A Performance Model for ATM Switches with Multiple Input Queues

Ge Nong, Jogesh K. Muppala and Mounir Hamdi
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract

An analytical model for the performance analysis of a novel input access scheme for an ATM switch is developed and presented in this paper. The interconnection network of the ATM switch is internally nonblocking and each input port maintains a separate queue for each output port so as to reduce the head-of-line (HOL) blocking of conventional input queuing switches. Each input is allowed to send only one cell per time slot, and each output port is allowed to receive only one cell per time slot. Using a tagged queue *approach, an analytical model with an underlying two-dimensional Markov chain with a state space of size (Queue Capacity × switch size) is constructed for evaluating the switch performance under i.i.d Bernoulli traffic for different offered traffic loads. The switch throughput, mean cell delay, and cell loss probability are computed from the analytical model. The accuracy of the analytical model is verified using simulation.*

1 Introduction

Input-queueing schemes have received a lot of attention in implementing high bandwidth ATM switches because of their simplicity in implementation. However they suffer from the well-known head of line (HOL) blocking problem which limits the switch throughput to a maximum of 58.6% [1]. One approach to overcome this problem is for each input port to maintain a separate queue of cells destined for each output port [2, 3, 4, 5] (see Figure 1). During a single time slot, a maximum of one cell per input port can be transferred, and a maximum of one cell per output port can be received. The selection of the cell from an input to transmit during a time slot to an output is accomplished using a scheduling algorithm. Of particular interest to us in this paper is an iterative matching algorithm named *parallel iterative matching* (PIM) [2] which finds the maximal matching between the inputs and outputs of the switch. This switch architecture has received a lot of attention from the research community, and commercial switches based on this queueing technique such as the DEC Systems AN2 switch [2] have already been built.

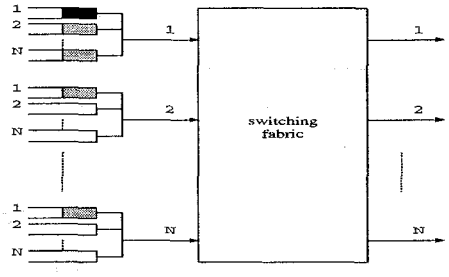


Figure 1: Architecture of the multiple input queues ATM switch ($N \times N$).

Performance evaluation of the PIM switches appearing in the literature so far have been based on simulation. A cell loss probability requirement of 10^{-9} is not uncommon for an ATM switch, and estimating the rare cell loss probability by simulation is inefficient and sometimes impossible [6]. Consequently, analytical models are of great importance in solving these problems.

In this paper we develop an analytical model of the PIM switch with finite input buffers using the *tagged queue* approach [7, 9] and verify the accuracy of the results using simulation. The remainder of this paper is organized as follows. Section 2 introduces the PIM scheduling algorithm. Section 3 develops the analytical model of PIM switches under i.i.d Bernoulli traffic based on the tagged queueing approach. Numerical results obtained from the analytical model are presented in Section 4 and compared with the results from simulation. Finally Section 5 gives the conclusions.

2 Parallel Iterative Matching

The PIM algorithm proposed by Anderson *et al.* [2] uses parallelism, randomness, and iteration to find

a maximal matching between the inputs that have queued cells for transmission and the outputs that have queued cells (at the inputs) destined for them. Interested readers may refer to [2, 3, 4] for details.

To facilitate the analysis of the PIM algorithm in the rest of the paper, we modify the original PIM algorithm as follows. The modified PIM algorithm iterates the following two steps until a maximal matching is found or until a fixed number of iterations are performed:

1. Each unmatched input chooses an output *uniformly* over all unmatched outputs for which it has queued cells and sends a request to it.
2. If an unmatched output receives any requests, it chooses one *uniformly* over all requests to grant and notifies each requesting input.

The two algorithms are **logically equivalent** for the purpose of analysis.

