Performance of ATM Switch Fabrics Using Cross-Point Buffers

Bin Zhou and **M. Atiquzzaman**
Dept. of Electrical & Computer Systems Engineering
Monash University, Clayton 3168, Australia.
atiq@eng.monash.edu.au

Abstract

A new analytical model of the finite cross-point buffered multistage ATM switch under uniform and nonuniform traffic patterns is developed in this paper. Three scheduling policies for selecting a cell to be transmitted are compared. We have studied the impact of hot spot traffic on the performance of the switch. The effect of cross-point buffering and output buffering on the performance of the switch is also reported. The results from the model were validated with simulation results. It has been shown that the model provides very accurate results.

1 Introduction

The asynchronous transfer mode (ATM) has been accepted as the transport mechanism in future broadband ISDN (B-ISDN) networks. Multistage switches (MS) consisting of a number of stage of small switching elements (SE) [1], have been proposed for ATM switching architectures. Buffers are provided internal to the switch to queue cells that lose contention during routing conflicts. The buffers can be located at inputs, outputs, cross-points or shared between inputs & output. Output buffered SEs have a higher throughput than input buffered SEs [2]. Moreover, they do not suffer from *buffer hogging* as is the case of shared buffering operating in a nonuniform traffic environment. However, output buffered SEs require internal speedup of buffer operation.

An alternative method for reducing the head of line blocking in input buffered SEs is to provide buffers at each crosspoint within the switch fabric[1]. This reduces the complexity of the controller, and the speed of the buffers can be equal to the speed of the input/output ports. There are a number of analytical models for MSs with output buffered SEs [3]. Goli [4] presented an analytical model of cross-point buffered switch fabrics under a uniform traffic. The model does not produce accurate results, especially when the input traffic load is high. To our knowledge, no research reported in the literature has accurately analyzed finite cross-point buffered MS in the presence of nonuniform traffic patterns.

Our objective is to study the performance of the finite cross-point buffered MS under *uniform* and *hot spot* traffic patterns. Section 2 describes two different buffering mechanisms and presents the modeling assumptions of the switch. A single buffer model is developed in Section 3. Section 4 discusses the analytical results, and finally Section 5 concludes the paper.

2 Switch Architecture

A MS switch (Fig. 1) connects N inputs to N outputs using $s = \log_2 N$ stages of $N/2$ switching elements per stage as shown in Fig. 1 for $N = 8$. Each SE is a 2×2 crossbar allowing any input link to be connected to any one of its output links. Each SE has a finite number of buffers.

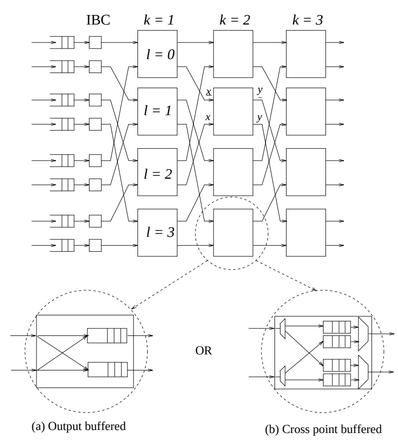

Figure 1: An 8×8 internally buffered ATM switch.

2.1 Buffering Strategies

Fig. 1 shows two different buffering strategies in the SEs. In the *output buffering* (Fig. 1(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. 1(b)), buffers are associated with each cross point of the SE. An SE consists of a cell distributor at each inlet, cell multiplexer at each outlet, and FIFO buffers for temporarily storing cells. It has the advantage of requiring only one read and one write operation on a buffer during a 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 enqueued in the appropriate buffer according to the destination tag. A cell leaves an outlet of an SE whenever any of the two associated buffers has a cell and the next stage can be accept it; if more than one buffer have cells, routing conflicts will occur. Scheduling policies to resolve the routing conflicts are discussed in Section 3.

