

Research

# Analysis and design of Banyan and crossbar switches with bypass queues

Jau-Hsiung Huang\*, Yo-Song Su

Communications and Multimedia Laboratory, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

(Received 15 December 1994; revised 20 October 1995)

## Abstract

Bypass queueing can overcome the problem of head of line blocking within an input queue. We study two different approaches for providing a bypass function in input buffers. They are *look-through* and *look-back* service disciplines, both of which can prevent out of sequence situations. Under uniform traffic, if the switch size is large, these two disciplines obtain almost the same throughput. However, under nonuniform traffic, especially a hot-spot traffic pattern, the *look-through* will achieve a much better throughput than *look-back* policy. This paper presents the throughput analysis and simulations for crossbar and Banyan switches under uniform and nonuniform traffic. Furthermore, to reduce contention across input queues, the parallel-plane ATM switch with bypass queues is also discussed and analyzed.

*Keywords:* Bypass queueing; Look-through/Look-back policies; Throughput

## 1. Introduction

Although switches using output queueing obtain better throughput and delay performance than switches using input queueing, the hardware cost is higher and the problem of output contention is still unavoidable [1–5]. On the other hand, input queueing is simple in the hardware aspect, while it suffers from the problem of head of line blocking [6–11]. This is the phenomenon where a packet in the input buffer is prevented from accessing an available output port because the packets ahead of it in the buffer are blocked. Hence, head of line blocking will reduce the throughput of the switch. This problem can be overcome by allowing other packets in the buffer to be transmitted when the leading packets are blocked. Such an input buffer is called a bypass queue [9,10,12,13].

The problem of output contention confines the throughput of crossbar switches. For Banyan switches, besides output contention, the problem of internal blocking also reduces the throughput. With bypass queueing, when the leading packet is blocked, the packet after it still has the chance to be transmitted. Hence, the throughput can be increased. However, a potential problem of bypass queueing is the out of sequence packets at the output port; thus, a special care should be provided in the design. This will be discussed in detail later.

As illustrated in Fig. 1, an  $N \times N$  non-buffered ATM switch with bypass queues consists of  $N$  input buffers,  $N$  input port controllers,  $N$  output port controllers, and an  $N \times N$  non-buffered fabric. The function of the input and output port controllers is to implement the bypass queueing function. The input port controller will allow a packet to be transmitted through the switch when the packet ahead of it is blocked. Moreover, outside the switch, the transmission line operates in *time slots* where one time slot is the time for the switch to transmit one ATM cell (i.e. 424 bits). Inside the switch, time is divided into *packet slots*, each of which contains several *offer slots* and one *transmission slot*, as illustrated in Fig. 2 [10]. One transmission slot is the time taken to transmit one ATM cell. In an offer slot, the packet routing tag is used to attempt to establish a path through the switch to the desired output port. Note that, because of the asynchronization between the packet and time slots, a 2-packet-deep buffer should be provided at each output port controller to synchronize packets with the time slot.

We further define the *window* size as the number of offer slots in a packet slot. At the beginning of the transmission slot, all packets which have successfully set up a path to their output ports during the offer slots are transmitted. If the switch is designed carefully, a  $\log_2 N$  bit period is required for an offer slot. Hence, the overhead of offer slots will reduce the throughput of an ATM switch by a fraction of  $424/(424 + f \log_2 N)$ , where 424 is the ATM cell length and  $f$  is the window size of a packet slot.

\* Email: jau@csie.ntu.edu.tw

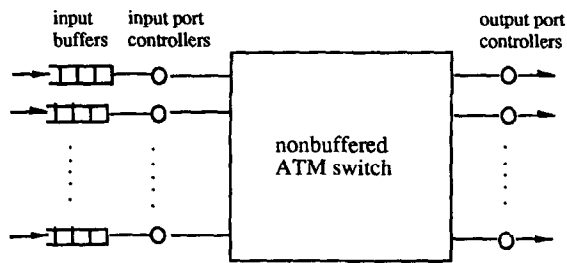


Fig. 1. Architecture of the ATM switch with bypass queue.

Clearly, a larger  $f$  provides more opportunities for packets in the input buffer to be transmitted, and hence can increase the switch throughput. On the other hand, a larger  $f$  also incurs more overhead for each packet slot, and hence will decrease the switch throughput. Therefore, the value of  $f$  should be carefully chosen to maximize the switch throughput.

Before discussing policies to overcome the packet out of sequence problem, we introduce definitions of nonuniform traffic patterns used in this paper. One of the nonuniform traffic patterns considered is hot-spot traffic, with  $h$  defined as the hot-spot ratio. This means that  $h\rho$  packets are directed to the hot-spot port and  $(1-h)\rho$  packets are directed uniformly to other output ports, where  $\rho$  is the load of an input port. Another nonuniform traffic pattern considered is the mixed single-source-single-destination (SSSD)/uniform traffic pattern [14]. This means that a specific input port sends all its packets to a single output port, but packets on other input ports are directed uniformly to the other output ports. Such traffic might exist if the network supports leased line services.

In the following we describe two policies to resolve the out of sequence problem and to enhance the throughput. With the *look-back* policy, when the window size is  $f$ , the input port controller will examine  $f$  packets at most in the front of the input buffer. Whenever a packet in the bypass window is allowed to be transmitted through the switch, the input port controller must check (or *look back*) whether any of the leading packets is directed to the same output port. If it does, the foremost packet with the same output address is transmitted in the transmission slot to avoid the problem of packets arriving out of sequence.

With the *look-through* policy, it is reasonable to assume that if a packet is blocked, then packets after it with the same output address will very likely be blocked in another offer slot, too. Based on this assumption, if the input port controller can skip (or *look through*) the packets which are directed to the same output ports as the leading packets, we

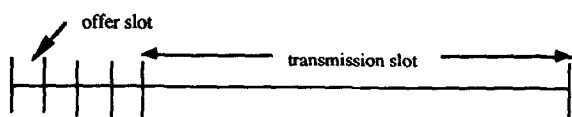


Fig. 2. Structure of a packet slot.

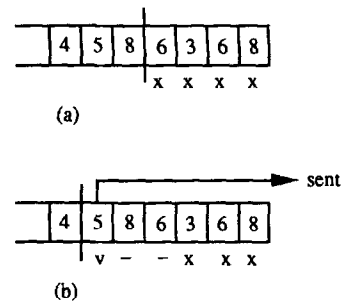


Fig. 3. Behavior of look-through and look-back policies when  $f = 4$ . (a) Look-back policy; (b) look-through policy. ('x' means attempt failure, 'v' means attempt success, '-' means ignore).

expect a better throughput. With look-through policy, when the window size is  $f$ , the input port controller will examine  $f$  packets at most, but these  $f$  packets are not necessarily the first  $f$  packets in the input buffer. It will be shown later that the look-through policy obtains a much higher throughput than the look-back policy under the hot-spot traffic pattern.

