ELSEVIER

Computer Communications 20 (1997) 1146–1159

computer communications

Performance of ATM switch fabric using cross-point buffers[1]

Bin Zhou[a,*], M. Atiquzzaman[b]

[a]*Motorola Australia Software Centre, 2 Second Avenue, Technology Park, Adelaide, South Australia 5095, Australia*
[b]*Department of Electrical and Computer Systems Engineering, Monash University, Clayton 3168, Melbourne, Australia*

Received 3 April 1996; accepted 13 November 1996

Abstract

Multistage switches consisting of a number of stages of small switching elements (SE) are often used in ATM switch fabrics. An ATM switch fabric performs the functions of switching and buffering of cells. The location of buffers inside a fabric is critical in the performance of the switch. Output buffered SEs have a higher throughput than input buffered SEs. Moreover, output buffered SEs do not suffer from buffer hogging as found in shared buffered SEs operating in a nonuniform traffic environment. However, output-buffered SEs require internal speedup, thereby requiring high-speed buffers and complex buffer management hardware. Crosspoint buffering, however, enjoys the advantages of output buffering without requiring an internal speedup of its buffers. The objective of this paper is to study the performance of a crosspoint buffered ATM switch operating under uniform and nonuniform traffic patterns. The performance of the crosspoint buffered switch is compared with that of the output buffered switch. Analytical models based on the Markov chains are developed to facilitate the study. The results from the model are validated using simulation. It is shown that the model provides very accurate results. © 1997 Elsevier Science B.V.

Keywords: ATM; Multistage switches; Buffer schemes; Markov chains; Performance

1. Introduction

The Asynchronous Transfer Mode (ATM) has been accepted as the transport mechanism in future broadband ISDN (B-ISDN). ATM is based on statistical multiplexing and can flexibly and efficiently support a wide range of services such as telephone speech, data, teleconferencing, entertainment video, etc. Multistage switches (MS) [1] have been proposed for ATM switching fabrics. An MS consists of a number of stages of SE connected together by some interconnection function. Buffers are provided inside the SEs to queue cells that lose contention during routing conflicts. The buffers can be located at inputs, outputs or crosspoints of SEs. They can also be fully shared by the inputs and outputs. Due to the head of line blocking, the maximum normalized throughput of an input-buffered switch under a uniform traffic pattern was shown to be 0.75 for a 2 × 2 SE [2,3]. Shared buffer switches [4,5] have a very high buffer utilization, but suffer from a phenomenon called *buffer hogging* in the case of a nonuniform traffic pattern. Moreover, buffer management in shared buffer switches is more complex and difficult to implement than input or output buffered switches. The high-performance switch architectures, in general, require that the buffers be placed at the outputs [6,7]. When compared with input buffering, output buffering provides increased throughput by getting rid of head-of-line blocking. However, the buffer management in output buffering is more complex and difficult to implement. The capability of inserting more than one cell into a buffer during a cycle adds considerable complexity and may increase the write time of the buffer, which may, in turn increase the cycle time of the switch.

Crosspoint buffering has been proposed [8,9] to eliminate the head of line blocking of input buffered switches and the internal speedup requirement of output buffered switches. Fig. 1 shows a 2 × 2 crossbar SE which has buffers preceded by address filters [1] at each crosspoint. A cell can pass a filter if the cell is destined to the output to which the address filter is connected. This reduces the complexity of buffer management, and the speed of the buffers can be equal to the speed of the input/output ports.

Although there are a number of analytical models for performance evaluation of MSs with input and output buffers [6,7,10–14], only a few studies for cross-points

* Corresponding author. Tel.: +61 8 203 3607; fax: +61 8 203 3501; e-mail: A12415@email.mot.com

[1] A preliminary version of this paper was presented at the APCC'95: 1995 Asia-Pacific Conference on Communications, June 13–16, 1995 Osaka, Japan.

PII S0140-3664(97)00066-2

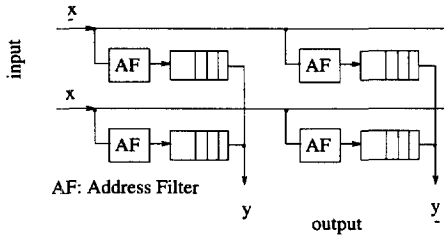


Fig. 1. A 2 × 2 crossbar switch with FIFO buffers at cross-points.

buffered MSs have been carried out. Input, output, and crosspoint buffering schemes for 2 × 2 crossbar switch architectures for the unbounded queue size and single queue size were analyzed in [8]. A bus matrix switch, with 16 Kb of FIFO buffer at each cross-point was described in [15]. The design of the switch was restricted to 2 × 2 SEs. A crossbar switch with FIFO buffers at each cross-point was also discussed in [16] under the name of Butterfly switch. Kato et al. [17] described the implementation detail of a crossbar switch having a dual port RAM at each cross-point. Goli [9] analyzed the performance of a cross-point buffered switch under a uniform traffic. The performance of the switch under a bursty traffic was studied using simulation. Gupta [18] developed simulation programs to investigate the performance of a nonblocking switch having input buffers and a limited amount of cross-points buffers within the switch fabric.

Simulation studies of cross-point buffered switches with a large number of stages and a considerable amount of buffer at each crosspoint are very costly in terms of computation time. The existing analytical models [8,9] do not produce accurate results, especially when the offered traffic load at the switch inputs is high. The inaccuracies are mainly due to simplifying assumptions regarding the blocked cells in the switch. To our knowledge, no research reported in the literature has accurately analyzed finite cross-point buffered MS in the presence of nonuniform traffic patterns.

The specific interest of this paper is to study the performance of the finite cross-point buffered switch under *uniform* and *hot spot* traffic patterns. Three different scheduling policies (viz., random selection (RS), new cell selection (NS), and blocked cell selection (BS)) are investigated to select a cell for transmission from a complementary buffer to a given output. The modeling results show that the BS policy results in the best performance under uniform traffic. The performance is almost the same under the RS and NS policies. When the hot spot probability is small, the NS policy performs the best, and the RS performs better than the BS policy. When the hot spot traffic is heavy, the performance of the three policies is identical.

The paper is organized as follows. Section 2 describes the output and crosspoint buffering schemes, followed by the modeling assumptions of the switch. The proposed model is developed in Section 3. Section 4 discusses the performance of the crosspoint buffered switch and compares with that of the output buffered switch. Finally, concluding remarks are given in Section 5.

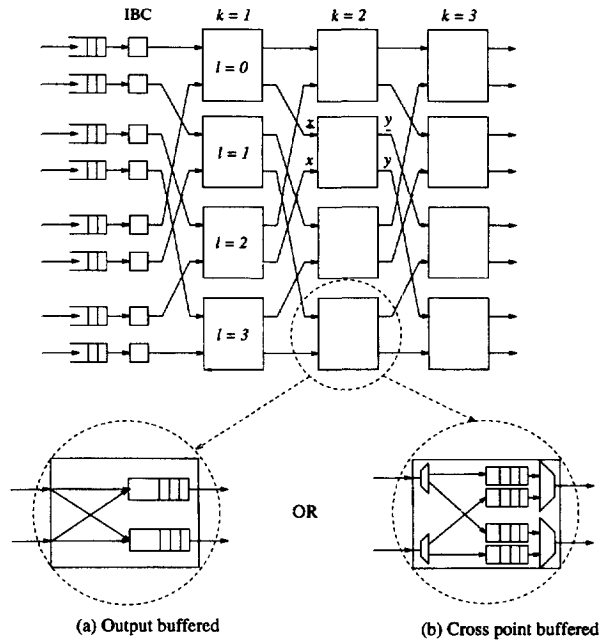


Fig. 2. An 8 × 8 internally buffered ATM switch.

2. Switch architecture

An MS switch (Fig. 2) connects N inputs to N outputs using $s = \log_2 N$ stages of $N/2$ switching elements per stage. Fig. 2 shows an 8 × 8 MS based on the Omega interconnection network. In this study, it is assumed that each SE is a 2 × 2 crossbar allowing any input to be connected to any one of its output links. Each SE has a finite number of buffers, which can be placed at different locations in the SE as described below.

2.1. Buffering strategies

Fig. 2 shows two different buffering strategies in the SEs. In the output buffering (shown in Fig. 2(a)), buffers are located at each output of an SE. Each buffer is capable of accepting two cells simultaneously in a clock cycle. Thus, an internal speedup of buffers is required to queue two cells during the same clock cycle. In *cross-point buffering* (Fig. 2(b)), separate buffers are associated with each cross point of the SE. In addition to the buffers, a crosspoint buffered SE consists of a cell distributor at each inlet and a cell multiplexer at each outlet. Crosspoint buffering has the advantage of requiring only one read and one write operation on a buffer during a clock cycle. This configuration uses more instance of a simpler type of buffer to approximate the performance of the multi-output buffers. Cells arriving at an input port are queued in the appropriate buffer according to the destination tag. A cell leaves an outlet of an SE whenever any or both of the two associated buffers has cell and the next stage can accept it; if both the buffers have cells, a routing conflict will occur. Scheduling policies to resolve such routing conflicts are discussed in Section 2.2.

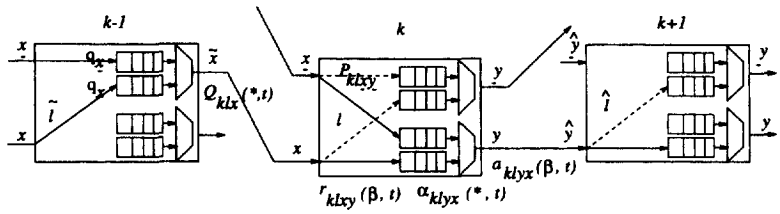


Fig. 3. The illustration of $Q_{klx}(\cdot,t)$, $\gamma_{klxy}(\beta,t)$, and $a_{klyx}(\beta,t)$.

2.2. Modeling assumptions

The following assumptions are made regarding the input traffic pattern and the operation of the switch.