3 Single-Buffer Model

For modeling purposes, as in [5], we assume that each input of the switch has an Input Buffer Controller (IBC) of size f and the switch operates *synchronously*. A *back-pressure* mechanism ensures that no cells are lost within the switch. Cell arrival process at each input of the switch is a *Bernoulli* process and each input of the switch has the same offered traffic load (λ). There is no *blocking* at the output links of the switch.

Fig. 2 shows three SEs at consecutive stages with multiple buffers at each crosspoint. Due to the space limitations, we will present our model for a single buffer at each crosspoint of an SE.

Fig. 1 shows how the SEs are specified by stage number k, SE number l, input ports x and $\underline{x}$, and output ports y and $\underline{y}$. Fig. 2 shows the relationship of three SEs in successive stages of the switch. The input and output ports of the l-th 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 k-th 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 k-th stage. Buffers $q_{\underline{x}}$ and q_x at an output port of an SE refer to buffers connected to inputs $\underline{x}$ and x of the SE. We refer to $q_{\underline{x}}$ and q_x as *complementary* buffers. A *new* cell in a buffer means that the cell has

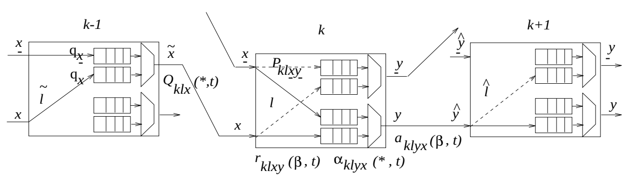

Figure 2: The illustration of $Q_{klx}(*,t)$ and $r_{kly}(\beta,t)$.

arrived during the current cycle, or it arrived in a previous cycle but has never won a routing conflict with a cell in the complementary buffer. A *blocked* cell means that the cell was not accepted by the SE in the next stage during a previous cycle. We define the following states variables to describe the state of a buffer at the *end* of a clock cycle.

State 0: Buffer is *empty*.

State n: Buffer contains a *new* cell.

State $\underline{y}$: Buffer contains a cell which was unsuccessful in a previous routing attempt to the *upper* output of the SE at the next stage.

State y: Buffer contains a cell which was unsuccessful in a previous routing attempt to the *lower* output of the SE at the next stage.

For modeling purposes, we split the clock cycle into two phases as in [3]. Cells are forwarded from a buffer during phase 1 and received during phase 2 of a cycle. The state of a buffer after phase 1 of a cycle will be called the *intermediate state* of the buffer.

Several scheduling policies are possible for selecting a cell from the buffers connected to an outlet of an SE. We consider the following three scheduling policies:

- *Random selection* (RS): the multiplexer randomly selects a cell from the buffer of contending cells for the given output.

- *New cell selection* (NS): the 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 on RS basis.

- *Blocked cell selection* (BS): the 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.

We define the following variables to describe the proposed model.

$B^0_{klvu}(0,t)$, ($\tilde{B}^0_{klvu}(0,t)$) = 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 t-th cycle (at the end of the phase 1 of the t-th cycle).

$B^1_{klvu}(\beta,t)$, ($\tilde{B}^1_{klvu}(\beta,t)$) = Probability the buffer at $klv, v \in \{y, \underline{y}\}$, connected to input u, $u \in \{x, \underline{x}\}$, contains a cell in state β, $\beta \in \{n, \underline{y}, y\}$, at the end of the t-th cycle (at the end of phase 1 of the t-th 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, $v \in \{\underline{y}, y\}$.

$Q_{klu}(*,t)$ = Probability that a new or blocked cell is ready to come to port klu, $u \in \{x, \underline{x}\}$, during clock cycle t.

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

$C^i_{klvu}(n,t)$, ($C^i_{klvu}(*,t)$) = Probability that i , $0 \le i \le 1$, cells which are in new (in new or blocked) state 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.

s = Number of stages in the switch.

