Performance Evaluation of Multiple Input-Queued ATM Switches With PIM Scheduling Under Bursty Traffic

Ge Nong, Mounir Hamdi, and Jogesh K. Muppala

***Abstract*—In this letter, we analyze the performance of multiple input-queued asynchronous transfer mode (ATM) switches that use parallel iterative matching for scheduling the transmission of head-of-line cells in the input queues. A queueing model of the switch is developed under independently, identically distributed, two-state *Markov modulated Bernoulli processes* bursty traffic. The underlying Markov chain of the queueing model is a *quasi-birth–death* (QBD) chain. The QBD chain is solved using an iterative computing method. Interesting performance metrics of the ATM switch such as the throughput, the mean cell delay, and the cell loss probability can be derived from the model. Numerical results from both the analytical model and simulation are presented, and the accuracy of the analysis is briefly discussed.**

I. Introduction

In THIS letter, we present an analytical model for a multiple input-queued asynchronous transfer mode (ATM) switch with virtual-output-queueing (VOQ), where each input of the switch maintains N separate queues, one for each of the N output ports. The switch operates synchronously and in each time slot the *head-of-line* (HOL) cells at the input queues can be selected for transmission across the switch with the constraint that, at most, one cell is able to be transmitted from/to any one input/output link. Specifically, the *parallel iterative matching* (PIM) algorithm used in DEC's *AN2* switches [2] is employed to schedule the queueing HOL cells to be forwarded to their destined output ports. For notational simplicity, the VOQ switch scheduled using the PIM algorithm is hereafter referred to as the PIM switch.

We extend our model for a PIM switch presented in [5] and [6], which considered the performance under independently, identically distributed (i.i.d.) Bernoulli traffic to the case of *i.i.d.* bursty traffic modeled by a two-state *Markov-modulated Bernoulli processes* (MMBPs) [1]. The remainder of this letter is organized as follows. In Section II, we develop and present the solution for the queueing model of the switch. In Section III, numerical results from the queueing model are presented and compared with the results from simulation. Finally, Section IV gives the conclusions.

II. Queueing Model and Analysis of the PIM Switch

A. Queueing Model

The queueing model is developed under the following assumptions. 1) The switch operates synchronously. 2) Every input queue has the same buffer size, namely b_i. 3) New cells arrive only at the beginning of the time slots and cells depart only at the end of the time slots. 4) Cells arrive at each input according to an ON–OFF bursty process [1] modeled by a two-state MMBP, where cells are generated only in the ON (1) state and the destinations of cells are uniformly distributed over all output ports. Only one cell can arrive at each input in a time slot. Times spent in the ON (1) and OFF (0) states are geometrically distributed with means of $(1-\alpha)^{-1}$ and $(1-\beta)^{-1}$, respectively. For an $N \times N$ switch, if an input's load is $N\lambda$, then every queue at this input has an offered load of λ. Given the mean burst length τ and the mean arrival rate λ, α and β can be calculated as $\alpha = 1 - 1/\tau$ and $\beta = 1 - N\lambda(2-\alpha)/1 - N\lambda$, respectively.

With the above assumptions, all the input queues' stochastic processes will be the same when the system attains the equilibrium steady state. The queue at input i, with output j as the destination, is denoted $Q(i,j)$. The occupancy of $Q(i,j)$ is taken as the *tagged input queue*. The number of HOL cells at input i is denoted as the ith *HOL input queue*, and the number of HOL cells addressed for output j is denoted as the jth *HOL output queue*. Fig. 1 summarizes this information.

B. Underlying Markov Chain

The queueing model is analyzed by constructing an underlying Markov chain Z in which the states are sampled at the end of each time slot. Each state is expressed as a 4-tuple (L, G, W_i, W_o), where L, G, W_i, and W_o refer to the length of the *tagged input queue*, the state of the traffic source at the tagged input queue, the length of the *virtual HOL input queue*, and the length of the *virtual HOL output queue*, respectively. Within each time slot, a sequence of operations are considered to occur in the following order: 1) generating new arrivals; 2) applying the PIM algorithm to select a set of cells from the currently queued HOL cells; and 3) transferring the HOL cells selected in the previous step to their output ports. Each transition probability in the Markov chain accounts for all the relevant events that occur in the aforementioned operations. The state space of this four-dimensional Markov chain is given by