1. There are $N = 2^s$ inputs and N outputs in the switch. Each input of the switch has an Input Buffer Controller (IBC) of size f.
2. The switch is operated *synchronously*. For modeling purposes, a clock cycle is split into two phases [7]:

- In *phase 1*, the buffers which contain cells send a request to the next stage indicating that a cell is available for transmission. The cell is forwarded, if there are no collisions with other cells and if the destination buffer at the next stage is ready to accept a cell.
- Depending on the availability of buffer space, a cell may be accepted from a preceding stage during *phase 2*.

3. A *backpressure* mechanism ensures that no cells are lost within the switch.
4. Cell arrival process at each input of the switch is a *Bernoulli* process.
5. The offered *traffic load* to each input of the switch is the same.
6. There is no *blocking* at the output links of the switch. The output links are therefore at least as fast as the internal links.
7. Since a cell spends at least one clock cycle in a buffer even when there is no waiting, the minimum possible mean cell delay of a cell is equal to $s + 1$, where s is the number of stages. It includes the mean cell delay at the IBC buffer.
8. For a uniform traffic pattern, an incoming cell is equally likely to be directed to any output of the switch. For a hot spot traffic pattern, the probability that an incoming cell is directed to a non-hot or a hot output are $(1 - h)/N$ or $h + (1 - h)/N$ respectively, where h is defined to be the hot spot probability.

For hot-spot traffic pattern, the cells directed for the hot output is defined as *hot cells*. The internal links of the switch which carry hot cells are called *hot links* (indicated by bold line in Fig. 5). Buffers carrying hot cells are called *hot buffers*. All SEs connected to hot links are called *hot SEs*.

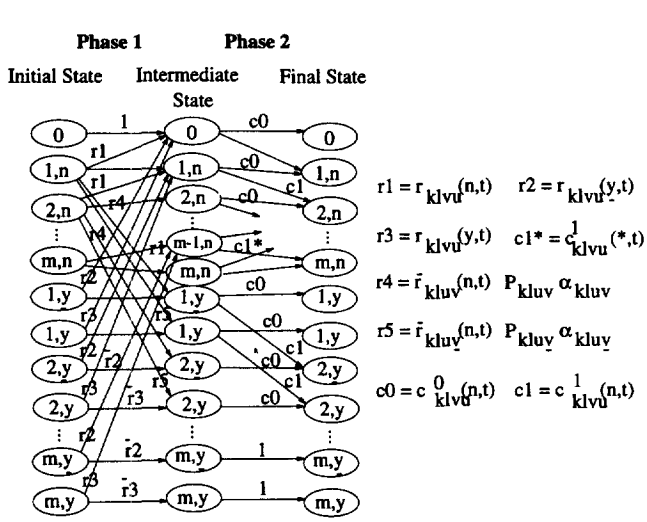


Fig. 4. Markov chain state transition diagram for internal stage buffers.

3. The model

Fig. 2 shows how the SEs are specified by stage number k, SE number l within a stage, input ports x and $\underline{x}$, and output ports y and $\underline{y}$. Fig. 3 shows three SEs in successive stages of the switch. The input and output ports of the lth SE at stage k are denoted by klx, $kl\underline{x}$, kly and $kl\underline{y}$. Let $(k+1)\hat{l}\hat{y}$ denote the input port of the SE at the $(k + 1)$th stage which is fed by output port kly at the kth stage. Also, let $(k - 1)\tilde{l}\tilde{x}$ denote the output port of the SE at the $(k - 1)$th stage which feeds input port klx at the kth stage. Buffers $q_{\underline{x}}$ and q_x at the output port of an SE refer to buffers connected to inputs $\underline{x}$ and x respectively of the SE. The $q_{\underline{x}}$ and q_x are referred as *complementary* buffers.

A *new* cell in a buffer is defined as a cell which has never attempted to go to the next stage, or has made an attempt but

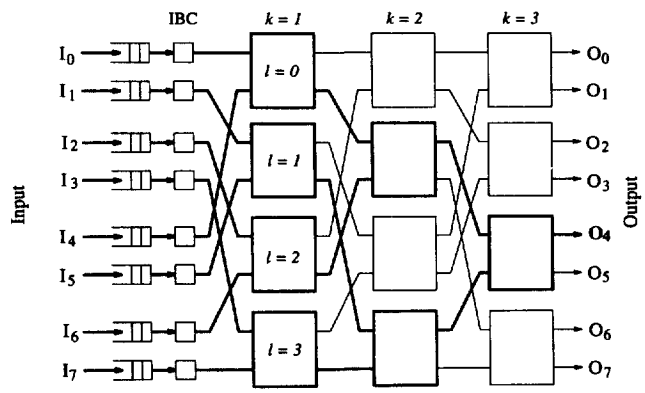


Fig. 5. An 8 × 8 ATM switch under hot spot traffic pattern.

has not won a routing conflict with a cell in the complementary buffer. For example, a cell which is behind other cells in a buffer is a new cell. A cell which has just moved to the head of a buffer, or has reached the head of the buffer in a previous cycle but has never won a routing conflict is a new cell. A *blocked* cell is defined as a cell which was not accepted by the SE in the next stage during a previous cycle. An example of a blocked cell is one which won a routing conflict but could not proceed forward because of a full buffer in the next stage. The following state variables are defined to describe the state of a buffer at the *end* of a clock cycle.

- State 0: the buffer is *empty*.
- State n: the cell at the head of the buffer contains a *new* cell.
- State $\underline{y}$: the cell at the head of the buffer contains a *blocked* cell which was unsuccessful in a previous routing attempt to the upper output of the SE at the next stage.
- State y: the cell at the head of the buffer contains a *blocked* cell which was unsuccessful in a previous routing attempt to the *lower* output of the SE at the next stage.
- State (i,β): the buffer contains i, $0 \le i \le m$, cells and the cell at the head of the buffer is in state β, $\beta \in \{n,\underline{y}, y\}$.

Cells are forwarded from a buffer to the next stage during phase 1 and received from the preceding stage during phase 2 of a clock cycle. The state of a buffer after phase 1 of a cycle will be called the intermediate state of the buffer. Note that phase 1 (2) in an SE in stage k coincides with phase 2 (1) with SE's in stage $k-1$ ($k+1$). The following variables are defined to describe the proposed model.

m = the buffer size.

λ = the probability that a cell comes to the IBC at the beginning of a cycle.

$\bar{z}$ = $1 - z$ for any variable z.

$B^0_{klvu}(0,t)$ = the probability that the buffer at klv, $v \in \{y,\underline{y}\}$, connected to input u, $u \in \{x,\underline{x}\}$, is in empty state at the end of the tth cycle.

$B_{klvu}{}^i(\beta,t)$ = the probability that the buffer at klv, $v \in \{\underline{y}, y\}$, connected to input u, $u \in \{x,\underline{x}\}$, contains i, $0 \le i \le m$, cells and is in state β, $\beta \in \{n,\underline{y}, y\}$, at the end of the tth cycle.

$\tilde{B}^0_{klvu}(0,t)$ = the probability that the buffer at klv, $v \in \{y,\underline{y}\}$, connected to input u, $u \in \{x,\underline{x}\}$, is in empty state at the end of phase 1 of the tth cycle.

$\tilde{B}_{klvu}{}^i(\beta,t)$ = the probability that the buffer at klv, $v \in \{\underline{y},y\}$, connected to input u, $u \in \{x,\underline{x}\}$, contains i, $0 \le i \le m$, cells and is in state β, $\beta \in \{n, \underline{y}, y\}$, at the end of phase 1 of the tth cycle.

P_{kluv} = Probability that a new cell, ready to come to klu, $u \in \{x,\underline{x}\}$, during cycle t, is destined to klv, $v \in \{\underline{y},y\}$.

$Q_{klu}(\cdot,t)$ = the probability that a cell (new or/and blocked) is ready to come to port klu, $u \in \{x,\underline{x}\}$, during cycle t.

$Q_{klu}(n,t)$ = the probability that a new cell is ready to come to port klu, $u \in \{x,\underline{x}\}$, during cycle t.

$Q_{klu}(v,t)$ = the probability that a cell which is blocked for port klv, $v \in \{\underline{y},y\}$, is ready to come to port klu, $u \in \{x,\underline{x}\}$, during cycle t.

$C^i_{klvu}(n,t)$ = the probability that i, $0 \le i \le 1$, cells which are in new state connected to input u, $u \in \{x,\underline{x}\}$, during cycle t.

$C_{klvu}{}^i(\cdot,t)$ = the probability that i, $0 \le i \le 1$, cells (new or/and blocked) are ready to come to the buffer at klv, $v \in \{\underline{y},y\}$, connected to input u, $u \in \{x,\underline{x}\}$, during cycle t.

The routing probability of a cell is defined as the probability that the cell can be routed to the next stage. The different routing probabilities are defined below.

$\gamma_{kluv}(\beta,t)$ = the probability that a cell from port klu, $u \in \{x,\underline{x}\}$, advances to the buffer at klv, $v \in \{\underline{y},y\}$, during phase 2 of clock cycle t, given that a cell in state β, $\beta \in \{n,\underline{y},y\}$, destined to klv is ready to come to klu during cycle t.

$a_{klvu}(\beta,t)$ = the probability that a cell in the buffer at klv, $v \in \{\underline{y},y\}$ connected to input u, $u \in \{x,\underline{x}\}$, is able to move forward during phase 1 of the tth cycle, given that the cell is in state β, $\beta \in \{n,\underline{y},y\}$.

$\alpha_{klvu}(n,t)$ = the probability that a new cell in the buffer at klv, $v \in \{y,\underline{y}\}$, connected to input u, $u \in \{x,\underline{x}\}$, is selected by the SE under the scheduling policy.

$\alpha_{klvu}(b,t)$ = the probability that a blocked cell, $b \in \{y,\underline{y}\}$, in the buffer at klv, $v \in \{y,\underline{y}\}$, connected to u, $u \in \{x,\underline{x}\}$, is selected by the SE under the scheduling policy.

$\mu_{klvu}(t)$ = the probability that a cell (new and/or blocked) leaves the buffer at klv, $v \in \{y,\underline{y}\}$, connected to u, $u \in \{x,\underline{x}\}$, during cycle t.

