An Analytical Model and Performance Analysis of Shared Buffer ATM Switches under Non-uniform Traffic

Simon Fong[*], *Samar Singh*[*], *Mohammed Atiquzzaman*[+]

[*] School of Computer Sci. & Computer Eng., La Trobe University, Bundoora 3083, Australia.

[+] Department of Electrical & Computer Sys. Eng., Monash University, Melbourne 3168, Australia.

{fong, samar}@leopard.cs.latrobe.edu.au, atiq@eng.monash.edu.au

Abstract

For uniform traffic, ATM switches employing shared buffering have lower cell loss probability and better buffer utilization than other buffering strategies. However, the performance of shared-buffer switches degrades dramatically under non-uniform traffic. Traffic nonuniformity is inherent in telecommunications, and is likely to be present in ATM networks because of the different characteristics of traffic sources such as video, voice and data.

A number of buffer management schemes which are designed to overcome the adverse effect of non-uniform traffic in shared-buffer ATM switches have been proposed and studied. In this paper, we analyse the pushout technique to purge hot-spot cells which unfairly occupy the buffer. We call it Hot-Spot Pushout (HSPO). We develop an analytical model for buffer sharing with and without HSPO under non-uniform traffic and validate it using simulation. We also simulate other buffer sharing schemes available in the literature and show that HSPO performs consistently better than all of them in the presence of non-uniform traffic.

Key words ATM switching, performance modelling, shared buffer, buffer management, non-uniform traffic.

1. Introduction

Since ATM (Asynchronous Transfer Mode) has emerged as the principle cell-switching technology to support future B-ISDN (Broadband Integrated Services Digital Network), there is considerable interest in designing suitable buffering strategies for ATM switches which will have small buffers and will have a very low cell loss probability. Cells which contend for the same output but cannot be serviced immediately are buffered in the switch instead of being lost. A large number of contributions proposing different buffering strategies have appeared in the literature. Among them are input buffering [1], output buffering [2], mixed input/output buffering [3] and shared-buffering [4] (this is also known as central-buffering). Among all these buffering strategies, shared-buffering is the best in terms of buffer utilization and cell loss probability because of its maximum degree of buffer sharing (while output buffering lacks this advantage). Switches employing a shared buffer no longer suffer from the head-of-line blocking which is a common problem in input buffering. The performance of shared-buffer switches has been extensively analysed in [5,6,7,8] for uniform traffic, and in [9,10,11,12] for non-uniform traffic. The non-uniform traffic used in this paper is a Bernoulli traffic with respect to the load at each input, but has a non-uniform output destination probability. The Hot-spot traffic pattern is a particular type of non-uniform traffic.

Practical designs of VLSI chips for ATM switches with shared-buffers and having a throughput of 12 Gbits/sec have been described in [13]. However, this is only shown to work well under uniform traffic. Atiquzzaman and Saleh [14] have demonstrated that the traffic in a B-ISDN environment, which is very likely to be non-uniform, has an adverse effect on the performance of a shared buffer, leading to *buffer hogging*. Hence, there is a need to design a buffer sharing scheme which will perform well under non-uniform traffic. We list some of these proposed schemes below:

(a) Complete Sharing (CS) [11] in which all of the buffer space is shared by the inputs and outputs. Incoming cells are accepted as long as there is space available. CS has no special ability to protect against cell-loss from a non-uniform traffic load. Here, in this paper, it acts as a reference for the performance of other schemes.

(b) Complete Partitioning (CP) [15] in which the buffer is partitioned among the output ports and no buffer sharing is possible.

(c) Sharing with Minimum Allocation (SMA) [16] in which a minimum number of spaces is allocated for each output port, and the rest of the buffer is shared between all output ports.

(d) Sharing with Maximum Queue (SMXQ) [17] in which a limit is imposed on the maximum number of spaces that can be shared by each output port.

(e) Complete Sharing with pushout from the longest queue (HSPO) [18].

These schemes have been extensively investigated in the past, and each of them has its own shortcomings. None of them performs well under all traffic conditions. In [18], we provided a detailed performance comparison of these schemes under *bursty* traffic using simulation. The advantages and disadvantages of each scheme were identified. It was shown that HSPO (Hot Spot Push Out) always performs better than other buffer management schemes in the presence of bursty traffic. In this paper, we study the HSPO scheme under a non-uniform traffic pattern. Our non-uniform traffic assumes one output port to be a hot-spot. Note the difference with bursty traffic where the hot-spot moves from one output port to another. In the next section, we describe the HSPO scheme in detail and relate it to other similar work.

1.1 Hot Spot Push Out and related work

An important aspect of any pushout strategy is to decide which cell to push out of a full buffer in order to make room for an incoming cell. HSPO features pushing out (from a full buffer) cells that are unfairly hogging the buffer. This should lead to improved fairness and reduced congestion in the switch. In the past, pushout has been considered as complex to implement, though it is good from a performance point of view. Nevertheless, it has been shown that it is a promising buffer sharing scheme for implementation in hardware [19, 20]. A VLSI architecture featuring programmable buffer management is proposed in [21].

The basic idea of pushout was initially used to deal with buffer space control for traffic classes of different priorities [22, 23, 24]. Wei et al., [25] proposed a similar dynamic allocation scheme called Drop-on-Demand, which has a cell purging action to prevent buffer hogging. Recently, Choudhury and Hahne [26] applied it as a means of sharing space among output ports in single priority traffic. Then they extended it to a scheme, called Delayed Pushout [27], which uses a pushout mechanism within each switch component and a backpressure mechanism between switch stages. These studies used simulation to evaluate their scheme. In [28], Cidon et al., have proved mathematically that pushout is the optimal buffer sharing method, but their optimization study is based on a two-port switch with uniform and symmetric traffic at the inputs (equal traffic on all ports). Moreover, their approximation results are inaccurate when the input rate is near unity, or when the buffer size is small.

Performance analysis of ATM switches requires an accurate model of the switching elements. The model should be flexible enough to permit arbitrary buffer sizes, switch sizes and a full range of possible input loads. Moreover, the low loss probabilities involved in the performance evaluation of ATM switches result in time-consuming simulation programs. Obtaining loss probabilities below 10^{-6} is difficult using simulations. Analytical models permit one to obtain very low cell-loss probabilities and can give results much faster than simulation. Since the available literature on pushout is limited to simulation studies, we have developed an analytical model in this paper. The objectives of this paper are as follows:

(1) Develop an analytical model for a shared buffered ATM switch using CS and HSPO under non-uniform traffic, and assess its accuracy using simulations.

(2) Investigate how the different buffer sharing schemes compare with HSPO under non-uniform traffic. It will be shown that HSPO outperforms the other schemes.

Our proposed model is similar to those in [24, 29, 30] but is different in that the departure probability is computed using an urn model approximation which turns out to be quite accurate. Details are described in Section 3.3. In contrast, for an $N \times N$ switch, one may approximate the joint distribution of the queue-size variables associated with the N-1 non-hot output ports, assuming that these N-1 output queues behave independently and identically to the tagged hot-spot queue. This has

been done in [30, 31]. Obviously, this approximation is not appropriate here since the tagged queue and the other queues are assumed to have different loads under non-uniform traffic. Another approach, known as *active output formulation* was first proposed by Turner [32], and was subsequently used in [33, 8]. It recursively evaluates the probability of having a number of distinct output ports from which cells will depart, given that there are a fixed number of cells in the buffer. However, this evaluation relies on the assumption that the addresses of the cells stored within the buffer are independent. Unfortunately, this is not the case in reality. While it is true that the destination addresses of cells arriving at a switch are independent, destination addresses of buffered cells are correlated as a result of having contended for outputs. The correlation is strongest when the switch size is small and the buffer size is large.

Our simulation method is similar to that used in [8]. The simulation results are used to check the accuracy of our analytical model within the limits of the confidence interval computed for the simulations.

The rest of the paper is organised as follows. Section 2 defines the ATM switch assumptions including the hot-spot traffic model. In Section 3, we present an analytical model of the HSPO scheme of buffer management. Section 4 describes the performance measures used to evaluate the performance of the switch. We compare the results obtained from our model and the simulations in Section 5. Conclusions drawn from the study are summarized in Section 6.

2. System Assumptions

2.1 Hot Spot Traffic Model

Hot spot traffic was described in [34]. It has been extensively used as an unbalanced input traffic pattern by Atiquzzaman [14]. We assume that the probability of an incoming cell to be destined to a *hot* output port is higher than the probability for a non-hot output port. For the uniform traffic pattern, an incoming cell is equally likely to be directed to any output port. For the hot-spot traffic pattern, let the probabilities of an incoming cell directed to the hot output port and to a non-hot output port be p_h and p_c respectively.

$$p_h = \text{Pr. \{ an incoming cell is addressed to the hot spot output \}} = h + \frac{1 - h}{N}$$

$$p_c = \text{Pr. \{ an incoming cell is addressed to a non-hot spot output \}} = \frac{1 - h}{N}$$

where h, $0 \le h \le 1$, is defined to be the hot spot probability. When $h = 0$, all incoming cells are destined to any output port with equal probability. For $h = 1$, all incoming cells are destined for the hot-spot output port. We use the terms hot-spot cells and non-hot-spot cells for cells which are destined to the hot-spot output port and the rest of the N-1 non-hot output ports respectively. We assume a single hot output port in a switch.

In any time slot, the probability that a cell will arrive at a given input port is p. Successive cell arrivals to an input port, and the arrivals at the different ports are assumed to be independent. Therefore, the arrival process is Bernoulli with one fixed hot spot.

Prob. { k cells arrive at the switch during a time slot } = $\binom{N}{k} p^k (1-p)^{N-k}$

Prob. {l hot spot cells and m non-hot-spot cells arrive at the switch during a time slot } =

$$p_a(l,m) = \binom{N}{l+m} p^{l+m} (1-p)^{N-l-m} \binom{l+m}{l} p_h{}^l (1-p_h)^m \qquad (2.1.1)$$

2.2 ATM Switch Assumptions

The system under study is an ATM switch operating in the presence of non-uniform (hot spot) traffic as described above.