In Fig. 3 we give an example for each policy with the window size  $f$  equal to 4. A look-back policy is shown in Fig. 3(a), where all attempts of the first four packets are failed. Note that there are two packets destined to '6' in the attempt window. Since the first '6' is failed, there is a good chance that the second '6' will also fail. In Fig. 3(b) a look-through policy is shown, where a successful attempt is found in the sixth buffer position. Note that in this figure there are two '8's in the first and fifth positions and two '6's in the second and fourth positions. Since both the first '8' and '6' are failed in their attempts, both the second '8' and '6' are skipped in the following attempts. Instead, the '5' in the sixth position is attempted and succeeds. This example shows that the look-through policy can achieve a higher throughput than the look-back policy.

To implement the look-back policy,  $(f - 1)$  registers are required. While the output address of the  $k$ th packet in the buffer is fed to the switch fabric, it is saved into the  $k$ th register and compared with the contents of the first  $(k - 1)$  registers in parallel. If the input port controller finds an unoccupied port for this packet, then the foremost packet with the same output address is transmitted in the transmission slot.

To implement the look-through policy, the input port controller should know the positions of these  $f$  packets, all with different output addresses, in the input buffer at the beginning of a packet slot. Since the length of the transmission slot is much larger than that of an offer slot and the input port controller has nothing to do during the transmission slot, the input port controller will find the positions of these  $f$  packets for the  $k$ th packet slot during the transmission slot of the  $(k - 1)$ th packet slot and memorize their positions by using registers. Hence, at the beginning of the  $k$ th packet slot, these  $f$  packets are available for the input port controller to feed the switch fabric.

For a large switch, the maximum throughputs of these two disciplines are almost the same under uniform traffic. It is because the output addresses of different packets are usually different that the leading  $f$  packets are normally heading for different output ports. However, under hot-spot traffic at heavy loads, hot-spot packets will be queued in front of input buffers because the switch cannot handle so many hot-spot packets as they come in. With the look-back policy, it is likely that the first  $f$  packets may all be hot-spot packets with the same destination, depending on the hot-spot ratio. However, with the look-through discipline, the input port controller will skip these hot-spot packets to find non-hot-spot packets. Hence, the look-through policy will achieve a much better throughput than the look-back policy.

Some previous works are limited to the look-back policy under a uniform traffic pattern. Hluchyj and Karol [15] presented simulation results of the maximum throughput for a nonblocking switch, such as crossbar switches, with bypass queues under various switch sizes and window sizes. The overhead incurred by offer slots was not considered, and no analytical model was presented. Sarkies [10] presented a structure of the input port controller for Banyan switches to overcome the problem of a packet arriving out of sequence with a few simulation results. We analyzed the maximum throughputs of single and parallel-plane Banyan switches with ‘look-back’ bypass queues under uniform traffic [16].

In this paper, we study the performance of Banyan and crossbar switches with bypass queues under uniform and nonuniform traffic. Both the look-back and look-through policies are considered and compared. An analytical model is proposed for the look-through policy under a uniform traffic pattern. The rest of the paper is organized as follows. Section 2 analyzes the maximum throughputs of Banyan and crossbar switches with bypass queues under uniform traffic. Simulation results accompany the analysis to show the preciseness of the analytical model. The performance under nonuniform traffic is presented in Section 3. Section 4 discusses and analyzes the parallel-plane ATM switches with bypass queues. Conclusions are given in Section 5.

## 2. Performance analysis of look-through policy under uniform traffic

In this section, we will find the maximum throughputs for switches with the look-through policy under a uniform traffic pattern. A Banyan switch will be examined first, and then the crossbar switch.

### 2.1. Banyan switch

With the look-through policy, it is reasonable to assume independence between packet slots, i.e. between packet slots the destinations of packets in the bypass window are

independent. Similarly, we also assume independence between offer slots. Based on these assumptions, the analysis will only consider the behavior of the switch in a packet slot. Furthermore, it is assumed that the two input links of a switching element are statistically independent; hence, the state of a stage can be reduced to that of a single input link of the stage [17]. We introduce the following notation for analysis:

- $N$  = switch size
- $n$  = number of stages in the switch ( $n = \log_2 N$ )
- $f$  = window size
- $j$  = number of paths which are already set up
- $\rho_i$  = Prob[there is a packet departing from state  $i$ ]
- $i$  = Prob[there is a packet coming to stage  $i + 1$ ]
- $\beta$  = overhead incurred by one offer slot. (Hence,  $\beta = \log_2 N/424$ .)
- $\alpha_j$  = prob[a connection setup is successful for a packet in the input buffer |  $j$  paths are already set up]
- $S(k)$  = total number of input links permitted to transmit a packet after the  $k$ th offer slot
- $B(k)$  = total number of blocked input buffers after the  $k$ th offer slot
- $TH(k)$  = throughput of the switch after the  $k$ th offer slot (after considering the overhead).

The throughput analysis under such a policy and uniform traffic pattern is the same as in our previous work [16]. However, we will briefly outline the key ideas in the following for completeness. The following is the analysis to obtain  $\alpha_j$ :

$$\begin{aligned}
 \rho_0 &= \rho \\
 \rho_1 &= \rho_0 \left[ 1 - \frac{j}{N} \cdot \frac{1}{2} - \left( 1 - \frac{j}{N} \right) \cdot \rho_0 \cdot \frac{1}{2} \cdot \frac{1}{2} \right] \\
 &\quad \vdots \quad \quad \quad \vdots \\
 \rho_n &= \rho_{n-1} \left[ 1 - \frac{j}{N} \cdot \frac{1}{2} - \left( 1 - \frac{j}{N} \right) \cdot \rho_{n-1} \cdot \frac{1}{2} \cdot \frac{1}{2} \right] \\
 \alpha_j &= \rho_n / \rho_0
 \end{aligned} \tag{1}$$

With  $\alpha_j$  found, we can derive the throughput as follows. To find the maximum throughput, we set  $\rho_0 = 1$  for all computations.

Initially,  $S(0) = 0$ ,  $B(0) = N$ ,  $TH(0) = 0$ .