$$\{(0,g,0,0),(l,g,w_i,w_o) \mid 1 \le l \le b_i, 0 \le g \le 1, \\ 1 \le w_i \le N, 1 \le w_o \le N\}$$

where all elements are ordered in a lexicographical order, i.e., $(0,0,0,0)$, $(0,1,0,0)$, $(1,0,1,1)$, $\ldots$, $(b_i,0,N,N)$, $(b_i,1,N,N)$. The set of states $\{(l,0,1,1), (l,1,1,1), \ldots, (l,1,2,2), \ldots, (l,1,N,N)\}$ will be labeled as states in level l of the Markov chain. The Markov chain Z is a quasi-birth–death (QBD) process with a $(2+2Lb_iN^2)\times(2+2Lb_iN^2)$ transition

Paper approved by P. E. Rynes, the Editor for Switching Systems of the IEEE Communications Society. Manuscript received July 23, 1999; revised May 31, 2000 and February 9, 2001. This work was supported in part by a grant from the Hong Kong Research Grants Council.

The authors are with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: nong@ieee.org; hamdi@cs.ust.hkand; muppala@cs.ust.hk).

Publisher Item Identifier S 0090-6778(01)06928-8.

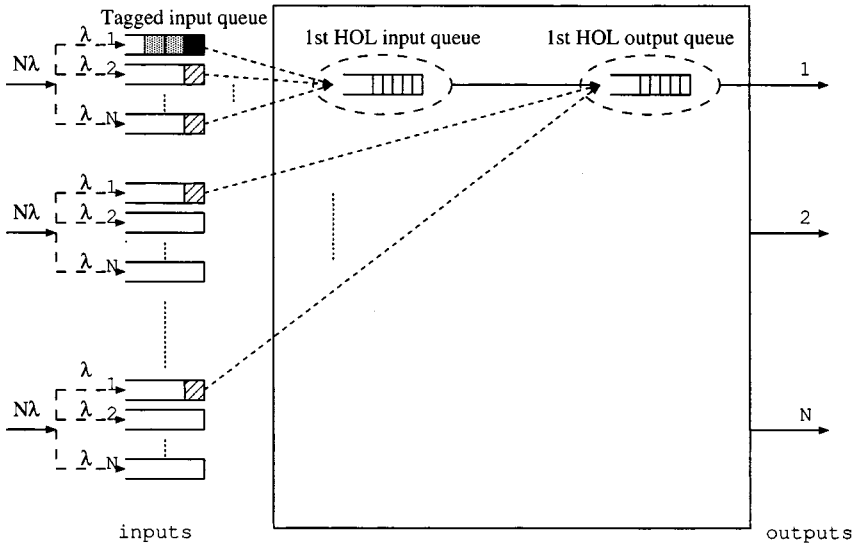

Fig. 1. An example of the queueing model for the PIM switch.

probability matrix that is sparse and block-partitioned as shown below

$$T = \begin{bmatrix} C_1 & C_2 & 0 & & & & & \\ C_0 & A_1 & A_2 & 0 & & & & \\ 0 & A_0 & A_1 & A_2 & 0 & & & \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & & \\ 0 & 0 & 0 & \cdots & 0 & A_0 & A_1 & A_2 \\ 0 & 0 & 0 & \cdots & 0 & 0 & D_0 & D_1 \end{bmatrix}$$

where $C_1[1,1]^T + C_2 e = [1,1]^T$ and $C_0 + (A_1 + A_2)e = (A_0 + A_1 + A_2)e = (D_0 + D_1)e = e$, e is a column vector of ones of length $2N^2$. To account for the transition probabilities in T, let $P_{\text{blo},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$ ($P'_{\text{blo},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$, respectively) denote the joint probability that the HOL cell of the queue $Q(i,j)$ is blocked and there is (is not, respectively) a new cell arrival at the queue $Q(i,j)$ at the beginning of the current time slot, and similarly $P_{\text{suc},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$ ($P'_{\text{suc},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$, respectively) denote the joint probability that the HOL cell of the queue $Q(i,j)$ is transmitted and there is (is not, respectively) a new cell arrival at the queue $Q(i,j)$ at the beginning of the current time slot, given the following: 1) at the end of the last time slot, the traffic source at input i is in state g and the lengths of the virtual HOL input and output queues are w_i and w_o, and 2) at the end of the current time slot, the lengths of the virtual HOL input and output queues are w_i' and w_o'.