3 Queueing Model and Analysis of Multiple Iterations PIM

The concept of *tagged input queue* has been successfully used to evaluate the FIFO input-queued switch model [7, 9]. As observed from the algorithm description of PIM, a HOL cell in an input queue will contend for transmission not only with the HOL cells of the same input, but also the HOL cells destined for the same output. This two stages contention process of PIM complicates our model when compared to the models in [7, 9]. The PIM switch model is developed under the following assumptions:

1. The switch operates synchronously.
2. Every input queue has the same buffer size, b_i.
3. Cells arrive at the inputs according to an i.i.d Bernoulli process with parameter $N\lambda$ and the cells' destinations are uniformly distributed over all the outputs.
4. New cells arrive only at the beginning of the time slots, and cells depart only at the end of the time slots.

Under the above assumptions, all the input queues will exhibit the same behavior when the system attains steady state. Let $Q(i,j)$ denote the queue at input i which contains cells with output j as the destination. Figure 2 shows an example of the queueing model for the PIM switch where the occupancy of $Q(1,1)$ is taken as the *tagged input queue*, and the number of HOL cells at input 1 is represented by the *1st HOL input queue*.

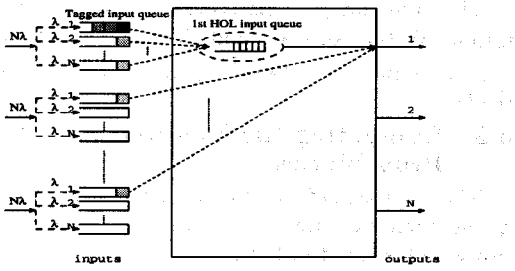


Figure 2: An example of the queueing model for the PIM switch.

3.1 Markov Model

The queueing model is analyzed by considering the underlying Markov chain Z whose states are expressed as a couplet (l, w_i), where l, w_i, are the lengths of the *tagged input queue* and *virtual HOL input queue*, respectively, at the end of a time slot. The states space of this two-dimensional Markov chain is $\{(0,0),(l,w_i)\}$, $(1 \le l \le b_i, 1 \le w_i \le N)$ and are ordered in a lexicographic order, that is, (0,0), (1,1), (1,2),...(2,1),... (b_i, N). This Markov chain is a *Quasi Birth and Death (QBD)* process having a block-partitioned form of transition probability matrix:

$$T = \begin{bmatrix} A_1' & A_2' & 0 & \cdots & & & \\ A_0' & A_1 & A_2 & 0 & \cdots & & \\ 0 & A_0 & A_1 & A_2 & 0 & \cdots & \\ \vdots & \vdots & \vdots & \cdots & & & \vdots \\ 0 & 0 & \cdots & 0 & A_0 & A_1 & A_2 \\ 0 & 0 & \cdots & 0 & 0 & S & B \end{bmatrix}$$

where $A_1' + A_2'e = 1$ and $A_0' + A_1e + A_2e = (A_0 + A_1 + A_2)e = e$ with $e = [1,1,1,\ldots,1]^T$. Let $P_{blo,W_t(w_i')|W_{t-1}(w_i)}/P_{suc,W_t(w_i')|W_{t-1}(w_i)}$ denote the probability that the HOL cell of a tagged queue is blocked/transmitted given that the remaining HOL cells of the last time slot is w_i and the remaining HOL cells at the end of the current time slot is w_i'. Define B as a $N \times N$ matrix whose element at position (r,c) is given by $P_{blo,W_t(c)|W_{t-1}(r)}$. Similarly, B_0 is defined as an $1 \times N$ matrix whose elements at position $(1,c)$ is $P_{blo,W_t(c)|W_{t-1}(0)}$. Also define S as a $N \times N$ matrix whose element at position (r,c) is given by $P_{suc,W_t(c)|W_{t-1}(r)}$. Similarly, S_0 is the probability that the HOL cell of the *tagged input queue* gets matched given that the *tagged input queue* is empty at the end of the last time slot. Here $Se + Be = e$ and $S_0 + B_0e = 1$. Then, $A_0' = (1-\lambda)Se$, $A_1' = (1-\lambda) + \lambda S_0$, $A_2' = \lambda B_0$, $A_0 = (1-\lambda)S$, $A_1 = \lambda S + (1-\lambda)B$, and $A_2 = \lambda B$.