1. We assume an NxN non-blocking ATM switch with shared buffer of capacity of B. The inputs have access to all the buffer spaces, and all the buffer spaces have access to all the output ports. We model the switch as having virtual output queues that contain the cells destined for a particular output port. The length of the virtual output queues can grow to any size but the sum of the lengths of the virtual queues must be less than or equal to B (see Figure 1).

2. The traffic rates at all the input ports are assumed to be identical. Routing of a cell from an input port to an output port is destination tag-based, with temporary internal storage provided by the shared buffer.

3. Only one output port is assumed to be a hot-spot. The remaining N-1 non-hot output ports are equally probable. The cell arrival process is Bernoulli, but the distribution of output addresses is non-uniform.

4. Fixed-length ATM cells enter and depart the switch during successive operational cycles; each such cycle is called a time slot. During each time slot, the operation of the switch can be viewed, for modelling purposes, as a sequence of three primitive operations.

   (1) **Send**: If there is any cell in the buffer which is tagged for an output port, the cell will leave through that port. If there is more than one cell tagged for the same port, then the one at the head of queue will be chosen to leave. The maximum number of cells that can leave the buffer during a time slot is N. At most one cell can leave an output port during a time slot.

   (2) **Check**: The number of hot and non-hot cells remaining in the buffer are checked.

   (3) **Receive**: Each input port is scanned and incoming cells are accepted or rejected (lost) according to the following criteria: A cell is accepted as long as there is space in the buffer. If the buffer is full when a cell arrives, a cell tagged for the hot-spot output port is pushed out to make room for the incoming cell, provided that the arriving cell is not destined to the hot output. The incoming cell is dropped only if it is destined to the hot-spot port. Thus a maximum of N cells can be received in all the input ports during a time slot, and each input port can receive at most one cell. Note that the pushout is only from the hot-spot queue because it is assumed to be the longest queue in the buffer.

5. If the buffer is not empty, at least one cell will leave during the next time slot. If the buffer is empty, no cell will leave during the next time slot. It follows that at least one cell can come into the buffer in every time slot.

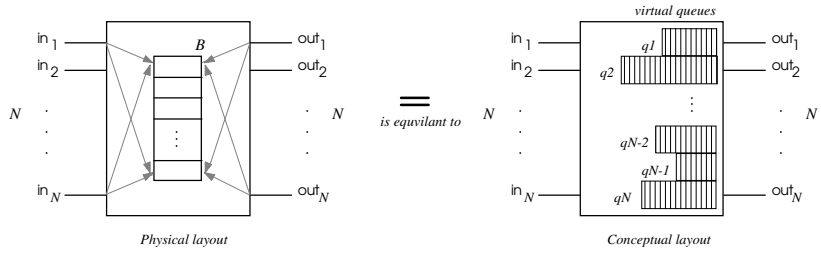

Figure 1. $N \times N$ shared buffer ATM switch of buffer size B, where $\sum_{i=1}^{N} q_i \leq B$

3. System Model with CS and HSPO

We use a discrete-time, finite-state Markovian Process to describe the queuing behaviour of the shared buffer switch. This is a natural choice since we assume a slotted system with a finite buffer capacity.

3.1. Model Construction

The state of the queuing system (the ATM switch) is denoted by a two-tuple $(X_h, , X_c)$, $0 \leq X_h + X_c \leq B$, where X_h represents the number of hot-spot cells in the buffer and X_c is the number of non-hot-spot cells in the buffer. A maximum of N cells can arrive at the buffer, and a maximum of N cells can leave the buffer during a time slot. This is equivalent to a classical queuing model of having N customers, N servers and B waiting room as shown in Figure 2.

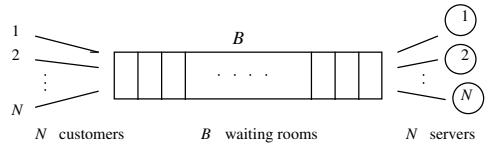

Figure 2. ATM buffer queuing model with N servers

The state space of the buffer occupancy can be represented by a tuple (l, m), $0 \leq l+m \leq B$. Let $\pi_{l,m}(k)$ be the state probability of the buffer at the kth time slot, i.e.

$\pi_{l,m}(k)$ = Prob. {l hot spot cells and m non-hot-spot cells in the buffer at the kth time slot}

The system will eventually stabilise because the queue is finite. Thus we know that a steady state probability distribution exists, i.e. $\lim_{k \to \infty} \pi_{l,m}(k) = \pi_{l,m}$ for all l and m.

3.2. Recursive State Probability Computation

Let π_n be the vector of state probabilities representing those states where n cells are in the buffer. Thus $\pi_n = [\pi_{0,n}, \pi_{1,n-1}, \ldots \pi_{n,0}]$ for each $n = 0,1,\ldots B$. The entire set of state probabilities represented by all the vectors π_n is $\pi = [\pi_0, \pi_1, \pi_2, \ldots \pi_B]$. The steady state behaviour of the queuing model is described by the state probability vector π, from which one can derive performance measures such as cell loss probabilities and average queue length.

The steady state probabilities are determined from the solution of $\pi = \pi \cdot P$ and the normalization condition $\pi \cdot e = 1$, where P is the state transition matrix and $e = [1,1,1, \ldots 1]^T$. Therefore, in order to find the steady state probabilities, we first define all the states of the system in π and subsequently generate the transition matrix P.

The state transition matrix P is of size $\frac{(B+1)(B+2)}{2}$ and is given by

$$P = \begin{bmatrix} M_{0,0} & M_{0,1} & M_{0,2} & \cdots & M_{0,B} \\ M_{1,0} & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ M_{B,0} & \cdots & \cdots & \cdots & M_{B,B} \end{bmatrix} \qquad (3.2.1)$$

Each element $M_{x,y}$ in the state transition matrix P is a submatrix defining the probability of transition from state x to state y. $M_{x,y}$ contains a set of $(x+1)(y+1)$ transition probabilities namely $M_{l,m;i,j}$ which represents the transition rate from state (l,m) to state (i,j) where $l+m=x$, $i+j=y$, $0 \le l+m \le B$ and $0 \le i+j \le B$. Replacing the elements of P in equation (3.2.1) by submatrices, the general form of the transition matrix P, for B=2 is given below.

$$P = \begin{bmatrix} \begin{bmatrix} M_{0,0;0,0} \end{bmatrix} & \begin{bmatrix} M_{0,0;0,1} & M_{0,0;1,0} \end{bmatrix} & \begin{bmatrix} M_{0,0;0,2} & M_{0,0;1,1} & M_{0,0;2,0} \end{bmatrix} \\ \begin{bmatrix} M_{0,1;0,0} \\ M_{1,0;0,0} \end{bmatrix} & \begin{bmatrix} M_{0,1;0,1} & M_{0,1;1,0} \\ M_{1,0;0,1} & M_{1,0;1,0} \end{bmatrix} & \begin{bmatrix} M_{0,1;0,2} & M_{0,1;1,1} & M_{0,1;2,0} \\ M_{1,0;0,2} & M_{1,0;1,1} & M_{1,0;2,0} \end{bmatrix} \\ \begin{bmatrix} M_{0,2;0,0} \\ M_{1,1;0,0} \\ M_{0,2;0,0} \end{bmatrix} & \begin{bmatrix} M_{0,2;0,1} & M_{0,2;1,0} \\ M_{1,1;0,1} & M_{1,1;1,0} \\ M_{2,0;0,1} & M_{2,0;1,0} \end{bmatrix} & \begin{bmatrix} M_{0,2;0,2} & M_{0,2;1,1} & M_{0,2;2,0} \\ M_{1,1;0,2} & M_{1,1;1,1} & M_{1,1;2,0} \\ M_{2,0;0,2} & M_{2,0;1,1} & M_{2,0;2,0} \end{bmatrix} \end{bmatrix} \quad (3.2.2)$$

Recall from Section 2.2 that the operation of the ATM switch is a sequence of send, check and receive operations. $M_{l,m;i,j}$ the transition from state (l,m) to state (i,j) can be extended to having an intermediate state, in between the sending and receiving operations. A time slot is thus assumed to be divided into two halves. Cell departures take place in the first half and cell arrival occurs in the second half. Therefore we can find the transition probability from state (l,m) to state (i,j) in the system as a result of two independent transitions. Consistency of transition between states and the sequence of switch operation is shown in Figure 3.

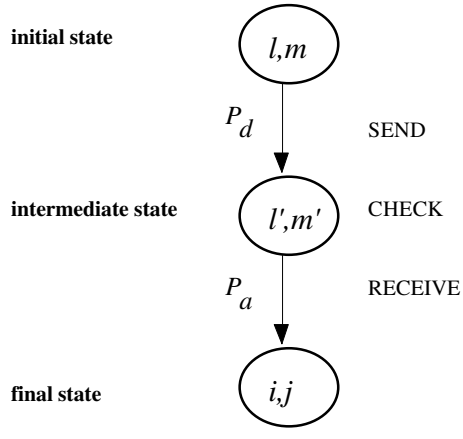

Figure 3. Transition between states as a sequence of switch operations between time slots k and $k+1$.

The first transition is due to cell departures, assuming no arrival of cells. The second transition is due to cell arrivals. Let the departure probability and arrival probability be P_d and P_a respectively.

P_d $(l',m' \mid l,m)$ = Prob. { l'-l hot-spot and m'-m non-hot-spot cells leave the buffer | there are l' hot-spot and m' non-hot-spot cells in the buffer }.

Clearly this probability is zero if $l'-l<0$ or $m'-m<0$. Similarly,

$P_a\,(i,j \mid l',m')$ = Prob. { a *suitable number* of hot-spot and non-hot-spot cells arrive to take the buffer to a state (i, j) | there are l' hot-spot and m' non-hot-spot cells in the buffer }.