For the case that there is a new arrival cell at the *tagged input queue* at the beginning of the current time slot, we define six matrices $B^{(g)}$, $B_0^{(g)}$ and $S^{(g)}$ ($g = 0$ or $g = 1$) as the equation shown at the bottom of the page.

In case there is no new cell arrival at the *tagged input queue* at the beginning of the current time slot, we define another six matrices $B'^{(g)}$, $B_0'^{(g)}$, and $S'^{(g)}$ similar to $B^{(g)}$, $B_0^{(g)}$, and $S^{(g)}$ by replacing $P_{\text{blo},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$ in $B^{(g)}$ with $P'_{\text{blo},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$, $P_{\text{blo},W_t(w_i',w_o')|W_{t-1}(g,0,0)}$ in $B_0^{(g)}$ with $P'_{\text{blo},W_t(w_i',w_o')|W_{t-1}(g,0,0)}$ and $P_{\text{suc},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$ in $S^{(g)}$ with $P'_{\text{suc},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$, respectively. Using the above

$$B^{(g)} = \begin{bmatrix} P_{\text{blo},W_t(1,1)|W_{t-1}(g,1,1)} & P_{\text{blo},W_t(1,2)|W_{t-1}(g,1,1)} & \cdots & P_{\text{blo},W_t(N,N)|W_{t-1}(g,1,1)} \\ P_{\text{blo},W_t(1,1)|W_{t-1}(g,1,2)} & P_{\text{blo},W_t(1,2)|W_{t-1}(g,1,2)} & \cdots & P_{\text{blo},W_t(N,N)|W_{t-1}(g,1,2)} \\ P_{\text{blo},W_t(1,1)|W_{t-1}(g,1,3)} & P_{\text{blo},W_t(1,2)|W_{t-1}(g,1,3)} & \cdots & P_{\text{blo},W_t(N,N)|W_{t-1}(g,1,3)} \\ \vdots & \vdots & \cdots & \vdots \\ P_{\text{blo},W_t(1,1)|W_{t-1}(g,N,N)} & P_{\text{blo},W_t(1,2)|W_{t-1}(g,N,N)} & \cdots & P_{\text{blo},W_t(N,N)|W_{t-1}(g,N,N)} \end{bmatrix}$$

$$B_0^{(g)} = \left[P_{\text{blo},W_t(1,1)|W_{t-1}(g,0,0)},\quad P_{\text{blo},W_t(1,2)|W_{t-1}(g,0,0)},\quad \ldots,\quad P_{\text{blo},W_t(N,N)|Wt-1(g,0,0)}\right]$$

$$S^{(g)} = \begin{bmatrix} P_{\text{suc},W_t(1,1)|W_{t-1}(g,1,1)} & P_{\text{suc},W_t(1,2)|W_{t-1}(g,1,1)} & \cdots & P_{\text{suc},W_t(N,N)|W_{t-1}(g,1,1)} \\ P_{\text{suc},W_t(1,1)|W_{t-1}(g,1,2)} & P_{\text{suc},W_t(1,2)|W_{t-1}(g,1,2)} & \cdots & P_{\text{suc},W_t(N,N)|W_{t-1}(g,1,2)} \\ P_{\text{suc},W_t(1,1)|W_{t-1}(g,1,3)} & P_{\text{suc},W_t(1,2)|W_{t-1}(g,1,3)} & \cdots & P_{\text{suc},W_t(N,N)|W_{t-1}(g,1,3)} \\ \vdots & \vdots & \cdots & \vdots \\ P_{\text{suc},W_t(1,1)|W_{t-1}(g,N,N)} & P_{\text{suc},W_t(1,2)|W_{t-1}(g,N,N)} & \cdots & P_{\text{suc},W_t(N,N)|W_{t-1}(g,N,N)} \end{bmatrix}.$$

definitions, the element matrices in the transition probability matrix T can be computed as shown below

