An Improved Buffer Sharing Scheme for ATM Switches under Bursty Traffic

Simon Fong, Samar Singh

Dept. of Computer Sci. & Comp. Eng.
La Trobe University
Bundoora, Australia, 3083

fong@leopard.cs.latrobe.edu.au
samar@latcs1.lat.oz.au

Mohammed Atiquzzaman

Dept. of Electrical Engineering
Monash University
Clayton, Australia, 3168

atiq@eng.monash.edu.au

Abstract

ATM has been proposed as the technique that will support B-ISDN services of the future. Thus the high-speed fabrics that switch the fixed-size ATM cells at switching nodes in the network, need to have efficient buffer management schemes to control size and therefore cost of buffers. Most of the buffer management schemes proposed to date do not perform well under non-uniform traffic. But it is well accepted by now that future ATM traffic is very likely to be non-uniform, although the nature of the non-uniformity is likely to vary over time. Each of the buffer management schemes proposed for the non-uniform case, performs better than the other for some values of the traffic parameters. In this paper we have proposed a new buffer-management scheme, called Hot-Spot-Push-Out (HSPO) which performs uniformly better than all the previously proposed schemes under bursty traffic. This is demonstrated by simulation.

Keywords ATM, performance evaluation, switching, buffer management, bursty traffic.

1 Introduction

An important way to offer consistent performance by a B-ISDN network in the presence of stochastically varying loads, is the use of an efficient buffer sharing scheme in an ATM switch. Cell buffering is common in most ATM switch architectures to accommodate cells whose service must be delayed due to a contention for the same output port. In practice, the amount of buffer space available in the switch is limited and costly, because it remains a challenge in today's VLSI technology to integrate large buffers into a tiny switch fabric. Thus this critical resource must be used efficiently in order to achieve the best possible switch performance.

Proceedings of the 19th ACSC Conference, Melbourne, Australia, January 31 - February 2 1996.

Previous researchers, have attempted to develop good buffer sharing schemes for ATM switches in general, [1] but none of the schemes to date offers a better solution to all the different traffic load conditions used in the B-ISDN environment. Ideally, a buffer sharing scheme for an ATM switch should satisfy several competing objectives. First, it should provide consistent performance to those applications that would generate varying and non-uniform traffic loads. By consistent performance, we mean that it allows high switch throughput, and low cell loss probability in all traffic conditions, even in the presence of bursty traffic streams. Second, it does not require a large amount of buffer in order to satisfy a set of performance requirements. In other words, it should provide a maximum level of buffer sharing with a minimum amount of buffer, and its utilization must be efficient. Third, which is the most serious issue of all, it should not make the switch congested in a way that most cells destined for the same output port disproportionately occupy the whole buffer. This is known as buffer hogging. Its seriously negative effect is that most of the switch outputs are starved, in spite of having low throughput and high cell loss rate. Fourth, it is highly desirable to have a generic design. That is a design which always works well regardless of the traffic loads, and switch size, and requires little reconfiguration whenever the switch parameters change. All the buffer sharing schemes we have seen for ATM switches have deficiencies with respect to at least one of these objectives.

We will review several traditional buffer sharing schemes which have been proposed and widely studied by various groups, plus the new scheme proposed in this paper. Our purpose here is to illustrate these distinctly different schemes, identify their strengths and weaknesses so that we can make use of them in constructing a new scheme, which will be described in detail later.

CS - Complete Sharing

The complete sharing scheme (CS) allows an unrestricted sharing of the entire buffer among the outputs [2] and its benefit in best utilization of buffer resource is demonstrated by Denzel et al. in [3]. They obtained good throughput performance with CS when the incoming traffic is assumed statistically uniform. However, under non-uniform traffic, arriving cells are accepted as long as there is spare buffer space available. CS cannot distinguish outputs which are at or near buffer saturation, resulting in the buffer space being monopolized by cells queuing for some highly utilized output while the other outputs are starved, since they are allowed little or no buffer space. This situation leads to a decrease in the average throughput with increasing load. CS has been shown to perform poorly in [4] when the incoming traffic is asymmetrically distributed across the outputs.