Each output link of an SE is connected to its buffers through a multiplexer. At most, one cell can be transmitted to an output of an SE per cycle. Several scheduling policies are possible for selecting a cell from the buffers belonging to output port. The following three scheduling policies are considered in this study.

- *Random selection* (RS): a multiplexer randomly selects a cell from one of the buffers connected to the multiplexer.
- *New cell selection* (NS): a multiplexer selects a cell from the buffer which has a new cell at the head of the queue. If there is no such cell, it selects a cell based on the RS policy. If there are more than one new cells, one is selected at random from the new ones.
- *Blocked cell selection* (BS): a multiplexer selects a cell from the buffer which has a blocked cell at the head of the queue. If there is no such cell, it selects a cell on RS basis.

The probability that a new cell is ready to come to input port klu, $1 \le k \le s$, $u \in \{\underline{x},x\}$, during cycle t is given by the probability that at least one of the upstream buffers is in a new state and the new cell is selected by the multiplexor. It is given by:

$$Q_{klu}(n,t) = \sum_{i=1}^{m} \left\{ B^i_{(k-1)\tilde{l}\tilde{u}\underline{x}}(n,t-1) \right\} \alpha_{(k-1)\tilde{l}\tilde{u}x}(n,t) + \sum_{i=1}^{m} \left\{ B^i_{(k-1)\tilde{l}\tilde{u}x}(n,t-1) \right\} \alpha_{(k-1)\tilde{l}\tilde{u}x}(n,t). \quad (1)$$

The probability that a blocked cell is ready to come to input port klu, $1 \le k \le s$, $u \in \{\underline{x},x\}$, during cycle t can be computed by the probability that the cell at the head of queue in the preceding stage is in a blocked state and the cell is selected or the complementary buffer is in a blocked

state and is selected by the multiplexor. It is therefore given by:

$$Q_{klu}(v,t) = \sum_{i=1}^{m}\left\{B^i_{(k-1)\bar{l}\bar{u}\underline{x}}(v,t-1)\right\}\alpha_{(k-1)\bar{l}\bar{u}x}(v,t) + \sum_{i=1}^{m}\left\{B^i_{(k-1)\bar{l}\bar{u}x}(v,t-1)\right\}\alpha_{(k-1)\bar{l}\bar{u}x}(v,t), \quad v \in \{\underline{y},y\}. \tag{2}$$

The probability that a cell is ready (or not ready) to come to the buffer at klv, $v \in \{\underline{y},y\}$, which is connected to input u, $u \in \{x,\underline{x}\}$, is given by

$$C^0_{klvu}(n,t) = 1 - Q_{klu}(n,t) + Q_{klu}(n,t)P_{klu\underline{v}}, \tag{3}$$

$$C^1_{klvu}(n,t) = Q_{klu}(n,t)P_{kluv}, \tag{4}$$

$$C^0_{klvu}(\cdot,t) = 1 - Q_{klu}(\cdot,t) + Q_{klu}(\underline{v},t) + Q_{klu}(n,t)P_{klu\underline{v}}, \tag{5}$$

$$C^1_{klvu}(\cdot,t) = Q_{klu}(v,t) + Q_{klu}(n,t)P_{kluv}. \tag{6}$$

For example, Eq. (5) is obtained from the probability that a new cell is not ready to come to input port klu, or a new cell is ready to come to input port klu but is destined to buffer $kl\underline{v}u$.

In the case of a routing conflict between cells at the two complementary buffers of an output, one of the three previously mentioned scheduling policies are used to select a cell. The probability that a cell is able pass an SE depends on the scheduling policy used and can be expressed as follows:

Random selection:

A cell in the buffer at $klvu$ can be selected by the multiplexor if the complementary buffer is empty, or it is not empty but the cell in the buffer at $klvu$ wins the conflict.

$$\alpha_{klvu}(n,t) = 0.5\sum_{i=1}^{m}\left\{B^i_{klv\underline{u}}(y,t) + B^i_{klv\underline{u}}(\underline{y},t) + B^i_{klv\underline{u}}(n,t)\right\} + B^0_{klv\underline{u}}(0,t), \tag{7}$$

$$\alpha_{klvu}(b,t) = 0.5\sum_{i=1}^{m}\left\{B^i_{klv\underline{u}}(y,t) + B^i_{klv\underline{u}}(\underline{y},t) + B^i_{klv\underline{u}}(n,t)\right\} + B^0_{klv\underline{u}}(0,t). \tag{8}$$

New cell selection:

A new cell in buffer at $klvu$ can be selected by the multiplexor if the complementary buffer is empty, or it is not empty but the cell at the head of queue is in a blocked state, or the cell at the head of queue is new and the cell in the buffer at $klvu$ wins the conflict. A blocked cell in the buffer at $klvu$ will be selected if the complementary buffer is empty, or it is not empty but the cell at the head of queue is in a blocked state and the cell in the buffer at $klvu$ wins the conflict.

$$\alpha_{klvu}(n,t) = \sum_{i=1}^{m}\left\{B^i_{klv\underline{u}}(y,t) + B^i_{klv\underline{u}}(\underline{y},t) + 0.5B^i_{klv\underline{u}}(n,t)\right\} + B^0_{klv\underline{u}}(0,t), \tag{9}$$

$$\alpha_{klvu}(b,t) = 0.5\sum_{i=1}^{m}\left\{B^i_{klv\underline{u}}(y,t) + B^i_{klv\underline{u}}(\underline{y},t)\right\} + B^0_{klv\underline{u}}(0,t), \quad b \in \{\underline{y},y\}. \tag{10}$$

Blocked cell selection:

$$\alpha_{klvu}(n,t) = 0.5\sum_{i=1}^{m}\left\{B^i_{klv\underline{u}}(n,t)\right\} + B^0_{klv\underline{u}}(0,t), \tag{11}$$

$$\alpha_{klvu}(b,t) = 0.5\sum_{i=1}^{m}\left\{B^i_{klv\underline{u}}(y,t) + B^i_{klv\underline{u}}(\underline{y},t)\right\} + \sum_{i=1}^{m}\left\{B^i_{klv\underline{u}}(n,t)\right\} + B^0_{klv\underline{u}}(0,t), \quad b \in \{\underline{y},y\}. \tag{12}$$

A new cell can be routed from input port klu to the buffer at $klvu$ if the buffer at $klvu$ has at least one space, or the buffer is full and a cell (new or blocked) can leave the buffer during the current cycle. The routing probabilities of a cell are therefore given by:

$$\gamma_{kluv}(n,t) = 1 - \left\{B^m_{kluv}(n,t-1) + B^m_{kluv}(y,t-1) + B^m_{klvu}(\underline{y},t-1)\right\} + B^m_{klvu}(n,t-1)a_{klvu}(n,t) + B^m_{klvu}(y,t-1)a_{klvu}(y,t) + B^m_{klvu}(\underline{y},t-1)a_{klvu}(\underline{y},t), \tag{13}$$

$$a_{klvu}(n,t) = \alpha_{klvu}(n,t)\left\{P_{(k+1)\hat{l}\hat{v}y}\gamma_{(k+1)\hat{l}\hat{v}y}(n,t) + P_{(k+1)\hat{l}\hat{v}\underline{y}}\gamma_{(k+1)\hat{l}\hat{v}\underline{y}}(n,t)\right\}. \tag{14}$$

When a cell in stage $k-1$ is in state $(i,\underline{y})$ or (i,y) at the end of cycle t, the cell's destination buffer in stage k after phase 1 of cycle $t+1$ will be either full or will have one space available. If a cell was blocked at stage $k-1$ at the end of cycle $t-1$, the cell's destination buffer at stage k must be full (new or blocked) at the end of cycle $t-1$. Under this condition:

- The cell's destination buffer in stage k at the end of cycle $t-1$ can be in state (m,n) only if it was in state (m,n) at cycle $t-2$ and a cell did not leave the buffer during phase 1 of cycle $t-1$ because a cell is selected from another buffer at the same SE in stage k.

- The cell's destination buffer in stage k at the end of cycle $t-1$ can be in state (m,y) or $(m,\underline{y})$ under one of the following conditions:

1. The buffer was full at the end of cycle $t-2$ and a cell is selected in this buffer but no cell left the buffer during phase 2 of cycle $t-1$.
2. The buffer had one space at the end of cycle $t-2$, no cell left during cycle $t-1$, and one cell came during cycle $t-1$.

Under the above circumstances, the intermediate state of the destination buffer in stage k during cycle t will be either full if it is blocked or one space available if a cell left.

$B'^{m}_{klv}(\beta,t)$, $\beta \in \{n,y,\underline{y}\}$, is defined as the probability of the buffer at klv being full and in state β at the end of cycle t, given that a cell at stage $k-1$ is blocked for this buffer at the end of cycle t.

$$B'^{m}_{klvu} = B^{m}_{klvu}(n,t-2)\bar{\alpha}_{klvu}(n,t-1), \tag{15}$$

$$r_{klvu}(v,t) = B'^{m}_{klvu}(n,t-1)a_{klvu}(n,t) + \{1 - B'^{m}_{klvu}(n,t-1)\} \times \{P_{(k+1)\hat{l}\hat{v}y}a_{klvu}(y,t) + P_{(k+1)\hat{l}\hat{v}\underline{y}}(y,t)\}, \tag{16}$$

$$a_{klvu}(y,t) = \alpha_{klvu}(y,t)r_{(k+1)\hat{l}\hat{v}y}(y,t), \tag{17}$$

$$a_{klvu}(\underline{y},t) = \alpha_{klvu}(\underline{y},t)r_{(k+1)\hat{l}\hat{v}\underline{y}}(\underline{y},t). \tag{18}$$

The above eqns (1)–(18) are valid for all the buffers in the switch and the IBC. However, those equations which depend on whether the buffers are at the IBC, the last stage or the internal stages of the switch are given below separately. We define all the stages other than the last stage and the stage containing the IBC as internal stages.