λ = Probability that a cell is ready to come to the IBC at the beginning of a cycle.

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

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.

$r_{kluv}(\beta, t)$ = 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 clock cycle t.

$a_{klvu}(\beta, t)$ = 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 t-th cycle, given that the cell is in state β, $\beta \in \{n, \underline{y}, y\}$.

$\alpha_{klvu}(n, t)$, **(**$\alpha_{klvu}(b, t)$**)** = Probability that a new cell (a blocked cell, $b \in \{y, \underline{y}\}$) in the buffer at klv, $v \in \{y, \underline{y}\}$, connected to input u, $u \in \{x, \underline{x}\}$, can pass the SE, given that there is a cell in buffer $klv\underline{u}$.

The probability that a cell is ready to come to klu during clock cycle t is given by

$$Q_{klu}(*, t) = 1 - \{B^0_{(k-1)\tilde{l}\tilde{u}\underline{x}}(0, t-1) B^0_{(k-1)\tilde{l}\tilde{u}x}(0, t-1)\}$$
$$Q_{klu}(n, t) = B^1_{(k-1)\tilde{l}\tilde{u}\underline{x}}(n, t-1)\alpha_{(k-1)\tilde{l}\tilde{u}\underline{x}}(n, t) + B^1_{(k-1)\tilde{l}\tilde{u}x}(n, t-1)\alpha_{(k-1)\tilde{l}\tilde{u}x}(n, t)$$
$$Q_{klu}(v, t) = B^1_{(k-1)\tilde{l}\tilde{u}\underline{x}}(v, t-1)\alpha_{(k-1)\tilde{l}\tilde{u}\underline{x}}(v, t) + B^1_{(k-1)\tilde{l}\tilde{u}x}(v, t-1)\alpha_{(k-1)\tilde{l}\tilde{u}x}(v, t), \qquad v \in \{\underline{y}, y\}$$

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

$$\begin{aligned}
C^0_{klvu}(n, t) &= 1 - Q_{klu}(n, t) + Q_{klu}(n, t) P_{klu\underline{v}} \\
C^1_{klvu}(n, t) &= Q_{klu}(n, t) P_{kluv} \\
C^0_{klvu}(*, t) &= 1 - Q_{klu}(*, t) + Q_{klu}(\underline{v}, t) \\
&\quad + Q_{klu}(n, t) P_{klu\underline{v}} \\
C^1_{klvu}(*, t) &= Q_{klu}(v, t) + Q_{klu}(n, t) P_{kluv}
\end{aligned}$$

The probability that a cell can pass an SE by winning any possible conflict in the SE depends on the scheduling policy.

<u>Random selection</u>

$$\begin{aligned}
\alpha_{klvu}(n, t) &= 0.5\{B^1_{klv\underline{u}}(y, t) + B^1_{klv\underline{u}}(\underline{y}, t) \\
&\quad + B^1_{klv\underline{u}}(n, t)\} + B^0_{klv\underline{u}}(0, t) \\
\alpha_{klvu}(b, t) &= 0.5\{B^1_{klv\underline{u}}(y, t) + B^1_{klv\underline{u}}(\underline{y}, t) + \\
&\quad B^1_{klv\underline{u}}(n, t)\} + B^0_{klv\underline{u}}(0, t), \ b \in \{\underline{y}, y\}
\end{aligned}$$

<u>New cell selection</u>

$$\begin{aligned}
\alpha_{klvu}(n, t) &= B^1_{klv\underline{u}}(y, t) + B^1_{klv\underline{u}}(\underline{y}, t) + 0.5 B^1_{klv\underline{u}}(n, t) \\
&\quad + B^0_{klv\underline{u}}(0, t) \\
\alpha_{klvu}(b, t) &= 0.5\{B^1_{klv\underline{u}}(y, t) + B^1_{klv\underline{u}}(\underline{y}, t)\} \\
&\quad + B^0_{klv\underline{u}}(0, t), \ b \in \{\underline{y}, y\}
\end{aligned}$$