Compute  $\alpha_0$  using Eq. (1):

$$S(1) = N \cdot \alpha_0, \quad B(1) = N - S(1),$$

$$TH(1) = S(1) / [N(1 + \beta)].$$

Compute  $\alpha_{S(1)}$  using Eq. (1):

$$S(2) = S(1) + B(1) \cdot \alpha_{S(1)}, \quad B(2) = N - S(2),$$

$$TH(2) = S(2) / [N(1 + 2\beta)].$$

$\vdots \quad \quad \quad \vdots$

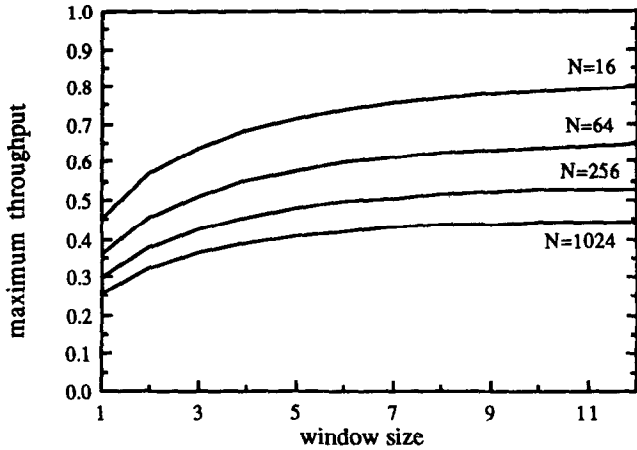


Fig. 4. Maximum throughput of Banyan switches under uniform traffic.

Computer  $\alpha_{S(f-1)}$  using Eq. (1):

$$S(f) = S(f-1) + B(f-1) \cdot \alpha_{S(f-1)},$$

$$B(f) = N - S(f), \quad TH(f) = S(f)/[N(1 + f\beta)]. \quad (2)$$

The maximum throughputs considering the window overhead are illustrated in Fig. 3. Note that with FIFO input queuing (window size = 1), the maximum throughput of a 10-stage Banyan switch is only 0.25. However, bypass input queuing can improve the throughput up to 0.44, with a window size equal to 12. Fig. 4 shows the simulation results of the maximum throughput for a 16 × 16 Banyan switch under various window sizes and policies, compared with the analytical results. From this figure, we observe that the analytical results fit well with the simulation results using the look-through policy. Besides, the assumptions of independence do not affect the estimation of the optimal window size.

### 2.2. Crossbar switch

For crossbar switches, at most one packet is permitted to

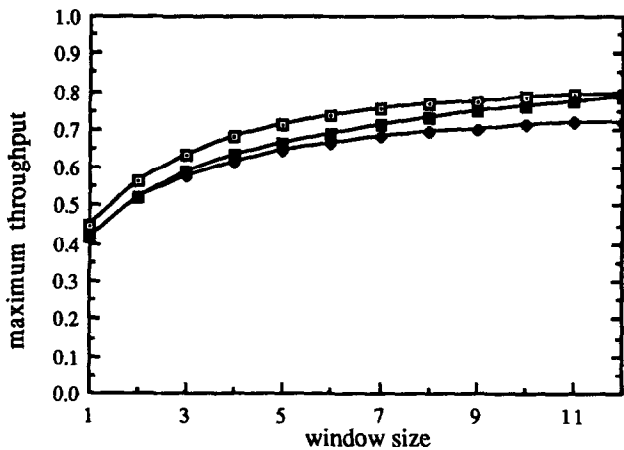


Fig. 5. Simulation results of the maximum throughput for a 16 × 16 Banyan switch under uniform traffic. □: analysis; ■: simulation (look-through); ◆: simulation (look back).

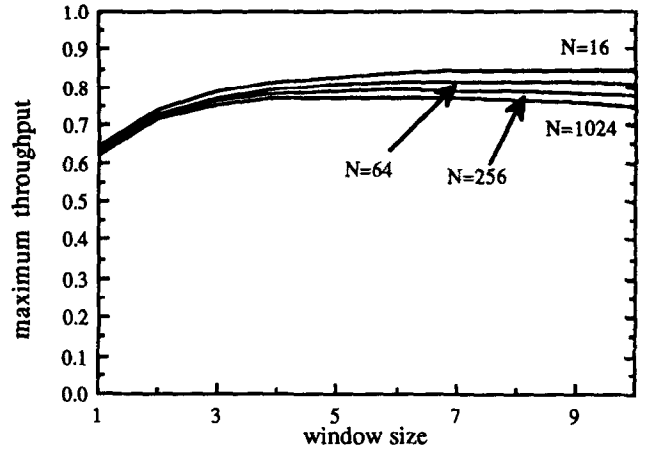


Fig. 6. Maximum throughput of crossbar switches under uniform traffic.

access each output port. If there are many packets directed to the same output port, one of them is randomly chosen to access this output port and the others are blocked in the input buffers [18]. In the following analysis, input buffers are checked one-by-one to find the maximum throughput. Some assumptions and notation are used as stated earlier:

Set  $j = 0$  and  $S(0) = 0$

Initial  $w = 1$

for\_each  $w \leq f$  do

begin

Initial  $p = 1$

for\_each  $p \leq N$  do

begin

$$\text{set } j = j + \left[ 1 - \frac{S(w-1)}{N} \right] \cdot \frac{N-j}{N};$$

end

Set  $S(w) = j$

$$\text{Set } TH(w) = \frac{S(w)}{N(1 + w\beta)}$$

end

Variable  $w$  is the index to calculate  $TH(f)$ , and  $p$  is the index variable used to check input buffers one-by-one. The term  $1 - S(w-1)/N$  in the inner loop is the probability that this input port controller is still looking for an unoccupied output port. That is, none of the leading packets in this input buffer has found an unoccupied output port. Note that this term is always 1 in the first loop to calculate  $TH(1)$ , because the packets checked are the first packets in the input buffers. The term  $(N-j)/N$  in the inner loop is the probability that the destination for the packet being checked is an unoccupied output port. After checking every input buffer, we find the value of variable  $j$  and use it to calculate  $TH(1)$ . Then,  $w$  is increased by one to calculate  $TH(2)$ . Using this method,  $f$  loops later, we will obtain  $TH(f)$ .