The remaining subsections show the computation of the success and blocking probabilities, $P_{suc,W_t(w_i')|W_{t-1}(w_i)}$ and $P_{blo,W_t(w_i')|W_{t-1}(w_i)}$, respectively.

3.2 Computing the Blocking and Success Probabilities

The transition of the state of the virtual HOL input queue from the state w_i to state w_i' is a two-steps process illustrated in Figure 3.

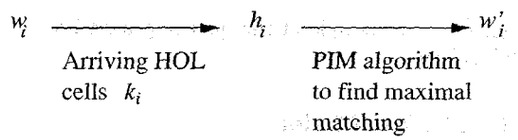


Figure 3: Transition of the virtual HOL input queue.

3.2.1 Arriving Cells at the Virtual HOL Input Queue

Let $K_t(k_i)$ denote the number of newly arriving HOL cells at the *virtual HOL input queue* (k_i new arrivals to the *virtual HOL input queue*), at the beginning of the current time slot t. $W_{t-1}(w_i)$ denotes the numbers of remaining HOL cells at the *virtual HOL input queue* (w_i is length of *virtual HOL input queue*), at the end of the previous time slot $t-1$. Let $H_t(h_i) = K_t(k_i) + W_{t-1}(w_i)$. Define, $a_{K(k_i)|W(w_i)} = Prob(K_t(k_i)|W_{t-1}(w_i))$. A cell that arrives at $Q(i,j)$ when $Q(i,j)$ is empty, will observe that another queue is non-empty with probability p_1 and let $p_0 = 1 - p_1$. Hence,

$$a_{K(k_i)|W(w_i)} = \begin{cases} \binom{N-1}{k_i-1} p_1^{k_i-1} p_0^{(N-k_i)} \\ 1 \le k_i \le N,\ w_i = 0 \\ \\ \binom{N-w_i}{k_i} \lambda^{k_i} (1-\lambda)^{N-(k_i+w_i)} \\ 0 \le k_i \le N - w_i,\ 1 \le w_i \le N. \end{cases}$$

3.2.2 Transition to $W_t(w_i')$

We define the following probabilities associated with the transitions:

$P_{blo\_0|H(h_i)} := Prob\{$*the HOL cell at the tagged input queue gets blocked, and* $W_t(w_i') = H_t(h_i)$ *given that* $H_t(h_i)\}$

$P_{blo\_1|H(h_i)} := Prob\{$*the HOL cell at the tagged input queue gets blocked, and* $W_t(w_i') = H_t(h_i - 1)$ *given that* $H_t(h_i)\}$

$P_{suc|H(h_i)} := Prob\{$*the HOL cell at the tagged input queue gets transmitted, and* $W_t(w_i') = H_t(h_i - 1)$ *given that* $H_t(h_i)\}$

Given $r_i = w_i' - w_i$, the blocking probability $P_{blo,W_t(w_i')|W_{t-1}(w_i)}$ is computed as:

$$\begin{cases} 0, \ for\ r_i < -1 \\ \\ a_{K(r_i)|W(w_i)} P_{blo\_0|H(w_i')} \\ +a_{K(r_i)|W(w_i)} P_{blo\_1|H(w_i')} (1 - \frac{l_0(r_i-1)}{r_i-1}) \\ +a_{K(r_i+1)|W(w_i)} P_{blo\_1|H(w_i')} \frac{l_0(r_i)}{r_i}, \\ for\ r_i \ge 1\ \ and\ \ w_i = 0 \\ \\ a_{K(r_i)|W(w_i)} P_{blo\_0|H(w_i')} \\ +a_{K(r_i)|W(w_i)} P_{blo\_1|H(w_i')} (1 - \frac{r_i + l_0(w_i-1)}{r_i+w_i-1}) \\ +a_{K(r_i+1)|W(w_i)} P_{blo\_1|H(w_i')} \frac{r_i+1+l_0(w_i-1)}{r_i+w_i}, \\ for\ r_i \ge -1\ \ and\ \ w_i > 0 \end{cases} \quad (1)$$