CP - Complete Partitioning

The complete partitioning scheme permanently partitions the entire buffer into N buffers of sizes B_i such that $\sum_{i=1}^{N} B_i = B$. No buffer is shared, so they are intrinsically fair. The advantage of CP is that, it is able to achieve relatively high throughput at very high traffic load [5], since a strict partitioning of the buffers may prevent some unbalanced cells from hogging all the buffers. However, the partitioned buffers cannot enjoy the statistical gains of sharing. Therefore, one obvious drawback is that it cannot take advantage of buffer sharing to increase throughput at light and moderate traffic loads [5]. Another drawback is that the buffers are used inefficiently. Each output has a fixed number of buffer spaces allocated to it. Once these spaces are filled, cells are dropped even though there may be free buffers associated in other outputs. It is known to waste buffer space up to a factor of five [6]. There is an analytical model for CP in [7].

Since CS and CP are problematic in certain situations, some notable modifications of CS and CP employing additional restrictions have been developed, such as sharing with minimum allocation (SMA) [8] and sharing with maximum queue length (SMXQ) [1].

SMA - Sharing with Minimum Allocation

In SMA, a minimum number of buffers are allocated to each output and the remaining buffer space is completely shared among all the outputs. Each output has a guaranteed amount of buffer resource, so in the worst-case non-uniform traffic, the overall switch performance would not suffer so badly. This approach is a mix of partitioning and sharing buffers, having the advantages of CS and CP.

SMXQ - Sharing with Maximum Queue

In SMXQ, a limit is imposed on the maximum number of buffers to be allocated to each output. This prevents any one of the output queues from dominating the whole buffer. Similar to SMA, it obtains the advantage of CS in sharing the buffer, and the advantage of CP in avoiding buffer hogging.

Recently, Tipper et al. [9] presented a simulation based performance comparison for various traffic source models for the above buffer sharing schemes. The study concluded that SMA or SMXQ may provide the best trade-off between throughput and buffer size. However, the amount of minimum allocation or the maximum queue length for using any of these restricted buffer sharing has to be resolved by using Latouche's square root rule [10], whenever the switch configuration (total buffer spaces and switch size) changes. This may not be appropriate for a generic scalable switch design. Further, each of these two schemes is better than the other for some range of the traffic parameters. This led [11] to propose a VLSI switch design in which the buffer management scheme is reprogrammable, even though a substantial amount of the circuitry is devoted to this aspect.

Nowadays, as indicated by the latest ATM switch design [3], a switch with CS is modular and scalable and has potential for very high-speed implementation on a single VLSI chip with an aggregate throughput of more than 12 Gb/s. However, it must implement a congestion control mechanism when the load is unbalanced yielding a Hot Spot (which is a non-uniform output reference pattern) in the switch, or the performance will suffer [12].

HSPO - Hot Spot Push Out

This motivated us to propose a new scheme, HSPO in [13] which is designed to remedy the shortages of all the schemes mentioned above, in that it has a pushout mechanism which purges Hot Spot cells from the buffer to give priority to other incoming cells entering to the buffer. This is how it works.

Rule 1: HSPO has an identical switch layout to CS. There is a shared buffer in the switch and it can be shared among all the outputs.

Rule 2: When the buffer is full and a new cell arrives, it will pushout one of the cells which are destined to the hot-spot output (usually the output with the most cells in the buffer), to make room for the arriving cell, if that is destined to a non-hot-spot output.

Rule 3: When the buffer is full and the new arriving cell is a Hot Spot cell, then it is dropped.

HSPO has many advantages. This scheme gives a maximum level of buffer utilization like CS does, and fairness in sharing the buffer among all the outputs is ensured as in CP. Therefore, less buffer spaces are required than CP; buffer hogging is avoided because pushing out the cells that occupy most of the buffer gives space priority to other cells, and consequently switch throughput is increased. It should have lower cell loss rates than other schemes because cells can only be lost when the entire buffer becomes full. Also unlike SMA and SMXQ, which are claimed to be the better than CS and CP [6, 9], HSPO does not need to predetermine the sharing 'limits' for different switch configurations.

Related Work

The idea of pushout is not entirely new. This approach is pioneered by Gravey and Hebuterne [14], and later Panwar et al. [15] applied it to study the problem of optimal buffer space priority control for cells of several traffic classes. Wei et al., [16] proposed a similar dynamic buffer allocation scheme called Drop-on-Demand, which has cell purging action to prevent buffer hogging, and they showed that it yields better performance than CS and CP. However, a comprehensive comparison of all schemes under bursty traffic has not yet been done. Recently in [17], Cidon et al., have proved that the optimal buffer management scheme in shared memory switches is of pushout type, but their optimization study is based on a 2-ported switch and uniform traffic or symmetric system (equal traffic on all ports). Choudhury and Hahne proposed a similar buffer management technique in [18] called Delayed Pushout which uses a pushout mechanism within each switch component and a backpressure mechanism between switch stages.