3.1. Internal stage buffers

Fig. 4 shows the state transition probabilities for a buffer in an internal stage. The intermediate state probabilities of the buffer at $klvu$, $1 \le k \le s-1$, are given by:

$$\tilde{B}^{0}_{klvu} = B^{0}_{klvu}(0,t-1) + B^{1}_{klvu}(n,t-1)a_{klvu}(n,t) + B^{1}_{klvu}(\underline{y},t-1)a_{klvu}(\underline{y},t) + B^{1}_{klvu}(y,t-1)a_{klvu}(y,t), \tag{19}$$

$$\tilde{B}^{i}_{klvui}(n,t) = B^{i+1}_{klvu}(n,t-1)a_{klvu}(n,t) + B^{i+1}_{klvu}(\underline{y},t-1)a_{klvu}(\underline{y},t) + B^{i+1}_{klvu}(y,t-1)a_{klvu}(y,t) + B^{i}_{klvu}(n,t-1)\bar{\alpha}_{klvu}(n,t), \quad 1 \le i \le m-1, \tag{20}$$

$$\tilde{B}^{m}_{klvu}(n,t) = B^{m}_{klvu}(n,t-1)\bar{\alpha}_{klvu}(n,t),$$

$$\tilde{B}^{i}_{klvu}(\underline{y},t) = B^{i}_{klvu}(n,t-1)\alpha_{klvu}(n,t)P_{(k+1)\hat{l}\hat{v}\underline{y}}\bar{\gamma}_{(k+1)\hat{l}\hat{v}\underline{y}}(n,t) + B^{i}_{klvu}(\underline{y},t-1)(\alpha_{klvu}(b,t)\bar{\gamma}_{(k+1)\hat{l}\hat{v}\underline{y}}(\underline{y},t) + \bar{\alpha}_{klvu}(b,t)), \quad 1 \le i \le m, \tag{22}$$

$$\tilde{B}^{i}_{klvu}(y,t) = B^{i}_{klvu}(n,t-1)\alpha_{klvu}(n,t)P_{(k+1)\hat{l}\hat{v}y}\bar{\gamma}_{(k+1)\hat{l}\hat{v}y}(n,t) + B^{i}_{klvu}(y,t-1)(\alpha_{klvu}(b,t)\bar{\gamma}_{(k+1)\hat{l}\hat{v}y}(y,t) + \bar{\alpha}_{klvu}(b,t)), \quad 1 \le i \le m. \tag{23}$$

For example, Eq. (19) states that the buffer at $klvu$ can be empty at the end of phase 1 of the tth cycle if it was empty at the $(t-1)$th cycle, or it contained a cell (new or blocked) at the $(t-1)$th cycle and the cell can leave during current clock cycle.

If the buffer at klv at the end of clock cycle t is not full, then there can be no blocked cell in its source buffers in stage $(k-1)$. Moreover, since at most one cell can leave a buffer during a cycle, the intermediate state of buffer klv during cycle t can not be $(m-2)$ if the buffer was full at the end of cycle $(t-1)$. Therefore, only new cells can come to buffer klv if the intermediate state of the buffer is more than $(m-1)$. The final state probabilities are therefore given by

$$B^{0}_{klvu}(0,t) = \tilde{B}^{0}_{klvu}(0,t)C^{0}_{klvu}(n,t), \tag{24}$$

$$B^{i}_{klvu}(n,t) = \tilde{B}^{i-1}_{klvu}(n,t)C^{1}_{klvu}(n,t) + \tilde{B}^{i}_{klvu}(n,t)C^{0}_{klvu}(n,t), \quad 1 \le i \le m-2, \tag{25}$$

$$B^{m-1}_{klvu}(n,t) = \tilde{B}^{m-2}_{klvu}(n,t) + \tilde{B}^{m-1}_{klvu}(n,t)C^{0}_{klvu}(\cdot,t), \tag{26}$$

$$B^{m}_{klvu}(n,t) = \tilde{B}^{m-1}_{klvu}(n,t)C^{1}_{klvu}(\cdot,t) + \tilde{B}^{m}_{klvu}(n,t), \tag{27}$$

$$B^{1}_{kvlu}(v,t) = \tilde{B}^{1}_{klvu}(v,t)C^{1}_{klvu}(n,t), \tag{28}$$

$$B^{i}_{klvu}(v,t) = \tilde{B}^{i-1}_{klvu}(v,t)C^{1}_{klvu}(n,t) + \tilde{B}^{i}_{klvu}(v,t)C^{0}_{klvu}(n,t) \quad 1 \le i \le m-2, \quad v\epsilon\{y,\underline{y}\}, \tag{29}$$

$$B^{m-1}_{klvu}(v,t) = \tilde{B}^{m-2}_{klvu}(v,t)C^{1}_{klvu}(n,t) + \tilde{B}^{m-1}_{klvu}(v,t)C^{0}_{klvu}(\cdot,t), \tag{30}$$

$$B^{m}_{klvu}(v,t) = \tilde{B}^{m-1}_{klvu}(v,t)C^{1}_{klvu}(\cdot,t) + \tilde{B}^{m}_{klvu}(v,t), \quad v\epsilon\{y,\underline{y}\}. \tag{31}$$

3.2. IBC buffers

The IBC, containing f buffers, can be modeled as follows.

Note that for the IBC buffers, $u = 0$, $k = 0$, and $v \in \{\underline{y}, y\}$ in the following equations:

$$\tilde{B}^0_{0lvu}(0,t) = B^0_{0lvu}(0,t-1) + B^1_{0lvu}(n,t-1)a_{0lvu}(n,t) + B^1_{0lvu}(\underline{y},t-1)a_{0lvu}(\underline{y},t) + B^1_{0lvu}(y,t-1)a_{0lvu}(y,t), \tag{32}$$

$$\tilde{B}^i_{0lvu}(n,t) = B^{i+1}_{0lvu}(n,t-1)a_{0lvu}(n,t) + B^{i+1}_{0lvu}(\underline{y},t-1)a_{0lvu}(\underline{y},t) + B^{i+1}_{0lvu}(y,t-1)a_{olvu}(y,t), \quad 1 \le i \le f-1, \tag{33}$$

$$\tilde{B}^i_{0lvu}(\underline{y},t) = B^i_{0lvu}(n,t-1)P_{1\hat{l}v_{\underline{y}}\bar{\gamma}1\hat{l}v_{\underline{y}}}(n,t) + B^i_{0lvu}(\underline{y},t-1)\bar{\gamma}_{0lvu}(\underline{y},t), \quad 1 \le i \le f, \tag{34}$$

$$\tilde{B}^i_{0lvu}(y,t) = B^i_{0lvu}(n,t-1)P_{1\hat{l}v_y}\bar{\gamma}_{1\hat{l}v_y}(n,t) + B^i_{0lvu}(y,t-1)\bar{\gamma}_{0lvu}(y,t), \quad 1 \le i \le f, \tag{35}$$

The final state probabilities at the end of a cycle are given by

$$B^0_{0lvu}(0,t) = \tilde{B}^0_{0lvu}(0,t)(1-\lambda), \tag{36}$$

$$B^i_{0lvu}(n,t) = \tilde{B}^{i-1}_{olvu}(n,t)\lambda + \tilde{B}^i_{0lvu}(n,t)(1-\lambda), \quad 1 \le i \le f-1, \tag{37}$$

$$B^f_{0lvu}(n,t) = \tilde{B}^{f-1}_{0lvu}(n,t)\lambda, \tag{38}$$

$$B^i_{0lvu}(v,t) = \tilde{B}^{i-1}_{olvu}(v,t)\lambda + \tilde{B}^i_{0lvu}(v,t)(1-\lambda), \quad 1 \le i \le f-1, \ v\epsilon\{y,\underline{y}\}, \tag{39}$$

$$B^f_{0lvu}(v,t) = \tilde{B}^{f-1}_{0lvu}(v,t)\lambda + \tilde{B}^f_{0lvu}(v,t), \quad v\epsilon\{y,\underline{y}\}. \tag{40}$$

3.3. Last stage buffers

Since there is no blocking at the output links of the switch, $a_{slvu}(y,t)$ and $a_{slvu}(\underline{y},t)$ are one. But a cell in the last stage can be stopped from advancing due to a conflict with a cell in the complementary buffer. The state probabilities are given as follows:

$$\tilde{B}^i_{slvu}(n,t) = \tilde{B}^{i+1}_{slvu}(n,t-1)\alpha_{slvu}(n,t) + B^i_{slvu}(n,t-1)\bar{\alpha}_{slvu}(n,t) \quad 0 \le i \le m-1, \tag{41}$$

$$\tilde{B}^m_{slvu}(n,t) = \tilde{B}^m_{slvu}(n,t-1)\bar{\alpha}_{slvu}(n,t), \tag{42}$$

$$B^0_{slvu}(0,t) = \tilde{B}^0_{slvu}(0,t)C^0_{slvu}(n,t), \tag{43}$$

$$B^i_{slvu}(n,t) = \tilde{B}^{i-1}_{slvu}(n,t)C^1_{slvu}(n,t) + \tilde{B}^i_{slvu}(n,t)C^0_{slvu}(n,t), \quad 1 \le i \le m-2, \tag{44}$$

$$B^{m-1}_{slvu}(n,t) = \tilde{B}^{m-2}_{slvu}(n,t)C^1_{slvu}(n,t) + \tilde{B}^{m-1}_{slvu}(n,t)C^0_{slvu}(\cdot,t), \tag{45}$$

$$B^m_{slvu}(n,t) = \tilde{B}^{m-1}_{slvu}(n,t)C^1_{slvu}(\cdot,t) + \tilde{B}^m_{slvu}(n,t). \tag{46}$$

3.4. Measures of performance

By solving the Markov chain equations described in the previous sections, we can find the normalized throughput (μ), mean cell delay (δ), cell loss probability (η) and the buffer occupancy at *klvu* (ξ_{klvu}) of the switch. The *throughput* of the switch at *slv* (μ_{slv}) is defined as the number of cells leaving the port *slv* per clock cycle. Since there is no blocked cell in the last stage, μ_{slv} is given by:

$$\mu_{slv} = 1 - \lim_{t\to\infty} B^0_{slu\underline{x}}(0,t)B^0_{slvx}(0,t). \tag{47}$$