$$C_0 = \begin{bmatrix} H_0 e_1 & \left(S'^{(0)} - H_0\right) e_1 \\ H_1 e_1 & \left(S'^{(1)} - H_1\right) e_1 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} \beta & 1-\beta-B_0^{(0)} e_1 \\ 1-\alpha & \alpha - B_0^{(1)} e_1 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} z_r & B_0^{(0)} \\ z_r & B_0^{(1)} \end{bmatrix}$$

$$A_0 = \begin{bmatrix} H_0 & S'^{(0)} - H_0 \\ H_1 & S'^{(1)} - H_1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} G_0 & S^{(0)} + B'^{(0)} - G_0 \\ G_1 & S^{(1)} + B'^{(1)} - G_1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} z_m & B^{(0)} \\ z_m & B^{(1)} \end{bmatrix}$$

$$D_0 = \begin{bmatrix} H_0 & S^{(0)} + S'^{(0)} - H_0 \\ H_1 & S^{(1)} + S'^{(1)} - H_1 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} G_0 & B^{(0)} + B'(0) - G_0 \\ G_1 & B^{(1)} + B'(1) - G_1 \end{bmatrix}$$

where e_1 is a column vector of ones of size N^2, z_r/z_c is a row/column vector of zeros of length N^2, z_m is an $N^2 \times N^2$ matrix of zeros and $H_0 = (S^{(0)}\beta)/((1-\beta)/N)$, $H_1 = (S^{(1)}(1-\alpha))/(\alpha/N)$, $G_0 = (B^{(0)}\beta)/((1-\beta)/N)$, and $G_1 = (B^{(1)}(1-\alpha))/(\alpha/N)$.

We omit the detailed procedures of deriving the transition probabilities in T due to space limitations. Provided that the transition probability matrix T is known, it is a routine matter to derive the steady-state equations for the Markov chain, and solving the equations to obtain the steady-state probability vector. The steady-state probability vector of the Markov chain Z is given by $\Pi = [\pi_{(0,g)}, \pi_{(1,g)}, \ldots, \pi_{(l,g)}, \ldots, \pi_{(b_i,g)}]$, where every element $\pi_{(l,g)} = [\pi_{(l,g,1,1)}, \pi_{(l,g,1,2)}, \ldots, \pi_{(l,g,N,N)}]$, $l > 0$, is a row vector of size N^2, except $\pi_{(0,g)}$ which is a scalar. The steady-state probabilities of the states in level l are denoted by $\pi_l = [\pi_{(l,0)}, \pi_{(l,1)}]$, where $\pi_{(l,0)}$ and $\pi_{(l,1)}$ are the probability vectors with the traffic source at input i in stage 0 and 1. Furthermore, we let $\overline{\pi_{(l,g)}} = \pi_{(l,g)} e_1$ and $\overline{\pi_0} = \pi_{(0,0)} + \pi_{(0,1)}$.

C. Solving the Markov Chain

We now derive the equations for computing the blocking probability $P_{\text{blo},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$ and the success probability $P_{\text{suc},W_t(w_i',w_o')|W_{t-1}(g,w_i,w_o)}$. The transition of the state of the virtual HOL input/output queues from state (w_i, w_o) to state (w_i', w_o') is a two-step process as illustrated in Fig. 2.

1) First, we account for the number (k_i, k_o) of the newly arriving HOL cells to the virtual HOL input/output queues.
2) Then, we consider the transition from the intermediate state (h_i, h_o) to the final state (w_i', w_o') after applying the PIM algorithm.

To utilize the concept of a *tagged queue*, the condition of independent and identical components must be satisfied. Studies indicate that such an assumption is reasonable for the moderate- or large-sized input-queued switches under *i.i.d* traffic [5], [6]. Here, we make the same assumption, that is, when a cell arrives

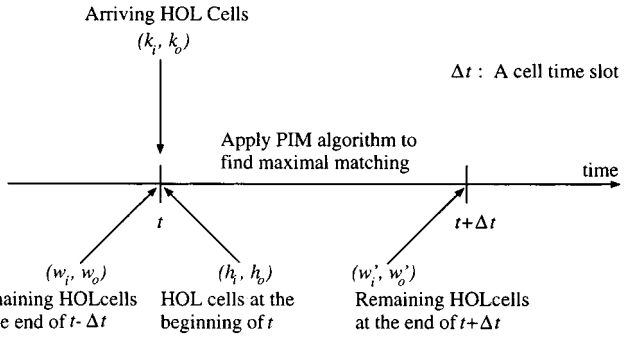

Fig. 2. Transition of the virtual HOL queues.

