Queueing Analysis of Shared Buffer Switches for ATM Networks

Mahmoud Saleh and Mohammed Atiquzzaman

Department of Computer Science and Computer Engineering
La Trobe University, Melbourne 3083, Australia
Email: *atiq@cs.latrobe.edu.au*

Abstract

A model for the analysis of ATM switches based on shared buffer switching for Broadband ISDN networks is developed, and the results are compared with the simulation. Switches constructed from shared buffer switching elements (SE) no longer suffer from the head of line blocking which is a common problem in simple input buffering. The analysis models the state of the entire switch and extends the model introduced by Turner to global flow control with backpressure mechanism. It is shown that buffer utilization is better and throughput improves significantly compared with the same network using local flow control policy.

Key Words- *Delta networks, shared buffer switches, analytical modeling, Markov chain, ATM switches.*

1 Introduction

In recent years, broadband ISDN (BISDN) has received increasing attention for its capability to provide a wide variety of services like video communication, graphic applications, and high speed data communications. One of the most promising approaches for BISDN is the Asynchronous Transfer Mode (ATM). An ATM network transfers all information in fixed length packets called *cells*, and is characterized by simplified protocols, high speed links, and high capacity switching nodes. The core of the switching fabric, referred to as *interconnection network* (IN), includes all the equipment required to route the cells through the switching fabric. Among the proposed architectures for ATM switches, multistage interconnection networks (MINs) have attracted a great deal of attention due to the features they offer, such as self-routing capability and suitability for VLSI implementation.

Multistage switches can be unbuffered [1, 2] or buffered to increase the throughput and reduce loss of cells. Buffers can be used at the inputs or outputs (or both) of a switch. Alternatively, dedicated buffers can used at the inputs or outputs of switching elements (SE). Switches constructed from input buffered SEs have been studied in [3, 4, 5, 6, 7]. Jenq [3] developed a model for analyzing Banyan networks consisting of 2×2 switching elements with single buffer slots at each input of the switching element (SE) and operating under a uniform traffic. Szymanski [8] extended Jenq's model to arbitrary SE sizes and buffer sizes. Input buffered SEs have lower throughput due to head-of-line blocking. Output buffered SEs have been investigated in [9, 10] and have been found to have higher throughput at the expense of internal hardware speedup. Both the input and output buffered SEs have poor buffer utilization due to buffers being dedicated to the input and output links respectively. Shared buffered SEs have better buffer utilization and higher throughput. Moreover, given the same amount of buffer, the shared buffer is the best choice in terms of cell loss rate.

Turner [11] developed a similar model for switches with shared buffering. His model assumes independence between buffer slots, and uses backpressure mechanism to avoid cell loss inside the switch. Monterosso [12] developed a new method based on the exact model of the switching element, and without backpressure mechanism. This model, while accurate, is computationally intractable for switches constructed from large size SEs. Bianchi [13] introduced an alternative approach to reduce the computation while maintaining the high accuracy. All of the models described in [11, 12, 13] use local flow control to forward cells between consecutive stages of the switch. In this paper we develop a model for a switch based on the Delta-b interconnection network [1] having shared buffering and using global flow control. Global flow control provides better buffer utilization, and improves the overall performance of the switch significantly.

The paper is organized as follows. In Section 2, we develop our model based on the assumptions we introduce there. Construction of the corresponding simulation program is explained in Section 3. In Section 4, we examine our model with some numerical examples, and compare the results with the simulation and local flow control. Concluding remarks and further possible work are given in Section 5.

2 Analysis of the Shared Buffer ATM Switch

In this section, we develop a model for shared buffer multistage switch based on the Delta interconnection network utilizing the *global* flow control in contrast to *local* flow control developed in [11]. In global flow control, acceptance of a cell in the next stage depends not only on the state of the SE in the next stage, but also on whether some cells in that SE are forwarded to its successors during the same cycle. This allows more efficient buffer utilization , and considerably better performance as explained in Section 4. A recursive definition of Delta interconnection network with shared buffering is illustrated in Fig. 1.