A mathematical model for HSPO with Bernoulli inputs and a single hot-spot was developed in [13] and validated using simulation. [13] also showed that HSPO out-performs other schemes uniformly in the possible range of traffic loads with Bernoulli inputs and one stationary hot-spot.

Here, our objective is to examine how bursty traffic contributes to performance degradation of all these schemes including HSPO. In bursty traffic, even if all the input sources have the same burstiness and are statistically distributed across the output ports in a long run, the loads will still differ temporally. The switch may experience momentary congestion during a burst. Bursty inputs should give rise to transient hot-spots that move around. We want to see how the scheme works under shifting bursts.

Section 2 defines the ATM switch assumptions. The bursty traffic model is described in Section 3. Section 4 describes the simulation setup. Section 5 presents numerical results showing that HSPO is superior to the other schemes even in bursty traffic. The last section contains the conclusion.

2 Assumptions of the Shared Buffer ATM Switch

The system under study is an ATM switch operating in the presence of bursty traffic.

1. *N*x*N* non-blocking synchronous ATM switch with shared buffer of capacity of *B*. See figure 1.

2. Switching of a cell from an input port to an output port is tag-based, with temporary internal storage provided by the shared buffer.

3. 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, operation of the switch can be viewed, effectively, as a sequence of three primitive operations.

(1) **Send** For each output port, if there is any cell in the buffer tagged for that port, the cell will leave through it. 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 number of cells that can leave the buffer during a time slot may be up to *N*, the number of output ports. At most one cell can leave through each output port at a time slot.

(2) **Arrive** Bursty input streams are generated on the input ports. (See Section 3 for details). A maximum of *N* cells can be generated in all the input ports, during each time slot, and each input port receives at most one cell.

(3) **Receive** Each input port is scanned and incoming cells are accepted into the buffer or rejected (lost) according to the following criteria: A cell is accepted as long as space remains in the buffer. Otherwise, Rule 2 and Rule 3 are applied.

4. If the buffer is not empty, at least one cell will leave in the next time slot. No cells will leave in the next time slot only if the buffer is empty. Because of this, it follows that at least one cell can come into the buffer in every time slot.

3 Traffic Model

A simple two-state Markov chain model is adopted to represent the bursty cell arrival process on each input. This approach for modelling bursty traffic has been used in [19, 20]. It stochastically alternates between an active state and an idle state with transition probabilities t_{00}, t_{01}, t_{10}, t_{11}. A cell arrival occurs at the utilization rate of p during a time slot if the Markov chain is in active state at the beginning of that time slot, and no arrival occurs otherwise. L is defined as the average burst length.

$$t_{01} = \frac{p}{L \cdot (1 - p)}, \quad t_{00} = 1 - t_{01} \qquad t_{10} = \frac{1}{L}, \quad t_{11} = 1 - t_{10}$$

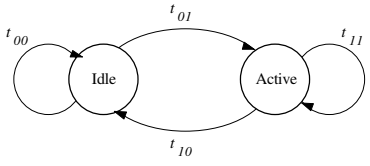


Figure 2. 2-state Markov model.

We assume that cell arrivals on the inputs of the switch are generated by N independent sources with identical statistical characteristics. The sources are uniform geometrically bursty traffic as described above. The maximum number of cells that arrive at the buffer in a time slot is equal to the number of inputs which are in the active state at the beginning of that time slot.

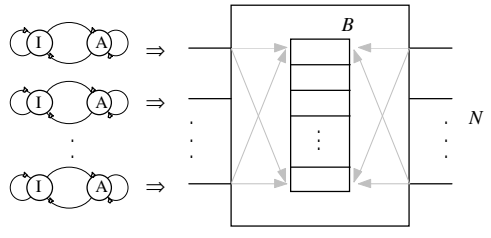


Figure 3. Bursty input streams to ATM switch.

All the cells belonging to the same burst arriving on an input are destined to the same output. The destination is chosen independently of other bursts and of the input on which the burst is arriving, and all destinations are equiprobable with probability $1/N$. The destinations are uniformly distributed between bursts so that all outputs receive an equal number of cells on average. An example is shown in figure 4.

