int. Switch. Symp., May 1984.
[13] J. H. Patel, "Performance of processor-memory interconnections for multiprocessors," *IEEE Trans. Comput.,* vol. C-30, pp. 771–780, Oct. 1981.
[14] Bjarne Stroustrup, *The C++ Programming Language.* New York: Addison-Wesley, 1986.
[15] J. S. Turner and L. F. Wyatt, "A packet network architecture for integrated services," in *Proc. IEEE GLOBECOM '83,* Nov. 1983, pp. 45–50.
[16] J. S. Turner, "Fast packet switching system," U.S. Pat. 4 491 945, Jan. 15, 1985.[2]
[17] ——, "Design of a broadcast packet switching network," *IEEE Trans. Commun.,* vol. 36, pp. 734–743, June 1988.
[18] ——, "Design of an integrated services packet network," *IEEE J. Select. Areas Commun.,* vol. SAC-4, pp. 1373–1380, Nov. 1986.
[19] J. S. Turner and L. F. Wyatt, "Alternate paths in a self-routing packet switching network," U.S. Pat. 4 550 397, Oct. 29, 1985.
[20] J. S. Turner, "Fluid flow loading analysis of packet switching networks," Washington Univ., Comput. Sci. Dep., Tech. Rep. WUCS-87-16.[3] Also, in Proc. Int. Teletraffic Congr., June 1988.
[21] J. E. Wirsching and T. Kishi, "Conet: A connection network model," *IEEE Trans. Comput.,* vol. C-30, Apr. 1981.

[2] Available upon request from the Commissioner of Patents and Trademarks, Washington, D.C.

[3] Available upon request from the Computer Science Department, Campus Box 1045, Washington University, St. Louis, MO 63130.

Richard G. Bubenik received the B.S. degrees in electrical engineering and computer science in 1984, and the M.S. degree in computer science in 1985, all from Washington University, St. Louis, MO. He is presently a doctoral candidate at Rice University in Houston, Texas.

His research interests include distributed operating systems, networking, and principles of database systems.

Mr. Bubenik is a student member of the IEEE Computer Society and the Association for Computing Machinery.

★

Jonathan S. Turner (M'77) received the M.S. and Ph.D. degrees in computer science from Northwestern University in 1979 and 1981.

He is an Associate Professor of Computer Science at Washington University, where he has been since 1983. His primary research interest is the design and analysis of switching systems, with special interest in systems with the flexibility to support a wide range of different applications. He has been awarded a dozen patents for his work on switching systems and has several widely cited publications. His research interests also include the study of algorithms and computational complexity, with particular interest in the probable performance of heuristic algorithms for NP-complete problems. From 1977 to 1983 he worked for Bell Laboratories in Naperville.