at an empty queue $Q(i,j)$, it will automatically observe another jth queue being empty with Bernoulli probability p_0 and another queue in input i being empty with Bernoulli probability $\overline{\pi_0}$. The introduction of p_0 plays an essential role in the solution of the Markov chain Z. However, the difficulty is that p_0 cannot be directly derived from the known system parameters, such as the switch size, buffer size, and traffic load. Instead of assuming p_0 as a known parameter, we use an iterative method to obtain p_0 from the known system parameters [4]. Equation (1) gives the equation on which the iterative computation is carried out

$$p_0 = \left(1 - \frac{1-\beta}{N}\right)\pi_{(0,0)} + \left(1 - \frac{\alpha}{N}\right)\pi_{(0,1)}. \tag{1}$$

Given p_0, the formula for the probability of the *virtual HOL queue*'s transition from (h_i, h_o) to (w_i, w_o) can be derived with some effort [5], [6]. Consequently, the steady-state probabilities of Z are given by

$$\pi_1 C_0 + \pi_0 C_1 = \pi_0 \tag{2}$$

$$\pi_0 \left([1,1]^T + \sum_{i=1}^{b_i} \prod_{j=1}^{i} \alpha_j e\right) = 1 \tag{3}$$

$$\pi_i = \pi_0 \prod_{j=1}^{i} \alpha_j, \qquad \text{for } 1 \le i \le b_i \tag{4}$$

where α_i is given as

$$\begin{cases} A_2(I - D_1)^{-1}, & \text{for } i = b_i \\ A_2\left(I - A_1 - \alpha_{b_i} D_0\right)^{-1}, & \text{for } i = b_i - 1 \\ A_2(I - A_1 - \alpha_{i+1} A_0)^{-1}, & \text{for } i \in [2, b_i - 2] \\ C_2(I - A_1 - \alpha_2 A_0)^{-1}, & \text{for } i = 1. \end{cases} \tag{5}$$

The elements of matrices in (5) are functions of $\pi_{(0,0)}$, $\pi_{(0,1)}$, $\overline{\pi_{(1,0)}}$, and $\overline{\pi_{(1,1)}}$. This naturally suggests an iterative solution [3], [5], [6]. Initially, both $\pi_{(0,0)}$ and $\pi_{(0,1)}$ are set to be $0.5(1-\lambda)$, which corresponds to the case that there is no new arriving cell at the *tagged input queue* at the beginning of a time slot, and both $\overline{\pi_{(1,0)}}$ and $\overline{\pi_{(1,1)}}$ are approximated by $0.5(1 - N^{-1})\lambda(\pi_{(0,0)} + \pi_{(0,1)})$. Then, the next $\pi_{(0,0)}$ and $\pi_{(0,1)}$ are obtained by finding the root for (2) and (3). Consequently, the new π_1 is computed by (4). As observed from our experiments, the converging rate is quite high and