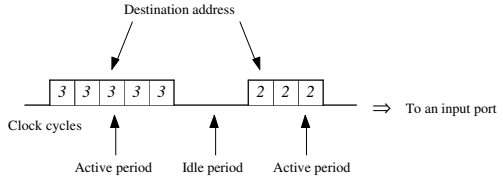


Figure 4. Bursts of cell arrival to input.

4 Simulation Setup

To cope with the high cost of long simulation runs, we choose to make a single long run, divide it into k portions (after having discarded the transient part) and then work with the sample of observations obtained from those portions [21]. 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 a 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

$$\text{Mean} = \mathrm{M}(k) = \frac{1}{K}\sum_{j=1}^{K}\overline{X}_j(k) = \frac{1}{kK}\sum_{j=1}^{K}\sum_{i=1}^{k}X_{ji}$$

$$\text{Variance} = \mathrm{var}(k) = \frac{1}{K-1}\sum_{j=1}^{K}\left[\overline{X}_j(k) - \mathrm{M}(k)\right]^2$$

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

In our experiments, we choose k = 10 (with the first warm-up portion discarded) and K = 5 runs, calculate the mean and use the variance to get a confidence interval* of 90%. The length of each simulation run is 5,000,000 cycles. Also for a fair comparison, we choose the buffer size for CP, B_i to be 5, at each output with a total of 4 outputs, which is equivalent to the buffer size of 20 for CS, SMA, SMXQ and HSPO. The minimum buffer allocation for SMA, Q_i for each output i ($1 \le i \le N$), and maximum queue length for SMXQ, X_j for each output j ($1 \le j \le N$), are calculated using the following Latouche's Square Root Formula for giving their best performance in the experiment [10].

$$\text{SMA}: Q_i = \lfloor B / N + \sqrt{N} \rfloor \forall i = 3 \ (\text{for } B = 20, N = 4)$$

$$\text{SMXQ}: X_j = \lfloor B / \sqrt{N} \rfloor \forall j = 10 \ (\text{for } B = 20, N = 4)$$

There are two main groups in our experiments. Figures 5-8 compare all the schemes under conditions of varying burstiness with fixed utilization rate. Figures 9-10 show the effect of increasing utilization rate on all the schemes with a fixed burst length. The fixed average burst length chosen is 8, since we want the ratio of average burst length to the number of buffers per output to be 2. At this ratio it is likely to have eighty percent of the buffers filled with cells that are destined to only one output (hot spot) at a high utilization rate. Such congestion should be sufficient to plague most sharing schemes [4]. Further, very long burst lengths (L = 11 to 20) are used in figures 7 and 8. One may argue that this extreme condition may not be realistic when we take into account the impact of a flow-control procedure for ATM networks. Our purpose is just to illustrate the fact that CP does really out perform CS under very heavy traffic [5].

* The confidence intervals are not displayed on the performance curves here for the sake of clarity in the graphs.

5 Simulation Results

In figure 5, the average loss probabilities of all the schemes are compared under bursty traffic with varying burst lengths. A longer burst length obviously yields higher congestion and more severe non-uniformity of traffic load. As the results show, loss probability is proportional to the degree of sharing of a scheme in case of bursty traffic. It was first observed in [20] that buffer-sharing is an effective way for dealing with bursty traffic. CP allows no sharing at all, so it has the highest loss rate. SMA has certain amount of buffers dedicated to each output, just like CP but less in amount. It has the second highest loss rate. SMXQ permits some degree of sharing, but not as much as CS, and so has respectively low loss rate. Similarly, HSPO allows the whole buffer to be shared plus the pushout function dynamically purges unfair cells. It achieves the lowest loss rate even in bursty traffic in which hot-spots keep shifting around continuously. It performs uniformly better all the way through. However CS continues to degrade further and further as the switch becomes more congested. This is due to the buffer hogging problem, and CS is incapable of handling it. Thus, some control in buffer-sharing is essential to guarantee satisfactory performance.

Figure 6, shows the throughput of all schemes with varying burst lengths. Throughput is inhibited when burst length is large and the switch becomes congested. Similar interpretations drawn from figure 5, apply here. Schemes with high sharing capacity give better throughput in bursty traffic, except that CS degrades severely when congestion increases. One interesting thing observed here is that SMA and SMXQ perform very well with short burst length. That is when the switch is not congested. Under this condition, CS and HSPO perform exactly the same because the push-out function associated with HSPO is not invoked.