The *suitable number* referred to above depends on the situation and needs to be computed. It has to take account of the fact that once the buffer is full, any cells that arrive are discarded (lost). If the discarding is done according to the pushout scheme, we have a slightly different expression which we will denote by $P_a^{\,pushout}\,(i,j \mid l',m')$. We will look at these probabilities closely in the later sections of the paper.

There may be conflict among non-hot-spot cells destined to the same output port for states represented by m, m' or j (due to the fact that a single number represents cells tagged to all N-1 ports). Thus, we cannot always have N cells leave the buffer through N output ports. We may have up to N cells leave the buffer provided it is not empty, and at least one cell will leave the buffer when l>0 or m>0. In Figure 4 below, we show all the possible intermediate states due to a transition from state (l,m) to state (i,j).

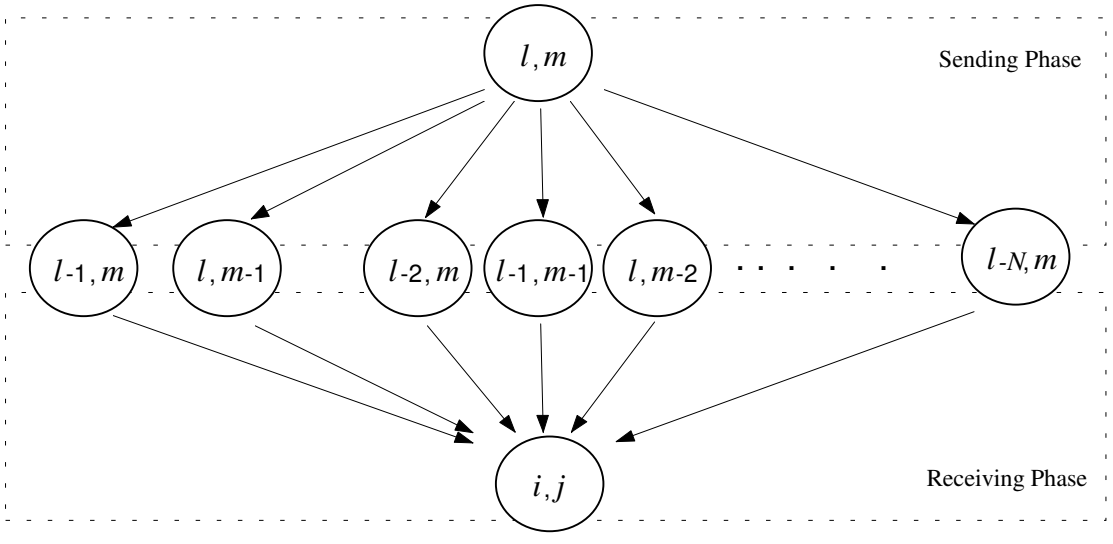

Figure 4. Markov state transition diagram.

Note that there are a number of transitions which cannot take place. For instance, l-l' must be exactly one, if there is at least one hot-spot cell in the buffer ($l \geq 1$) and l' must be 0 if l is 0; otherwise the transition is not valid. Also $M_{x,y}$ is a null matrix if it goes beyond the upper bound on the number of cell arrivals or the lower bound on the number of cell departures in a time slot, i.e.

$M_{x,y} = 0$ if $x < (y\text{-}N)$ or $x > (y+N)$, implying that such a transition is not possible. All these conditions will be specified clearly in the descriptions of P_d and P_a. The elements of $M_{l,m;i,j}$ are defined as follows:

Case 1. ($l+m=0$ and $i+j<B$)

When the starting state is empty and the finishing state is less than the size of the buffer, i hot-spot cells and j ordinary cells can arrive with the probability p_a as in Eqn. (2.1.1).

$$M_{l,m;i,j} = p_a(i, j) \tag{3.2.3}$$

Case 2. ($l+m=0$ and $i+j=B$)

The starting state of the buffer is empty and the finishing state is full. In this case, a weighted sum of arrival probabilities, P_a which is described more fully in Eqn. (3.4.1) later, is obtained by considering the fact that the number of cell arrivals may exceed the exact number of cells required to reach the desired state. Cells arriving after the buffer fills up are discarded.

$$M_{l,m;i,j} = P_a(i, j \mid 0, 0) \tag{3.2.4}$$

Case 3. ($l+m>0$ and $i+j<B$)

The starting state is not empty and the finishing state is not full. Since there are some cells in the buffer to start with, we would expect cells to depart from the buffer, and then to have cells arrive to the buffer. Therefore, this transition is a product of P_d which represents cell departure (described in detail in Eqn. (3.3.1)) and p_a which represents cell arrival from Eqn. (2.1.1).

$$M_{l,m;i,j} = \sum_{a=1}^{\min(N,\, l+m)} \sum_{b=0}^{a} P_d(l\text{-}a+b,\ m-b \mid l,\ m) \cdot p_a(i\text{-}l+a\text{-}b,\ j\text{-}m+b) \tag{3.2.5}$$

Case 4. ($l+m>0$ and $i+j=B$)

The starting state of the buffer is not empty and the finishing state is full. As in Eqn. (3.2.5), this transition is composed of two probabilities; cell departure P_d and cell arrival P_a. P_a is used here since the finishing state is full, and cell losses may occur when the buffer overflows and more cells arrive.

$$M_{l,m;i,j} = \sum_{a=1}^{\min(N,\, l+m)} \sum_{b=0}^{a} P_d(l-a+b,\ m-b \mid l,\ m) \cdot P_a(i,\ j \mid\ l-a+b,\ m-b) \quad (3.2.6)$$

Equations (3.2.3) to (3.2.6) above are valid for all $l, m, i, j \in \{0,1, .., B\}$ provided $l+m \le B$ and $i+j \le B$. The transition probabilities for cell arrival, P_a, (or $P_a^{pushout}$ in case pushout is functioning) and cell departure, P_d, that appear in the above equations are defined in the following sections.

3.3. Departure Probability

Let $P_d(l', m' \mid l, m)$ be the probability of transition from state (l, m) to (l', m'). This is also equivalent to the probability of having $(l-l')$ hot cells and $(m-m')$ non-hot cells leave the buffer. The probability is defined by the conditions under which the cells can leave the buffer, provided l is the number of hot- spot cells, and m is the number of non-hot-spot cells.

To work out this probability, we need a reasonable assumption about the state of the non-hot-spot cells. We use an urn model (as in [35]) for this purpose. We assume that each of the non-hot-spot cells can be destined for any of the N-1 non-hot-spot ports. But we are not concerned with the ordering of these cells. Rather we consider the total number of ways in which the m cells can be allotted to the N-1 ports, without regard to their positions in the queue. It is well known [35], that we can represent this by $Q(k,m) = \frac{(k+m-1)!}{(k-1)!m!}$, with k=N-1. We assume that each of these configurations is equally likely. On the other hand, if only $m-m'$ of these cells leave, that means only $m-m'$ of the queues were non-empty. There are $\binom{N-1}{m-m'}$ ways of choosing the non-empty queues. And within these non-empty queues any configuration of cells will lead to the departure of $m-m'$ cells. The number of such configurations is clearly the total number of ways of distributing the remaining m' in the $m-m'$ queues (allowing also for empty queues amongst these). This is nothing but $Q(m-m', m')$. Thus altogether, there are $\binom{N-1}{m-m'} Q(m-m', m')$ ways in which exactly $m-m'$ cells will leave. Thus, we may say that

Prob. {m-m' cells leave from the N-1 non-hot-spot ports | there are m non-hot-spot cells in the buffer}

$$= \binom{N-1}{m-m'} \frac{Q(m-m',\ m')}{Q(N-1,\ m)}.$$

Notice that, as long as the number of hot-spot cells is not zero, exactly one such cell will leave. Thus we can define $p_s(i, j, z)$ = Prob. {z cells will leave the buffer | i hot-spot and j non-hot-spot cells are in the buffer }. We have seen that if $i>0$, $j>0$ and $z \le N$, then only one hot-spot cell and z-1 non-hot-spot cells will leave.

This gives $p_s(i, j, z) = \binom{N-1}{z-1} \frac{Q(z-1,\ j-z+1)}{Q(N-1,\ j)}$. In the case that $i=0$, we must have $z \le N$-1 and then $p_s(0, j, z) = \binom{N-1}{z} \frac{Q(z,\ j-z)}{Q(N-1,\ j)}$. Using this result, we can compute the departure probabilities $P_d(l', m' \mid l, m)$ as follows.

$$P_d(l', m' \mid l, m) = \begin{cases} 1 & \text{if } (l=0) \text{ and } (l'=0) \text{ or,} \\ & \text{if } (l>0) \text{ and } (l-l'=1), \text{ and} \\ & \text{if } (m=0) \text{ and } (m'=0) \text{ or,} \\ & \text{if } (m=1) \text{ and } (m-m'=1) \\ \binom{N-1}{m-m'} \frac{Q(m-m',\ m')}{Q(N-1,\ m)} & \text{if } (l=0) \text{ and } (l'=0) \text{ or,} \\ & \text{if } (l>0) \text{ and } (l-l'=1), \text{ and} \\ & \text{if } (m=0) \text{ and } (m'=0) \text{ or,} \\ & \text{if } (m=1) \text{ and } (m-m'=1) \text{ or,} \\ & \text{if } (m \ge 2) \text{ and } (m-m' \ge 1) \text{ or,} \\ & \text{if } (m > N-1) \text{ and } (m'-m \le N-1) \text{ or,} \\ & \text{if } (m \le N-1) \text{ and } (m-m' \le m) \\ 0 & \text{otherwise} \end{cases} \tag{3.3.1}$$

The first and second case represent exactly one transition and more than one transition respectively while the third case represents an invalid transition.