The maximum throughputs considering the window overhead are illustrated in Fig. 5, which shows that, with FIFO input queuing (window size = 1), the maximum throughput of a 1024 × 1024 crossbar switch is 0.62. With the aid of

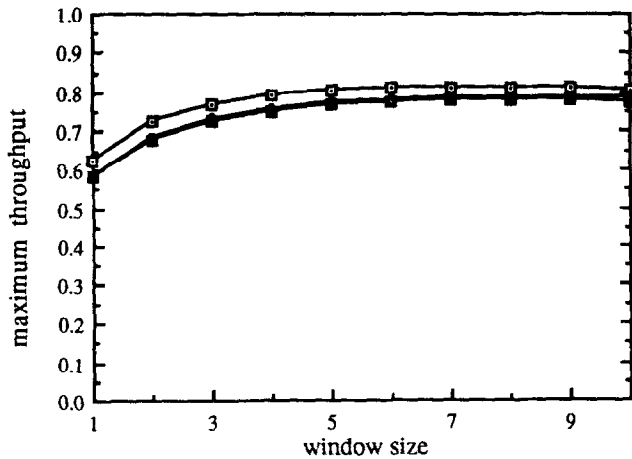


Fig. 7. Simulation results of the maximum throughput for a  $64 \times 64$  crossbar switch under uniform traffic.  $\square$ : analysis;  $\blacklozenge$ : simulation (look-through);  $\blacksquare$ : simulation (look-back).

bypass input queueing, the throughput can be improved to 0.77 with an optimal window size equal to 5. Fig. 6 is the simulation results of the maximum throughput for a  $64 \times 64$  crossbar switch under various window sizes and policies, compared with the analytical results. It is shown in this figure that the analytical results match well with the simulation results. We also observe that the look-through and look-back policies achieve almost the same throughput, since the switch size is large and the traffic pattern is uniform.

### 3. Simulation results under nonuniform traffic

For nonuniform traffic, it is very difficult to find the maximum throughput analytically. Hence, we run numerous simulations to learn the impact of traffic pattern on the throughput under different switch architectures and service policies.

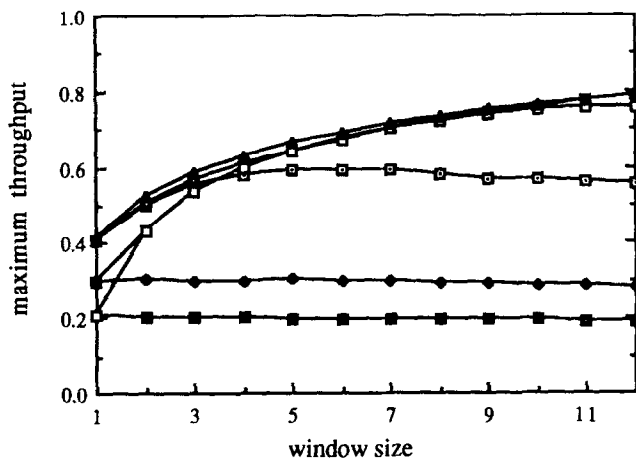


Fig. 8. Simulation results of the maximum throughput for a  $16 \times 16$  Banyan switch when the hot-spot ratio equals 0.1, 0.2 and 0.3, respectively.  $\blacktriangle$ : look-through (uniform);  $\blacklozenge$ : look-through ( $h = 0.1$ );  $\blacksquare$ : look-through ( $h = 0.2$ );  $\square$ : look-through ( $h = 0.3$ );  $\square$ : look-back ( $h = 0.1$ );  $\blacklozenge$ : look-back ( $h = 0.2$ );  $\blacksquare$ : look-back ( $h = 0.3$ ).

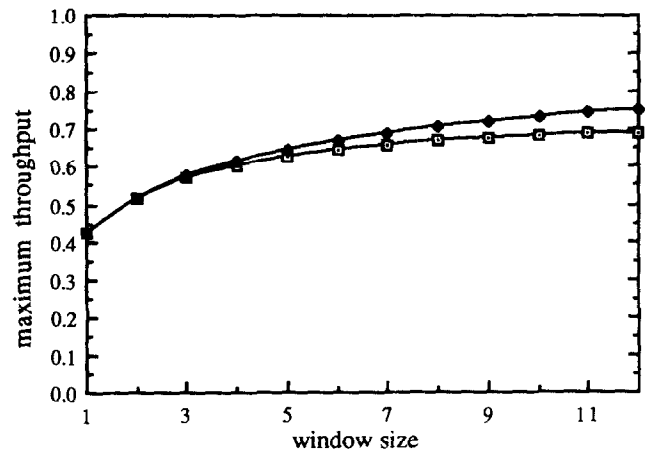


Fig. 9. Simulation results of the maximum throughput for a  $16 \times 16$  Banyan switch under mixed SSSD/uniform traffic.  $\square$ : look-back;  $\blacklozenge$ : look-through.

#### 3.1. Banyan switch

The switch simulated is a  $16 \times 16$  Banyan switch. Fig. 7 shows the simulation results of the maximum throughput under various window sizes and policies, providing that the hot-spot ratios are 0.1, 0.2 and 0.3, respectively. From this figure, we observe that the maximum throughput obtained by the look-through policy is much higher than that obtained by the look-back policy under higher hot-spot ratios. Moreover, notice that a higher hot-spot ratio only slightly decreases the throughput for the look-through policy as long as the window size is properly chosen and the hot-spot ratio is not too high. For the look-back policy, a higher hot-spot ratio will significantly decrease the throughput. The reason why the maximum throughput of the switch with look-back under a higher hot-spot ratio is almost independent of window sizes is because there are many hot-spot packets queued in the front of the input buffers, so the input port controllers cannot find more non-hot-spot packets in the bypass windows to send, even for a large window size.

For mixed SSSD/uniform traffic, Fig. 8 shows the simulation results of the maximum throughput under various window sizes and policies. We observe that the look-through policy achieves a slightly better throughput than the look-back policy.

#### 3.2. Crossbar switch

The switch simulated here is a  $64 \times 64$  crossbar switch. Fig. 9 shows the simulation results of the maximum throughput under various window sizes and policies, providing that the hot-spot ratios are 0.1, 0.2 and 0.3, respectively. Again, we observe that the throughput obtained by the look-through policy is much higher than that obtained by the look-back policy. Similar to the result of Banyan switches, the throughput is only slightly decreased for a higher hot-spot ratio for the look-through policy.

