# Throughput Analysis and Optimal Design of Banyan Switches with Bypass Queues

Yo-Song Su and Jau-Hsiung Huang

***Abstract*—A Banyan switch with bypass queue can overcome the problem of head of line blocking. This paper presents the throughput analysis for such a switch and the optimal window size for the bypass queue is also obtained. Moreover, to reduce the effect of the internal blocking and to increase the throughput, a parallel Banyan switch with bypass queue is also discussed and analyzed.**

## I. Introduction

ALTHOUGH a switch using output queueing has the characteristic of internal nonblocking and has much better throughput and delay performance than switches using input queueing, hardware cost is high for large switches [1], [2]. On the other hand, input queueing is simple in the hardware aspect, while it suffers from the problem of head of line blocking. This is the phenomenon that a packet in the input buffer is prevented from accessing an available output port because the packet ahead of it in the buffer is 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 packet is blocked. Such an input buffer is called a bypass queue [3].

An $N \times N$ nonbuffered Banyan switch consists of $N$ input buffers, $N$ output buffers, $N$ input port controllers, $N$ output port controllers, and an $N \times N$ nonbuffered Banyan fabric. The $N$ input buffers and $N$ output buffers are used to store and to synchronize packets. The function of the input port controllers 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.

Inside the switch, time is divided into *packet slots* which contains several *offer slots* and one *transmission slot* as illustrated in Fig. 1. In an offer slot, the packet routing tag is used to attempt to establish a path through the switch to the desired output port. Hence, if the switch is designed carefully, at most $\log_2 N$ bit period are required for an offer slot. We define the *window* size as the number of offer slots in a packet slot. At the end of the offer slots, all packets which have successfully set up a path to their output ports are transmitted during the transmission slot. Note that the 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 packet length and $f$ is the window size of a packet slot.

Paper approved by E. G. Sable, the Editor for Communication Switching of the IEEE Communications Society. Manuscript received July 27, 1992; revised September 27, 1992.

The authors are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

IEEE Log Number 9404713.

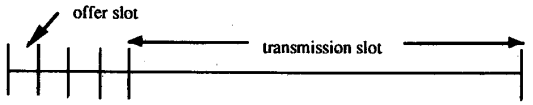


Fig. 1. The structure of a packet slot.

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 in order to maximize the switch throughput.

## II. Performance Analysis of the Switch

We assume that packets arriving at each input port are destined uniformly for all output ports and the loads for all input ports are the same. We also assume that the two input links of a switching element are statistically independent. Under these assumptions, The state of a stage can be reduced to that of a single input link of the stage [4]. We introduce the following notation before the analysis.

- $n$ = number of stages in the switch (i.e., $n = \log_2 N$).
- $\rho_{i-1}$ = Prob[there is a packet coming to stage $i$].
- $\rho_i$ = Prob[there is a packet departing from stage $i$].
- $\beta$ = 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 following is the analysis to obtain $\alpha_j$ which is the key parameter in the following throughput analysis.

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

The second term in the square bracket is the probability that the request in an input link of a switching element fails since

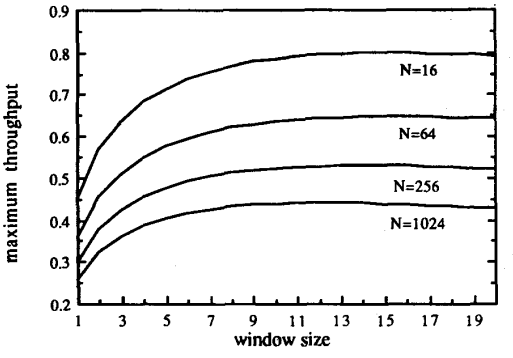


Fig. 2. Maximum throughput for various window sizes.

there is a path already set up for the other input link (with probability $j/N$) and with probability $1/2$ that the set up path has the same output as our requesting packet. The third term in the square bracket is the probability that the request fails since there is another request (not an already set up path) from the other input link (with probability $(1-\frac{j}{N})\times\rho_{i-1}$ ) and with probability 1/2 both input requests are destined for the same output link and with probability 1/2 our requesting packet will lose and be dropped.

To find the throughput of the switch, we have to consider the overhead caused by the offer slots. This overhead will reduce the throughput by a fraction $1/(1+k\beta)$ if $k$ offer slots are used. The computation of the maximum throughput of the switch using $f$ offer slots follows. In order to find the maximum throughput, we set $\rho_o = 1$ for all computation. That is, the switch is operated under saturation.

