Performance analysis of input queueing ATM switches with parallel iterative matching scheduling

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.
Email: {nong,muppala,hamdi}@cs.ust.hk

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 is provided with N input queues per each input port for a switch of size $N \times N$. That is, 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 just one cell per slot time, and each output port is allowed to accept just one cell per slot time. Under saturated conditions the switch was analyzed and a closed-form solution for the maximum throughput is derived. Using a ***tagged input queue*** approach, an analytical model for evaluating the switch performance under an *i.i.d.* Bernoulli traffic for different offered traffic loads is developed. 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.

Keywords

ATM switch, analytical modeling, performance evaluation, computer simulation

1 INTRODUCTION

Input queueing is preferred in implementing switching architectures for ATM (Awdeh *et al.* 1995) because of its simplicity. However, they suffer from the ***head-of-line*** (HOL) blocking problem which limits the throughput of each input port to a maximum of 58.6% under uniform traffic, and much lower for bursty traffic (Pattavina *et al.* 1993). Several approaches have been proposed

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35353-1_28

D. Kouvatsos (ed.), *Performance Analysis of ATM Networks*

to overcome this problem: adopting a switch expansion, a windowing technique, or a channel grouping technique (Awdeh *et al.* 1995). Of particular interest to us in this paper is a recent technique termed ***parallel iterative matching*** (PIM) algorithm and its variants (Anderson *et al.* 1993, McKeown 1994) which 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. Each input queue of the switch contains a random access buffer consisting of N FIFO queues, each of which stores the cells destined for one of the N output ports. The first cell in each queue can be selected for transmission across the switch in each time slot, with the following constraints: (i)Only one cell from any of the N queues in an input port can be transmitted in each time slot. (ii)At most one cell can be transmitted from the N input ports to an output port of the switch in any given time slot.

To facilitate mathematical analysis, we modify the original PIM algorithm into a *logically equivalent* algorithm. 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 the requests and notifies each requesting input.

The remainder of this paper is organized as follows. Section 2 presents recursive equations for the maximum throughput of the switch. Section 3 develops an analytical model based on the tagged queuing approach. Equations for computing interesting performance measures including throughput, mean cell delay, and mean cell loss probability are derived in this section. Numerical results obtained from the analytical model are presented for switches of different sizes in Section 4, and compared with the results from simulation. Finally conclusions are presented in Section 5.

2 MAX THROUGHPUT OF MULTIPLE ITERATIONS PIM

Under saturated conditions, all the queues at each input will have at least one cell so that each output will have requests from every unmatched input. An output selects one uniformly among the input requests. The throughput of the ATM switch with 1 iteration PIM scheduling, $\rho(1)$, is equal to the probability that an output O_j gets matched after the first iteration. The probability of an input request being accepted by an output, $p = 1/N$. Then,

$$\rho(1) = 1 - (1 - \frac{1}{N})^N, \qquad \lim_{N \to \infty} \rho(1) = 1 - e^{-1} = 0.632. \tag{1}$$

Let $Pr\{m(1)\}$ and $Pr\{n(1)\}$ respectively be the probabilities that $m(1)$ inputs (outputs) get matched or remain unmatched and output O_j remains unmatched after the first iteration. Then, $Pr\{n(1)\} = Pr\{m(1) = N - n(1)\}$ and $Pr\{m(1)\} = \binom{N-1}{m(1)} m(1)! \mathcal{S}_N^{(m(1))}/N^N$, where $\mathcal{S}_n^{(m)}$ is the *stirling number* of the second kind which gives the number of ways of partitioning a set of n elements into m non-empty subsets (Abramowitz *et al.* 1972): $\mathcal{S}_n^{(m)} = \frac{1}{m!}\sum_{k=0}^{m}(-1)^{m-k}\left(\begin{array}{c} m \\ k \end{array}\right)k^n$.

The throughput of two iterations PIM scheduling is equal to the sum of $\rho(1)$ and the probability that output O_j gets matched in the second iteration, that is

$$\begin{aligned}
\rho(2) &= \rho(1) + Pr\{O_j\ gets\ matched\ in\ the\ second\ iteration\ | \\
&\qquad\qquad O_j\ wasn't\ matched\ in\ the\ first\ iteration\} \\
&= \rho(1) + \sum_{n(1)=1}^{N-1}(1-(1-\frac{1}{n(1)})^{n(1)})Pr\{n(1)\} \\
Pr\{n(i)\} &= \sum_{n(i-1)=n(i)+1}^{N-(i-1)} Pr\{m(i) = n(i-1) - n(i)\}Pr\{n(i-1)\} \qquad (2)
\end{aligned}$$

where $Pr\{m(i)\} = \binom{n(i-1)-1}{m(i)}\frac{m(i)!\mathcal{S}_{n(i-1)}^{(m(i))}}{(n(i-1))^{n(i-1)}}$. Using Eq (2), the throughput of i iterations PIM scheduling $\rho(i)$ is

$$\rho(i) = \rho(i-1) + \sum_{n(i-1)=1}^{N-(i-1)}(1-(1-\frac{1}{n(i-1)})^{n(i-1)})Pr\{n(i-1)\}$$

Figure 1 shows the results for maximum throughput as function of switch size and number of iterations. As shown in this figure, the maximum throughput of a ATM switch with 1 iteration PIM scheduling converges to 0.63 (which corresponds to Eq (1)) when the switch size grows. Furthermore, the throughput increases significantly after each iteration of PIM scheduling. Four iterations are sufficient for achieving maximum throughput of about 99% for a switch of any size.