Under mixed SSSD/uniform traffic, a  $64 \times 64$  crossbar switch is composed of a leased line and a  $63 \times 63$

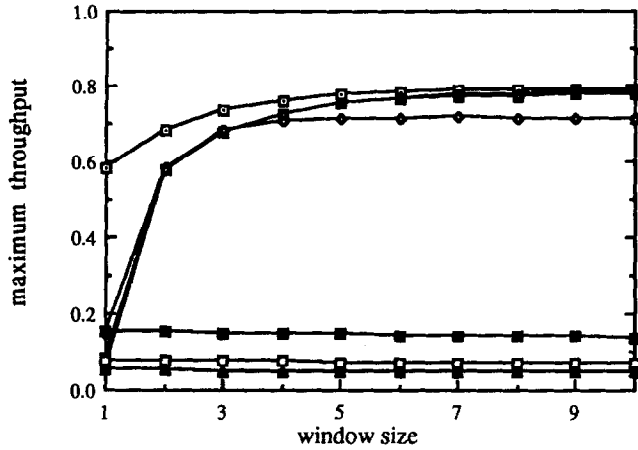


Fig. 10. Simulation results of the maximum throughput for a  $64 \times 64$  crossbar switch when the hot-spot ratio equals 0.1, 0.2 and 0.3, respectively.  $\square$ : look-through (uniform);  $\blacklozenge$ : look-through ( $h = 0.1$ );  $\blacksquare$ : look-through ( $h = 0.2$ );  $\blacklozenge$ : look-through ( $h = 0.3$ );  $\blacksquare$ : look-back ( $h = 0.1$ );  $\square$ : look-back ( $h = 0.2$ );  $\blacktriangle$ : look-back ( $h = 0.3$ ).

crossbar switch. Fig. 10 shows the simulation results of the maximum throughput under various window sizes and policies. We observe that the look-through and look-back policies achieve almost the same throughput. This is because the switch size is large and the traffic pattern, except the leased line, is uniform.

#### 4. Parallel-plane ATM switches

Bypass queueing is efficient to overcome the problem of head of line blocking within an input queue. However, when many packets in different input buffers contend for the same internal circuit of the switch, only one of them can be transmitted in the transmission slot. To relieve this situation, we propose a parallel-plane architecture [7-10] to reduce the contention across input queues. In this section, we will find the maximum throughput analytically under uniform traffic.

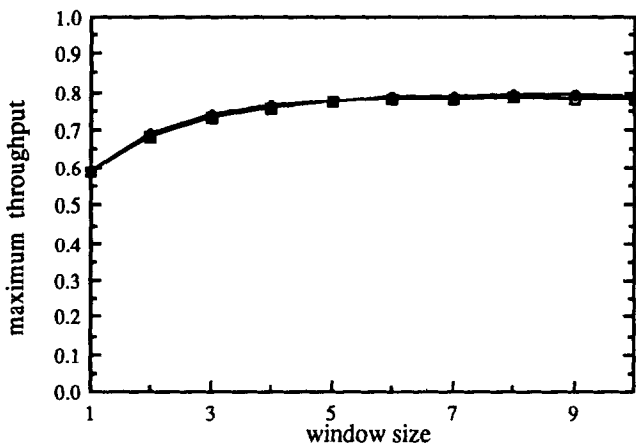


Fig. 11. Simulation results of the maximum throughput for a  $64 \times 64$  crossbar switch under mixed SSSD/uniform traffic.  $\blacklozenge$ : look-through;  $\square$ : look-back.

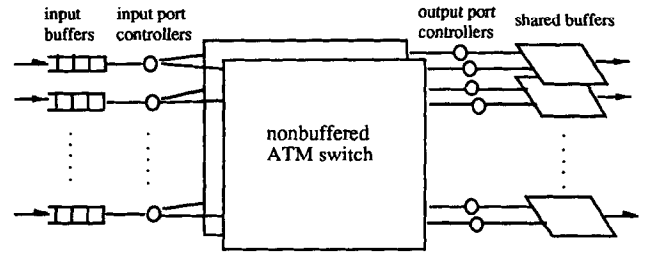


Fig. 12. Architecture of the parallel-plane ATM switch with bypass queues.

##### 4.1. Banyan switch

Although the bypass queue has improved the switch throughput significantly, the throughput is still not high enough. For instance, the throughput for a  $1024 \times 1024$  Banyan switch with an optimal bypass window is only 0.44, as stated earlier. To increase the throughput, the parallel-plane ATM switch is proposed. Fig. 11 shows the architecture of the parallel-plane ATM switch. Its operation is similar to the single-plane switch except that the parallel-plane switch provides multiple paths. A cell will be sent to all switch planes for connection setup. If more than one connection is established within one offer slot, only one connection will be chosen and the others will immediately be released at the end of that offer slot. Since the parallel-plane switch will cause the problem of output contention, a form of output queueing, the *shared buffer*, is used to overcome this problem [1]. To keep the packet loss probability lower than a specified value, the size of the shared buffer must be calculated, and can be found in Ref. [19].

With the same assumptions as stated in Section 2, and assuming  $j$  paths have already been set up before an offer slot, it is reasonable to assume that these  $j$  paths are uniformly distributed among all planes. Defining  $m$  as the number of switch planes, the following is the analysis to obtain  $\alpha_j$ , as can be found in Ref. [16]:

$$\begin{aligned}
 \rho_0 &= \rho \\
 \rho_1 &= \rho_0 \left[ -\frac{j}{Nm} \cdot \frac{1}{2} - \left(1 - \frac{j}{Nm}\right) \cdot \rho_0 \cdot \frac{1}{2} \cdot \frac{1}{2} \right] \\
 &\vdots \\
 \rho_n &= \rho_{n-1} \left[ -\frac{j}{Nm} \cdot \frac{1}{2} - \left(1 - \frac{j}{Nm}\right) \cdot \rho_{n-1} \cdot \frac{1}{2} \cdot \frac{1}{2} \right] \\
 \alpha_j &= 1 - (1 - \rho_n / \rho_0)^m
 \end{aligned} \tag{3}$$

Based on the assumption that  $j$  paths are uniformly distributed among these  $m$  planes, there are  $j/m$  paths for each plane. We use this term to calculate  $\rho_n$  as we did in Section 2. Then,  $\alpha_j$  is obtained by Eq. (3). The second term in Eq. (3) is the probability that all connection attempts in  $m$  planes are failed for a packet in the input buffer.

After obtaining  $\alpha_j$ , Eq. (2) is again applied to find the

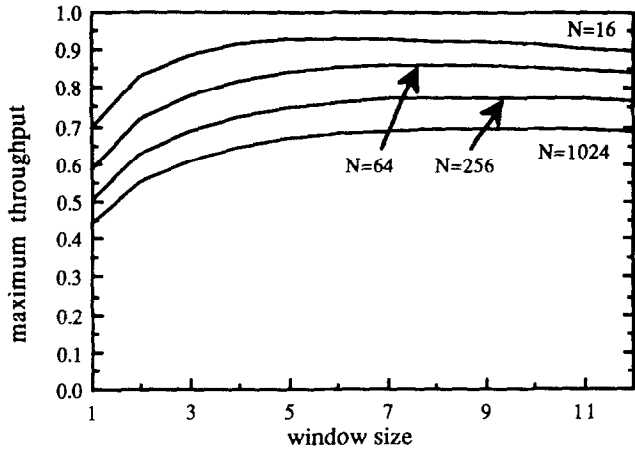


Fig. 13. Maximum throughput of 2-plane Banyan switches under uniform traffic.

system throughput. The maximum throughputs for  $m = 2$  and 3 are illustrated in Figs. 12 and 13, respectively. Note that, as the number of switch planes increases, the optimal window size decreases such that the switch complexity is reduced. Fig. 14 shows the simulation and analytical results of the maximum throughput for a  $16 \times 16$  Banyan switch with the look-through policy under various window sizes and  $m = 2$ . From this figure, we observe that the analytical results conform to the simulation results perfectly. As the number of planes grows, the analytical results conform even better to the simulation results.