The normalized throughput (μ) is given by

$$\mu = \frac{1}{N}\sum_{\text{all outputs}} \mu_{slv}. \tag{48}$$

Let $\delta_{klvu}(t)$ be the expected delay of a cell in the buffer *klvu* of an SE in stage k at cycle t. From Little's law, it can be expressed as

$$\delta_{klvu}(t) = \frac{\sum_{i=1}^m i\{B^i_{klvu}(n,t) + B^i_{klvu}(\underline{y},t) + B^i_{klvu}(y,t)\}}{\mu_{klvu}(t)}, \tag{49}$$

where $\mu_{klvu}(t)$ is given by

$$\mu_{klvu}(t) = \sum_{i=1}^m \{B^i_{klvu}(n,t)a_{klvu}(n,t) + B^i_{klvu}(\underline{y},t)a_{klvu}(y,t) + B^i_{klvu}(y,t)a_{klvu}(\underline{y},t)\}, \quad u\epsilon\{x,\underline{x}\}, \ 1 \le k \le s. \tag{50}$$

The *equivalent delay* is determined by considering the two buffers at output port *klv* as one buffer called the *equivalent buffer*. Let $\bar{\delta}_{klv}$ denote the equivalent delay induced by the equivalent buffer at *klv*. Since the buffer is work conserving and non-preemptive, the conservation law [19] implies that $\bar{\delta}_{klv}$ satisfies the following equation:

$$\bar{\delta}_{klv} = \lim_{t\to\infty} \frac{\mu_{klu\underline{x}}(t)\delta_{klu\underline{x}}(t) + \mu_{klux}(t)\delta_{klux}(t)}{\mu_{klu\underline{x}}(t) + \mu_{klux}(t)}. \tag{51}$$

The delay at an IBC buffer is given by

$$\delta_{IBC} = \lim_{t\to\infty} \frac{\sum_{i=1}^{m} i\{B^i_{0lvu}(n,t) + B^i_{0lvu}(\underline{y},t) + B^i_{0lvu}(y,t)\}}{\sum_{i=1}^{m}\{B^i_{0lv}(n,t)r_{1lv}(n,t) + B^i_{0lv}(\underline{y},t)r_{1lv}(\underline{y},t) + B^i_{0lv}(y,t)r_{1lv}(y,t)\}}. \quad (52)$$

The *cell delay* (δ_i) is defined to be the number of clock cycles a cell spends in the switch from the instant it enters the switch until it exits the switch. δ_i is therefore given by

$$\delta_i = \sum_{k=1}^{s} \bar{\delta}_{klv} + \delta_{IBC}. \quad (53)$$

The mean cell delay is therefore given by

$$\delta = \frac{\sum_{i=0}^{N-1}\delta_i}{N}. \quad (54)$$

Cell loss probability at input $0lv$ (η_{0lv}) is defined as the probability of a cell being lost and is given by the probability that the IBC buffer is full:

$$\eta_{0lv} = \lim_{t\to\infty} B^f_{0lv}(0,t). \quad (55)$$

For uniform traffic, $\mu_{slv} = \mu$ for all l and v. The cell loss probability at any input for a uniform traffic is therefore given by:

$$\eta_{0lv} = \frac{\lambda - \mu}{\lambda}, \quad \text{for all } l. \quad (56)$$

Buffer occupancy (ξ_{klvu}) at $klvu$ is defined as the average number of cells in the buffer at klv, $v \in \{y,\underline{y}\}$, connected to u, $u \in \{x,\underline{x}\}$ and can be given by the probability of i cells in the buffer as follows:

$$\xi_{klvu} = \lim_{t\to\infty} \sum_{i=1}^{m} i\{B^i_{klvu}(n,t) + B^i_{klvu}(\underline{y},t) + B^i_{klvu}(y,t)\}. \quad (57)$$

The single queue analyses are made consistent by forcing the single queue variables to yield certain known long term flows. Since the equations describing the dynamics of the switch are described by recurrence relations, the solution is obtained by an iterative method [10].

3.5. Destination port probabilities

P_{kluv} is 0.5 for a uniform traffic pattern. On the contrary, a general traffic pattern implies that P_{kluv} may not be 0.5. This general traffic pattern is modeled by finding a mapping scheme that transforms a given output referencing pattern into a set of P_{kluv}'s which reflects the given referencing pattern.

As an example, let us take an 8×8 ATM switch as shown in Fig. 5. Since all the inputs are assumed to have the same general traffic pattern, only the mapping scheme for one input is discussed. The referencing pattern of input I_i can be represented in terms of the output destination probability β_{ij}, the probability that a cell chooses output O_j as its destination. Consider a cell arriving at input I_0 and observe the path it takes as it travels through the switch to reach an output. The cell chooses output O_0 with probability β_{00} which equals $P_{10\underline{x}y}P_{20\underline{x}y}P_{30\underline{x}y}$. Similarly, the cell from input 0 chooses output $\underline{1}$ with probability $\beta_{01} = P_{10\underline{x}y}P_{20\underline{x}y}(1 - P_{30\underline{x}y})$. Using these two equations, $P_{30\underline{x}y}$ is found in terms of β_{00} and β_{01}:

$$P_{30\underline{x}y} = \frac{\beta_{00}}{\beta_{00} + \beta_{01}}. \quad (58)$$

The other output port probabilities in the SEs can be found in a similar manner, once the memory referencing patterns P_{kluv}'s are known:

$$P_{3l\underline{x}y} = \frac{\beta_{i(2l)}}{\beta_{i(2l)} + \beta_{i(2l+1)}}, \quad (59)$$

$$P_{3l\underline{x}y} = 1 - P_{3l\underline{x}y}, \quad (60)$$

$$P_{3l\underline{x}y} = \frac{\beta_{i(2l+1)}}{\beta_{i(2l)} + \beta_{i(2l+1)}}, \quad (61)$$

$$P_{3l\underline{x}y} = 1 - P_{3lxy}, \quad l = 0,1,2,3, \quad (62)$$

$$P_{2l\underline{x}y} = P_{2l\underline{x}y} = \frac{\beta_{i0} + \beta_{i1}}{\beta_{i0} + \beta_{i1} + \beta_{i2} + \beta_{i3}}, \quad l = 0,2, \quad (63)$$

$$P_{2lxy} = \frac{\beta_{i2} + \beta_{i3}}{\beta_{i0} + \beta_{i1} + \beta_{i2} + \beta_{i3}}, \quad l = 0,2, \quad (64)$$

$$P_{2l\underline{x}y} = P_{2l\underline{x}y} = \frac{\beta_{i4} + \beta_{i5}}{\beta_{i4} + \beta_{i5} + \beta_{i6} + \beta_{i7}}, \quad l = 1,3, \quad (65)$$

$$P_{2lxy} = \frac{\beta_{i6} + \beta_{i7}}{\beta_{i4} + \beta_{i5} + \beta_{i6} + \beta_{i7}}, \quad l = 1,3, \quad (66)$$

$$P_{1l\underline{x}y} = \frac{\beta_{i0} + \beta_{i1} + \beta_{i2} + \beta_{i3}}{\beta_{i0} + \beta_{i1} + \beta_{i1} + \beta_{i2} + \beta_{i3}\beta_{i4} + \beta_{i5} + \beta_{i6} + \beta_{i7}}, \quad l = 0,1,2,3, \quad (67)$$

$$P_{1l\underline{x}y} = P_{1l\underline{x}y}, \quad (68)$$

$$P_{1lxy} = \frac{\beta_{i4} + \beta_{i5} + \beta_{i6} + \beta_{i7}}{\beta_{i0} + \beta_{i1} + \beta_{i2} + \beta_{i3} + \beta_{i4} + \beta_{i5} + \beta_{i6} + \beta_{i7}}, \quad l = 0,1,2,3. \quad (69)$$

One of the nonuniform traffic patterns to be considered in this paper is the hot spot traffic pattern. In the hot spot traffic pattern, there is an output port which is accessed more often than other output ports. For example, many telephone callers may contend to call a popular location; many nodes may report synchronously some information to one node (say, the switch control center) for administrative purposes. Such a traffic can be characterized by a single hot-spot of a higher access rate, superimposed on a background of uniform traffic. Let h be the fraction of cells directed to the hot-spot output O_4. Then we have

$$P_{klxy} = \begin{cases} \dfrac{\lambda\left(h + \left(2^{(s-k)} - 1\right)\dfrac{1-h}{N}\right)}{2^{(s-k)}\lambda((1-h)/N) + \lambda\left(h + \left(2^{(s-k)} - 1\right)((1-h)/N)\right)} & \text{if } y \to O_4, \\ \dfrac{2^{(s-k)}\lambda((1-h)/N)}{2^{(s-k)}\lambda((1-h)/N) + \lambda\left(h + \left(2^{(s-k)} - 1\right)((1-h)/N)\right)} & \text{if others.} \end{cases} \tag{70}$$

Fig. 5 shows an 8 × 8 ATM switch under the hot spot traffic pattern. The switching elements and links that carry hot traffic are shown in bold. The average cell arrival rates for the O_jth output are

$$P\{O_j\} = \begin{cases} \lambda[1 + (N-1)h] & \text{if } j = 4, \\ \lambda(1-h) & \text{if } j \neq 4. \end{cases} \tag{71}$$

As long as $h > 0$, we have $P\{O_j\} > \lambda$ for $j = 4$, and $P\{O_j\} < \lambda$ for $j \neq 4$. It implies that the hot-spot output port (O_4) is overloaded and other outputs ($j \neq 4$) are underloaded. Only one cell can be transmitted to an output link in one time slot. Therefore, in order to guarantee a stable output queueing, the traffic load λ must be limited such that $P\{O_j\} \leq 1$ for $j = 4$. Therefore, from Eq. (71),

$$\lambda \leq \frac{1}{1 + (N-1)h}. \tag{72}$$

4. Results

In this section, we present results for the performance of an MS consisting of 2 × 2 crosspoint buffered switching elements. The performance criteria are normalized throughput, mean cell delay, cell loss probability, and buffer occupancy. We obtain the results using simulation and the analytical model we have developed. Results obtained from the proposed model are compared with simulation results. First, the analytical and simulation results are presented for small switches (64 × 64), followed by analytical results for large switches (1024 × 1024). Because of the excessive computing time and memory requirements, it was not possible to simulate large switches. In the graphs presented in this section, the dotted lines correspond to simulation results and the solid lines represent results obtained from the proposed analytical model.