3 QUEUEING MODEL AND ANALYSIS OF MULTIPLE ITERATIONS PIM

In this section, we model the ATM switch with PIM scheduling using queueing theory and analyze the underlying Markov chain. Our method uses the concept of *tagged queues* in modeling the PIM switch leading to a smaller state space. The concept of *tagged input queue* has been successfully used to evaluate the FIFO input-queued switch model (Pattavina *et al.* 1993, Youn *et*

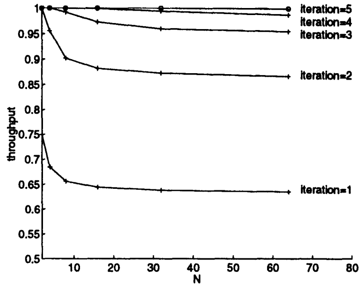


Figure 1 Maximum throughput as function of switch size and number of iterations.

al. 1994). These switches involve a single stage of contention resolution. On the other hand, for the switch with PIM scheduling, the contention resolution process consists of two stages. As observed from the algorithm descriptions 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. As a result, the corresponding model is more complicated than for the FIFO input-queued switch. We make the following assumptions in developing the PIM switch model: 1. The switch operates synchronously. 2. Every input queue has the same buffer size, namely b_i. 3. Cells arrive at every input queue according to an *i.i.d.* Bernoulli process with probability λ. 4. New cells arrive only at the beginning of the time slots, and cells depart only at the end of the time slots.

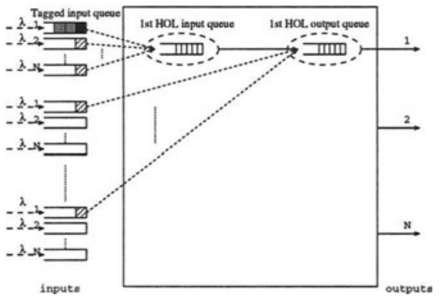


Figure 2 An example of the queueing model for the PIM switch.

Under the above assumptions, all the input queues will exhibit the same behavior when the system attains steady state. A queue at input i with output j as the destination is denoted by $Q(i, j)$. Figure 2 shows an example of the queueing model for the PIM switch. In this example, the occupancy of $Q(1, 1)$ is taken as the *tagged input queue*, the number of HOL cells at input 1 is represented by the *1st HOL input queue*, and the number of HOL cells addressed for output 1 is denoted by the *1st HOL output queue.* Both the *HOL input queue* and the *HOL output queue* are virtual queues which don't exist in a real PIM switch but are useful for our mathematical analysis.

3.1 Markov model

Analyzing the queueing model of the PIM switch requires the construction of the underlying Markov chain Z. The states of the Markov chain Z are sampled at the end of the time slots and can be expressed as a triplet (l, w_i, w_o), where l, w_i, and w_o refer to the lengths of the *tagged input queue, virtual HOL input queue,* and *virtual HOL output queue*, respectively. The state-space of this three-dimensional Markov chain is

$$\{(0,0,0),(l,w_i,w_o)|1 \leq l \leq b_i, 1 \leq w_i \leq N, 1 \leq w_o \leq N\}$$

and are ordered in a lexicographic order, that is, $(0,0,0)$, $(1,1,1)$, $...(b_i,N,N)$. The set of states $\{(l,1,1),(l,1,2),...(l,2,1),...(l,N,N)\}$ will be labelled as states in level l of the Markov chain. This Markov chain is a *Quasi Birth and Death (QBD)* process with block-partitioned form of transition probability matrix T as

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

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 $P_{blo,W_t(w_i',w_o')|W_{t-1}(w_i,w_o)}$ denote the probability that the HOL cell of the tagged queue is blocked, and $P_{suc,W_t(w_i',w_o')|W_{t-1}(w_i,w_o)}$ denote the probability that the HOL cell of the tagged queue is transmitted given that the remaining HOL cells at the end of the last time slot is (w_i, w_o) and the remaining HOL cells at the end of the current time slot is (w_i', w_o'). Define the matrice B, B_0 and S_0 as $B = [P_{blo,W_t(w_i',w_o')|W_{t-1}(w_i,w_o)}]$, $B_0 = [P_{blo,W_t(w_i',w_o')|W_{t-1}(0,0)}]$ and $S = [P_{suc,W_t(w_i',w_o')|W_{t-1}(w_i,w_o)}]$, where $0 \leq w_i', w_o', w_i, w_o \leq N$.

Let S_0 be 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 last time slot. From the definitions of B_0, S_0, S, and B, we can show that:

$$S_0 + B_0e = 1 \tag{3}$$

$$S_c + B_c = e \tag{4}$$

where $S_c = Se$, $B_c = Be$. As illustrated in the appendix of this paper, Eq (3) and Eq (4) help us solve the Markov chain using the Matrix-Geometric approach by simply focusing the computation on matrix B and vector B_0. By using the above equations, the element matrices in the transition probability matrix T can be computed as:

$$\begin{array}{lll} A_0' = (1-\lambda)S_c & A_1' = (1-\lambda) + \lambda S_0 & A_2' = \lambda B_0 \\ A_0 = (1-\lambda)S & A_1 = \lambda S + (1-\lambda)B & A_2 = \lambda B \end{array}$$

The remaining subsections will cover the computation of the success and blocking probabilities, $P_{suc,W_t(w_i',w_o')|W_{t-1}(w_i,w_o)}$, and $P_{blo,W_t(w_i',w_o')|W_{t-1}(w_i,w_o)}$ respectively. Once these probabilities are computed, the transition probability matrix T can be constructed. Once the transition probability matrix is known, it is a routine matter to derive the steady state equations by utilizing the properties of Markov chains, and solving the equations to obtain the steady-state probability vector. Detailed procedures are presented in the appendix of this paper.

3.2 Computing the blocking and success probabilities

We now derive the equations for computing the blocking and success probabilities. The transition of the state of the virtual HOL input/output queues from the state (w_i, w_o) to state (w_i', w_o') is a two step process illustrated in Figure 3: First, we account for the newly arriving HOL cells to the virtual HOL input/output queues. Then, we consider the transition from the intermediate state to the final state after applying the PIM algorithm.

$$\begin{tikzcd}(w_i,w_o) \arrow[r, "\text{Arriving HOL cells } (k_i,k_o)"'] & (h_i,h_o) \arrow[r, "\text{PIM algorithm to find maximal matching}"'] & (w_i',w_o')\end{tikzcd}$$

Figure 3 Transition of the virtual HOL queues.