The same experiment is carried out in figures 7 and 8, except that larger burst lengths are used. We want to show that CP eventually out performs CS when the switch is heavily congested. Also, HSPO is shown to perform best over the other schemes under these extreme conditions.

Figures 9 and 10 are performance comparisons of all schemes with fixed burst length of 10, and varying utilization rate (0.5 to 1.0) in terms of average loss probability and throughput respectively. The exact meaning of this parameter is explained in section 3. The long term average utilization of the input is lower. The results are very similar to those obtained in figures 5-8. It suggests that the effect of burstiness is more or less similar to the effect of utilization rate, as long as the switch remains at a similar level of congestion for our comparison.

Although HSPO so far has been shown to be uniformly superior in figures 5-10, the gain in better cell loss rate and throughput is obtained at the price of having a longer cell delay. This is due to the fact that HSPO allows the whole buffer to be shared. Intuitively, having more cell buffering will lead to longer cell delay. CP has the best cell delay, because no buffer is shared. Then come SMA, SMXQ respectively. CS has the worst delay and HSPO can slightly perform better than CS because congestion is relieved by HSPO. See figure 11 and figure 12. Given that HSPO has certain advantages in cell loss rate and throughput in all kinds of traffic, and is generic in design, it is still a good candidate in implementing shared-buffer ATM switches or ATM switches with shared output queues, which is the latest switch design that [3] capitalizes on.

It is obvious that the relative performances of these schemes are heavily dependent upon the ratio of the average burst length of traffic to the number of buffers per output (ie. dependent upon the combination of switch size, number of buffers, and average burst length of traffic). Our work presented here is just a typical example of how well each schemes performs under bursty traffic. In general, we may summarize some important observations in table 1.

7 Conclusions and Future Research

Burstiness is known to be a major factor in determining the performance of the various buffer sharing schemes. This paper focuses on a performance comparison of CS, CP, SMA, SMXQ and HSPO schemes in shared-buffer ATM switches under bursty traffic conditions. Specifically, we investigate how well HSPO performs in bursty traffic since HSPO was proved in our previous study to be uniformly superior in non-uniform traffic with a single fixed hot-spot associated with an output port. The simulation results of this paper show that HSPO can also achieve the best performance out of these schemes in the presence of bursty traffic. As might be expected, the better buffer utilization leads to longer delays, for cells that might have otherwise been lost. HSPO thus appears to be a robust scheme that performs well in all situations. It can do without the reprogramming suggested in [11].

We are extending this work by developing a mathematical model for performance evaluation.

References

[1] 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.

[2] Yamashita, H. Perrors, "Performance Modelling of a Shared Buffer ATM Switch Architecture", *Proc. of 13 Int. Teletraffic Cong.: Teletraffic and Datatraffic in a Period of Change, ITC-13. pp. 993-998.*

[3] 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.

[4] Jeffrey W. Causey, Hyong S. Kim, "Comparison of Buffer Allocation Schemes in ATM Switches: Complete Sharing, Partial Sharing, and Dedicated Allocation", *Proc. of ICC '94*, pp. 1164-1168, 1994.

[5] A. Thareja, S. Tripathi, "Buffer Sharing in Dynamic Load Environment", *Proc. of IEEE INFOCOMM '84*, pp. 369-375, 1984.

[6] N. Endo, T. Ohuchi, T. Kozaki, H. Kuwahara and M. Mori, "Traffic Characteristic Evaluation of a Shared Buffer ATM Switch", *Proc. of IEEE GLOBECOM '90*, vol. 3, pp. 1913-1918, 1990.

[7] A. Mahapatra, "Analysis of Output Buffer Management Strategies in B-ISDN", M. S. Thesis Computer Eng., Clemson University, 1992.

[8] M. L. Irland, "Buffer Management in a Packet Switch", *IEEE Trans. Communication*, vol. COM-26, no. 3, pp. 328-337, Mar. 1978.

[9] D. Tipper, S. Bhagwat, K. Balakrishnan, A. Mahapatra, "Comparative Evaluation of Output Buffer Management Schemes in ATM Networks", *Proc. of IEEE ICC '94*, pp. 1174-1178, 1994.

[10] G. Latouche, "Exponential Servers Sharing a Finite Storage: Comparison of Space Allocation Policies", *IEEE Trans. on Communications*, vol. COM-28, no. 6, pp. 910-915, June 1980.