<u>Blocked cell selection</u>

$$\begin{aligned}
\alpha_{klvu}(n, t) &= 0.5 B^1_{klv\underline{u}}(n, t) + B^0_{klv\underline{u}}(0, t) \\
\alpha_{klvu}(b, t) &= 0.5\{B^1_{klv\underline{u}}(y, t) + B^1_{klv\underline{u}}(\underline{y}, t)\} + \\
&\quad B^1_{klv\underline{u}}(n, t) + B^0_{klv\underline{u}}(0, t), \quad b \in \{\underline{y}, y\}
\end{aligned}$$

The routing probabilities of a cell are therefore given by:

$$\begin{aligned}
r_{kluv}(n, t) &= B^0_{klvu}(0, t-1) + B^1_{klvu}(n, t-1) \\
&\quad a_{klvu}(n, t) + B^1_{klvu}(y, t-1) a_{klvu}(y, t) \\
&\quad + B^1_{klvu}(\underline{y}, t-1) a_{klvu}(\underline{y}, t) \\
a_{klvu}(n, t) &= \alpha_{klvu}(n, t)\{P_{(k+1)\hat{l}\hat{v}y} r_{(k+1)\hat{l}\hat{v}y}(n, t) \\
&\quad + P_{(k+1)\hat{l}\hat{v}\underline{y}} r_{(k+1)\hat{l}\hat{v}\underline{y}}(n, t)\}
\end{aligned}$$

When a cell in stage $k-1$ is in state $(1, \underline{y})$ or $(1, y)$ 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 be empty. 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 destination buffer in stage k at the end of cycle $t-1$ can be in state $(1, n)$ only if it was in state $(1, n)$ at the end of 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 destination buffer in stage k at the end of cycle $t-1$ can be in state $(1, y)$ or $(1, \underline{y})$ when 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 cycle $t-1$.

Under the above circumstances, the intermediate state of the destination buffer in stage k during cycle $(t-1)$ will be either full if it is blocked or empty if a cell left. We therefore define $B'^1_{klv}(\beta, t)$, $\beta \in \{n, \underline{y}, y\}$, as the probability of the destination buffer being full and in state β at the end of cycle t, given that the cell at stage $k-1$ is blocked at the end of cycle t. We can obtain $B'^1_{klv}(\beta, t-1)$, from

$$B'^1_{klvu}(n, t-1) = B^1_{klvu}(n, t-2)\overline{\alpha}_{klvu}(n, t-1)$$
$$r_{kluv}(b, t) = B'^1_{klvu}(n, t-1) a_{klvu}(n, t) + (1 - B'^1_{klvu}(n, t-1))\{P_{(k+1)\hat{l}\hat{v}y} a_{klvu}(y, t) + P_{(k+1)\hat{l}\hat{v}\underline{y}} a_{klvu}(\underline{y}, t)\}$$
$$a_{klvu}(y, t) = \alpha_{klvu}(y, t) r_{(k+1)\hat{l}\hat{v}y}(y, t)$$
$$a_{klvu}(\underline{y}, t) = \alpha_{klvu}(\underline{y}, t) r_{(k+1)\hat{l}\hat{v}\underline{y}}(\underline{y}, t)$$

The above equations are valid for all the buffers in the SEs and the IBC. However, there are other equations which are different for the buffers at the first stage, the last stage, the intermediate stages and the IBC. Those equations are given below separately.