(a) Arriving cells at the virtual HOL queues

Let $K_t(k_i, k_o)$ denote the number of newly arriving HOL cells at the *virtual HOL input/output queues* (k_i/k_o new arrivals to the *virtual HOL input/output queue*), at the beginning of current time slot t. $W_{t-1}(w_i, w_o)$ denotes the numbers of remaining HOL cells at the *virtual HOL input/output queue* (w_i/w_o is length of *virtual HOL input/output queue*), at the end of the previous time slot

$t-1$. Let $H_t(h_i, h_o) = K_t(k_i, k_o) + W_{t-1}(w_i, w_o)$. Define $a_{K(k_i,k_o)|W(w_i,w_o)} =$ *Prob(*$K_t(k_i, k_o)|W_{t-1}(w_i, w_o)$*)*. Let p_0 be the probability that a queue is empty in a time slot, and $p_1 = 1 - p_0$. 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. If the current state is (l, w_i, w_o), $(N - w_i)$ queues of input i and $(N - w_o)$ jth queues of inputs will be non-empty with probability p_1. Hence,

$$a_{K(k_i,k_o)|W(w_i,w_o)} = \begin{cases} 0, \qquad k_i < 0 \textit{ or } k_o < 0. \\ \binom{N-1}{k_i-1}\binom{N-1}{k_o-1} p_1^{k_i+k_o-2} p_0^{2N-(k_i+k_o)} \\ 1 \leq k_i \leq N,\ 1 \leq k_o \leq N,\ w_i = w_o = 0. \\ \binom{N-w_i}{k_i}\binom{N-w_o}{k_o} p_1^{k_i+k_o}(1-p_1)^{2N-(k_i+k_o+w_i+w_o)} \\ 0 \leq k_i \leq N - w_i,\ 0 \leq k_o \leq N - w_o,\ 1 \leq w_i, w_o \leq N. \end{cases}$$

(b) Transition to $W_t(w_i', w_o')$

Having determined the number of cell arrivals to the virtual HOL queues, we now consider the transition from the intermediate state to the final state after applying the PIM algorithm. Given the *tagged input queue* $Q(i,j)$, the inputs excluding input i are divided into two subsets E and F according to whether the jth queue of the inputs is empty or not. The cardinality of these sets are $(N - w_o)$ and $(w_o - 1)$ respectively. The state of set E and set F will affect the transitions of *virtual HOL input queue* and *virtual HOL output queue*. For the HOL cell of the tagged input queue, its contention process can be split into two stages. In the first stage, the tagged input queue contends with other non-empty queues at the same input. If it succeeds in the first stage contention, it joins the second stage contention with all successful *jth* queues from other inputs. Let $Q(i,k)(k \neq j)$ be the successful queue at input i if $Q(i,j)$ is blocked in the first contention stage. We define the following probabilities associated with the second transition step in Figure 3:

- $P_{blo\_00|H(h_i,h_o)}$ = *Prob*{*the HOL cell at the tagged input queue gets blocked, and* $W_t(w_i', w_o') = H_t(h_i, h_o)$ *given* $H_t(h_i, h_o)$}

- $P_{blo\_01|H(h_i,h_o)}$ = *Prob*{*the HOL cell at the tagged input queue gets blocked, and* $W_t(w_i', w_o') = H_t(h_i, h_o - 1)$ *given* $H_t(h_i, h_o)$}

- $P_{blo\_10|H(h_i,h_o)}$ = *Prob*{*the HOL cell at the tagged input queue gets blocked, and* $W_t(w_i', w_o') = H_t(h_i - 1, h_o)$ *given* $H_t(h_i, h_o)$}

- $P_{blo\_11|H(h_i,h_o)}$ = *Prob*{*the HOL cell at the tagged input queue gets blocked, and* $W_t(w_i', w_o') = H_t(h_i - 1, h_o - 1)$ *given* $H_t(h_i, h_o)$}