Initially, $S(0) = 0$, $B(0) = N$, $\mathrm{TH}(0) = 0$.
Compute $\alpha_0$ using Eq. (1).
$S(1) = N \times \alpha_0$, $B(1) = N - S(1)$,
$\mathrm{TH}(1) = S(1)/N(1+B)$.
Compute $\alpha_{S(1)}$ using (1).
$S(2) = S(1) + B(1) \times \alpha_{S(1)}$, $B(2) = N - S(2)$,
$\mathrm{TH}(2) = S(2)/N(1+2\beta)$.
: : :
Compute $\alpha_{S(f-1)}$ using (1).
$S(f) = S(f-1) + B(f-1) \times \alpha_{S(f-1)}$,
$B(f) = N - S(f)$,
$\mathrm{TH}(f) = S(f)/N(1+f\beta)$. (2)

The maximum throughput considering the window overhead is illustrated in Fig. 2. Fig. 2 shows that, with FIFO input queueing (window size = 1), the maximum throughput of a 10-stage Banyan switch is only 0.25. However, bypass input queueing can improve the throughput up to 0.44, with the optimal window size equals 12.

## III. Parallel Banyan Switches

Although the bypass queue has improved the switch throughput significantly, the throughput is still low. For instance, the throughput for a $1024 \times 1024$ switch is only 0.44 as stated earlier. To reduce the effect of the internal blocking and to increase the throughput, the parallel Banyan switch is proposed in [3] as illustrated in Fig. 3. Its operation is similar

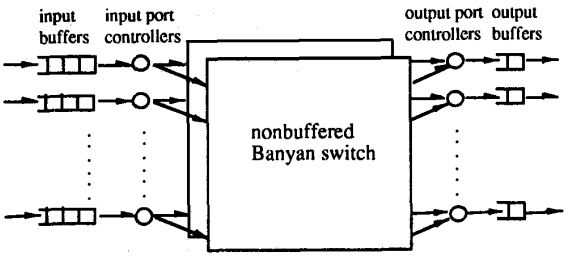


Fig. 3. Architecture of the parallel Banyan switch with bypass queue.

to the single-plane Banyan switch except that the parallel Banyan switch provides multiple paths. In this architecture, a cell will be sent to all switch planes for connection setup. However, if more than one connection is established, only one connection will be chosen and the others will be released. It is reasonable to assume that those $j$ paths which are already set up are evenly distributed among all planes; hence, the probability to pass through one of the planes increases. Therefore, the value of $\alpha_j$ and the switch throughput can be increased.

The throughput analysis of the parallel Banyan switch is similar to that of the single-plane Banyan switch, except for the computation of $\alpha_j$. We define $m$ as the number of the switch planes and the following is the analysis to obtain $\alpha_j$.

$$\rho_0 = \rho$$

$$\rho_1 = \rho_0 \times \left[1 - \frac{j}{Nm} \times \frac{1}{2} - \left(1 - \frac{j}{Nm}\right) \times \rho_0 \times \frac{1}{2} \times \frac{1}{2}\right]$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$\rho_n = \rho_{n-1} \times \left[1 - \frac{j}{Nm} \times \frac{1}{2} - \left(1 - \frac{j}{Nm}\right) \times \rho_{n-1} \times \frac{1}{2} \times \frac{1}{2}\right]$$

$$\alpha_j = 1 - (1 - \rho_n/\rho_0)m \tag{3}$$

The maximum throughputs for $m = 2$, 3, 4 are illustrated in Figs. 4, 5, and 6 respectively. Note that as the number of switch planes increases, the optimal window size decreases and the maximum throughput gets more sensitive with regard to the window size. Hence, the window size should be more carefully chosen. In Fig. 7, we show the maximum throughput for various numbers of switch planes using the optimal window size. Note that the throughput increment is insignificant when more than four switch planes are used. Also note that the maximum throughput for $N = 1024$ can be as high as 0.85 when $m \geq 4$. Further, the throughput is higher than 0.9 for $N = 256$ when $m \geq 4$.