2.1 Assumptions

We model each SE as a $B+1$ state Markov chain, where B is the total buffer space in an SE. The following assumptions are made regarding the switch, and its operation:

- The switch operates *synchronously*, i.e. the cells are submitted to the switch at the beginning of the time slots. This reflects an ATM switch in which all packets have fixed lengths, and fit exactly into one time slot.
- *Destination tag* is used to route a cell. The conflict inside the switch is resolved randomly, i.e. if two or more cells are destined to the same output, one is chosen at random.

0-7803-1820-X/94 $4.00 © 1994 IEEE

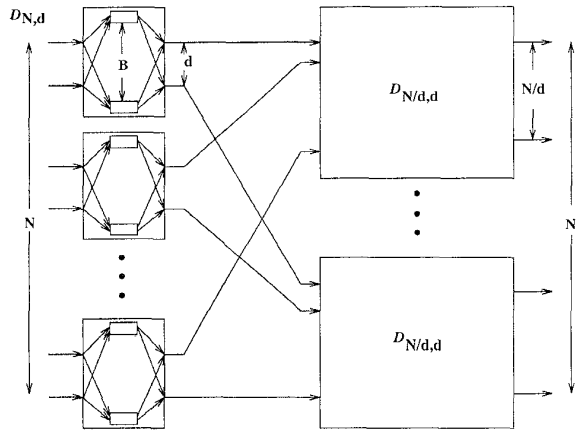

Figure 1: Recursive definition of Delta interconnection network with shared buffering.

- The switching elements (SEs) at a particular stage are statistically *indistinguishable*, and the state of a stage is determined by the state of an SE in the stage.
- The arrival of cells at each input of the switch is a *Bernoulli* process, i.e., the probability that a cell arrives during a time slot is constant, and the arrivals are independent of each other. Destination addresses are distributed uniformly.
- A *backpressure* mechanism with global flow control ensures that no cell is lost inside the switch. Thus, a cell leaves a stage if there is a space for it in the next stage's SE, or if a space becomes available during the same cycle. An acknowledgment policy is used to advise the receipt of a cell in the next stage's SE. Unacknowledged cells contend with other cells in subsequent cycles.
- There is no blocking at an output link of the switch, i.e., an output can always accept a cell.

For the purpose of analysis, we assume that each time slot is divided in two phases as illustrated in Fig. 2. During the *forward phase*, cells in an SE are forwarded to the next stage, and the SE goes to an *intermediate* state. During the *receive phase*, the available cells at the inputs of an SE are placed in the buffers, corresponding acknowledgments are issued, and the SE goes to the final state. If the number of arriving cells is greater than the available space, a number of cells equal to the number of available space are selected randomly.

2.2 Analysis of the ATM Switch

The following notations will be used in the model.

$\lambda_i(s1, s2)$: Probability that a stage i SE contains $s2$ cells at the beginning of the next cycle given that it contained $s1$ cell at the beginning of the current cycle.

$\tau_i(s1, s3)$: Probability that a stage i SE contains $s3$ cells at the end of the forward phase of the current cycle given that it contained $s1$ cells at the beginning of the cycle, where $s1 \geq s3$.

$\sigma_i(s3, s2)$: Probability that a stage i SE contains $s2$ cells at the end of the current cycle given that it contained $s3$ cells at the beginning of the receive phase, where $s3 \leq s2$.

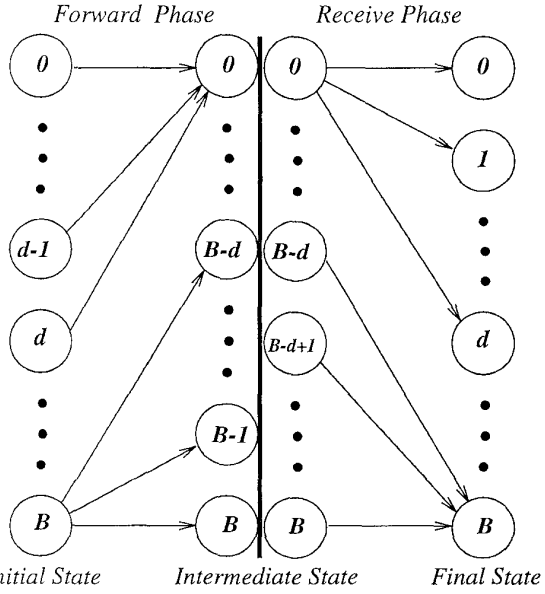

Figure 2: State diagram of a two phase switch operation.