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

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

$$\begin{cases} 0,\ \mathit{for}\ r_i < -1\ \mathit{or}\ r_o < -1 \\[2ex] \begin{array}{l} a_{K(r_i,r_o)|W(w_i,w_o)} P_{blo\_00|H(w_i',w_o')} \\ +a_{K(r_i,r_o+1)|W(w_i,w_o)} P_{blo\_01|H(w_i',w_o'+1)} l_0(r_o)/w_o' \\ +a_{K(r_i,r_o)|W(w_i,w_o)} P_{blo\_01|H(w_i',w_o')} (1 - \frac{l_0(r_o-1)}{w_o'-1}) \\ +a_{K(r_i+1,r_o)|W(w_i,w_o)} P_{blo\_10|H(w_i'+1,w_o')} l_0(r_i)/w_i' \\ +a_{K(r_i,r_o)|W(w_i,w_o)} P_{blo\_10|H(w_i',w_o')} (1 - \frac{l_0(r_i-1)}{w_i'-1}) \\ +a_{K(r_i+1,r_o+1)|W(w_i,w_o)} P_{blo\_11|H(w_i'+1,w_o'+1)} \frac{l_0(r_i)l_0(r_o)}{w_i' \cdot w_o'} \\ +a_{K(r_i+1,r_o)|W(w_i,w_o)} P_{blo\_11|H(w_i'+1,w_o')} \frac{l_0(r_i)(r_o-1-l_0(r_o-1))}{w_i' \cdot (w_o'-1)} \\ +a_{K(r_i,r_o+1)|W(w_i,w_o)} P_{blo\_11|H(w_i',w_o'+1)} \frac{(r_i-1-l_0(r_i-1))l_0(r_o)}{(w_i'-1) \cdot w_o'} \\ +a_{K(r_i,r_o)|W(w_i,w_o)} P_{blo\_11|H(w_i',w_o')} \frac{(r_i-1-l_0(r_i-1))(r_o-1-l_0(r_o-1))}{(w_i'-1) \cdot (w_o'-1)} \\ ,\ \mathit{for}\ r_i \geq 1\ \mathit{and}\ r_o \geq 1\ \mathit{and}\ w_i = 0\ \mathit{and}\ w_o = 0 \end{array} \\[2ex] \begin{array}{l} a_{K(r_i,r_o)|W(w_i,w_o)} P_{blo\_00|H(w_i',w_o')} \\ +a_{K(r_i,r_o+1)|W(w_i,w_o)} P_{blo\_01|H(w_i',w_o'+1)} (r_o + 1 + l_0(w_o - 1))/w_o' \\ +a_{K(r_i,r_o)|W(w_i,w_o)} P_{blo\_01|H(w_i',w_o')} (w_o - 1 - l_0(w_o - 1))/(w_o' - 1) \\ +a_{K(r_i+1,r_o)|W(w_i,w_o)} P_{blo\_10|H(w_i'+1,w_o')} (r_i + 1 + l_0(w_i - 1))/w_i' \\ +a_{K(r_i,r_o)|W(w_i,w_o)} P_{blo\_10|H(w_i',w_o')} (w_i - 1 - l_0(w_i - 1))/(w_i' - 1) \\ +a_{K(r_i+1,r_o+1)|W(w_i,w_o)} P_{blo\_11|H(w_i'+1,w_o'+1)} \frac{(r_i+1+l_0(w_i-1))(r_o+1+l_0(w_o-1))}{w_i' \cdot w_o'} \\ +a_{K(r_i+1,r_o)|W(w_i,w_o)} P_{blo\_11|H(w_i'+1,w_o')} \frac{(r_i+1+l_0(w_i-1))(w_o-1-l_0(w_o-1))}{w_i' \cdot (w_o'-1)} \\ +a_{K(r_i,r_o+1)|W(w_i,w_o)} P_{blo\_11|H(w_i',w_o'+1)} \frac{(w_i-1-l_0(w_i-1))(r_o+1+l_0(w_o-1))}{(w_i'-1) \cdot w_o'} \\ +a_{K(r_i,r_o)|W(w_i,w_o)} P_{blo\_11|H(w_i',w_o')} \frac{(w_i-1-l_0(w_i-1))(w_o-1-l_0(w_o-1))}{(w_i'-1) \cdot (w_o'-1)}, \\ \mathit{for}\ r_i \geq -1\ \mathit{and}\ r_o \geq -1\ \mathit{and}\ w_i > 0\ \mathit{and}\ w_o > 0 \end{array} \end{cases} \tag{5}$$

in which

$$l_0(w) = \begin{cases} 0, & \mathit{for}\ w = 0 \\ \sum_{u=1}^{w} \binom{w}{u} P_{l1}^{u} (1 - P_{l1})^{w-u} & \mathit{for}\ w > 0 \end{cases} \tag{6}$$

represents the number of input queues that contain only one buffered cell, and P_{l1} in Eq (6) is the probability that an input queue length is equal to one (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 $W_{t-1}(w_i, w_o) = (0, 0)$, the blocking probability $P_{blo,W_t(w_i',w_o')|W_{t-1}(0,0)}$ can be computed by Eq (5) provided that the function P_{l1} in Eq (6) is replaced by $P_{l1}' = (\lambda\pi_0 + (1 - \lambda)\pi_1 e)/p_1$.

We can compute the probability $P_{suc,W_t(w'_i,w'_o)|W_{t-1}(w_i,w_o)}$ as:

$$P_{suc,W_t(w'_i,w'_o)|W_{t-1}(w_i,w_o)} = a_{K(k_i,k_o)|W(w_i,w_o)} P_{suc|H(k_i+w_i,k_o+w_o)}$$

(c) Applying the PIM algorithm

We now compute the probabilities in Figure 3 by considering each iteration of the PIM scheduling algorithm. The state of the switch at the beginning and end of each iteration ϕ is characterized by the following parameters:

- $n(\phi)$: 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.
- $h_o(\phi)$: the number of non-empty jth queues in $n(\phi)$ inputs (including input i) at the beginning of ϕth iteration matching.
- $m(\phi)$: the number of inputs/outputs that get matched at the end of ϕth iteration, $m(\phi) = n(\phi) - n(\phi+1)$.
- $\Delta h_i(\phi)$: the number of outputs whose corresponding non-empty queues in input i that get matched at the end of ϕth iteration, $\Delta h_i(\phi) = h_i(\phi) - h_i(\phi+1)$.
- $\Delta h_o(\phi)$: the number of inputs in set F that get matched at the end of ϕth iteration, $\Delta h_o(\phi) = h_o(\phi) - h_o(\phi+1)$.

For the sake of simplicity, we do not mention the iteration number in the following discussion. If no iteration number is mentioned, then the current iteration ϕ is implied.

Let $x_i x_j$ represent the state of the matching process for input i and output j of the switch, where $x_i, x_j \in \{0,1\}$ with 0 representing that the input/output is unmatched and 1 representing that the input/output is matched at the end of the current iteration. The possible states of the matching process are 00, 01, 10, and 11. However, the state 11 should explicitly consider if the tagged input queue $Q(i,j)$ at input i is matched. Thus the state 11 is split into two: 11_{suc} and 11_{blo} respectively. Given the current state of the switch $(n(\phi), h_i(\phi), h_o(\phi))$ and the current state of the matching process $x_i x_j$, the resulting state of the switch $(n(\phi+1), h_i(\phi+1), h_o(\phi+1))$ and the resulting state of the matching process $x'_i x'_j$ is controlled by the transition probabilities as in Figure 4. These probabilities are functions of the current state of the switch $(n(\phi), h_i(\phi), h_o(\phi))$ and are defined as:

- $P_{blo\_x'_i x'_j|x_i x_j}$=*Prob*{*at the end of current iteration, the HOL cell at the tagged input queue $Q(i,j)$ gets blocked; and $x'_i x'_j / x_i x_j$ represent whether input i (x'_i or x_i) and output j (x'_j or x_j) remain unmatched (represented by 0) or get matched (represented by 1) at the end/beginning of the current iteration*}.