In Fig. 15, we show the maximum throughputs under various numbers of switch planes and switch sizes, providing the optimal window size. Note that the throughput increment becomes insignificant when more than four switch planes are used; therefore, only three or four planes are needed to obtain a very high throughput. For  $N = 1024$  the throughput is as high as 0.85 when  $m \geq 4$ . For  $N = 256$ , the throughput is higher than 0.9 when  $m \geq 4$ . In general, a switch with hundreds of input ports is

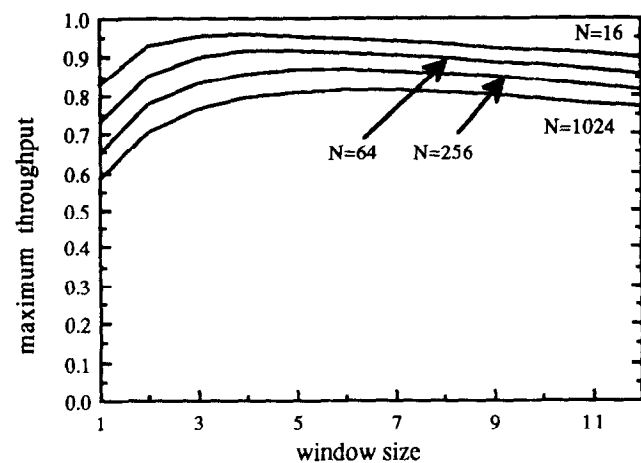


Fig. 14. Maximum throughput of 3-plane Banyan switches under uniform traffic.

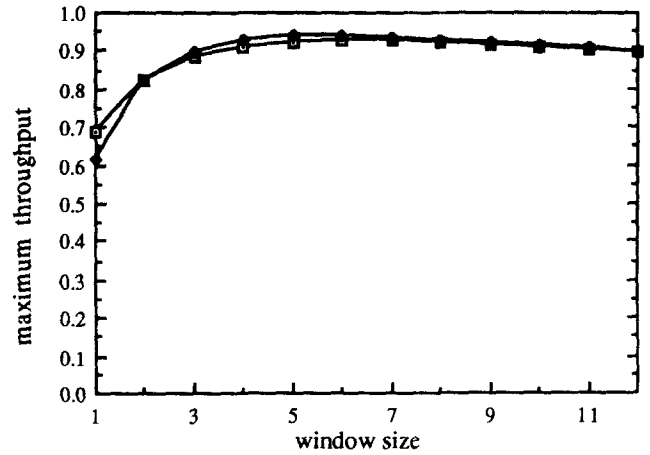


Fig. 15. Simulation results of the maximum throughput for a 2-plane  $16 \times 16$  Banyan switch under uniform traffic. □: analysis; ◆: simulation (look-through).

considered a large switch; hence, this architecture obtains a high throughput even for large switches.

For parallel-plane switches under non-uniform traffic pattern, Fig. 16 gives the results for a three-plane  $16 \times 16$  switch under hot-spot traffic with a hot-spot ratio equal to 0.3. This figure shows that, for the look-through policy, the maximum throughput reaches 0.88 when the window size equals only three. If only one plane is used, the window size would be as high as ten to achieve a maximum throughput at 0.75. Note that a smaller window size will lead to a simpler architecture for the input port controller, and also reduce the delay time incurred. Also note that for a window size equal to three, the three-plane switch obtains a throughput at 0.88 while the single-plane switch only obtains 0.52. These results show that parallel-plane with bypass queuing is very effective for non-uniform traffic. For the look-back policy, Fig. 17 shows that the throughput of the three-plane architecture is about

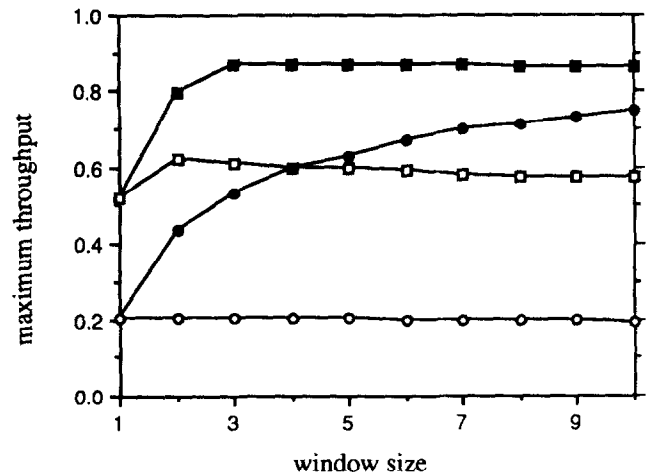


Fig. 16. Maximum throughput of a 3-plane  $16 \times 16$  Banyan switch under hot-spot traffic with hot-spot ratio = 0.3. ■: look-through (3-plane); □: look-back (3-plane); ●: look-through (1-plane); ○: look-back (1-plane).

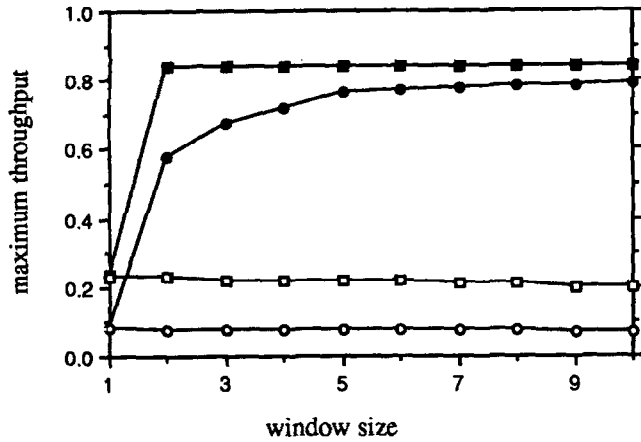


Fig. 17. Maximum throughput of a 3-plane 64 x 64 crossbar switch under hot-spot traffic with hot-spot ratio = 0.2. ■: look-through (3-plane); □: look-back (3-plane); ●: look-through (1-plane); ○: look-back (1-plane).

three times that of the single-plane architecture. Hence, the parallel-plane architecture is indeed very effective for non-uniform traffic.