in which

$$l_0(w) = \begin{cases} 0, & for\ w = 0 \\ \sum_{u=1}^{w} \binom{w}{u} P_{l1}^u (1-P_{l1})^{w-u}, & for\ w > 0 \end{cases} \quad (2)$$

and P_{l1} in Eq (2) is the probability that an input queue length is equal to 1 (there is only one buffered cell in this input queue) during a time slot, and is given by:

$$P_{l1} = (1-\lambda)\pi_1 e/(1-\pi_0)$$

For $W_{t-1}(w_i) = 0$, the blocking probability $P_{blo,W_t(w_i')|W_{t-1}(0)}$ can be computed by Eq (1) provided that the function P_{l1} in Eq (2) is replaced by P_{l1}'.

$$P_{l1}' = (\lambda\pi_0 + (1-\lambda)\pi_1 e)/p_1$$

The probability $P_{suc,W_t(w_i')|W_{t-1}(w_i)}$ can be computed as

$$P_{suc,W_t(w_i')|W_{t-1}(w_i)} = a_{K(r_i)|W(w_i)} P_{suc|H(r_i+w_i)}$$

3.2.3 Applying the PIM Algorithm

The transition probabilities $P_{blo\_0|H(h_i)}$, $P_{blo\_1|H(h_i)}$ and $P_{suc|H(h_i)}$ defined above is computed by considering the transition probabilities for each iteration of the PIM scheduling algorithm. For the ϕ th iteration, following parameters are defined:

$n(\phi)$=the number of unmatched inputs/outputs at the beginning of ϕth iteration
$h_i(\phi)$=the number of non-empty queues in input i at the beginning of ϕth iteration, whose outputs are still unmatched
$m(\phi)$=$n(\phi) - n(\phi+1)$
$\Delta h_i(\phi)$=$h_i(\phi) - h_i(\phi+1)$

Given $(n(\phi), h_i(\phi), h_o(\phi))$ and $(n(\phi+1), h_i(\phi+1), h_o(\phi+1))$, we derive the following blocking probabilities $P_{blo\_x'_i x'_j|0x_j}$ and successful probability $P_{suc|00}$. Here the subscript *blo* means that at the end of current iteration, the HOL cell at the ***tagged input queue*** $Q(i,j)$ gets blocked; and $0x_j/x'_i x'_j$ represents whether input i and output j remain unmatched (represented by 0) or get matched (represented by 1) at the beginning/end of the current iteration, $x_j, x'_i, x'_j \in \{0,1,2\}$. For convenience, the following two functions of p_0 are defined:

$$P_{suc1} = \frac{1-p_0^n}{n}$$

$$P_{suc2} = \sum_{v=1}^{n-1} \binom{n-1}{v} p_1^v p_0^{(n-1-v)}/(v+1)$$

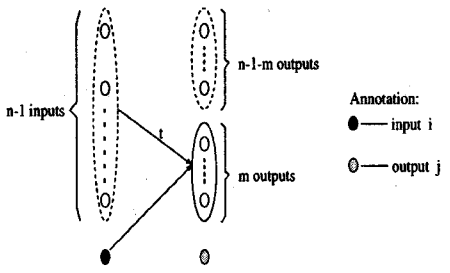


Figure 4: Diagram for derivation of $P_{blo\_00|00}$

Let t $(m \le t \le n-1)$ be the number of queues excluding the queue from input i that succeed in the first stage of contention, and m is the number of outputs contended by that t inputs. Figure 4 shows the situation where output j remains unmatched at the end of the current iteration. There are two sub-problems to be considered in computing the transition probabilities:

1. What is the probability that t inputs contend for m outputs?
2. What is the probability that Δh_i out of h_i outputs whose corresponding queues in input i are non-empty get matched?