- $P_{suc|00}$ = *Prob{at the end of current iteration, the HOL cell at the tagged input queue $Q(i,j)$ gets matched with output j, given that input i and output j were unmatched at the beginning of current iteration}.*

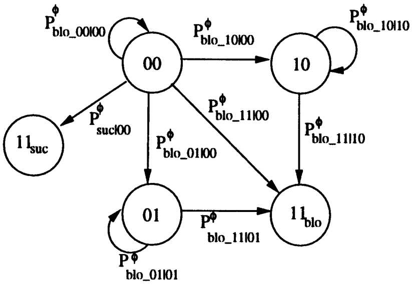


Figure 4 The matching process state transition diagram.

To derive equations for the transition probabilities, we define the following probabilities associated with the first stage of contention for a cell and give the computing formulas of them as below:

- P_{suc1_e} = *Prob{the HOL cell at* kth*($k \neq j$) queue of an input in set E succeeds in the first stage contention}.*
- P_{suc1_ft} = *Prob{the HOL cell at* jth *queue of an input in set F succeeds in the first stage contention}.*
- P_{suc1_fe} = *Prob{the HOL cell at* kth*($k \neq j$) queue of an input in set F succeeds in the first stage contention}.*

$$P_{suc1\_ft} = \sum_{v=1}^{n} \binom{n-1}{v-1} p_1^{(v-1)} p_0^{(n-v)} / v$$

$$P_{suc1\_e} = \frac{1-p_0^{n-1}}{n-1} \qquad P_{suc1\_fe} = \frac{1-P_{suc1\_ft}}{n-1}$$

Let t ($\max(m, h_o - 1) \leq t \leq n-1$) 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 for by the t inputs. There are three sub-problems to be considered in computing the transition probabilities in ϕth iteration given $(n(\phi), h_i(\phi), h_o(\phi))$ and $(n(\phi+1), h_i(\phi+1), h_o(\phi+1))$: 1. What is the probability that t inputs contend for m outputs? 2. What is the probability that Δh_o inputs in set F (whose cardinality is $h_o - 1$) get matched? and 3. What is the probability that Δh_i out of h_i outputs whose corresponding queues in input i are non-empty get matched? The equations below consider each of the sub-problems in computing the transition probabilities.

Computing $P_{blo\_00|00}$*:* In this case both input i and output j remain unmatched. Given t and m, the probability that the queue that succeeds in the first stage of contention at input i gets blocked at its corresponding output is

$$\begin{aligned} P_{t\rightarrow m|blo\_00\_00} &= (1-\frac{m}{t+1})(m!S_t^m + (m-1)!S_t^{m-1})P_{sucl\_e}^{t-h_o+1} \\ &\quad \cdot p_0^{(n-1)(n-1-t)} P_{sucl\_fe}^{h_o-1} \end{aligned}$$

Among the m outputs that get matched, Δh_i of them will see their corresponding queues in input i being non-empty. The number of combinations satisfying this condition is $C_{\Delta h_i|blo\_00\_00} = \binom{h_i-2}{\Delta h_i-1}\binom{n-h_i}{m-\Delta h_i}$.

Given that input i is blocked, it is clear that each combination of m out of t inputs gets matched with equal probability. The probability that Δh_o inputs which are elements of the set F get matched is $P_{\Delta h_o|blo\_00\_00} = \frac{\binom{h_o-1}{\Delta h_o}\binom{t-h_o+1}{m-\Delta h_o}}{\binom{t}{m}}$.

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

$$\begin{aligned} P_{blo\_00|00} &= (1-1/h_i) \sum_{t=max(m,h_o-1)}^{n-1} \binom{n-h_o}{t-h_o+1} C_{\Delta h_i|blo\_00\_00} \\ &\quad \cdot P_{\Delta h_o|blo\_00\_00} P_{t\rightarrow m|blo\_00\_00} \end{aligned}$$

Computing $P_{blo\_10|00}$*:* In this case, input i gets matched while output j remains unmatched. So we compute only the aggregated probability over the set of all possible Δh_o. The probability that the queue that succeeds in the first stage of contention at input i succeeds in getting matched in the second stage of contention is

$$\begin{aligned} P_{t\rightarrow m|blo\_10\_00} &= (1-\frac{m}{t+1})(m!S_t^m + (m-1)!S_t^{m-1})P_{sucl\_e}^{t-h_o+1} \\ &\quad \cdot p_0^{(n-1)(n-1-t)} P_{sucl\_fe}^{h_o-1} \end{aligned}$$

The probability that Δh_o inputs which are elements of set F get matched is $P_{\Delta h_o|blo\_10\_00} = \frac{\binom{h_o-1}{\Delta h_o-1}\binom{t-h_o+1}{m-\Delta h_o}}{\binom{t}{m-1}}$. Therefore, $P_{blo\_10|00}$ is given by,

$$\begin{aligned} P_{blo\_10|00} &= (1-1/h_i)\binom{n-2}{m-1} \sum_{t=max(m-1,h_o-1)}^{n-1} \binom{n-h_o}{t-h_o+1} \\ &\quad \cdot P_{\Delta h_o|blo\_10\_00} P_{t\rightarrow m|blo\_10\_00} \end{aligned}$$