Performance of an output buffered switch is often taken as a benchmark for comparing the performance of different switching architectures. Consequently, we have compared the performance of crosspoint and output buffered switches. For a fair and meaningful comparison, the total buffer space in an SE has been kept the same for both the crosspoint and output buffered switches. For example, when using a buffer space of six for an output buffer, we have used a buffer size of three for each of the crosspoint buffers.

For a uniform traffic, the normalized throughput as a function of the buffer size (m) is depicted in Fig. 6. It is found that the normalized throughput is very close under the RS and NS policies but is higher for the BS policy. For a buffer size of 20, the normalized throughput for the RS, NS and BS scheduling policies are 0.861, 0.858 and 0.906 respectively. For $m = 2$, the normalized throughput is rather poor (approximately 0.5). As expected, if the buffer size increases, the normalized throughput increases. However, beyond $m = 6$, the normalized throughput increases only marginally. Thus, additional buffers beyond this point can only be justified if the performance improves further under other traffic conditions.

The normalized throughput and mean cell delay under a uniform traffic pattern is shown in Fig. 7 for three different

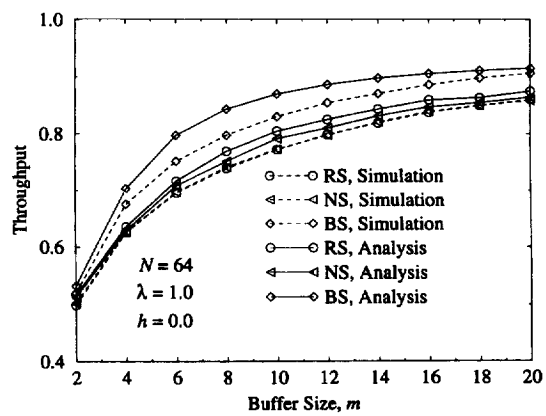


Fig. 6. Normalized throughput versus buffer size for $N = 64$ and uniform traffic.

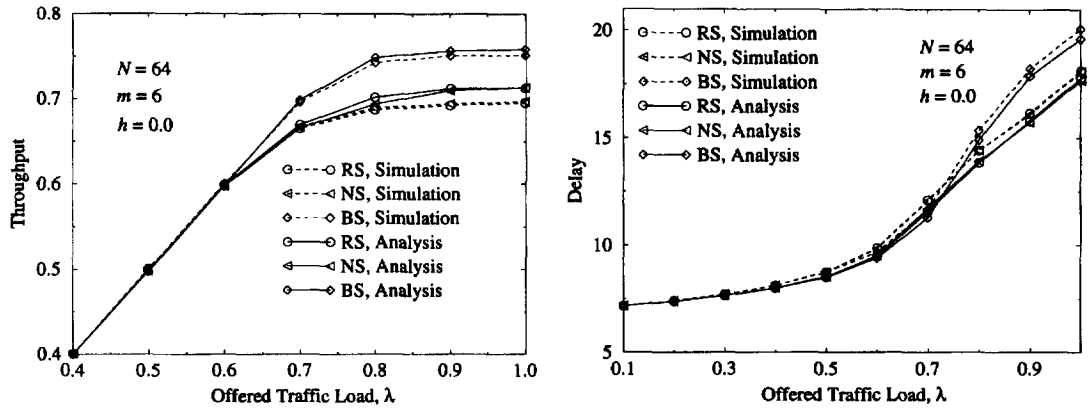


Fig. 7. Normalized throughput and mean cell delay versus traffic load for $h = 0.0$ and $N = 64$.

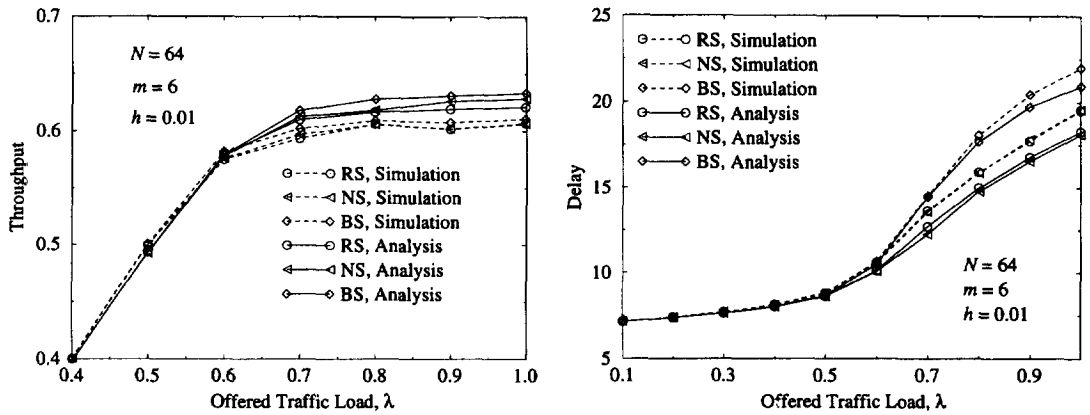


Fig. 8. Normalized throughput and mean cell delay versus traffic load for $h = 0.01$ and $N = 64$.

scheduling policies. It is seen that the normalized throughput using the BS policy is higher than the other two policies which are very close. For example, a 64 × 64 switch with a buffer size of six has a normalized throughput of 0.751 for the BS policy in comparison to about 0.695 for the RS policy and 0.697 for the NS policy under a uniform traffic pattern. Comparison of simulation and analytical results presented in the above figures shows that the proposed model is very accurate. Fig. 8 shows the normalized throughput and mean cell delay versus offered traffic load for a hot spot probability of 0.01. The figure shows that the switch has a small mean cell delay up to an offered traffic load of 0.6.

Fig. 9 compares the three scheduling schemes for a 1024 × 1024 switch under a uniform traffic load. It is seen that our proposed model is accurate for large switch sizes also. The normalized throughput of the BS scheme is the highest, and the RS scheme is better than the NS scheme. Fig. 10 plots the normalized throughput and mean cell delay for hot spot probabilities ranging from 0 to 0.01. An

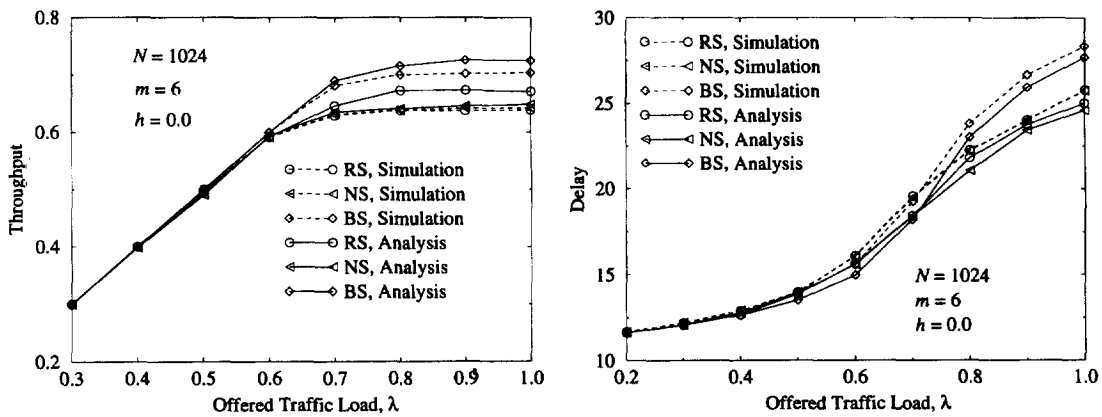


Fig. 9. Normalized throughput and mean cell delay versus traffic load for $N = 1024$ and uniform traffic.

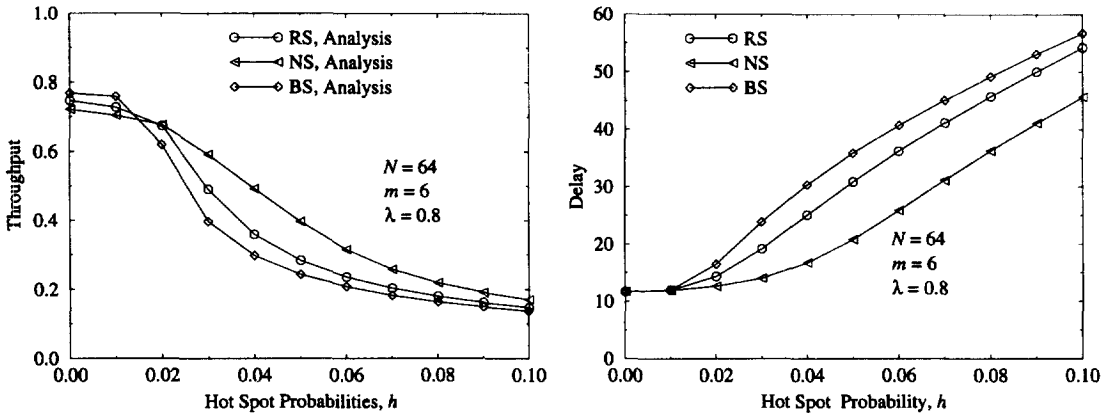


Fig. 10. Normalized throughput and mean cell delay versus hot spot probability for $\lambda = 0.8$.