$\theta_i(n1, n2)$: Probability that a stage i SE contains $n2$ cells at the beginning of the receive phase of the current cycle given that it contained $n1$ cells at the beginning of the receive phase of the previous cycle.

$\pi_i(s)$: Steady state probability that a stage i SE contains exactly s cells at the beginning of a cycle.

$\tilde{\pi}_i(s)$: Steady state probability that a stage i SE contains exactly s cells at the beginning of the receive phase.

a_i : Probability that a cell is ready to enter a stage i buffer.

b_i : Probability that a successor of a stage i SE provides an acknowledgment to an output line given that a cell was submitted to the successor during the same cycle.

$Y_d(r, s)$: Probability that a switch that contains s cells, contains cells for exactly r distinct outputs.

$SUCC_i$: the successor SE of a stage i SE.

The objective is to calculate the steady state vector Π_i for every stage. The steady state vector Π_i is obtained by solving the matrix equation:

$$\Pi_i = \Pi_i \Lambda_i, \tag{1}$$

where Λ_i is the transition matrix of a stage i SE.

$$\Pi_i = [\pi_i(s)], s = 0, 1, \cdots, B$$

$$\Lambda_i = [\lambda_i(s1, s2)], s1 = 0, 1, \cdots, B; s2 = 0, 1, \cdots, B.$$

The state transition matrix elements are obtained by:

$$\lambda_i(s1, s2) = \sum_{s3=\max(0, s1-d)}^{s1} \tau_i(s1, s3)\sigma_i(s3, s2), \tag{2}$$

where d is the size of an SE.

$$\tau_i(s1, s3) = \sum_{r=s1-s3}^{\min(d, s1)} Y_d(r, s1)\binom{r}{s1 - s3} \times$$

$$b_i^{(s1-s3)}(1-b_i)^{r-(s1-s3)}. \quad (3)$$

$$b_i = \sum_{h=d}^{B} \tilde{\pi}_{i+1}(B-h) + \sum_{h=1}^{d-1} \tilde{\pi}_{i+1}(B-h) \times \left[\sum_{r=0}^{h-1} \binom{d-1}{r} {a_{i+1}}^r (1-a_{i+1})^{d-1-r} + \sum_{r=h}^{d-1} \binom{d-1}{r} {a_{i+1}}^r (1-a_{i+1})^{d-1-r} . \frac{h}{r+1} \right]. \quad (4)$$

Eq. (4) states that given a cell was submitted to a $SUCC_i$ through a particular output link, the link definitely receives an acknowledgment if $SUCC_i$ has greater than or equal to d buffer spaces by the end of the forward phase or the total number of submitted cells to that $SUCC_i$ is less than the number of available spaces by that time. Otherwise, receiving of an acknowledgment depends on the number of submitted cells, and the intermediate state of the $SUCC_i$. The value of b_i for $i = k, k = \log_d N$, is always equal to 1 due to the assumption that the output of the switch can always accept a cell.

$$\sigma_i(s3,s2) = \begin{cases} \sum_{w=s2-s3}^{d} \binom{d}{w} {a_i}^w (1-a_i)^{d-w} & , s2 = B \\ \binom{d}{s2-s3} {a_i}^{(s2-s3)} (1-a_i)^{d-(s2-s3)} & , s2 < B \end{cases} \quad (5)$$