[11] R. Velamuri, P. Landsberg, C. Zukowski, "A Multi-Queue Flexible Buffer Manager Architecture", *Proc. of IEEE GLOBECOM '93*, pp. 1401-1405, 1993.

[12] M. Saleh, M. Atiquzzaman, "Buffer Occupancy in ATM Switches with Single Hot Spot", *Electronics Letters*, Vol. 31, Iss. 1, pp. 13-15, Jan. 1995.

[13] S. Fong, M. Atiquazzaman, S. Singh, "An Improved Buffer Sharing Scheme for ATM Switches under Non-uniform Traffic", Tech. Report Computer Eng., La Trobe University, 1995.

[14] A. Gravey and G. Hebuterne, "Mixing time and loss priorities in a single server queue,", *in Proc. 13th ITC*, Copenhagen, June 1991, pp. 147-152.

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

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

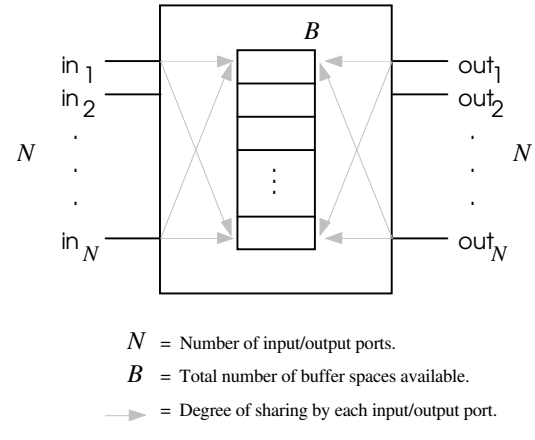
[17] I. Cidon, A. Khamisy and L. Georgiadis, "Optimal Buffer Sharing", *IEEE INFOCOM '95* , pp. 24-31, 1995.

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

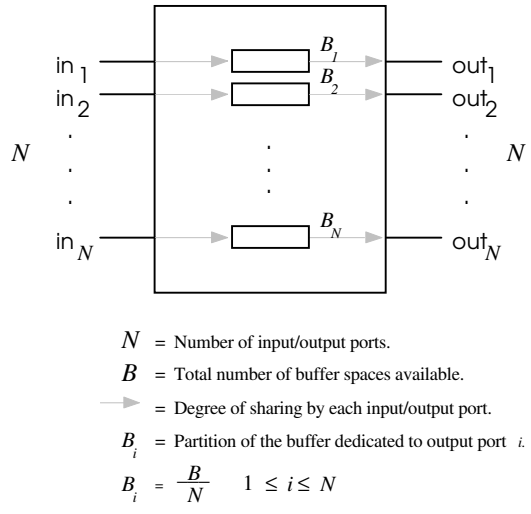
[19] F. Bonomi, S. Montagna, R. Paglino, "Busy Period Analysis for an ATM Switching Element Output Line", *Proc. of IEEE INFOCOM '92*, pp. 544-551, 1992.

[20] S. Liew, "Performance of Input-Buffered and Output-Buffered ATM Switches under Bursty Traffic: Simulation Study", *Proc. of IEEE INFOCOM '90*, pp. 1919-1925, 1990.

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

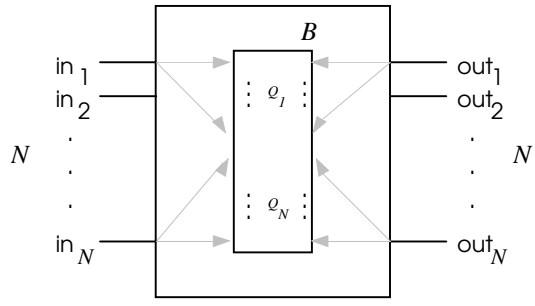


(a) CS



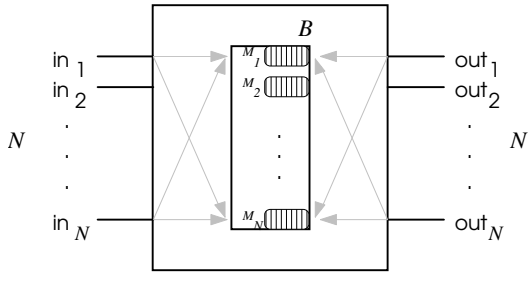
N = Number of input/output ports.
B = Total number of buffer spaces available.
= Degree of sharing by each input/output port.
B_i = Partition of the buffer dedicated to output port i.
$B_i = \frac{B}{N}$ $1 \le i \le N$