3.4. Arrival Probability (for CS)

Let $P_a(i, j \mid l', m')$ be the probability of transition from (l',m') to (i,j). This is equivalent to the probability of having (i-l') hot-spot cells arrive, and (j-m') non-hot-spot cells enter the buffer. There are two cases for the arrival pattern. Normally, for the system to reach state (i,j), we need

$(i+j)$-$(l'+m')$ cells to arrive, so the arrival probability is $P_a = p_a(i-l', j-m')$. Now consider the case where the transition to state (i,j) results in the buffer becoming full. The determination of the number of cells required for the transition is the same as in the case presented above, except that in the computation of P_a, one must account for the fact that the desired state is reached even if the number of cell arrivals exceeds the exact requirement. This is due to the fact that all the arrivals after the buffer becomes full are discarded.

$$P_a(i, j \mid l', m') = \begin{cases} p_a(i-l', j-m') & \text{if } (i+j) < B \\ \sum_{b=j-m'}^{b \le N-i+l'} \sum_{a=i-l'}^{a \le N-b} \frac{\binom{a}{i-l'}\binom{b}{j-m'}}{\binom{a+b}{i-l'+j-m'}} \cdot p_a(i-l', j-m') & \text{if } (i+j) = B \\ 0 & \begin{array}{l}\text{if } (i-l') < 0 \text{ and,} \\ \text{if } (i-m') < 0 \text{ and,} \\ \text{if } (i+j)-(l'+m') > N\end{array} \end{cases} \tag{3.4.1}$$

3.5. Arrival Probability with Pushout (for HSPO)

Now consider the pushout case under the HSPO scheme. Since pushout will only affect the arrival probability, we use a modified form of P_a, which is called $P_a^{pushout}$. It has a very similar form to P_a, except that the number of hot-spot cells in the final state (after the receive operation) can be less than the number of hot-spot cells in the intermediate state, because some hot-spot cells in the intermediate state may have been pushed out by non-hot-spot cells during the receive operation. The number of hot-spot cells being pushed out must be equal to the number of additional non-hot-spot cells that now fill the buffer. Of course both types of cells can arrive and can be pushed out. However, non-hot-spot cells are discarded only if there are no hot-spot cells in the buffer to be pushed out.

$$P_a^{pushout}(i, j \mid l', m') = \begin{cases} p_a(i-l', j-m') & i+j < B \\ & 0 \le (i+j)-(l'+m') \le N \\ & i \ge l' \\ & j \ge m' \\ \begin{cases} N & \text{if } i = 0 \\ j-m' & \text{if } i > 0 \end{cases} \sum_{b=j-m'} \sum_{a=0}^{N-b} \cdot p_a(a, b) & \\ & i+j = B \\ & 0 \le (i+j)-(l'+m') \le N \\ & i < l' \\ & j \ge l'+m-i \\ \begin{cases} N-i+l' & \text{if } i = 0 \\ B-i-j & \text{if } i > 0 \end{cases} \sum_{b=j-m'} \sum_{a=i-l'}^{N-b} \frac{\binom{a}{i-l'}\binom{b}{j-m'}}{\binom{a+b}{i-l'+j-m'}} \cdot p_a(a, b) & \\ & i+j < B \\ & 0 \le (i+j)-(l'+m') \le N \\ & i \ge l' \\ & j \ge m'-i \\ 0 & \text{otherwise} \end{cases} \tag{3.5.1}$$

4. Performance Evaluation

The performance measures of interest are the cell loss probability L, the switch throughput T, and the average cell delay within the switch D.

4.1 Cell Loss Probability

The cell loss probability is obtained by dividing the average number of cells lost during a time slot by the average number of cell arrivals during the time slot (known as balanced input load [30]), i.e.

$$L_{bal} = L / (p \cdot N) \tag{4.1.1}$$

where L is the probability that one cell is lost per time slot and is defined as follows.

$$L = \sum_{x=2}^{N} p_a(x) \sum_{y=B-x+2}^{B} e(y) \sum_{z=1}^{x-l-(B-y)} p_s(i, j, z)\ (x-z-B+y) \tag{4.1.2}$$

where

$e(y)$ = the steady-state probability of y cells in the buffer at the time of check operation, and

$p_S(i, j, z)$ = the probability that z cells will leave the buffer, given that i hot-spot cells and j non-hot-spot cells are present in the buffer.

4.2 Throughput

The switch throughput, T, is defined as the average number of cells that leave the switch per time slot, where $0 \le T \le N$.

$$T = \sum_{z=1}^{N} \sum_{S_B(i,j)} e(i, j)\, p_s(i, j, z) \tag{4.2.1}$$

where $e(i,j)$ is the steady-state probability of i hot-spot cells and j non-hot-spot cells being in the buffer. $S_B(i, j) = \{i, j \mid i+j \le B,\ 0 \le i \le B \text{ and } 0 \le j \le B\}$ is a set of buffer states with all the possible combinations of i and j.

4.3 Average Cell Delay

The average cell delay, D, within the switch can be computed using the Little's formulae as follows.

$$D = \frac{\sum_{k=0}^{B} k \cdot e(k)}{T / N} \tag{4.3.1}$$

where $e(k)$ represents the steady-state probability of k cells in the buffer.

5. Discussion of Results

In this section we compare the performance of the HSPO scheme with some of the schemes in the literature, and validate the HSPO model with simulation. Figures 5-20 show the numerical results we have obtained from our model and from the simulation program with total run time of 10,000,000 cycles.

5.1 Simulation Experiment Design

As is usual in simulation experiments, we choose to make a single long run, divide it into k portions (after having discarded the initial transient part) and then work with the sample of observations obtained from those portions [36]. Furthermore, high accuracy in the simulation results is required since the performances of some of the schemes to be compared are very close (they sometimes differ in the third decimal place or less). The simulation run is repeated K times with independent random number streams. Let X_{ji} be the ith observation in the jth run. For example, in the jth run, we obtained the batched means, X_{j1}, X_{j2}, ..., X_{jk}.. Then the estimates of the mean and its variance are given by

$$\text{Mean} = M(k) = \frac{1}{K}\sum_{j=1}^{K}\overline{X}_j(k) = \frac{1}{kK}\sum_{j=1}^{K}\sum_{i=1}^{k}X_{ji}$$
$$\text{Variance} = \text{var}(k) = \frac{1}{K-1}\sum_{j=1}^{K}\left[\overline{X}_j(k) - M(k)\right]^2 \qquad (5.1.1)$$

where $\overline{X}_j(k)$ = mean of jth run.

In our experiments, we chose k = 10 (with the first transient portion discarded) and K = 5 runs. We calculate the mean and use the variance to get a confidence interval* of 90% with a typical offset value of approximately ±1% of the mean. Also for a fair comparison, for CP, we choose the buffer size B_n to be 5 (where $B = \sum_{n=1}^{N} B_n$) at each output with 4 outputs, which is the same as the buffer size of 20 for CS, SMA, SMXQ and HSPO. The minimum buffer allocation of SMA, (Q_i) for each output i ($1 \le i \le N$), and maximum queue length of SMXQ, (X_j) for each output j ($1 \le j \le N$), are calculated using the following Latouche's Square Root Formula [16, 17] for obtaining their best performance in the experiment.

$$\begin{aligned}\text{SMA} &: Q_i = \lfloor B / N + \sqrt{N} \rfloor \; \forall i = 3 \;\; (\text{for } B = 20,\ N = 4)\\ \text{SMXQ} &: X_j = \lfloor B / \sqrt{N} \rfloor \; \forall j = 10 \;\; (\text{for } B = 20,\ N = 4)\end{aligned} \qquad (5.1.2)$$

* However, they are not displayed on the performance curves here for the sake of clarity in the graphs.

5.2 Numerical Results

Figures 5 to 8 show the accuracy of the model for CS in terms of buffer occupancy, cell loss probability, throughput and delay. Similarly Figures 9 to 12 show those for HSPO. The simulation data in the above two sets of figures were obtained by simulating five runs with 5,000,000 clock cycles in each run. The comparison for HSPO is done for a fixed h=0.4. In general, the results are in good agreement for both small (B=6) and large (B=20) buffer sizes and for a range of traffic loads. We also observe that throughput and cell loss probability of a shared-buffer switch cannot be improved under non-uniform traffic even if a large buffer is used (Figures 6 and 7), unless HSPO is applied (see Figures 10 and 11). A larger buffer will consequently lead to an adverse increase in delay when the switch is congested. Figures 13 to 16 show the accuracy of the model for HSPO at different values of p_h. The input load, p, was kept constant at 0.7.

The rest of the graphs show comparisons between the different buffer management schemes using simulations only. Figures 17 and 18 show comparisons of the average cell loss probability and throughput respectively for the different schemes with varying traffic load and a fixed hot-spot ratio (h=0.4). In Figure 17, CS allows the buffer to efficiently accommodate cells until the traffic load exceeds 0.6, after which it suffers cell loss almost linearly as the traffic load increases. This indicates that the buffer is congested and any arriving cell will be lost. The other schemes CP, SMA, SMXQ, HSPO perform far better than CS because they are able to avoid buffer hogging. Among all the schemes HSPO performs slightly better, as it can achieve lower loss probability all the way along the curve. Figure 18 shows the throughput, measured by the average number of cells that can leave the switch during each time slot. Again when the traffic load is near 0.6, the switch with CS becomes congested and the throughput is limited constantly with increasing traffic load. Note that SMXQ performs better than CP which is better than SMA at high loads. Figures 19 and 20 show the degree of performance degradation as the hot-spot ratio increases. In essence, the order of the overall performance of each scheme under non-uniform traffic and moderate traffic load can be summarized as follows.

$$\text{CS} \ll \text{CP} \underset{\textit{slightly}}{<} (\text{SMA} \approx \text{SMXQ}) \underset{\textit{slightly}}{<} \text{HSPO}$$

Since the results in the graphs are very close together, Table 1 gives loss probabilities of the various schemes for a selected set of parameter values. It can be seen that the % improvements over CS are quite large.

	HSPO	CP	SMA	SMXQ
Av. Cell Loss Probability	0.0438419841	0.0544162302	0.0454448016	0.0454897222
% Improvement over CS	59.89	0.22	58.43	58.39

Table 1. Numerical values of average loss probability for various schemes and the percentage improvement over CS for $N = 4$, $B = 20$, $p = 0.7$ and $p_h = 0.4$.