Computing $P_{blo\_01|00}$*:* In this case, output j gets matched while input i remains unmatched. We compute only the aggregated probability over the set of all possible Δh_i. There are two cases to be considered here: (i) $Q(i,j)$

fails the first stage of contention, and (ii) $Q(i,j)$ survives the first stage of contention. Therefore $P_{blo\_01|00} = P_{blo\_01\_B|00} + P_{blo\_01\_S|00}$, where $P_{blo\_01\_B|00}$ and $P_{bloc\_01\_S|00}$ are probabilities for the two cases (i) and (ii) respectively. Case (i): $Q(i,j)$ fails in the first stage of contention

$$\begin{aligned} P_{blo\_01\_B|00} &= (1 - 1/h_i) \sum_{t=max(m,h_o-1)}^{n-1} \binom{n-h_o}{t-h_o+1} \\ &\cdot \sum_{u=1}^{min(h_o-1,t-m+1)} \binom{h_o-1}{u} C_{\Delta h_i|blo\_01\_B\_00} P^u_{t\to m|blo\_01\_B\_00} \end{aligned}$$

where

$$\begin{aligned} P^u_{t\to m|blo\_01\_B\_00} &= (1 - \frac{m-1}{t-u+1})((m-1)!S^{m-1}_{t-u} + (m-2)!S^{m-2}_{t-u}) \\ &\cdot P^u_{sucl\_ft} P^{h_o-1-u}_{sucl\_fe} P^{t-h_o+1}_{sucl\_e} p_0^{(n-1)(n-1-t)} \\ C_{\Delta h_i|blo\_01\_B\_00} &= \binom{h_i-2}{\Delta h_i - 2}\binom{n-h_i}{m-\Delta h_i} \end{aligned}$$

Case (ii): $Q(i,j)$ is successful in the first stage of contention

$$P_{blo\_01\_S\_00} = \frac{1}{h_i} \sum_{t=max(m,h_o-1)}^{n-1} \binom{n-h_o}{t-h_o+1} C_{\Delta h_i|blo\_01\_S\_00} P_{t\to m|blo\_01\_S\_00}$$

where

$$\begin{aligned} P_{t\to m|blo\_01\_S\_00} &= \sum_{k=1}^{min(h_o-1,t-m+1)} \binom{t}{k} (1 - \frac{1}{k+1})(m-1)!S^{m-1}_{t-k} \\ &\cdot P^k_{sucl\_ft} P^{h_o-1-k}_{sucl\_fe} P^{t-h_o+1}_{sucl\_e} p_0^{(n-1)(n-1-t)} \\ C_{\Delta h_i|blo\_01\_S\_00} &= \binom{h_i-1}{\Delta h_i - 1}\binom{n-h_i}{m-\Delta h_i} \end{aligned}$$

Computing $P_{suc|00}$*:* The state 11_{suc} in Figure 3 is an absorbing state, so this transition probability is computed without consideration on m.

$$P_{suc|00} = \frac{1}{h_i} \sum_{u=0}^{h_o-1} \binom{h_o-1}{u} \frac{1}{u+1} P^u_{suc\_ft} (1 - P_{suc\_ft})^{h_o-1-u}$$

Computing $P_{blo\_11|00}$*:* Then, $P_{blo\_11|00}$ is computed from the boundary condition as $P_{blo\_11|00} = 1 - (P_{blo\_00|00} + P_{blo\_01|00} + P_{blo\_10|00} + P_{suc|00})$.

Computing $P_{blo\_01|01}$: In this case input i remains unmatched, while output j is already matched. Then,

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

where

$$P_{t\to m|blo\_01\_01} = (1 - \frac{m}{t+1})(m!S_t^m + (m-1)!S_t^{m-1})P_{suc\_0}^t p_0^{n(n-1-t)}$$

$$P_{suc\_0} = \frac{1-p_0^n}{n} \qquad C_{\Delta h_i|blo\_01\_01} = \binom{h_i - 1}{\Delta h_i - 1}\binom{n-h_i}{m - \Delta h_i}$$

Computing $P_{blo\_11|01}$: In this case input i gets matched while output j has already been matched at the beginning of the iteration.

$$P_{blo\_11|01} = \sum_{u=0}^{n-1} \binom{n-1}{u} \frac{1}{u+1} P_{suc\_0}^u (1 - P_{suc\_0})^{n-1-u}$$

Computing $P_{blo\_10|10}$: In this case input i is already matched, while output j remains unmatched at the end of the iteration. Then,

$$P_{blo\_10|10} = \binom{n-1}{m} \sum_{t=max(h_o-1,m)}^{n} \binom{n-h_o+1}{t-h_o+1} P_{\Delta h_o|blo\_10\_10} P_{t\to m|blo\_10\_10}$$

where

$$\begin{aligned} P_{t\to m|blo\_10\_10} &= m!S_t^m P_{suc1\_e}^{t-h_o+1} p_0^{(n-1)(n-t)} P_{suc1\_fe}^{h_o-1} \\ P_{\Delta h_o|blo\_10\_10} &= \frac{\binom{h_o-1}{\Delta h_o}\binom{t-h_o+1}{m-\Delta h_o}}{\binom{t}{m}} \end{aligned}$$

Computing $P_{blo\_11|10}$: In this case output j gets matched at the end of the iteration. This is feasible only if at least one of j th queues of the $h_o - 1$ inputs in set F succeed in the first stage of contention at their respective inputs.

$$P_{blo\_11|10} = 1 - (1 - P_{suc1\_ft})^{h_o - 1}$$

The states of the switch at the end of each iteration, $(n(\phi), h_i(\phi), h_o(\phi), x_i x_j)$, can be viewed as a weighted tree with the nodes of the tree corresponding to the switch states. The root of the tree is the initial state of the switch $(N, h_i, h_o, 00)$. All states in level ϕ of the tree correspond to the states of the switch at the end of the ϕ th iteration of the PIM algorithm. Weights are assigned to the arcs between the states, and are equal to the transition probabilities $P_{blo\_x'_i x'_j|x_i x_j}$ or $P_{suc|x_i x_j}$. Each state $(n(\phi), h_i(\phi), h_o(\phi), x_i x_j)$ is assigned a probability $Pr(n(\phi), h_i(\phi), h_o(\phi), x_i x_j)$ equal to the product of the transition probabilities along the arcs from the root to the state. The probabilities $P_{blo\_00|H(h_i,h_o)}$, $P_{blo\_01|H(h_i,h_o)}$, $P_{blo\_10|H(h_i,h_o)}$, and $P_{suc|H(h_i,h_o)}$ at the end of Φ iterations of the PIM algorithm can be computed as

$$P_{blo\_x_i x_j|H(h_i,h_o)} = \sum_{n(\Phi),h_i(\Phi),h_o(\Phi)} Pr(n(\Phi), h_i(\Phi), h_o(\Phi), x_i x_j)$$
$$P_{suc|H(h_i,h_o)} = \sum_{n(\Phi),h_i(\Phi),h_o(\Phi)} Pr(n(\Phi), h_i(\Phi), h_o(\Phi), 11_{suc})$$