4.2. Crossbar switch

With the same assumptions as stated in Section 2, and assuming that the paths which are already set up are uniformly distributed among all planes, the throughput analysis is similar to that of Section 2:

```

Set  $j = 0$  and  $S(0) = 0$ 
Initial  $w = 1$ 
for_each  $w \leq f$  do
begin
  Initial  $p = 1$ 
  for_each  $p \leq N$  do
begin

```

$$\text{set } j = j + \left[ 1 - \frac{S(w-1)}{N} \right] \cdot \left[ 1 - \left( \frac{j}{Nm} \right)^m \right]$$

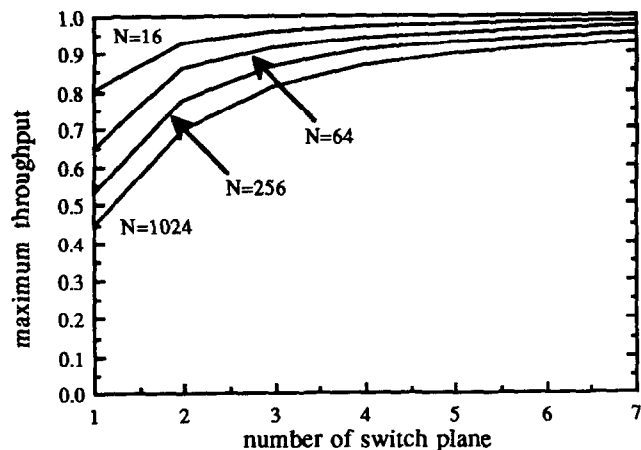


Fig. 18. Maximum throughput of parallel-plane Banyan switch using optimal window size for various switch planes under uniform traffic.

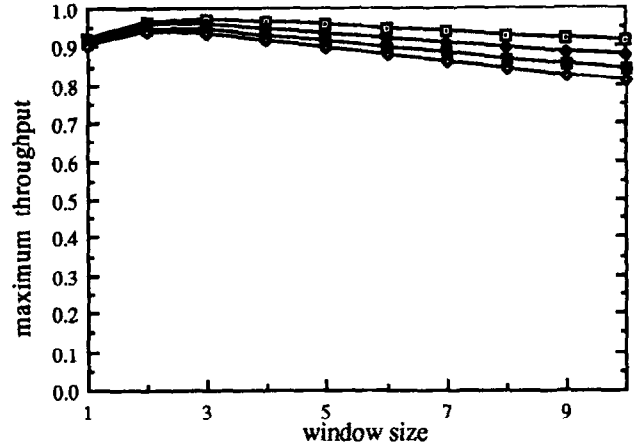


Fig. 19. Maximum throughput of 2-plane crossbar switches under uniform traffic. □: 16 x 16; ◆: 64 x 64; ■: 256 x 256; ○: 1024 x 1024.

```

end
Set  $S(w) = j$ 
Set  $TH(w) = \frac{S(w)}{N(1 + w\beta)}$ 
end

```

The term  $(j/Nm)^m$  in the inner loop means the probability that  $m$  requests for transmission in  $m$  planes are all failed. Hence,  $[1 - (j/Nm)^m]$  is the probability that there is at least one available output port for the packet examined. The maximum throughputs for  $m = 2$  are illustrated in Fig. 18. Fig. 16 contains the simulation results of the maximum throughput for a 64 x 64 crossbar switch with the look-through policy under various window sizes and  $m = 2$ . From this figure, we can observe the preciseness of the analysis.

In Fig. 19, we show the maximum throughputs under various numbers of switch planes and switch sizes, providing the optimal window size. Note that the throughput increment becomes insignificant when more than two switch planes are used; therefore, only two or three planes

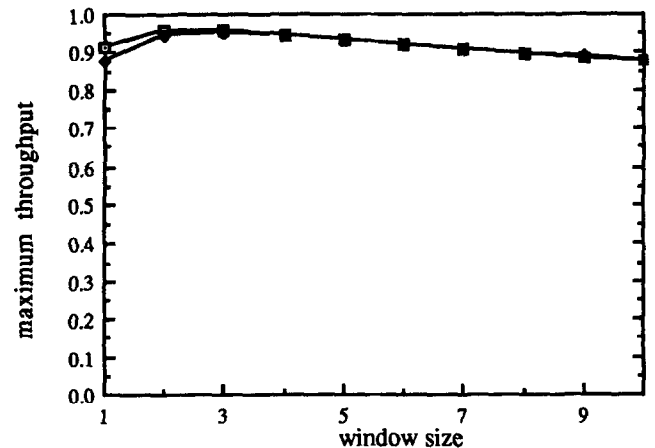


Fig. 20. Simulation results of the maximum throughput for a 2-plane 64 x 64 crossbar switch under uniform traffic. □: analysis; ◆: simulation (look-through).

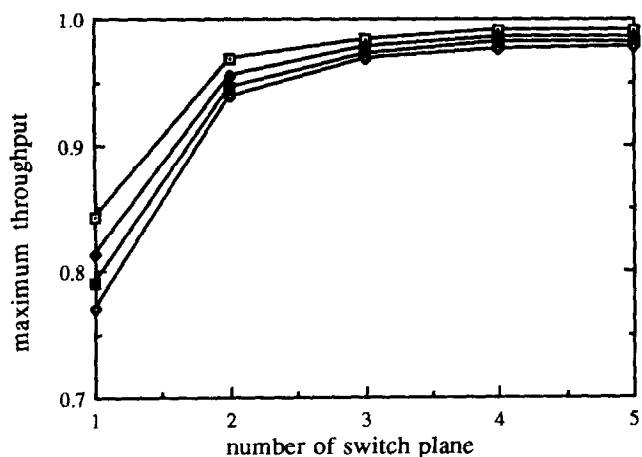


Fig. 21. Maximum throughput of parallel-plane crossbar switch using optimal window size for various switch planes under uniform traffic. □:  $N = 16$ ; ◆:  $N = 64$ ; ■:  $N = 256$ ; ◇:  $N = 1024$ .

are needed to obtain a very high throughput. Also, note that the maximum throughput for  $N = 1024$  can be as high as 0.94 when  $m \geq 2$ . Fig. 20 shows the simulation results of the maximum throughput for a two-plane  $64 \times 64$  crossbar switch under uniform traffic.

For a non-uniform traffic pattern, Fig. 17 gives the results for a three-plane  $64 \times 64$  switch under hot-spot traffic with a hot-spot ratio equal to 0.2. This figure shows that the maximum throughput reaches 0.85 when the window size equals only two for the look-through policy. If only one plane is used, the window size would be as high as ten to achieve a maximum throughput at 0.79. This figure also shows that, for the look-back policy, the throughput of the three-plane architecture is about three times that of the single-plane architecture. These results again show that the parallel-plane architecture is effective under non-uniform traffic.