6. Conclusions

A new analytical model for evaluating shared-buffered ATM switches operating under a non-uniform (Hot Spot) traffic model has been developed in this paper. This model can handle either simple CS or the HSPO mechanism to manage the buffer during congestion. Although approximations have been used to reduce the number of states in the analytical model, comparison with simulation results show that the model provides accurate results. The model has been used to obtain the cell loss, throughput, and cell delay of a switch for different traffic parameters.

We have compared the performance of the HSPO scheme with other buffer management schemes available in the literature. Because of no buffer sharing, the CP scheme performs badly when traffic load is light, resulting in poor buffer utilization. The CS scheme was shown in [25] to be unacceptable as it is prone to congestion under unbalanced loading conditions. It was also shown that restricted sharing schemes like SMA and SMXQ avoid unfair domination while having the advantage of sharing of buffer space. However, they are not flexible for generic switch design because the amount of buffer restriction has to be determined. Neither of these can respond dynamically to different loads. HSPO can cope dynamically with different degrees of non-uniformity in the traffic, and as in the CS scheme, it still enjoys the gain resulting from statistical buffer sharing. This is important because recent switch designs have capitalised on the advantages gained by using CS. Our results indicate that additional complexity in the switch to allow for reprogramming of the buffer

management scheme as in [21], is not required. HSPO is shown to be superior to the other schemes for all parameter values of interest.

7. References

[1] Y. Oie, T. Suda, M. Masayuki, H. Miyahara, "Survey of The Performance of Nonblocking Switches with FIFO Input Buffers", ICC 90, pp.737-741, 1990.

[2] B. Zhou, M. Atiquzzaman, "Performance Analysis of Finite Output-Buffered Multistage ATM Switching Fabrics", IEEE INFOCOM 95, pp.16-23, 1995.

[3] E. Del Re, R. Fantacci, "Performance Evaluation on Input and Output Queueing Techniques in ATM Switching Systems", IEEE Trans. Comm. Vol. 41, No. 10, pp. 1565-1575, 1993.

[4] A. Jajszczyk, W. Kabacinski, "A Growable Shared-Buffer-Based ATM Switching Fabric", IEEE INFOCOM 93, pp.29-33, 1993.

[5] J. Schormans, C. Phillips, R. Smith, L. Cuthbert, "Analysis of a Finite Shared Buffer", Electronics Letters, Vol. 30, No. 1, pp.8-9, Jan, 1994.

[6] L. Gavrilovska, "Fast Packet Switches with Shared Buffer Memory", ITI 94, 16th International Conference on Information Technology Interfaces, pp. 185-190, 1994.

[7] G. Bianchi, J. Turner, "Improved Queueing Analysis of Shared Buffer Switching Networks", IEEE Trans. on Networking, Vol. 1, No. 4, pp.482-490, 1993.

[8] M. Saleh, M. Atiquzzaman, "Queueing Analysis of Shared Buffer Switches for ATM Networks", IEEE GLOBECOM 94, pp.1070-1074, 1994.

[9] E. Valdimarsson, "Queueing Analysis for Shared Buffer Switching Networks for Non-Uniform Traffic", IEEE INFOCOM 95, pp.8-15, 1995.

[10] M. Akhtar, M. Atiquzzaman, "Performance of Buffered Multistage Interconnection Networks in Non-Uniform Traffic Environment", 7th International Parallel Processing Symposium, California, pp.762-767, April 1993.

[11] S. Hong, G. Perros, H. Yamashita, "A Discrete-Time Queueing Model of the Shared Buffer ATM Switch with Bursty Arrivals", Telecommunication Systems Modeling, Analysis, Design and Management, Vol. 2, No. 1, pp.1-20, Dec. 1993.

[12] H. Yamashita, "Discrete-Time Analysis of a Classified Multi-Server Queue with Burst Arrivals and a Shared Buffer", Queueing Systems Theory and Applications, Vol. 18, No. 1-2, pp.167-182, Oct. 1994.

[13] W. Denzel, A. Engbersen, I. Iliadis, "A Flexible Shared-Buffer Switch for ATM at Gb/s rates", Computer Networks and ISDN Systems, Vol. 27, pp.611-623, 1995.

[14] M. Saleh, M. Atiquzzaman, "Buffer Occupancy in ATM Switches with Single Hot Spot", Electronics Letters, Vol. 31, No. 1, pp.13-15, Jan, 1995.

[15] A. Thareja, S. Tripathi, “Buffer Sharing in Dynamic Load Environment”, IEEE INFOCOM 94, pp.369-375, 1984.

[16] M. Irland, “Buffer Management in a Packet Switch”, IEEE Trans. on Communications, Vol. COM-6(3), pp.328-37, March 1978.

[17] F. Kamoun, L. Kleinrock, “Analysis of Shared Finite Storage in a Computer Network Node Environment Under General Traffic Conditions”, IEEE Trans. on Communications, Vol.28, No.7, pp.992-1003, July 1980.

[18] S. Fong, S. Singh, M. Atiquzzaman, "An Improved Buffer Sharing Scheme for ATM Switches under Bursty Traffic", 19th Australian Computer Science Conference, Melbourne, pp.26-34, Jan, 1996.

[19] J. George, “A Hardware Implementation of a Pushout Scheme for ATM Output Buffer Management”, M.S. Thesis, Computer Engineering, Clemson University, 1993.

[20] M. Katevenis, P. Vatsolaki, A. Efthymiou, “Pipelined Memory Shared Buffer for VLSI Switches”, ACM SIGCOMM 95, Cambridge, USA, pp.39-48, August 1995.

[21] R. Velamuri, P. Landsberg, C. Zukowski, “A Multi-Queue Flexible Buffer Manager Architecture”, IEEE GLOBECOM 93, pp. 1401-1405, 1993.

[22] A. Gravey, G. Hebuterne, "Analysis of a Priority Queue with Delay and/or Loss Sensitive Customers", 7th ITC Seminar, Morristown, Oct. 1990.

[23] L. Tassiulas, Y. Hung, S. Panwar, "Optimal buffer Control During Congestion in an ATM Network Node", IEEE Trans. on Networking, Vol.2, No.4, pp.374-386, August 1994.

[24] M. Atiquzzaman, “Buffer Dimensioning in ATM Networks Using No-priority and Pushout Space Priorities”, IEEE GLOBECOM 95, pp.388-392, 1995.

[25] S. Wei, E. Coyle, M. Hsiao, "An Optimal Buffer Management Policy for High-Performance Packet Switching", IEEE GLOBECOM 91, pp.924-928, 1991.

[26] A. Choudhury, E. Hahne, "Space Priority Management in a Shared Memory ATM Switch", IEEE GLOBECOM 93, Vol. 3, pp. 1375-1383, Dec. 1993.

[27] A. Choudhury, E. Hahne, "Buffer Management in a Hierarchical Shared Memory Switch", IEEE INFOCOM 94, pp.1410-1419, 1994.

[28] I. Cidon, L. Guerin, A. Khamisy, "Optimal Buffer Sharing", IEEE INFOCOM 95, pp.24-31, 1995.

[29] S. Suri, D. Tipper, G. Meempat, “A Comparative Evaluation of Space Priority Strategies in ATM Networks”, IEEE INFOCOM 94, pp.516-523, 1994.

[30] J. Meyer, S. Montagna, R. Paglino, “Dimensioning of an ATM Switch with Shared Buffer and Threshold Priority”, Computer Networks and ISDN Systems, Vol.26, pp.95-108, 1993.

[31] M. Marco, A. Pattavina, “Performance Analysis of ATM Multistage Networks with Shared Queueing under Correlated Traffic”, ITC 14th, pp.601-10, 1994.

[32] J. Turner, “Queueing Analysis of Buffered Switching Networks”, International Teletraffic Congress, Copenhagen, Denmark, pp.35-40, June 1991.

[33] J. Turner, “Queueing Analysis of Buffered Switching Networks”, IEEE Trans. on Comm., Vol. 41, No. 2, pp.412-420, February 1993.

[34] G. F. Pfister, V. A. Norton, "Hot Spot Contention and Combining in Multistage Interconnection Networks", IEEE Transactions on Computers, Vol. C-34, No. 10, pp. 943-948, October 1985.

[35] W. Feller, "An Introduction to Probability Theory and Its Application", Wiley, 1968, New York.

[36] I. Mitrani, "Simulation Techniques for Discrete Event Systems", Cambridge University Press, 1982, Cambridge, New York, chapter 5, pp.86-105.

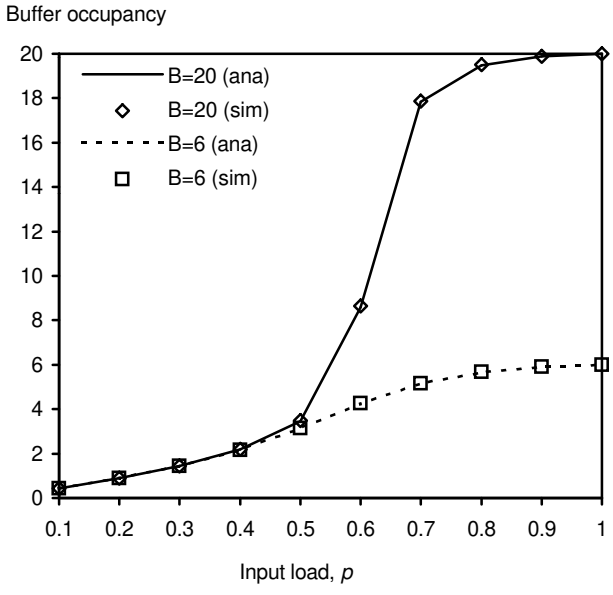

Figure 5. Buffer occupancy of CS hot-spot traffic with varied input load with p_h = 0.4 and N=4.