3.3 Solving the Markov chain

As can be seen from the above equations, p_0, π_0 and π_1 must be known in advance in order to compute the steady state probabilities. From the Appendix of this paper, the steady state probabilities are given by:

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

Notice that in steady state the following equation holds

$$p_0 = (1-\lambda)\pi_0 \tag{7}$$

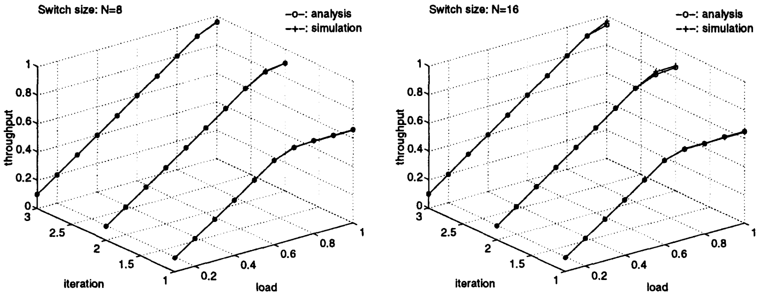
This naturally suggests an iterative solution (Youn *et al.* 1994). Initially, π_0 is set to zero, which corresponds to the case of saturated offered loads. Then p_0 can be obtained by using Eq (7). Since p_0 is known, the next value of π_0 is computed again. This iterating process continues until both p_0 and π_0 converge, leading to the values of steady state probabilities π.

3.4 Computing the performance metrics

Once the steady state probabilities are known, then interesting performance parameters, such as throughput, mean queue length and mean cell loss probability can be computed directly by using the known parameters. Let ρ, $\bar{Q}$, $\bar{D}$ and P_{loss} be the throughput, mean queue length, mean cell delay and mean cell loss probability respectively, then

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

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



(a) 8 × 8 switch

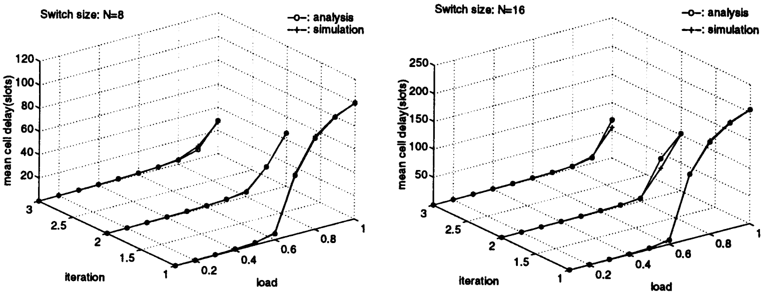
(b) 16 × 16 switch

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

4 NUMERICAL RESULTS

Both mathematical analysis and simulation results are presented in this section in order to investigate the accuracy of the above queueing model. Figure 5 shows the switch throughput as function of offered load λ for PIM switch sizes 8 and 16 with various PIM scheduling iteration numbers 1, 2 and 3, respectively. It can be seen that when the switch size increases, the throughput of the switch decreases under high offered load (greater than 60% when maximum iteration is 1). Also from this figure, we can see that the saturation throughput will increase as the PIM scheduling iteration increases. It is expected that with more iterations, more HOL cells get matched during a scheduling iteration. The curves show that 3 iterations are enough to get a high throughput > 90%. Comparing Figure 5 with Figure 1, we can see that even under saturated traffic loads, our queueing model approximates the original system quite well.

Figure 6 shows the mean cell delay as a function of offered load λ for the different PIM switch sizes 8 and 16 with various PIM iteration numbers 1, 2, and 3. 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 just one iteration. For a 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 cluster into a single point. It isn't difficult to understand that, under low traffic



(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.

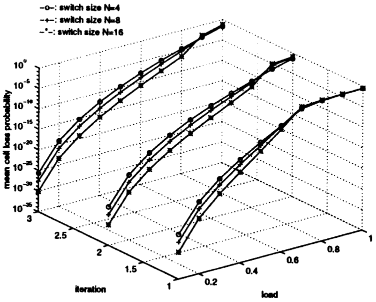


Figure 7 The mean cell loss probability of a PIM switch, as a function of offere d load, with a buffer size b_i=10.

load, the opportunity that more than one HOL cell contend for a common input/output is small. That is, single iteration PIM scheduling is typically enough to find a maximal matching. When the traffic load grows, the chances of conflits increase and more iterations are needed using PIM scheduling to achieve a maximal matching. In this case, the analysis results diverse from the simulation results significantly when compared to the case that the traffic load is low. This phenomenon is due to the approximation in computing the transition probability of multiple iteration PIM algorithm.

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 CONCLUSION

In order to make the original switch model tractable for analysis, a number of assumptions have been added. The most important is that the random traffic, that is, cells are arriving at each input according to an *i.i.d.* Bernoulli process, and the destinated output of arriving cells are distributed over all outputs uniformly. In case of non-random traffic loads, the analysis will be more complicated than the one for random traffic loads. In future research direction, we will try to apply this method to analyze the same kind of ATM switches with bursty and correlated traffic.

The contribution of this paper is two fold. First, the throughput of an ATM switch with multiple iteration PIM scheduling in case of saturated traffic load is analyzed mathematically. Second, a theoretical analysis for various performance parameters including throughput, mean cell delay, and mean cell loss probability, of a ATM switch using a PIM scheduling scheme is presented. Such theoretical analysis is lacking in existing literature on ATM switches with PIM or variations of PIM scheduling (Anderson *et al.* 1993, McKeown 1994, Mckeown *et al.* 1994, LaMaire *et al.* 1994).