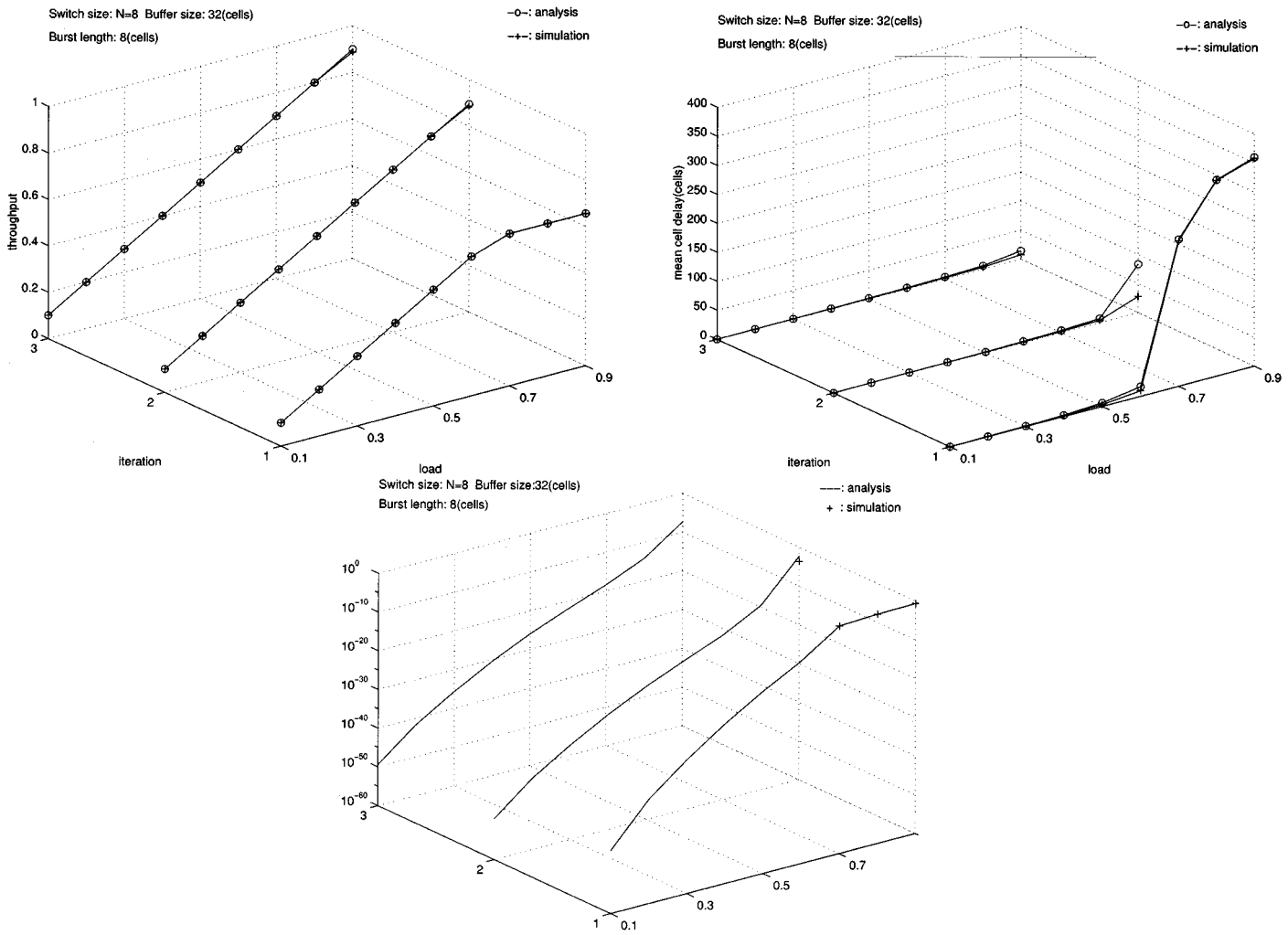

Fig. 3. The throughput, mean cell delay and mean cell loss probability of an 8×8 PIM switch with the buffer size $b_i = 32$, as a function of offered loads with the mean burst length $\tau = 8$.

an accuracy of 10^{-5} for π_0 can be attained within 15 iterative computations in most of cases.

D. Computing the Performance Metrics

The performance parameters of throughput ρ, mean queue length $\bar{Q}$, mean cell delay $\bar{D}$, and mean cell loss probability P_{loss} can be expressed in term of the steady-state probabilities given as follows:

$$\rho = \left[\pi_{(0,0)}\left(\lambda_0 - B_0^{(0)}e_1\right) + \pi_{(0,1)}\left(\lambda_1 - B_0^{(1)}e_1\right)\right] + \sum_{l=1}^{b_i}\left[\pi_{(l,0)}(S^{(0)} + S'^{(0)}) + \pi_{(l,1)}(S^{(1)} + S'^{(1)})\right]e_1$$

$$\bar{Q} = \sum_{l=1}^{b_i} l\pi_l e$$

$$\bar{D} = \frac{\bar{Q}}{\rho}$$

$$P_{\text{loss}} = (\pi_{(b_i,0)}\lambda_0 + \pi_{(b_i,1)}\lambda_1)\frac{e_1}{\lambda}$$