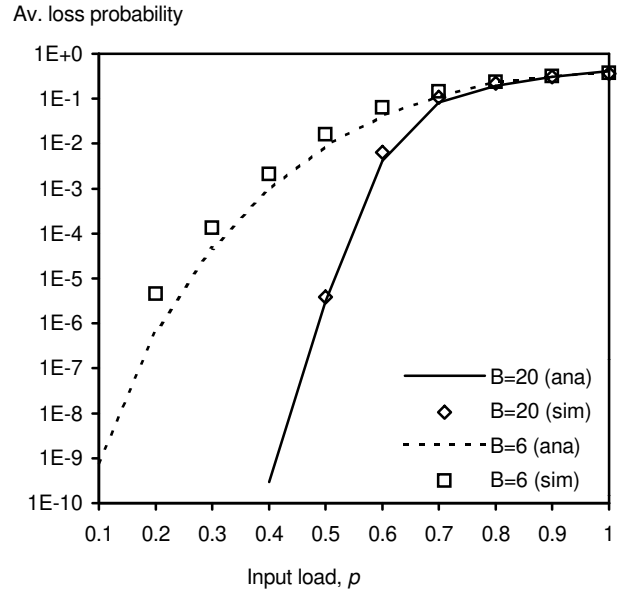

Figure 6. Cell loss probability of CS hot-spot traffic with p_h = 0.4 and N=4.

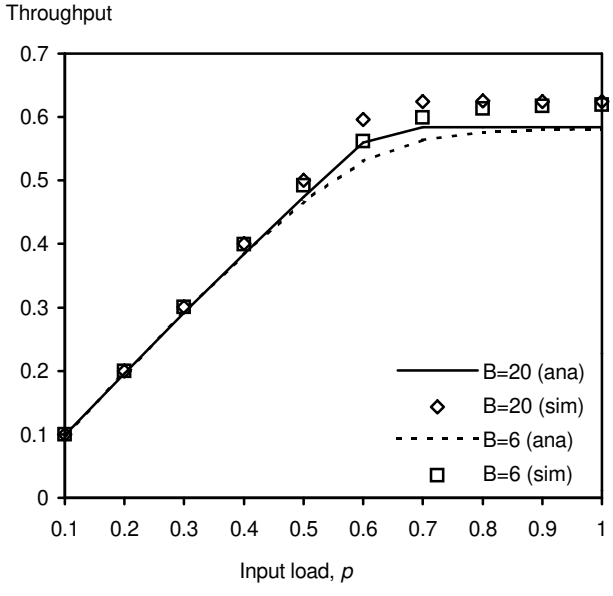

Figure 7. Throughput of CS hot-spot traffic with p_h = 0.4 and N=4.

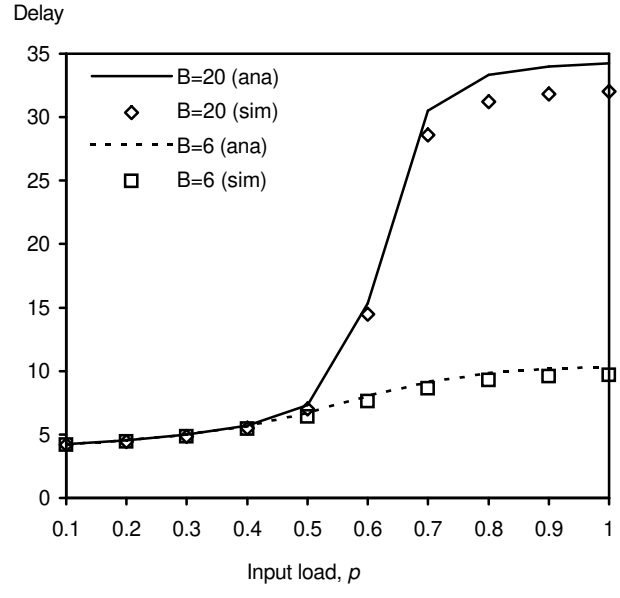

Figure 8. Delay of CS hot-spot traffic with p_h = 0.4 and N=4.

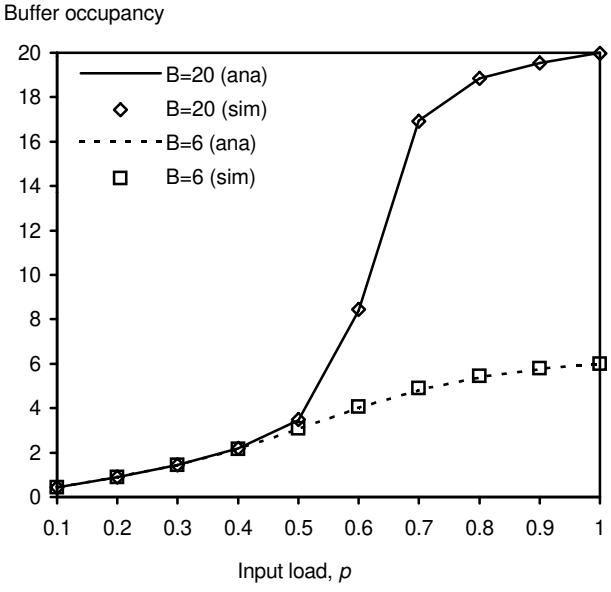

Figure 9. Buffer occupancy of HSPO hot-spot traffic with p_h = 0.4 and N=4.

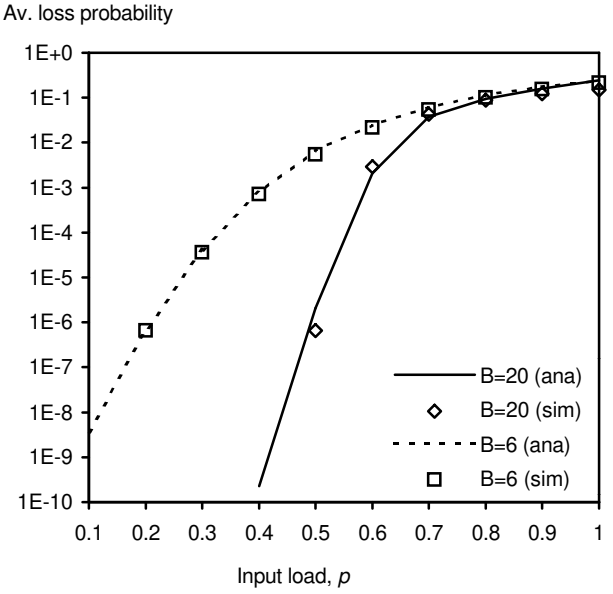

Figure 10. Cell loss probability of HSPO hot-spot traffic with p_h = 0.4 and N=4.

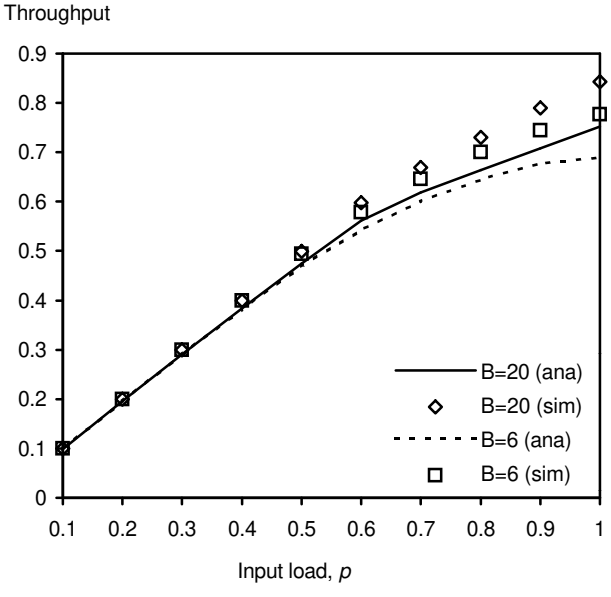

Figure 11. Throughput of HSPO hot-spot traffic with p_h = 0.4 and N=4.

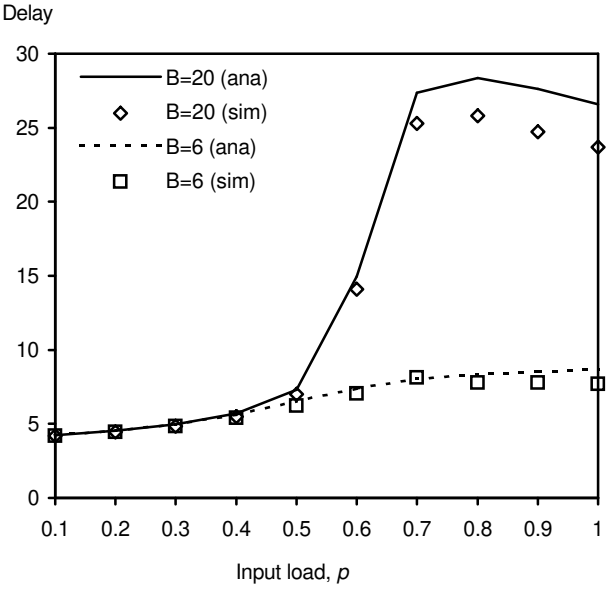

Figure 12. Delay of HSPO hot-spot traffic with p_h = 0.4 and N=4.

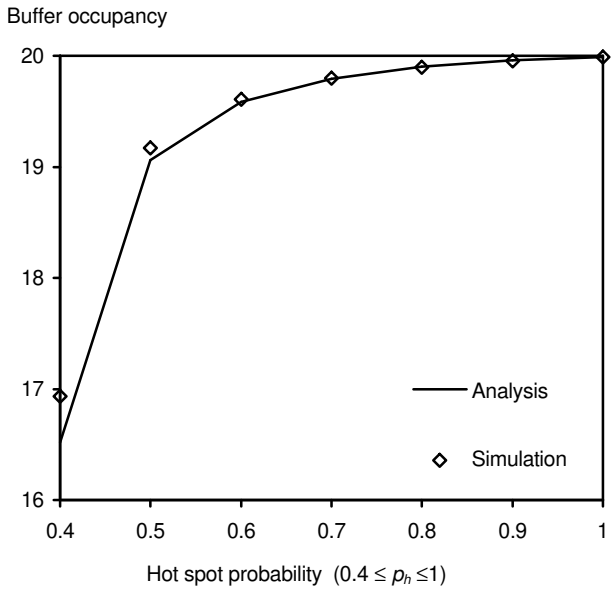

Figure 13. HSPO - Buffer occupancy vs. hot-spot probability with switch size of N=4, B=20, p=0.7.