## IV. Simulation Results and Discussions

Figs. 8 to 11 are the simulation results of the mean delay time for a $16 \times 16$ switch under various loads and window sizes for $m = 1$ to $m = 4$. The delay time is measured in *time slots* where one time slot is the time for the switch to transmit an ATM cell (i.e., 424 bits). By comparing Figs. 2, 4,

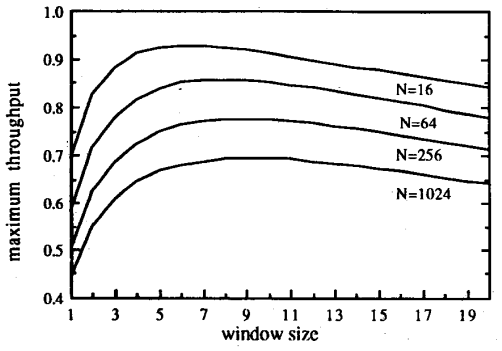


Fig. 4. Maximum throughput of a 2-plane switch for various window sizes.

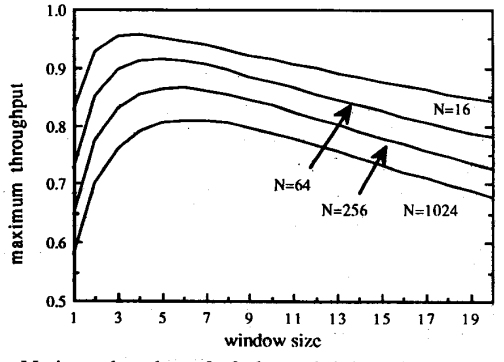


Fig. 5. Maximum throughput of a 3-plane switch for various window sizes.

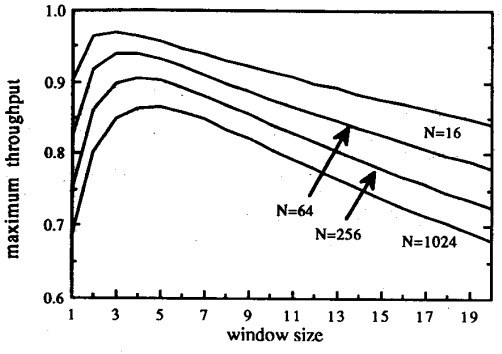


Fig. 6. Maximum throughput of a 4-plane switch for various window sizes.

5, and 6 with Figs. 8–11, we can see that the analytical results match fairly well with the simulation results. For instance, for $m = 3$, we see that the maximum throughput is obtained around 0.95 when the window size equals 3 or 4 from Fig. 5; this result is also observed in Fig. 10 for $\rho = 0.95$. Fig. 12 shows the mean delay time for various switch planes using the optimal window size under various loads. From this figure, we observe that the mean delay time improves a lot when the number of switch planes increases from 1 to 2. However, when the number of switch planes changes from 3 to 4, the improvement in the mean delay time becomes insignificant. This result also suggests that it is not necessary to use more than 4 switch planes.

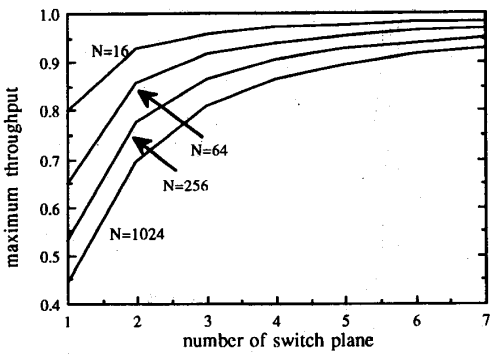


Fig. 7. Maximum throughput using optimal window size for various switch planes.

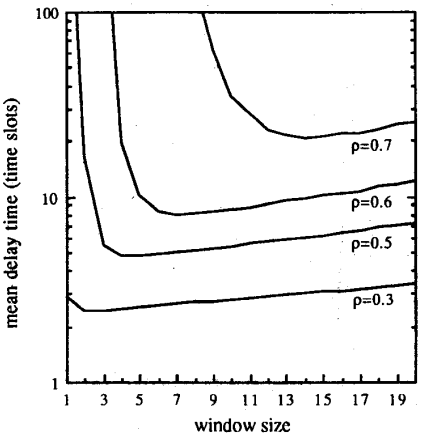


Fig. 8. Simulations for mean delay time for a $16 \times 16$ switch when $m = 1$.