$Y_d(r,s)$ is recursively calculated using the following equation [11]

$$Y_d(r,s) = \begin{cases} 1, & s = r = 0 \\ 0, & (s > 0 \wedge r = 0) \vee s < r \\ \frac{r}{d} Y_d(r, s-1) + & \\ \frac{d-(r-1)}{d} Y_d(r-1, s-1), & 0 < r \leq s \end{cases} \quad (6)$$

Eq. (6) is independent of the stages, thus a table of required values can be created once and used for the rest of the calculations.

$$a_i = \begin{cases} \rho & , i = 1 \\ \sum_{j=0}^{B} \pi_{i-1}(j) \left[1 - (1-1/d)^j \right] & , \text{otherwise} \end{cases} \quad (7)$$

where ρ is the offered load to the switch. The intermediate state vector is calculated in the same way as the initial state vector. In particular

$$\tilde{\Pi}_i = \tilde{\Pi}_i . \Theta_i .$$

where $\tilde{\Pi}_i = [\tilde{\pi}_i(s)], s = 0, 1, \cdots, B$ and $\Theta_i = [\theta_i(n1, n2)]$, $n1 = 0, 1, \cdots, B$, $n2 = 0, 1, \cdots, B$, is given by

$$\theta_i(n1,n2) = \sum_{k=0}^{B} \sigma_i(n1,k) \tau_i(k,n2). \quad (8)$$

To calculate Π_i and $\tilde{\Pi}_i$ we need Eqs. (2) to (7). But Eqs. (4) and (7) depend on the values of $\tilde{\Pi}_i$ and Π_i respectively. Thus obtaining the steady state values requires an iterative computation. Though we are considering the steady state condition of the switch, our experience shows that a fast convergence to the steady state values could be obtained by starting the calculation from the rest condition of the switch. In particular, at the beginning of the iterations all of the stages can accept cells, and none have any cell to submit to their successor stages. Thus the value of b_i for all of the stages is initially 1, and the value of a_i for every stage, except the first is 0. Having these values we can calculate σ_i and τ_i for every stage, and calculate $\tilde{\Pi}_i$ and Π_i thereafter. Then the new values of $\tilde{\Pi}_i$ and Π_i are used to calculate the new values for a_i and b_i. The iteration continues until the steady state conditions are reached. After finding the steady state values, our merits of measurements could be obtained. The throughput of the SE is given by

$$\sum_{s1=0}^{B} \sum_{s3=0}^{s1} (s1-s3) \tau_k(s1,s3) \pi_k(s1). \quad (9)$$

The average delay at a stage i SE is given by the Little's law. In particular, the delay per stage is obtained from

$$\frac{\sum_{s=0}^{B} s\pi_i(s)}{\sum_{s3=0}^{B} \sum_{s2=s3}^{B} (s2-s3) \sigma_i(s3,s2) \tilde{\pi}_i(s3).} \quad (10)$$

In Eq. (10), the numerator represents the equivalent average queue length in a stage i SE, and the denominator is the average arrival rate at a stage i SE. The total delay is obtained by summing the delays over all the stages.

3 Simulation Study

We validate the model presented in Section 2 with a simulation study. The same assumptions as made for the analysis apply to the simulation of the switch. Moreover, the following considerations are carried out too:

- At each cycle, a cell is generated with probability ρ (offered load to the switch input). The generated cell is independent of the cells generated at previous cycles and other input ports. Each cell consists of the following information
  1. a source tag which represents the address at which it is generated,
  2. a destination tag, the address to which the cell is destined,
  3. the current clock, used for measurement of the instantaneous delay.
- Simulation results from the first 500 cycles of the switch operation are ignored to allow the switch to reach a steady state condition. The simulation is then allowed to run until the change in the average throughput between the cycles becomes less than 10^{-6}.
- Conflict in the buffers for accessing a particular output as well as contention to seize a buffer space in the next stage is resolved using a random number generator with a different seed value from that of the cell generator.