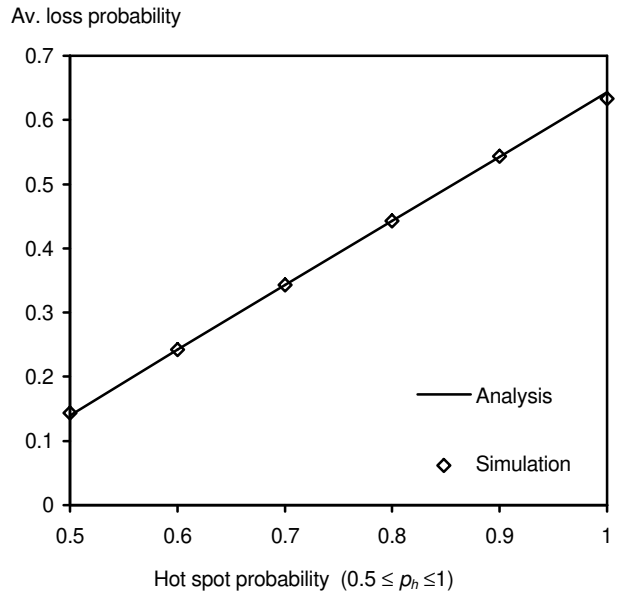

Figure 14. HSPO - Average cell loss probability vs. hot-spot probability with switch size of N=4, B=20, p=0.7.

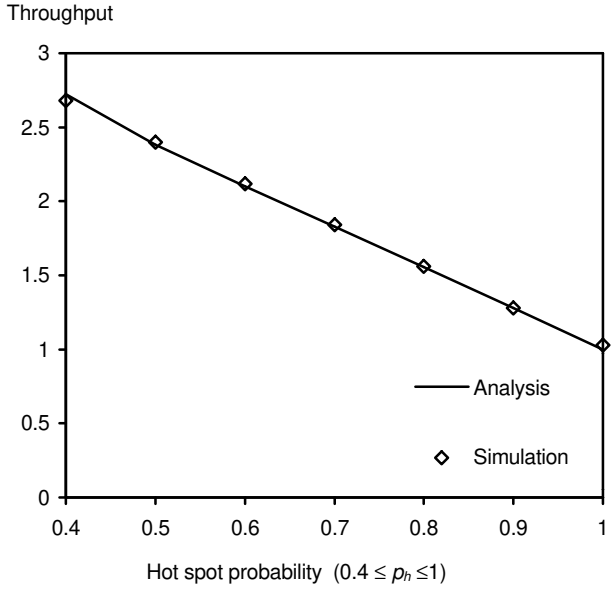

Figure 15. HSPO - Throughput vs. hot-spot probability with switch size of N=4, B=20, p=0.7.

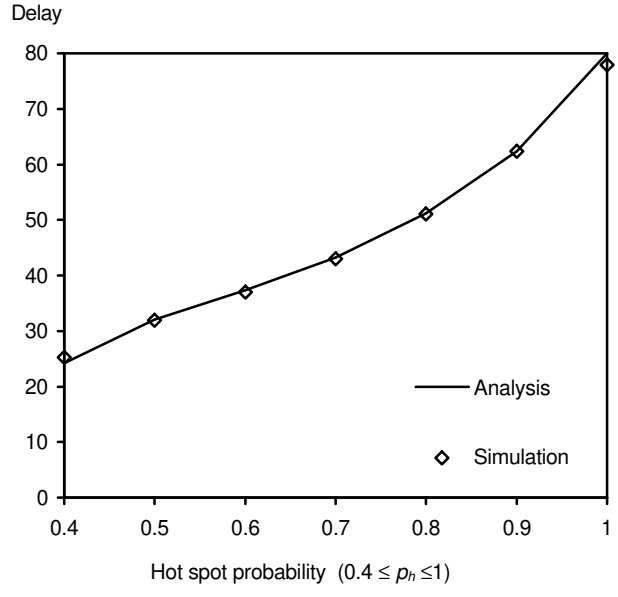

Figure 16. HSPO - Delay vs. hot-spot probability with switch size of N=4, B=20, p=0.7.

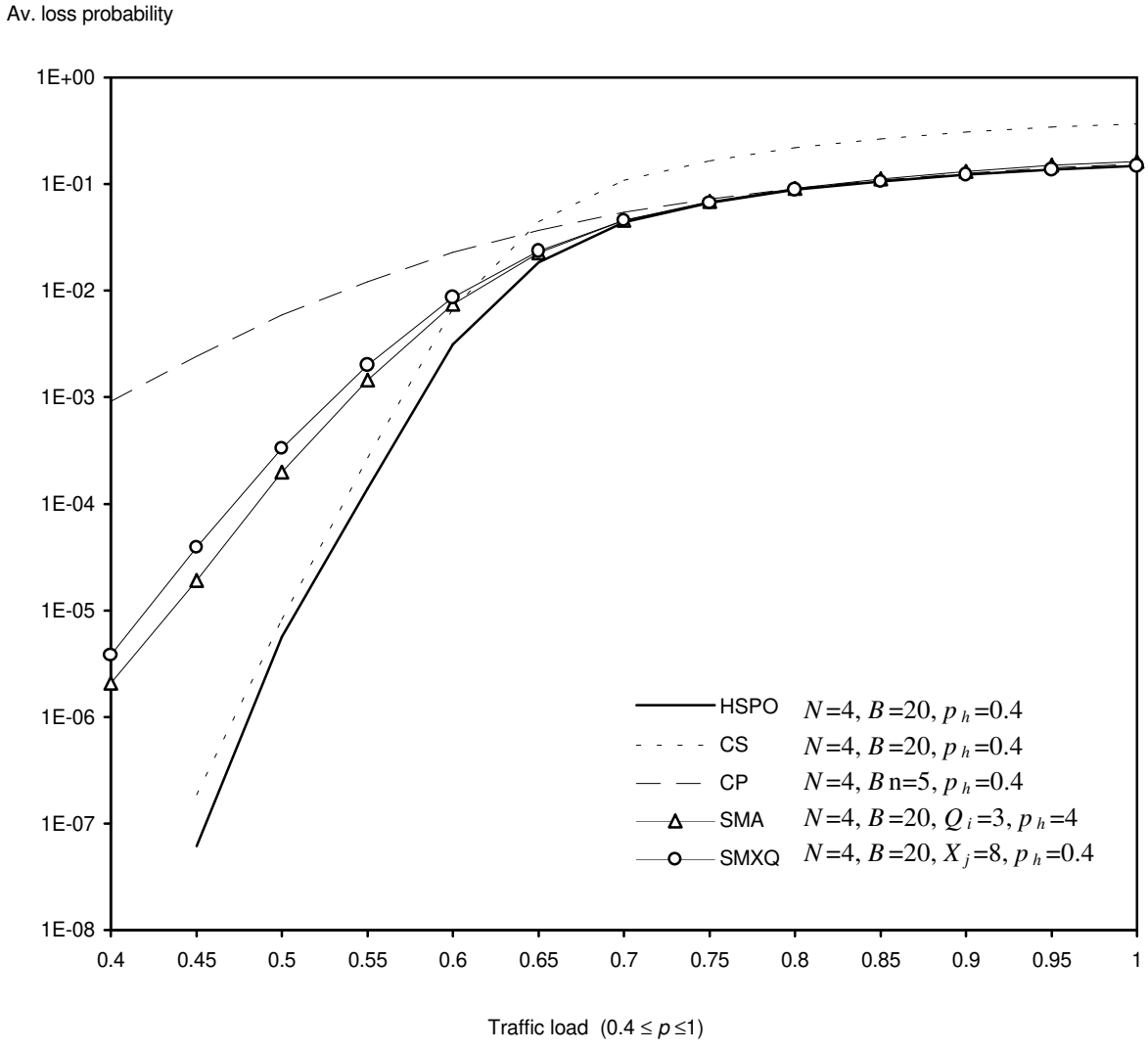

Figure 17. Average cell loss probability vs traffic load for the different schemes.