3.1 Intermediate Stage Buffers

The intermediate states of the buffer at $klvu$ are:

$$\begin{aligned}
\tilde{B}^0_{klvu}(0, t) &= B^0_{klvu}(0, t-1) + B^1_{klvu}(n, t-1) \\
&\quad a_{klvu}(n, t) + B^1_{klvu}(\underline{y}, t-1) a_{klvu}(\underline{y}, t) \\
&\quad + B^1_{klvu}(y, t-1) a_{klvu}(y, t)
\end{aligned}$$

$$
\begin{aligned}
\tilde{B}^1_{klvu}(n,t) &= B^1_{klvu}(n,t-1)\overline{\alpha}_{klvu}(n,t) \\
\tilde{B}^1_{klvu}(\underline{y},t) &= B^1_{klvu}(n,t-1)\alpha_{klvu}(n,t)P_{(k+1)\hat{l}\hat{v}\underline{y}} \\
&\quad \overline{r}_{(k+1)\hat{l}\hat{v}\underline{y}}(n,t) + B^1_{klvu}(\underline{y},t-1)\{ \\
&\quad \alpha_{klvu}(b,t)\overline{r}_{(k+1)lvu}(\underline{y},t) + \overline{\alpha}_{klvu}(b,t)\} \\
\tilde{B}^1_{klvu}(y,t) &= B^1_{klvu}(n,t-1)\alpha_{klvu}(n,t)P_{(k+1)\hat{l}\hat{v}y} \\
&\quad \overline{r}_{(k+1)\hat{l}\hat{v}y}(n,t) + B^1_{klvu}(y,t-1)\{ \\
&\quad \alpha_{klvu}(b,t)\overline{r}_{(k+1)lvu}(y,t) + \overline{\alpha}_{klvu}(b,t)\}
\end{aligned}
$$

If the buffer at klv at the end of clock cycle t is empty,

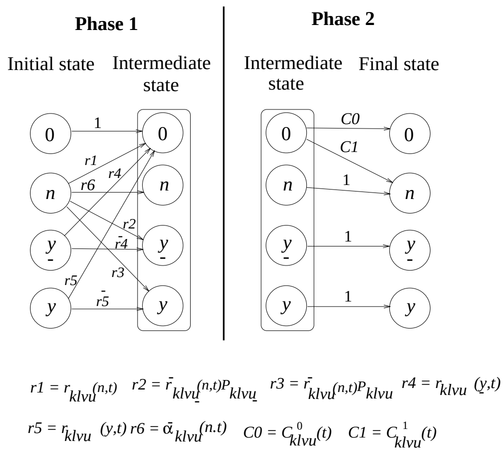

Figure 3: Markov chain state transition for single buffer.

then there can be no blocked cell in its source buffers in stage $(k-1)$. Therefore, only new cells can come to buffer klv if the intermediate state of the buffer is zero. The final state probabilities are given by

$$
\begin{aligned}
B^0_{klvu}(0,t) &= \tilde{B}^0_{klvu}(0,t)C^0_{klvu}(*,t) \\
B^1_{klvu}(n,t) &= \tilde{B}^0_{klvu}(0,t)C^1_{klvu}(*,t) + \tilde{B}^1_{klvu}(n,t) \\
B^1_{klvu}(\underline{y},t) &= \tilde{B}^1_{klvu}(\underline{y},t) \\
B^1_{klvu}(y,t) &= \tilde{B}^1_{klvu}(y,t)
\end{aligned}
$$

3.2 IBC and Last Stage Buffers

The IBC can be modeled in the same way as the buffers in the SEs (see [5] for details). Note that for the IBC buffers, $u = 0$, $k = 0$, and $v \in \{\underline{y}, y\}$ in the equations. Since there is no blocking at the output links of the switch, $a_{nlvu}(y,t)$ and $a_{nlvu}(\underline{y},t)$ are one. But a cell can be blocked by a conflict in the output line of the SE in the last stage. The state probabilities are given as follows, where $k = n$.