III. Numerical Results

Both mathematical analysis and simulation results are presented in this section in order to investigate the accuracy of the above queueing model and to evaluate the performance of the PIM switch under bursty traffic. Fig. 3(a)–(c) shows the switch throughput, mean cell delay, and mean cell loss probability as a function of the offered load with a mean burst length of eight cells for an 8×8 PIM switch with various PIM scheduling iteration numbers 1, 2, and 3, respectively. In Fig. 3(c), the simulation results for the mean cell loss probability are given only for the case when the switches are overloaded. This is because the simulation results are meaningful only in these cases. Simulation cannot be used to estimate very low cell loss probability values with good accuracy. It can be seen from these figures that the mathematical analysis results closely approximate the simulation results. Noticeable deviations between the analysis and simulation appear only in cases where the switch with multiple iterations is *overloaded*.

From Figs. 4 and 3(b), we can see that when the number of PIM scheduling iterations is bigger than one, the mean delay increases slowly with the traffic load as compared with just one iteration. For a single iteration PIM scheduling, the mean cell

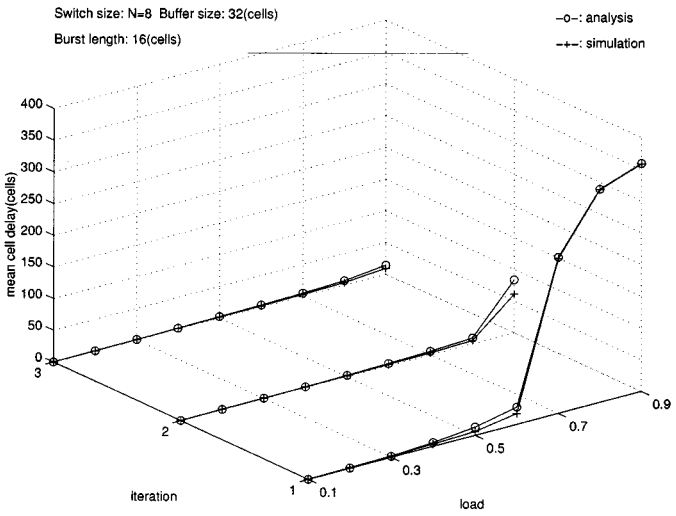

Fig. 4. The mean cell delay of an 8×8 PIM switch with the buffer size $b_i = 32$, as a function of offered loads with the mean burst length of $\tau = 16$.

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 two and three iterations PIM, this overloaded traffic point is about 0.8. This trend can also be observed in Fig. 3(a) and (c). As a result, it is advised to iterate the PIM scheduling algorithm more than once to get a good performance using these switches under bursty traffic.

IV. Conclusion

The presented analysis provides a unifying framework to build queueing models for PIM switches under *i.i.d* traffic. In addition, the queueing model can be extended using the same technique to the situation where complicated bursty traffic with more states are inserted to the switch. Recalling our previous work in [5] and [6], we conclude that our suggested queueing model works well not only in the case of the *i.i.d* Bernoulli traffic, but also in the case of the *i.i.d* burst traffic where the cells' arrival process is correlated in a long term.

References

[1] A. Adas, "Traffic models in broadband networks," *IEEE Commun. Mag.*, vol. 35, pp. 82–89, July 1997.

[2] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed switch scheduling for local-area networks," *ACM Trans. Comput. Syst.*, vol. 11, no. 4, pp. 319–352, Nov. 1993.

[3] Y. C. Jung and C. K. Un, "Performance analysis of packet switches with input and output buffers," *Comput. Netw. ISDN Syst.*, vol. 26, no. 12, pp. 1559–1580, Sept. 1994.

[4] M. K. Mehmet-Ali, M. Youssefi, and H. T. Nguyen, "The performance analysis and implementation of an input access scheme in a high-speed packet switch," *IEEE Trans. Commun.*, vol. 42, pp. 3189–3199, Dec. 1994.

[5] G. Nong, J. K. Muppala, and M. Hamdi, "A performance model for ATM switches with multiple input queues," in *Proc. IC3N'97*, Las Vegas, Sept. 1997, pp. 222–227.

[6] ——, "Analysis of nonblocking ATM switches with multiple input queues," *IEEE/ACM Trans. Networking*, vol. 7, pp. 60–74, Feb. 1999.