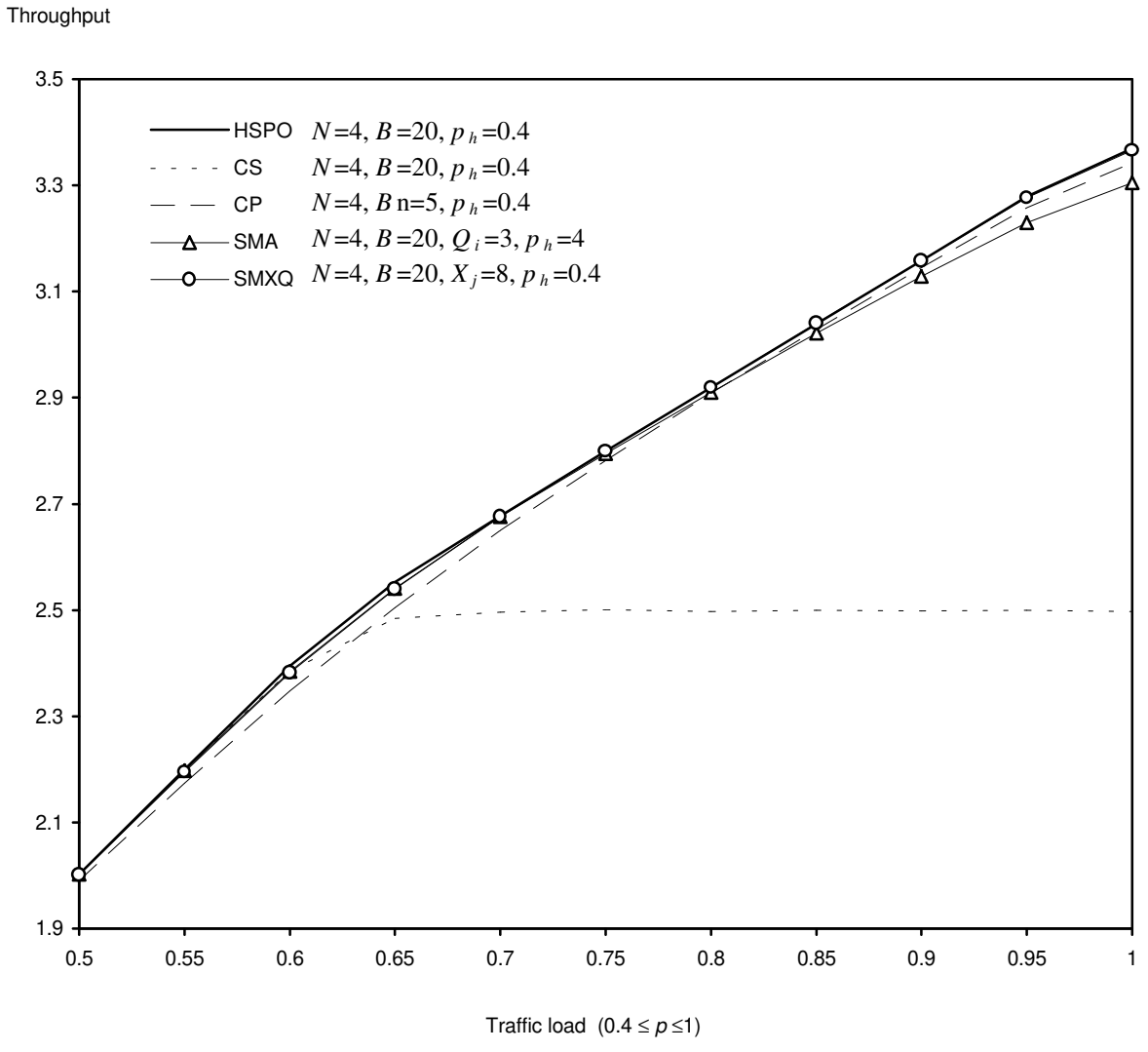

Figure 18. Throughput vs traffic load for the different schemes.

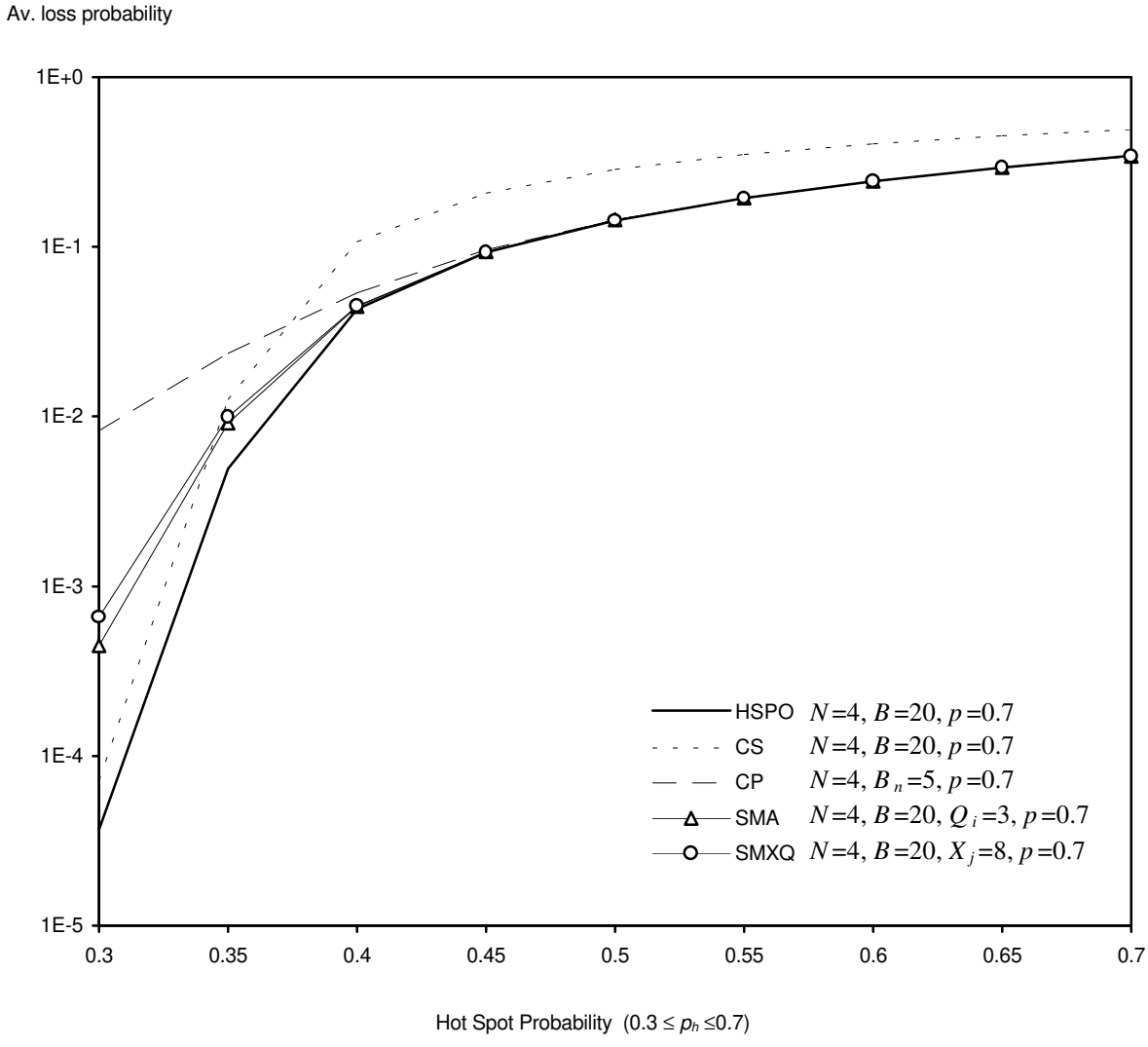

Figure 19. Average cell loss probability vs hot spot probability for the different schemes

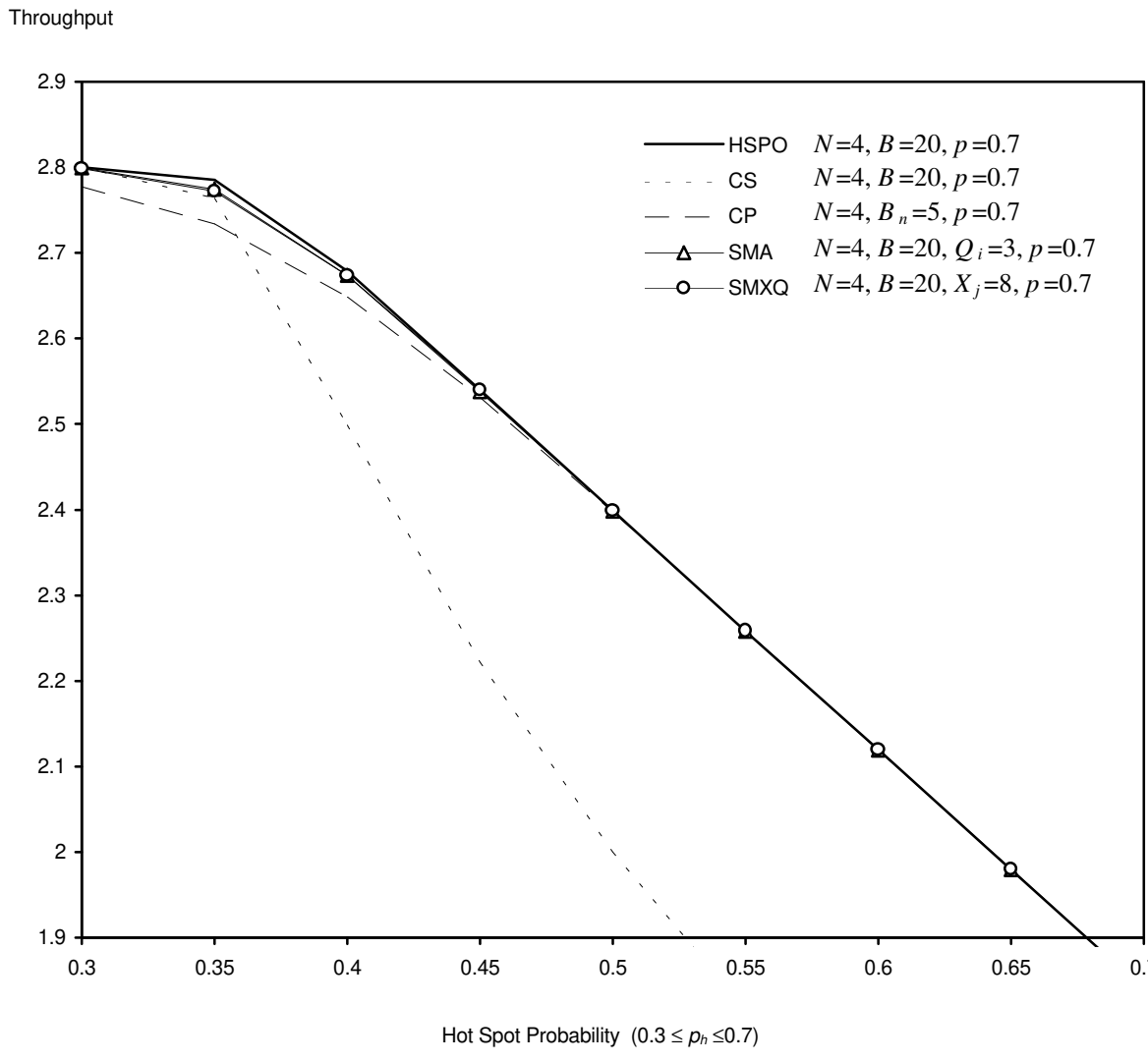

Figure 20. Throughput vs hot spot probability for the different schemes