The derivation below considers each of the sub-problems in computing the transition probabilities.

Computing $P_{blo\_00|00}$: From Figure 4, given t and m, the probability that input i gets blocked is

$$P_{t\to m} = (1-\frac{m}{t+1})((m-1)!S_t^{m-1}+m!S_t^m)P_{suc1}^t p_0^{n(n-1-t)}$$

Among these m outputs, Δh_i of them will see their corresponding queues in input i being non-empty. The number of combinations satisfying this condition is

$$C_{\Delta h_i|blo\_00\_00} = \binom{h_i-2}{\Delta h_i-1}\binom{n-h_i}{m-\Delta h_i}$$

Knowing the above probabilities, $P_{blo\_00|00}$ can be easily computed as

$$P_{blo\_00|00} = (1-\frac{1}{h_i})\sum_{t=m}^{n-1}\binom{n-1}{t}C_{\Delta h_i|blo\_00\_00}P_{t\to m}$$

Computing $P_{blo\_12|00}$: In this case, input i is matched. The probability is given by

$$P_{blo\_12|00} = (1-\frac{1}{h_i})P_{suc2}$$

Computing $P_{blo\_01|00}$: In this case, output j gets matched while input i remains unmatched. Only the aggregated probability over the set of possible Δh_i is counted. There are two candidate transition paths depending on whether $Q(i,j)$ survives the first stage of contention or not. Therefore $P_{blo\_01|00}$ is a sum of two probabilities:

$$P_{blo\_01|00} = P_{blo\_01\_B|00} + P_{blo\_01\_S|00}$$

where $P_{blo\_01\_B|00}$ and $P_{bloc\_01\_S|00}$ are the probabilities for the two candidate transition paths, respectively.

$$P_{blo\_01\_B\_00} = (1-\frac{1}{h_i})\sum_{t=m}^{n-1} C_{\Delta h_i|blo\_01\_B\_00}P_{t\to m}$$

$$P_{blo\_01\_S\_00} = \frac{1}{h_i}\sum_{t=m}^{n-1} C_{\Delta h_i|blo\_01\_S\_00}P_{t\to m}$$

where,

$$C_{\Delta h_i|blo\_01\_B\_00} = \binom{h_i-2}{\Delta h_i-2}\binom{n-h_i}{m-\Delta h_i}$$

$$C_{\Delta h_i|blo\_01\_S\_00} = \binom{h_i-1}{\Delta h_i-1}\binom{n-h_i}{m-\Delta h_i}$$

If output j has been matched, it should be relatively easy to find:

Computing $P_{blo\_01|01}$:

$$P_{blo\_01|01} = \sum_{t=m}^{n-1}\binom{n-1}{t}C_{\Delta h_i|blo\_01\_01}P_{t\to m}$$

$$C_{\Delta h_i|blo\_01\_01} = \binom{h_i - 1}{\Delta h_i - 1}\binom{n - h_i}{m - \Delta h_i}$$

Computing $P_{blo\_12|01}$:

$$P_{blo\_12|01} = P_{suc2}$$

Computing $P_{suc|00}$: Finally, $P_{suc|00}$ is given as,

$$P_{suc|00} = 1 - (P_{blo\_00|00} + P_{blo\_01|00} + P_{blo\_12|00})$$

The state transition path can be expressed as a weighted tree of depth d, where d is the maximum number of iterations and the weight is the probabilities, the states $\{blo\_00,\ blo\_01,\ blo\_12,\ suc\}$ are the leaves and the iteration numbers are the levels. To search the entire tree according to desired states' leaves and compute the weight of paths, we get the blocking probabilities $P_{blo\_0|H(h_i)} = P_{blo\_00|H(h_i)} + P_{blo\_01|H(h_i)}$ and $P_{blo\_1|H(h_i)} = P_{blo\_12|H(h_i)}$ as well as the successful probability $P_{suc|H(h_i)}$.