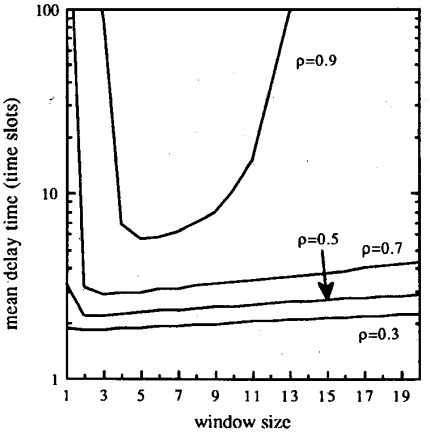


Fig. 9. Simulations for mean delay time for a $16 \times 16$ switch when $m = 2$.

The potential problem of such a switch is that packets may be out of sequence at the receiver. To resolve this problem, a more careful design should be provided in the input port controller. However, as long as the window size and the number of switch planes are small, the hardware overhead required for the bypass queue will not be significant.

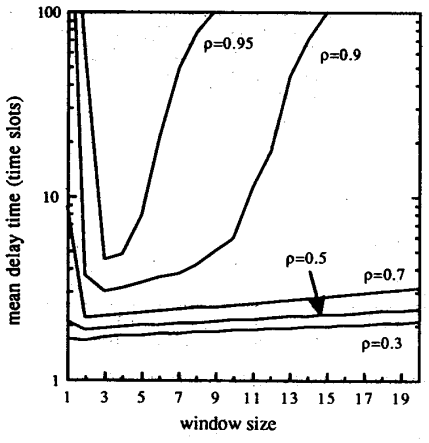


Fig. 10. Simulations for mean delay time for a $16 \times 16$ switch when $m = 3$.

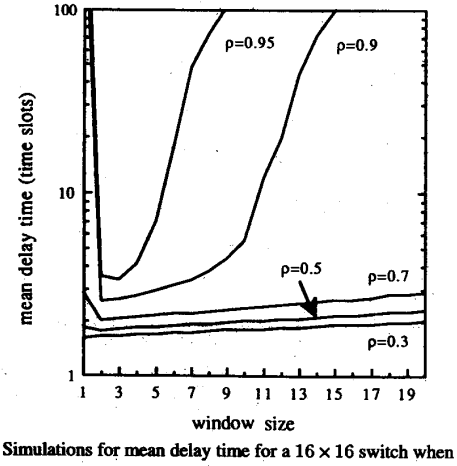


Fig. 11. Simulations for mean delay time for a $16 \times 16$ switch when $m = 4$.

## V. Conclusion

Bypass queueing is efficient to overcome the problem of head of line blocking. According to the analysis, the optimal window size for such a switch is suggested. Moreover, the analysis is also performed on the parallel Banyan switch architecture to reduce the problem of internal blocking. It is shown that by using four switch planes with window size equals four, the Banyan switch with bypass queue can obtain a throughput as high as 0.9 for a large switch. This throughput is higher than that of switches with output queueing. By examining the hardware complexity, the Banyan switch has a complexity of switching elements of order $N \log N$ which is much smaller than that of output queueing switches (of order $N^2$). Hence, the hardware complexity for Banyan switch with bypass queue is still low compared to output queueing switches even if four switch planes are used.

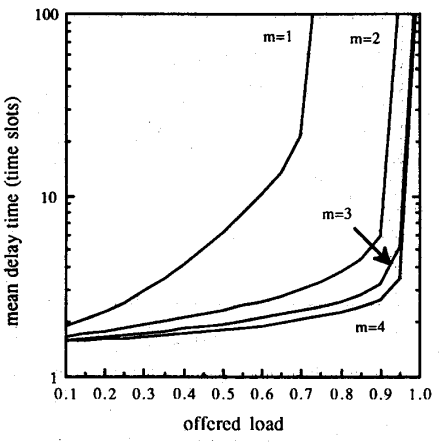


Fig. 12. Simulations for mean delay time using optimal window size for a $16 \times 16$ switch.

## References

[1] M. G. Hluchyj and M. Karol, "Queueing in high-performance packet switching," *IEEE J. Select. Areas Commun.*, vol. 6, Dec. 1988.
[2] 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.*, vol. SAC-5, pp. 1274–1283, Oct. 1987.
[3] K. W. Sarkies, "The bypass queue in fast packet switching," *IEEE Trans. Commun.*, vol. 39, May 1991.
[4] Y. C. Jeng, "Performance analysis of a packet switch based on a single-buffered Banyan network," *IEEE J.Select.Areas Commun.*, vol. SAC-1, pp. 1014–1021, Dec. 1983.