The switch operates as follows:

1. The cells at the last stage's buffers are sent to the output links of the switch, and the instantaneous throughput and delay are measured for every link.
2. For each stage from stage $k-1$ to 0:

- The SE's buffer is examined for every output link of the SE, a copy of all cells destined to different outputs are placed in different lists, and the lists are sent to the corresponding input links of the next stage.
- If the number of available buffer spaces in an SE at $SUCC_i$ is less than the number of cells in the different lists at the inputs to that SE, a number of cells equal to the number of available spaces are chosen at random from the available lists.

3. A new set of cells is generated at stage 1 with probability ρ, and cells are placed in the first stage's buffer if there is any room. Unaccepted cells are discarded.

4 Numerical Results

Figs. 3 to 4 compare the results obtained from our model with the simulation, and with the results obtained from the model described in [11] for switch size $N = 256$. The analytical model is more accurate when the offered load to the switch is low, and when a bigger buffer size is used. However, the model is optimistic under high input load due to the fact that the model does not assume any correlation between the cells in the buffers, whereas the correlation becomes significant under high offered loads. As the results in Fig. 3 suggest, global flow control policy provides significantly higher throughput than local flow control, especially when B/d is small. This is because global flow control uses the buffer spaces more efficiently.

In Fig. 4, the throughput and delay versus switch size for $B/d = 1$ and $B/d = 3$ for $\rho = 1$ are given. According to the results, under full input load, the delay per stage decreases as the buffer size increases, since the total number of buffer spaces increases inside the switch. However, for a particular switch size and B/d, the delay per stage grows when the SE size increases.

5 Conclusion

In this paper, we have developed a model for the analysis of multistage switch based on shared buffer switching elements for ATM networks. Our model employs global flow control and acknowledgment mechanism in contrast to local flow control described in [11]. In local flow control, a cell can be forwarded to the next stage depending on the state of the corresponding switch at the beginning of a cycle, whereas in global flow control, the simultaneous operation of forwarding and receiving the cells is allowed. This results in better buffer utilization, and higher performance. The new model is quite accurate under low to medium input load probability, however, due to the output correlation of cells in a switch, its inaccuracy grows when the input load probability approaches 1. An extension to this work could be to consider this correlation which leads to more accurate results. Other extensions could include the analysis of shared buffer switches under bursty traffic and other nonuniform traffic patterns.

References

[1] J.H. Patel, "Performance of processor-memory interconnections for multiprocessors," *IEEE Transactions on Computers*, vol. C-30, no. 10, pp. 771–780, October 1981.

[2] M. Atiquzzaman and M.S. Akhtar, "Effect of nonuniform traffic on the performance of unbuffered multistage interconnection networks," *IEE Proceedings - Computer and Digital Techniques*, vol. 141, no. 3, pp. 169–176, May 1994.