$$
\begin{aligned}
\tilde{B}^0_{klvu}(0,t) &= B^0_{klvu}(0,t-1) + B^1_{klvu}(n,t-1) \\
&\quad \alpha_{klvu}(n,t) \\
\tilde{B}^1_{klvu}(n,t) &= B^1_{klvu}(n,t-1)\overline{\alpha}_{klvu}(n,t) \\
B^0_{klvu}(0,t) &= \tilde{B}^0_{klvu}(0,t)C^0_{klvu}(*,t) \\
B^1_{klvu}(n,t) &= \tilde{B}^0_{klvu}(0,t)C^1_{klvu}(*,t) + \tilde{B}^1_{klvu}(n,t)
\end{aligned}
$$

3.3 Performance Measures

The throughput (μ_{nlv}) of the switch at the output nlv is given by the number of cells leaving the port nlv during a cycle.

$$
\mu_{nlv} = 1 - \lim_{t\to\infty} B^0_{nlv\underline{x}}(0,t)B^0_{nlvx}(0,t)
$$

The *normalized throughput* (μ) and the probability of a cell being lost (η) at a switch input can be given by

$$
\mu = \frac{1}{N}\sum_{\text{all outputs}} \mu_{nlv}, \qquad \eta = \frac{\lambda - \mu}{\lambda}
$$

Let $\delta_{klvu}(t)$ be the 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

$$
\begin{aligned}
\delta_{klvu}(t) &= \frac{B^1_{klvu}(n,t) + B^1_{klvu}(\underline{y},t) + B^1_{klvu}(y,t)}{\mu_{klvu}(t)}, \\
\mu_{klvu}(t) &= B^1_{klvu}(n,t)a_{klvu}(n,t) + B^1_{klvu}(\underline{y},t) \\
&\quad a_{klvu}(\underline{y},t) + B^1_{klvu}(y,t)a_{klvu}(y,t), \\
&\quad u \in \{x, \underline{x}\}, v \in \{y, \underline{y}\}, 1 \le k \le n
\end{aligned}
$$

For the *mean delay*, we consider the two buffers at the output port klv as one buffer and is defined as an *equivalent buffer*. Let $\overline{\delta}_{klv}$ denote the mean delay induced by the equivalent buffer at klv. Since the buffer is work conserving and non-preemptive, the conservation law [6] implies that $\overline{\delta}_{klv}$ satisfies the following equation.

$$
\overline{\delta}_{klv} = \lim_{t\to\infty} \frac{\mu_{klv\underline{x}}(t)\delta_{klv\underline{x}}(t) + \mu_{klvx}(t)\delta_{klvx}(t)}{\mu_{klv\underline{x}}(t) + \mu_{klvx}(t)}
$$

The delay at the IBC buffer (δ_{IBC}) is given by

$$
\lim_{t\to\infty} \frac{B^1_{0lv}(n,t)+B^1_{0lv}(\underline{y},t)+B^1_{0lv}(y,t)}{B^1_{0lv}(n,t)r_{0lv}(n,t)+B^1_{0lv}(\underline{y},t)r_{0lv}(\underline{y},t)+B^1_{0lv}(y,t)r_{0lv}(y,t)}
$$

The total mean delay δ in the switch is therefore given by $\delta = \sum_{k=1}^{n} \overline{\delta}_{klv} + \delta_{\mathrm{IBC}}$.

4 Results

Due to space limitations, we presented the single buffer model in this paper. Extension of the model to a multibuffered model can be found in [6]. Results obtained from the multibuffered extension of the model are validated by comparison with simulation results. We also compare the performance of the crosspoint buffered switch with those of the output buffered switch [5]. In the comparison, the total buffer space in an SE is the same for both the crosspoint and output buffering schemes. For example, a buffer space of six for an output buffer implies that each cross-point buffer is of size three. The normalized throughput of the MS using three different scheduling policies under a uniform traffic load is compared in Fig. 4. We see that the normalized throughput using the BS policy is better than the other two policies, which are very close. As illustrated, the normalized throughput in the NS policy is slightly better than the RS policy. Fig. 5 shows the mean delay (in terms of cycles) encountered by a cell passing through a switch of size 64×64 for three scheduling policies under hot spot traffic. It is to be noted that when the input