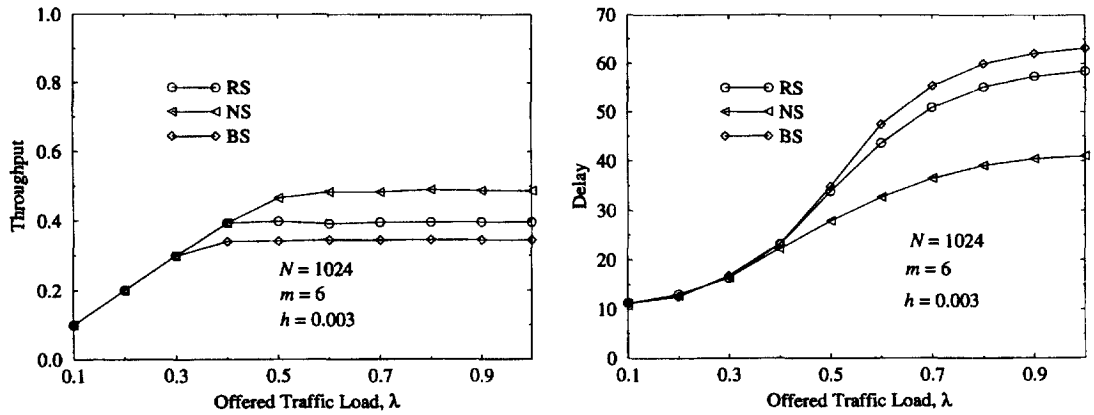


Fig. 11. Normalized throughput and mean cell delay versus traffic load for $h = 0.003$ and $N = 1024$.

interesting phenomenon occurs in this figure. When the hot spot probability is very small (h is between 0.0 to 0.001), the BS policy performs the best followed by the RS policy which is better than the NS policy. The normalized throughput for the BS policy drops sharply (as compared to the RS and NS policies) as the hot spot probability increases. When h reaches 0.02. the normalized throughput of the NS policy is higher than the other two policies. It is seen that the BS policy has the lowest normalized throughput and the highest mean delay under a heavy hot spot traffic.

Fig. 11 compares the normalized throughput and mean cell delay in a 1024 × 1024 switch for the three scheduling policies under a hot spot probability of 0.003. It is seen that the NS policy has the highest normalized throughput and the lowest mean delay for high values of the hot spot probability. This is because the mean cell delay is calculated as an average of the delays encountered by cells leaving the outputs of the switch. To illustrate it further, Table 1 shows the individual throughput and delay encountered by cells leaving the different outputs, where O_0 is the hot output. Comparing output O_0 with O_1 and O_2, it is seen that a higher throughput corresponds to a higher cell delay. However, comparison of the mean delay and normalized throughput (in the last column of Table 1) for hot spot probabilities of 0.2 and 0.4 shows a higher normalized throughput corresponding to a lower mean delay.

Table 1
Delay and throughput at each output

	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	Mean
$h = 0.2$									
Delay	19.7025	17.5691	13.7156	13.7156	8.0498	8.0498	8.0498	8.0498	12.1128
Throughput	0.9905	0.3445	0.3479	0.3479	0.3426	0.3426	0.3426	0.3426	0.4252
$h = 0.4$									
Delay	28.2823	25.6606	20.2500	20.2500	10.8862	10.8862	10.8862	10.8862	17.2485
Throughput	0.9982	0.1609	0.1593	0.1593	0.1548	0.1548	0.1548	0.1548	0.2621

Crosspoint buffering RS, $N = 8$, $m = 2$ and $\lambda = 1.0$.

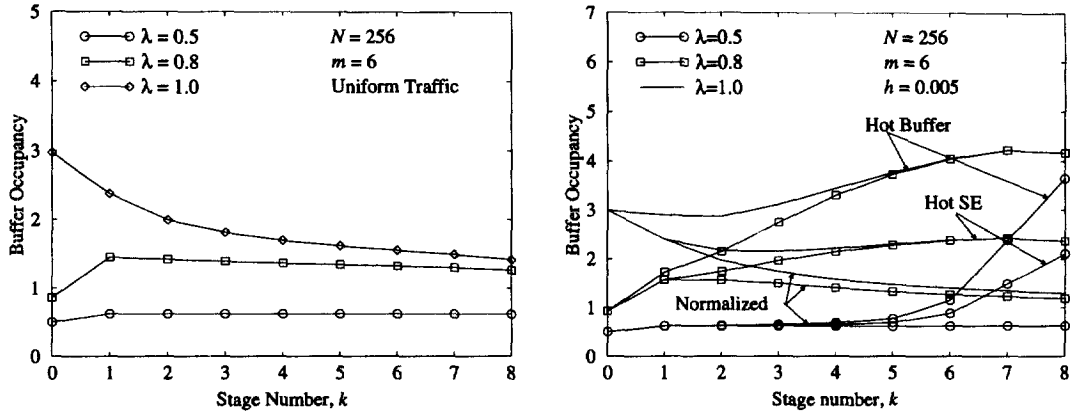


Fig. 12. Buffer occupancy versus stages number for uniform traffic and $h = 0.005$.

Fig. 12 shows buffer utilization (measured by buffer occupancy), at every stage of the MS under uniform and hot spot traffic patterns. The hot buffer and hot SE referred to in the figure are defined in Section 2.2. The offered traffic load, λ, is varied from 0.5 to 1.0. The results are for a switch size of 256 × 256 and λ = 0.5, 0.8 and 1.0. Note that when h is 0.005 (see Fig. 12), the buffer occupancy falls off at the last stage for the hot SE and the hot buffer. This is because a cell in the last stage can always leave the output port of the switch. Under uniform traffic, the offered traffic load is uniform distributed to all the outlets. Therefore, the buffer occupancy of the buffers in the same stage are identical, and the occupancy decreases with increasing stage number in the switch when λ = 1. The occupancy increases when λ is 0.8 or less. For small values of λ and uniform traffic, there is no blocking inside the switch and the occupancy of the IBC buffer is lower than the buffers in the internal stages. When λ is large, because of cell blocking inside the switch, the occupancy of the IBC buffer is much higher than the internal buffers. Consequently, there is a possibility of cells being lost at the switch inputs.

For an output buffered ATM switch (see Fig. 2), an output buffer is shared by all the inputs resulting in a higher buffer utilization than the crosspoint buffered switch. However, this proportion changes in favor of the hot traffic, when some hot spot value is introduced. Fig. 13 compares the buffer occupancy of the crosspoint and output buffering [20] by showing the buffer occupancy of the hot buffer, hot SE (average) and the normalized (for a stage) at every stage of the switch. It is seen that the hot traffic saturates the hot buffers for an input load of as low as 0.5. Again, increasing the buffer size makes little improvement on this effect. Higher buffer utilization is an advantage of output buffered switches as compared to switches using the crosspoint buffer discipline.

In Fig. 14, the normalized throughput of crosspoint and output buffered switches are compared for various values of the buffer size. Normalized throughput and mean cell delay for the crosspoint and output buffering [7] for different hot spot probabilities are shown in Fig. 15. For the crosspoint buffering, only the RS policy is shown. For a uniform traffic, the results show that cross-point buffering provides comparable performance to output buffering at offered traffic loads below 60%. The normalized throughput of output buffering is approximately 10% higher than crosspoint buffering at higher loads. For a hot spot traffic, the performance of output buffering is much better than cross-point buffering.

Fig. 16 shows the effect of buffer size on the cell loss probability for 64 × 64 and 1024 × 1024 crosspoint and output buffered switches operating under a uniform traffic

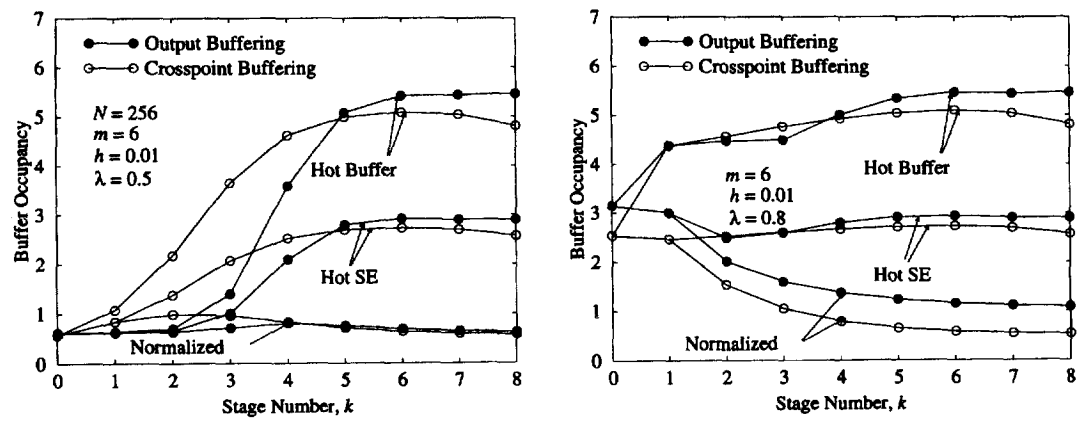


Fig. 13. Comparison of buffer occupancy of crosspoint and output buffering with λ = 0.5 and 0.8.

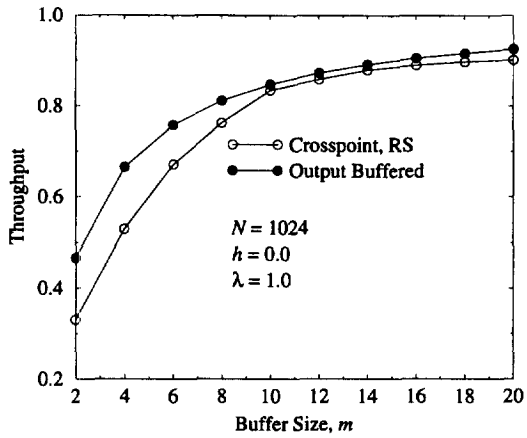


Fig. 14. Comparison of throughput for cross-point and output buffering for different buffer sizes.

with various offered traffic loads. For cross-point buffering, only results obtained from the RS policy are shown. The cell loss probability linearly decreases with increasing buffer size m. For $N = 64$ and an offered load of 0.4, the cell loss probability can be kept low (less than 10^{-7}) when the buffer size is more than six for output buffering and 11 for cross-point buffering. If such a low cell loss probability is desired at an offered load of 0.7, the buffer size has to be increased to 14 for output buffering and 19 for cross-point buffering. For a switch size (N) of 1024, the cell loss probability decreases slowly at large values of the offered traffic load but drops sharply at small values of the offered traffic load. As can be seen, the cell loss probability for output buffering is lower than the crosspoint buffering scheme. With an output buffer size of 12, it is possible to have a cell loss probability of 10^{-8} under an offered traffic load of 0.6. On the contrary, crosspoint buffering requires 18 buffers to achieve this value.