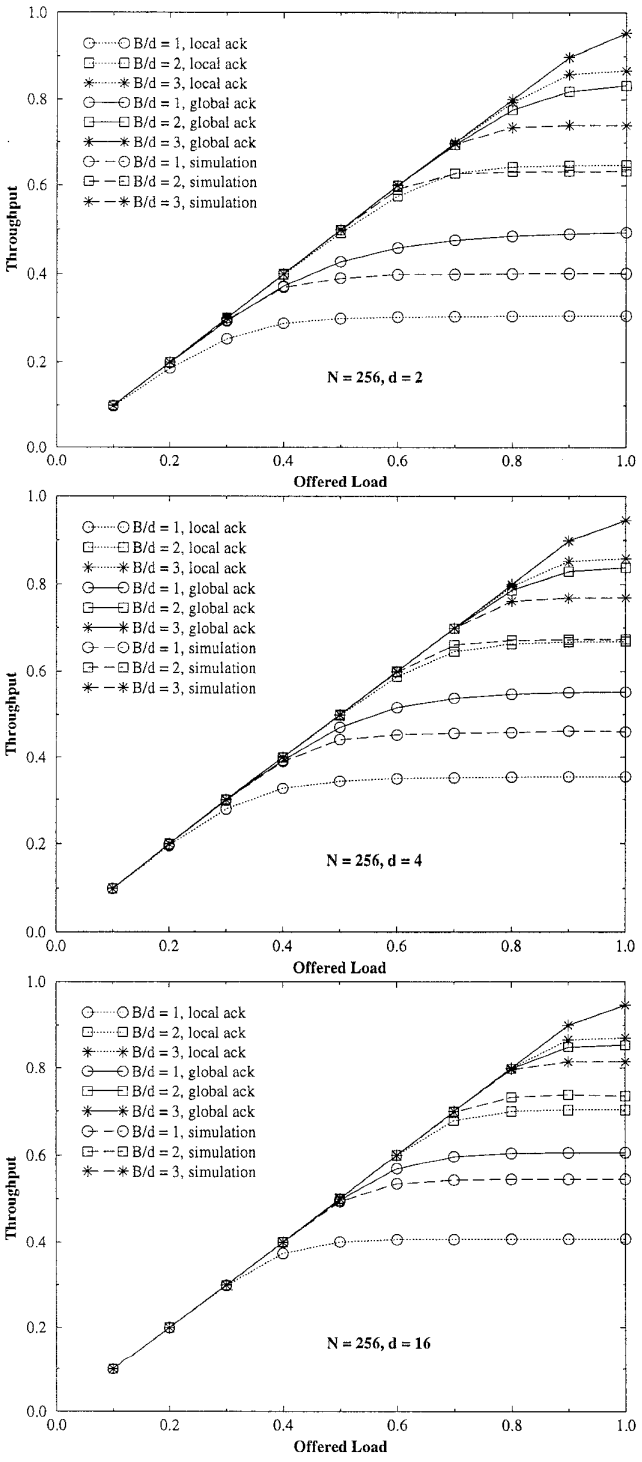

Figure 3: Throughput versus offered load (ρ) for $N = 256$.