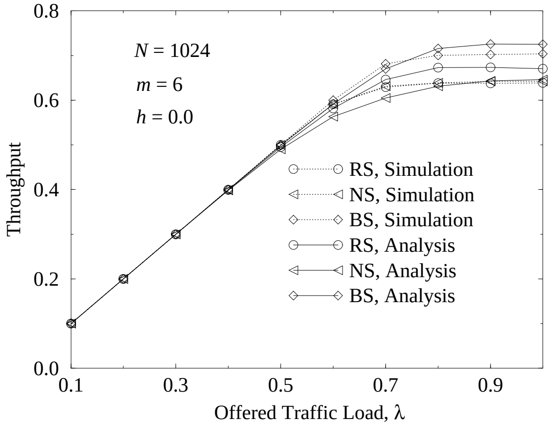

Figure 4: Normalized throughput versus offered traffic load for N=1024.

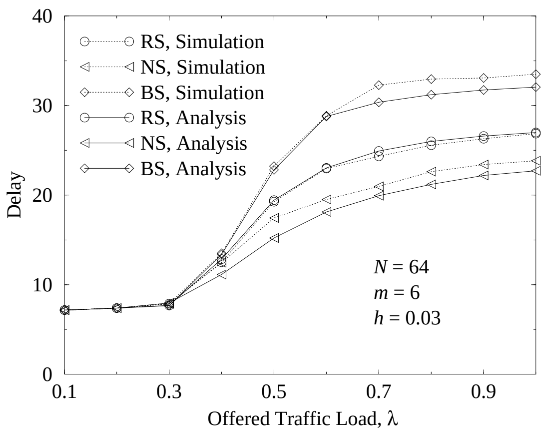

Figure 5: Mean delay versus offered load for h=0.03 and N=64.

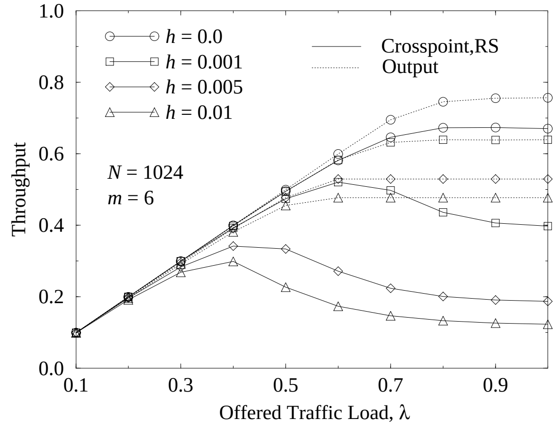

Figure 6: Normalized throughput comparison of output and crosspoint buffering.

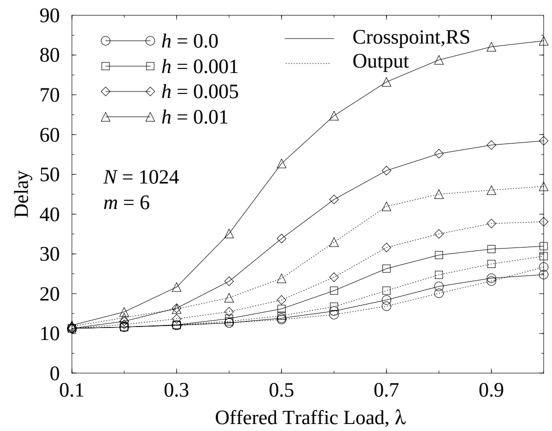

Figure 7: Mean delay comparison of output and cross-point buffering.

load is low to medium, the proposed model is very accurate. The difference between the results from the simulation and the proposed model arises when the input load is larger than 0.6.