3.3 Solving the Markov Chain

Using *Matrix-Geometric* solution method [8] for the Markov chain the steady state probabilities are given by:

$$\pi_0 = 1/(1 + \alpha \sum_{l=1}^{b_i - 1} \beta^{l-1} e + \alpha\beta^{b_i - 2}\lambda B(I - B)^{-1} e)$$

$$\pi_l = \begin{cases} \pi_0 \alpha \beta^{l-1} & , for\ 0 < l < b_i \\ \pi_0 \alpha \beta^{b_i - 2} \lambda B(I - B)^{-1} & , for\ l = b_i \end{cases}$$

where $e = [1, 1, 1, \ldots, 1]^T$, I is the identity matrix, $I_1 = ee_1$, $e1 = [1, 0, 0, \ldots, 0]$, $\alpha = \lambda B_0(I - \lambda I_1 - (1 - \lambda)B)^{-1}$, and $\beta = \lambda B((1 - \lambda)(I - B))^{-1}$. Notice that for steady state, the following equation should hold

$$p_0 = (1 - \lambda)\pi_0$$

This naturally suggests an iterative solution [9].

3.4 Computing the Performance Metrics

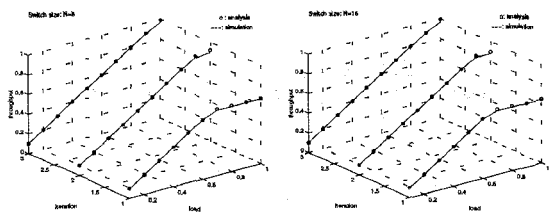
Let ρ, $\bar{Q}$, $\bar{D}$ and P_{loss} be throughput, mean queue length, mean cell delay and mean cell loss probability respectively, then

$$\rho = \lambda\pi_0(1 - B_0 e) + \sum_{l=1}^{b_i} \sum_{u=1}^{N} \pi_{(l,u)} P_{suc|W(u)}$$

$$\bar{Q} = \sum_{l=1}^{b_i} l\pi_l e, \qquad \bar{D} = \bar{Q}/\rho, \qquad P_{loss} = \lambda\pi_{b_i} e$$

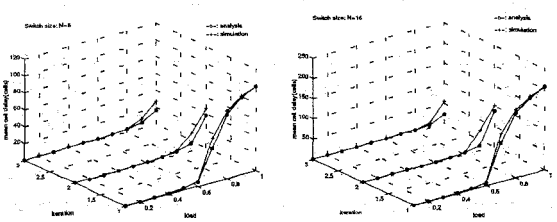
4 Numerical Results

Figure 5 shows the switch throughput as function of offered load λ for PIM switch sizes 8 and 16 with for different number of PIM iterations respectively. Under high offered load (greater than 60% when maximum iteration is 1), the throughput of the switch decreases when the switch size increases. As more HOL cells get matched during each iteration, the saturation throughput will increase as the number of iterations increases. The figures show that three iterations are sufficient to get a high throughput (> 90%).



(a) 8 × 8 switch (b) 16 × 16 switch

Figure 5: The throughput of the PIM switch as a function of offered load with a buffer size b_i=10.



(a) 8 × 8 switch (b) 16 × 16 switch

Figure 6: The mean cell delay of the PIM switch as a function of offered load with a buffer size b_i=10.