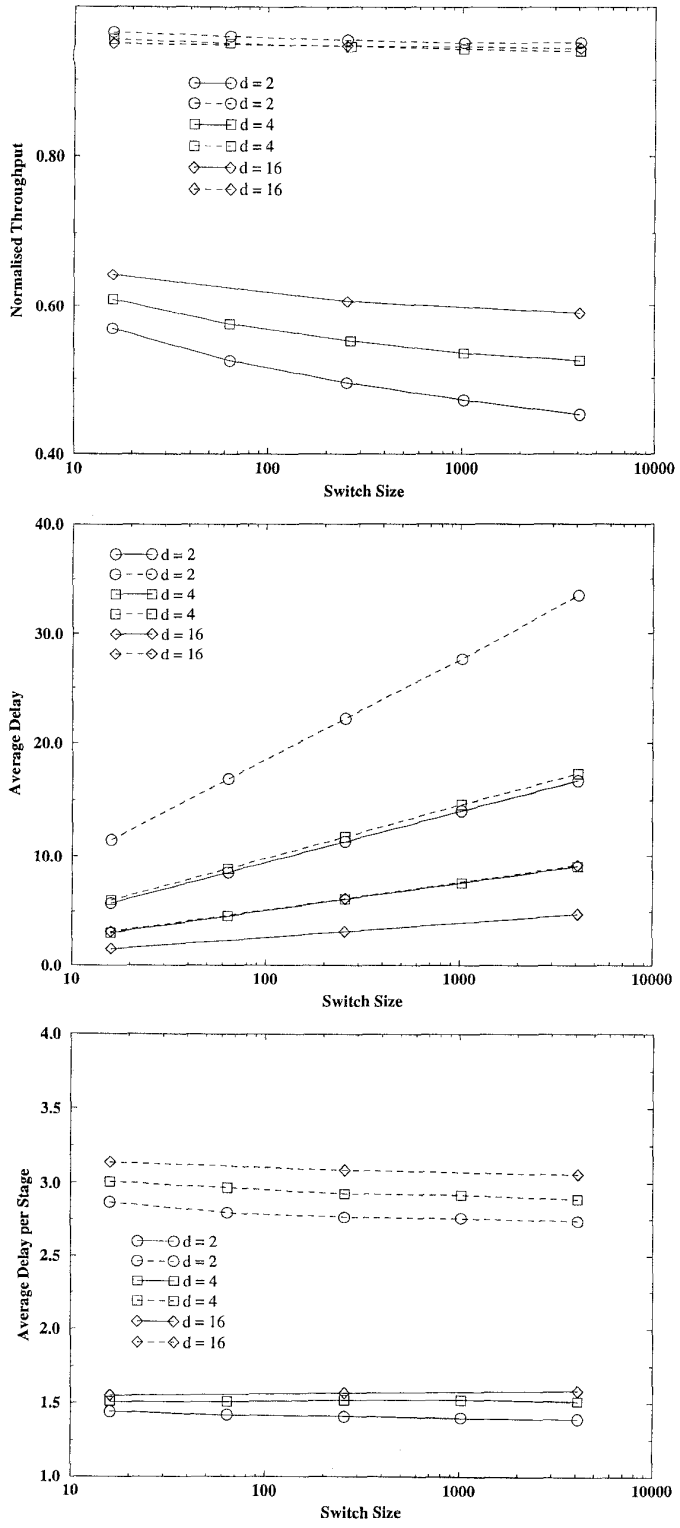

Figure 4: Throughput and delay vs switch size (N) for different values of B/d, and $\rho = 1$. Solid curves represent $B/d = 1$ and dashed curves represent $B/d = 3$.

[3] Y-C. Jenq, "Performance analysis of a packet switch based on single-buffered Banyan network," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 6, pp. 1014–1021, December 1983.

[4] H. Yoon, K.Y. Lee, and M.T. Liu, "Performance analysis of mulitbuffered packet-switching networks in multiprocessor systems," *IEEE Transactions on Computers*, vol. 39, no. 3, pp. 319–327, March 1990.

[5] T.H. Theimer, E.P. Rathgeb, and M.N. Huber, "Performance analysis of buffered Banyan networks," *IEEE Transactions on Communications*, vol. 39, no. 2, pp. 269–277, February 1991.

[6] S.H. Hsiao and R.Y. Chen, "Performance analysis of single-buffered multistage interconnection networks," *Third IEEE Symposium on Parallel and Distributed Processing*, pp. 864–867, December 1-5, 1991.

[7] M. Atiquzzaman and M.S. Akhtar, "Performance of buffered multistage interconnection networks in non uniform traffic environment," *Seventh International Parallel Processing Symposium*, California, pp. 762–767, April 13-16, 1993.

[8] T. Szymanski and S. Shaikh, "Markov chain analysis of packet-switched Banyans with arbitrary switch sizes, queue sizes, link multiplicities and speedups," *IEEE INFOCOM '89: 8th Annual Joint Conference of the IEEE Computer and Communication Societies*, Ontario, Canada, pp. 960–971, April 25-27, 1989.

[9] B. Zhou and M. Atiquzzaman, "Improved performance model of multibuffered multistage interconnection network under general traffic patterns," *IEEE INFOCOM '94: Conference on Computer Communications*, Toronto, Canada, pp. 1448–1455, June 12-16, 1994.

[10] T. Lin and L. Kleinrock, "Performance analysis of finite-buffered multistage interconnection networks with a general traffic pattern," *1991 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, San Diego, CA, pp. 68–78, May 21-24, 1991.

[11] J.S. Turner, "Queueing analysis of buffered switching networks," *International Teletraffic Congress*, Copenhagen, Denmark, pp. 35–40, June 1991.

[12] A. Monterosso and A. Pattavina, "Performance analysis of multistage interconnection networks with shared-buffered switching elements for ATM switching," *IEEE INFOCOM '92: Conference on Computer Communications*, Florence, Italy, pp. 124–131, May 4-8, 1992.

[13] G. Bianchi and J.S. Turner, "Improved queueing analysis of shared buffer switching networks," *IEEE INFOCOM '93: Conference on Computer Communications*, San Franciso, California, pp. 1392–1399, March 30 - April 1, 1993.