Normalized throughput and mean delay obtained from the proposed model, and output buffered model [3] under different hot spot probabilities, are shown in Figs. 6 and 7. For the two buffering strategies, we only show RS scheme. These figures compare the hot spot probabilities of 0.0, 0.001, 0.005 and 0.01. for a switch size of 1024 × 1024. Under the uniform traffic (h=0.0), the results shown cross-point buffering provides performance comparable to output buffering at normal operating loads below 60%, and the normalized throughput of the output buffering is 10% higher than that for the cross-point buffering at the high load. It is seen that the delay increases with an increase in the value of h. For the hot spot traffic, the performance of output buffering is better than that of the cross-point buffering.

Fig. 8 shows the cell loss probability with respect to the number of buffer in uniform traffic situations. The traffic load, λ, is varied from 0.3 to 0.7. The switch size, N, is 64 and 1024. The solid lines are cross-point buffering results and the dotted lines are obtained from output buffering. For cross-point buffering, only results obtained from the RS scheme are shown. The cell loss probability linearly decreases with increasing buffer size m. When the offered load is 0.4, the cell loss can be kept low (below 10^{-7}) when the buffer size is more than seven for the output buffering and thirteen for the cross-point buffering. If such a low cell loss 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 . The cell loss probability decreases slowly at a heavy offer load of networks but sharply at a small load of networks. As can be seen, the cell loss probability can be lower using the output buffering comparing with those from crosspoint buffering schemes. With a output buffer size of 12, it is possible to have a cell loss probability of 10^{-8} under offer load of 0.6. But 18 buffers is required for crosspoint buffering scheme to achieve this value.

5 Conclusions

We propose an efficient Markov chain model to evaluate the performance of multistage ATM switch fabrics using cross-point buffered SEs. We studied the effect of varying the buffer sizes and of varying the switch sizes. We analyzed the switch performance under uniform and hot spot traffic conditions. We have compared the results obtained from the proposed model and those of output buffered switch model for both uniform and hot spot traffic pattern. Comparison with simulation indi-

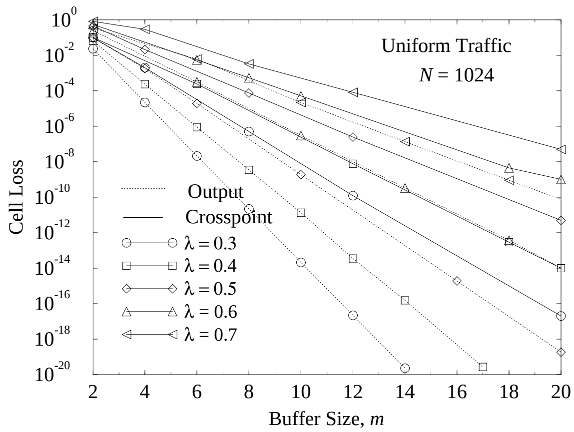

Figure 8: Comparison of cell loss of output and crosspoint buffering for $N = 1024$.

cates that the analytical model is very accurate. Three scheduling policies were considered for selecting a cell to be transmitted to a given output.

References

[1] H. Ahmadi and W.E. Denzel, "A survey of modern high-performance switching techniques," *IEEE Journal on Selected Areas in Commun.*, vol. 7, no. 7, pp. 1091–1103, Sept. 1989.

[2] B. Zhou and M. Atiquzzaman, "A performance comparison of buffering schemes for multistage switches," *ICA3PP'95, First International Conference on Algorithms And Architectures for Parallel Processing*, Brisbane, Australia, April 19-21, 1995.

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

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

[5] B. Zhou and M. Atiquzzaman, "Performance analysis of finite output-buffered multistage ATM switching fabrics," *IEEE INFOCOM'95*, Massachusetts, U.S.A., April 2-6, 1995.

[6] B. Zhou and M. Atiquzzaman, "Performance of ATM switch fabrics using cross-point buffers," Tech. Rep., Monash University, Melbourne, Department of Electrical and Computer Systems Engineering, 1995.