Since a smaller window size will lead to a simpler architecture for the input port controller, it also reduces the delay time incurred. This result shows that parallel-plane with bypass queueing is very effective for non-uniform traffic. For the look-back policy, Fig. 17 also shows that the three-plane architecture is about three times that of the single-plane architecture; hence, the parallel-plane architecture performs very well indeed. Fig. 21 shows the maximum throughput of the parallel-plane crossbar switch using an optimal window size for various switch planes under uniform traffic.

## 5. Conclusions

With the assumption of a uniform traffic pattern, many elegant switch architectures have been proposed, such as the Knockout [1, 2] and Christmas tree [4] switches. However, with a non-uniform traffic pattern, especially hot-spot traffic, the elegance of these switches will soon disappear. For example, the size of the concentrator of Knockout switches will grow rapidly as the hot-spot ratio rises. Unfortunately, a

hot-spot traffic pattern is common in real networks because there are many databases and file servers distributed in the network. Banyan and crossbar switches have simpler architectures but lower throughputs compared to a Knockout switch. However, with the aid of a bypass input queue and parallel-plane architecture, the throughput can be improved significantly without adding too much complexity, even with a hot-spot traffic pattern. Hence, such a switch architecture is of much practical use. All in all, we can summarize the paper as follows:

- A bypass queue can improve throughput significantly.
- The look-through policy obtains a much higher throughput than the look-back policy under a hot-spot traffic pattern.
- As long as the hot-spot ratio is not high, the throughput of the look-through policy will decrease only slightly compared to that under uniform traffic.
- Based on the above observation, the throughput analysis under a non-uniform traffic pattern can be approximated by the analytical results under uniform traffic as long as the hot-spot ratio is moderate.
- The parallel-plane architecture can further improve the throughput.
- Both the optimal window size and the optimal number of planes are small (smaller than 4), hence the additional switch complexity is low.

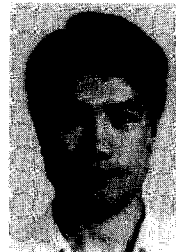
## References

- [1] Y.S. Yeh, M.G. Hluchyj and A.S. Acampora, The Knockout switch: a simple, modular architecture for high-performance packet switching. *IEEE J. Select. Areas Commun.*, (October 1987) 1274–1283.
- [2] K.Y. Eng, M.G. Hluchyj and Y.S. Yeh, A Knockout switch for variable-length packets. *IEEE J. Select. Areas Commun.*, (December 1987) 1426–1435.
- [3] K.Y. Eng, M.J. Karol and Y.S. Yeh, A growable packet (ATM) switch architecture: design principles and applications. *IEEE Trans. Commun.*, 40(2) (February 1992) 423–430.
- [4] Weijia Wang and F.A. Tobagi, The Christmas-tree switch: an output queueing space-division fast packet switch based on interleaving distribution and concentration functions. *IEEE INFOCOM '91*, 1991, 0163–0170.
- [5] E.W. Zegura, Architectures for ATM switching systems. *IEEE Commun. Magazine* (February 1993) 28–37.
- [6] F.A. Tobagi, Fast packet switch architectures for broadband integrated services digital networks. *Proc. IEEE*, 78(1) (January 1990) 133–167.
- [7] H. Ahmadi and W.E. Denzel, A survey of modern high performance switching techniques. *IEEE J. Select. Areas Commun.*, 7 (September 1989) 1091–1103.
- [8] Y. Oie et al., Survey of switching techniques in high-speed networks and their performance. *IEEE INFOCOM '90*, 1990, 1242–1251.
- [9] P. Newman, A fast packet switch for the integrated services backbone network. *IEEE J. Select. Areas Commun.*, 6(9) (December 1988) 1468–1479.
- [10] K.W. Sarkies, The bypass queue in fast packet switching. *IEEE Trans. Commun.*, 39(5) (May 1991) 766–774.
- [11] J.Y. Hui and E. Arthurs, A broadband packet switch for integrated transport. *IEEE J. Select. Areas Commun.*, 5(8) (October 1987) 1264–1273.

- [12] R.G. Bubenik and J.S. Turner, Performance of a broadcast packet switch. *IEEE Trans. Commun.*, 37(1) (January 1989) 60–69.
- [13] K. Shiomoto et al., Performance evaluation of cell bypass queueing discipline for buffered Banyan type ATM switches. *IEEE INFOCOM '90*, 1990, 677–685.
- [14] H.S. Kim and A. Leon-Garcia, Performance of buffered Banyan networks under nonuniform traffic patterns. *IEEE Trans. Commun.*, 38(5) (May 1990) 648–658.
- [15] M.G. Hluchyj and M. Karol, Queueing in high-performance packet switching. *IEEE J. Select. Areas Commun.*, 6(9) (December 1988) 1587–1597.
- [16] Y.-S. Su and J.-H. Huang, Throughput analysis and optimal design of Banyan switches with bypass queues. *IEEE Trans. Commun.*, 42(10) (October 1994) 2781–2784.
- [17] Y.C. Jeng, Performance analysis of a packet switch based on a single-buffered Banyan network. *IEEE J. Select. Areas Commun.*, 1(6) (December 1983) 1014–1021.
- [18] S. Nojima et al., Integrated services packet network using bus matrix switch. *IEEE J. Select. Areas Commun.*, 5 (October 1987) 1284–1292.
- [19] Y.-S. Su, Performance and optimal design of ATM switches with bypass queues, MS thesis, Department of Computer Science and Information Engineering, National Taiwan University, June, 1993.



*Yo-Song Su received his BS in computer science and information engineering from Tatung Institute of Technology, Taipei, Taiwan in 1988. From 1988 to 1990 he served in the Army as an Electronics Officer. In 1991 he attended the graduate school and received an MS in computer science and information engineering from National Taiwan University, Taipei, Taiwan in 1993. Since then he has worked for Syscom Computer Engineering Co. as an engineer in Taiwan. His interest is focused on the design of the ATM switch and its performance evaluation.*



*Jau-Hsiung Huang received the BS in electrical engineering from National Taiwan University, Taipei, Taiwan in 1981 and the MS and PhD in computer science from the University of California, Los Angeles in 1985 and 1988, respectively. Since 1988 he has been a member of the faculty in the Department of Computer Science and Information Engineering, National Taiwan University, where he is currently a professor. He has published over 40 technical papers in the areas of multimedia networking, high speed networking, parallel and distributed systems and the performance evaluation of computing systems.*