Figure 6 shows the mean cell delay as a function of offered load λ for the switch sizes 8 and 16 for different number of PIM iterations. The figures indicate that the mean cell delay increases as the switch size increases and also as the offered load increases. But when the number of PIM scheduling iterations is increased, even from 1 to 2, the mean delay increased slowly with the traffic load as compared with

the case of just one iteration. For single iteration PIM scheduling, the mean cell delay increases dramatically when the offered load exceeds 60%, which indicates that PIM switches with single iteration PIM scheduling will be overloaded when the traffic load is greater than 60%. However, for 2 and 3 iteration PIM, this *overloaded traffic point* is about 0.8. This phenomenon can also be observed in figure 5. Notice that when the traffic load is extremely low, such as 0.1, all curves coincide. Under low traffic load, the case for more than one HOL cell contending for a common input/output is very low such that single iteration PIM scheduling is typically sufficient to find a maximal matching. However, when the traffic load increases, the chances of conflicts increase and more iterations are needed for using PIM to attain maximal matching.

From the figures for throughput and mean cell delay, it can be seen that our queueing model gives somewhat optimistic results than simulation under high offered load.

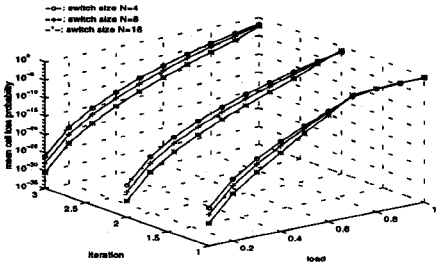


Figure 7: The mean cell loss probability of a PIM switch, as a function of offered load, with a buffer size b_i=10.

In figure 7, the mean cell loss probabilities of PIM switches with queue size of 10 cells are given as a function of offered load. It can be seen that, for a medium size PIM switch with 3 iterations PIM scheduling (such as 16-by-16) with traffic load less than 60%, a buffer size of 10 cells per queue is sufficient to guarantee a cell loss probability $< 10^{-9}$.

5 Conclusions

The proposed queueing model provides a general method for analyzing non-blocking ATM switches with multiple input queues in terms of throughput, mean queue length, mean cell delay, and mean cell loss probability given the switch size, queue buffer size, and offered load (i.i.d Bernoulli arrivals). The results from the analytical model match closely with the simulation results. This model will be extended in the future for more realistic traffic patterns such as bursty and correlated traffic. The analysis procedure developed in this paper is considered as the foundation for solving the more difficult problem involving bursty and correlated traffic [10].

References

[1] A. Pattavina and G. Bruzzi, Analysis of Input and Output Queueing for Nonblocking ATM Switches, *IEEE/ACM Trans. on Networking*, vol. 1, No. 3, June 1993, pp. 314-328.

[2] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed Switch Scheduling for Local-area Networks," *ACM Transactions on Computer Systems*, Vol. 11, No. 4, Nov. 1993, pp. 319-352.

[3] N. W. McKeown, *Scheduling Algorithms for Input-Queued Cell Switches*, Ph.D. thesis, University of California at Berkeley, 1994.

[4] B. Prabhakar, N. Mckeown, and R. Ahuja, "Multicast Scheduling for Input-Queued Switches," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 5, June 1997, pp. 855-866.

[5] N. McKeown, M. Izzard, A. Mekkittikul, W. Ellersick and M. Horowitz, "Tiny Tera: a packet switch core," *IEEE Micro*, Vol.17, No.1, Jan.-Feb. 1997, pp. 26-33.

[6] M. J. Lee and D. S. Ahn, "Cell Loss Analysis and Design Trade-Offs of Nonblocking ATM Switches," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 2, April 1995, pp. 199-210.

[7] P. Achille and B. Giacomo, "Analysis of Input and Output Queueing for nonblocking ATM Switches," *ACM Trans. on Networking*, vol. 1, No. 3, June 1993.

[8] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, 1981.

[9] Y.C. Jung and C. K. Un, "Performance Analysis of Packet Switches with Input and Output Buffers," *Computer Networks and ISDN Systems*, Vol. 26, 1994, pp. 1559-1580.

[10] X. R. Cao and D. Towsley, "A Performance Model for ATM switches with General Packet Length Distributions," *IEEE/ACM Trans. on Networking*, Vol. 3, No. 3, June 1995, pp. 299-309.