In essence, a truly accurate model must consider all the buffers in the switch simultaneously, and this is not always possible because of the excessive computational resource requirements. This paper considered three stages simultaneously and each clock cycle has been split into three cycles. As result of the above modeling assumptions, the analytical results in some of the cases are slightly different from the simulation results.

5. Conclusion

We have developed a Markov chain model to study the performance of multistage ATM switch fabrics using finite cross-point buffered switching elements. The model is

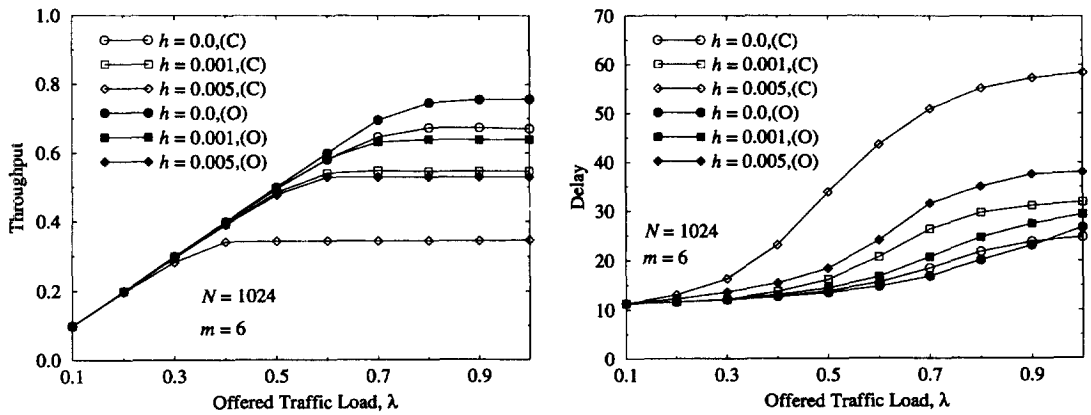


Fig. 15. Comparison of normalized throughput and mean cell delay for cross-point and output buffering.

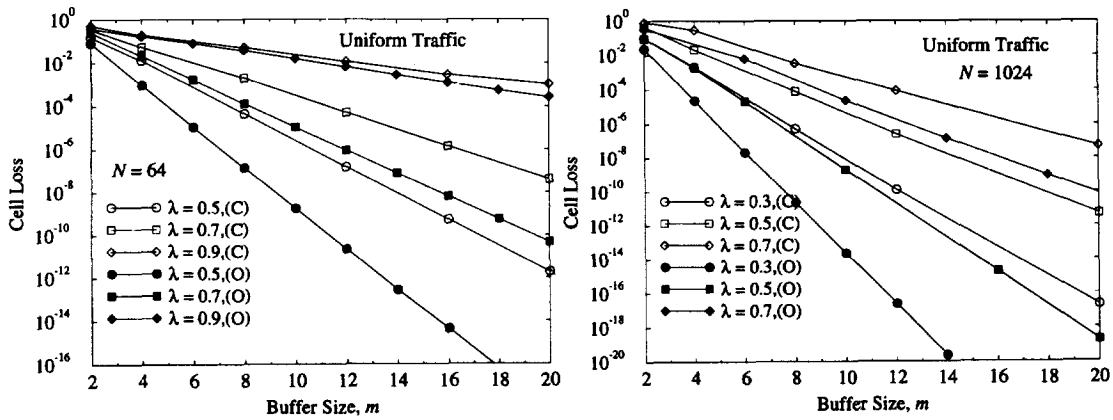


Fig. 16. Comparison of cell loss for cross-point and output buffering.

capable of handling uniform and hot spot traffic patterns. An iterative technique has been used to solve the model which provides steady state values for the normalized throughput, mean cell delay and cell loss probability. patterns. Comparison with simulation results indicates that the analytical model is very accurate.

The most important feature of cross-point buffering is that the switch fabric operates at the same speed as the input-output ports, which is desirable in high-speed switching systems. The effect of varying the buffer size and switch size has been studied for uniform and hot spot input traffic. The performance of crosspoint and output buffering have been compared and contrasted.

Three scheduling policies, viz. new cell selection, blocked cell selection and random cell selection were considered for selecting a cell from the complementary buffers to be transmitted to a given output. Among these, the policy which schedules a blocked cell first results in the best performance (higher normalized throughput and higher mean cell delay) under a uniform traffic. The performance of the random selection and new selection policies are similar when the hot spot probability is small. However, the new selection policy has the best performance followed by the random and blocked selection schemes when the hot spot traffic is heavy.

For a uniform traffic, cross-point buffering has a comparable performance to output buffering at offered loads below 60%. For a hot spot traffic, performance of the output buffering is much better than the cross-point buffering. For high offered loads, the normalized throughput of the output buffering is higher than the cross-point buffering. The proposed model can be extended to evaluate the performance of cross-point buffered switches operating under other non-uniform traffic patterns.

References

[1] H. Ahmadi, W.E. Denzel, A survey of modern high-performance switching techniques, IEEE Journal on Selected Areas in Commun. 7 (7) (1989) 1091–1103.

[2] J. Hui, E. Arthurs, A broadband packet switch for integrated transport, IEEE Journal Select Areas Commun. 5 (8) (1987) 1264–1273.

[3] M. Hluchyj, M. Karol, S. Morgan, Input versus output queueing on a space-division packet switch, IEEE Transactions Commun. 35 (12) (1987) 1347–1356.

[4] M. Saleh, M. Atiquzzaman, Queueing analysis of shared buffer switches for ATM networks, GLOBECOM 94: IEEE Global Telecommunications Conference, San Francisco, California, Nov 27–Dec 1, 1994, pp. 1070–1074.

[5] J.S. Turner. Queueing analysis of buffered switching networks, IEEE Transactions on Communications 41 (2) (1993) 412–420.

[6] B. Zhou, M. Atiquzzaman, Performance of output-multibuffered multistage interconnection networks under nonuniform traffic patterns, International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'94), North Carolina, USA, Jan. 31–Feb. 2, 1994, pp. 405–406.

[7] B. Zhou, M. Atiquzzaman, Performance of output-multibuffered multistage interconnection networks under general traffic patterns, IEEE INFOCOM '94: Conference on Computer Communications, Toronto, Canada, June 14–16, 1994, pp. 1448–1455.

[8] O.E. Percus, S.R. Dickey, Performance analysis of clock-regulated queues with output multiplexing in three different 2 × 2 crossbar switch architectures, Journal of Parallel and Distributed Computing 16 (1) (1992) 27–40.

[9] P. Goli, V. Kumar, Performance of a crosspoint buffered ATM switch fabric, IEEE INFOCOM'92, 1992, pp. 426–435.

[10] Y.C. Jenq, Performance analysis of a packet switch based on single-buffered banyan network, IEEE Journal Select. Areas Commun. 1 (1983) 1014–1021.

[11] H.S. Kim, A.L. Garcia, Performance of buffered banyan networks under nonuniform traffic patterns, IEEE Transactions Commun. 38 (5) (1990) 648–658.

[12] H. Yoon, K.Y. Lee, M.T. Liu, Performance analysis of multibuffered packet-switching networks in multiprocessor systems, IEEE Transactions Commun. 39 (3) (1990) 319–327.

[13] H.S. Kim, I. Widjaja, A. Leon-Garcia, Performance of output-buffered banyan networks with arbitrary buffer sizes, IEEE INFOCOM '91: Conference on Computer Communications, Bal Harbour, Florida, 1991, pp. 701–710.

[14] T.H. Theimer, E.P. Rathgeb, M.N. Huber, Performance analysis of buffered banyan networks, IEEE Transactions Commun. 39 (2) (1991) 269–277.

[15] S. Nojima, E. Tsutsui, H. Fukudav, M. Hashimoto, Integrated services packet network using bus matrix switch, IEEE Journal Selected Areas Comm. 5 (8) (1987) 1284–1292.

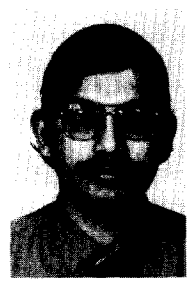
[16] E.P. Rathgeb, T.H. Theimer, M.N.Huber, Buffering concepts for ATM switching networks, Proc. IEEE GLOBECOM'88, Hollywood, FL, 1988, p. 39.3.

[17] Y.K. Kato, H. Hayami, J. Kamoi M. Takeno, K. Murakami, S. Hattori, A VLSIC for the ATM switching system, Proc. ISS'90, 1990, pp. 3–5.

[18] A.K. Gupta, L.O. Barbosa, N.D. Georganas, Switching modules for ATM switching systems and their interconnection networks, Computer Networks and ISDN Systems 26 (1993) 433–445.

[19] I. Stavrakakis, Statistical multiplexing under non-I.I.D. packet arrival processes and different priority policies, in: G. Pujolle, R. Puigjaner (Eds.), Data Communication Systems and their Performance, North-Holland, 1991, pp. 277–289.

[20] B. Zhou, M. Atiquzzaman, Performance analysis of finite output-buffered multistage ATM switching fabrics, IEEE INFOCOM'95, Massachusetts, U.S.A., April 2–6, 1995, pp. 16–23.



Mohammed Atiquzzamaan received his M.Sc. and Ph.D. degrees in Electrical Engineering and Electronics from the University of Manchester Institute of Science and Technology, England in 1984 and 1987, respectively. He had been academic staff member at La Trobe University, Melbourne and King Fahd University of Petroleum & Minerals, Saudi Arabia. Currently he is an academic staff member in the Department of Electrical and Computer Systems Engineering at Monash University, Melbourne, Australia. He has also served as technical program committee member of several international conferences including IEEE Infocom. His current research interests are in Broadband ISDN and ATM networks, multiprocessor systems, interconnection networks, parallel processing and image processing.