6 APPENDIX: COMPUTATION OF THE STEADY STATE PROBABILITIES

Following the steps given in (Youn *et al.* 1994), we give the procedures to compute the steady state probabilities of the Markov chain. The method presented in (Youn *et al.* 1994) is based on the algorithmic approach given in (Neuts 1981). From the definition of the transition probability matrix, we know that $\Pi T = \Pi$. By expanding this equation, we have:

$$\pi_0((1-\lambda)+\lambda S_0)+\pi_1(1-\lambda)S_c=\pi_0 \tag{8}$$

$$\pi_0\lambda B_0+\pi_1(\lambda S+(1-\lambda)B)+\pi_2(1-\lambda)S=\pi_1 \tag{9}$$

$$\pi_{i-1}\lambda B+\pi_i(\lambda S+(1-\lambda)B)+\pi_{i+1}(1-\lambda)S=\pi_i\ ,\quad for\ 1<i<b_i-1 \tag{10}$$

$$\pi_{b_i-2}\lambda B+\pi_{b_i-1}(\lambda S+(1-\lambda)B)+\pi_{b_i}S=\pi_{b_i-1} \tag{11}$$

$$\pi_{b_i-1}\lambda B+\pi_{b_i}B=\pi_{b_i} \tag{12}$$

Multiplying Eq (10) by e results in:

$$\pi_{i-1}\lambda B_c=\pi_i(1-\lambda)S_c, \tag{13}$$

The solution for $\pi_i(1 < i < b_i)$ in terms of π_{i-1} can be obtained by multiplying Eq (10) by I_1 and using Eq (13), where $I_1 = ee_1$ and $e_1 = [1, 0, 0, ..., 0]$.

$$\pi_i(I_1 - \lambda SI_1 - BI_1) = \pi_{i-1}\lambda B_c e_1 \tag{14}$$

Multiplying Eq (4) by e_1 and substitute it into Eq (14), we have a recursive formula for π_i in terms of matrix B.

$$\pi_i \ = \ \pi_1(\lambda B((1-\lambda)(I-B))^{-1})^{i-1} \tag{15}$$

From Eq (12), we have:

$$\pi_{b_i} = \pi_{b_i-1}\lambda B(I-B)^{-1} \tag{16}$$

Let $\alpha = \lambda B_0(I - \lambda I_1 - (1-\lambda)B)^{-1}$ and $\beta = \lambda B((1-\lambda)(I-B))^{-1}$. Using Eq (8), we get:

$$\pi_1 = \lambda B_0(I - \lambda I_1 - (1-\lambda)B)^{-1}\pi_0 = \pi_0\alpha \tag{17}$$

Using Eq (15, 16, 17),

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

Notice that $\pi_0 + \sum_{l=1}^{b_i}\pi_l e = 1$, we have:

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

REFERENCES

Abramowitz, M. and Stegun, I.A. (1972) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* John Wiley & Sons.

Awdeh, R.Y. and Mouftah, H.T. (1995) Survey of ATM Switch Architectures. *Computer Networks and ISDN Systems*, **Vol. 27, No. 12**, 1567-1613.

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

LaMaire, R.O. and Serpanos, D.N. (1994) Two-Dimentional Round-Robin Schedulers for Packet Switches with Multiple Input Queues. *IEEE/ACM Transactions on Networking*, **Vol.2, No.5**, 471-82.

McKeown, N. (1994) *Scheduling Algorithms for Input-Queued Cell Switches.* Ph.D. thesis, University of California at Berkeley.

Mckeown, N., Varaiya, P., and Walrand, J. (1994) Scheduling Cells in an Input-Queued Switch. *Electronics Letters*, **Vol. 29, No. 25**, 2174-2175.

Neuts, M.F. (1981) *Matrix-Geometric Solutions in Stochastic Models.* Johns Hopkins Universi ty Press.

Pattavina, A. and Bruzzi, G. (1993) Analysis of Input and Output Queueing for Nonblocking ATM Switches. *IEEE/ACM Trans. on Networking*, **Vol. 1, No. 3**, 314–328.

Youn Chan Jung and Chong Kwan Un (1994) Performance Analysis of Packet Switches with Input and Output Buffers. *Computer Networks and ISDN Systems*, **Vol. 26, No. 12**, 1559-1580.

7 BIOGRAPHY

Ge Nong received the B.E. degree from the NanJing Aeronautical Institute, China, in 1992 and the M.E. degree from the South China Univ. of Sci. and Technol., China, in 1995, all in Computer Engineering. He is currently pursuing his Ph.D degree in the Computer Science Department (CSD), The Hong Kong Univ. of Science and Technology (HKUST), Kowloon, Hong Kong. His current research interests include performance modeling of ATM switches and architectures of high-speed packet switching.

Jogesh K. Muppala received the Ph.D degree in Elec. Eng. from Duke Univ., Durham, NC in 1991, the M.S. degree in Computer Eng. from Univ. of Southwestern Louisiana, LA in 1987 and the B.E. degree in Elec. and Comm. Eng. from Osmania Univ., India in 1985. He is currently an Assistant Prof. in the CSD, HKUST. His research interests include performance and dependability modeling, high speed networking, distributed systems, and stochastic Petri nets.

Mounir Hamdi received the BSc degree with distinction in Electrical Engineering from the Univ. of Southwestern Louisiana in 1985, and MSc and PhD degrees in Electrical Engineering from the Univ. of Pittsburgh in 1987 and 1991, respectively. In 1991 he joined the CSD, HKUST as an Assistant Prof.. He is now an Associate Prof. of Computer Sci. and the Director of the Computer Eng. Programme. His main areas of research are Parallel Computing, High-Speed Networks, ATM Packet Switching Architectures, and Wireless networking.