(b) CP



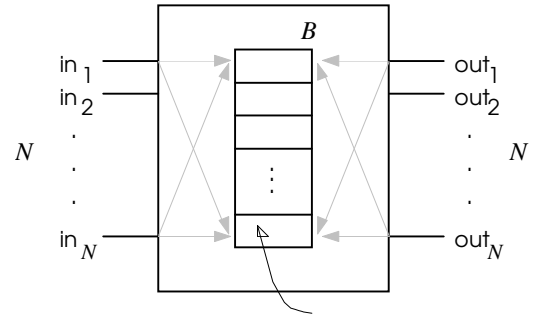
N = Number of input/output ports.
B = Total number of buffer spaces available.
= Degree of sharing by each input/output port.
Q_i = Maximum queue that can be shared by port i.
$1 < Q_i < B$ $1 \le i \le N$

(d) SMXQ



N = Number of input/output ports.
B = Total number of buffer spaces available.
= Degree of sharing by each input/output port.
M_i = Minimum allocation of buffer for output port i.
$0 < M_i < \frac{B}{N}$ $1 \le i \le N$

(c) SMA



HSPO mechanism implemented in the buffer.

N = Number of input/output ports.
B = Total number of buffer spaces available.
= Degree of sharing by each input/output port.

(e) HSPO

Figure 1. NxN Shared-buffer ATM switch with (a) CS scheme, (b) CP scheme, (c) SMA schem, (d) SMXQ scheme and (e) HSPO scheme.

Traffic	Parameters	Loss probability comparison of the schemes
Moderate	p = 0.5 to 0.75 or L = 4 to 7	CP > SMA > SMXQ > CS > HSPO
Heavy	p = 0.8 to 0.95 or L = 4 to 7	CP > CS > SMA > SMXQ > HSPO
Very heavy	p = 1.0 or L = 13 to 20	CS > CP > SMA > SMXQ > HSPO

Table 1. Loss probability summary of all schemes.

Switch and traffic parameters used.

schemes	configuration
HSPO	$N = 4,\ B = 20,\ p = 0.9,\ L = 2,3 \ldots 10$
CS	$N = 4, B = 20,\ p = 0.9,\ L = 2,3 \ldots 10$
CP	$N = 4, B_a = 5,\ p = 0.9,\ L = 2,3 \ldots 10$
SMA	$N = 4, B = 20,\ Q_i = 3,\ p = 0.9,\ L = 2,3 \ldots 10$
SMXQ	$N = 4, B = 20,\ X_i = 8,\ p = 0.9,\ L = 2,3 \ldots 10$

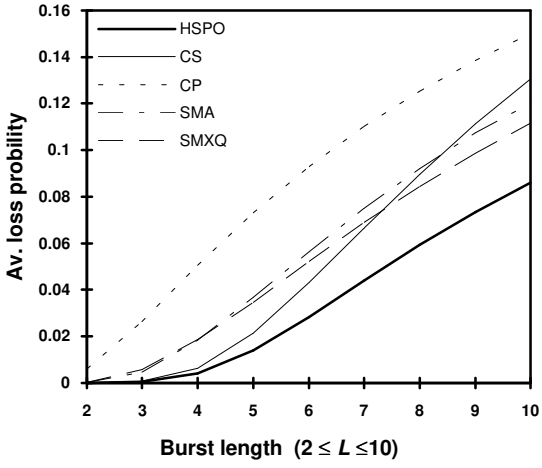


Figure 5. Performance comparison of all schemes under bursty traffic with various burst lengths - Average cell loss probability vs burst length. t_{01}=0.7, t_{00}=0.3 and 5,000,000 clock cycles in each run with 5 runs.

Switch and traffic parameters used.

schemes	configuration
HSPO	$N = 4,\ B = 20,\ p = 0.9,\ L = 11,12 \ldots 20$
CS	$N = 4, B = 20,\ p = 0.9,\ L = 11,12 \ldots 20$
CP	$N = 4, B_a = 5,\ p = 0.9,\ L = 11,12 \ldots 20$
SMA	$N = 4, B = 20,\ Q_i = 3,\ p = 0.9,\ L = 11,12 \ldots 20$
SMXQ	$N = 4, B = 20,\ X_i = 8,\ p = 0.9,\ L = 11,12 \ldots 20$

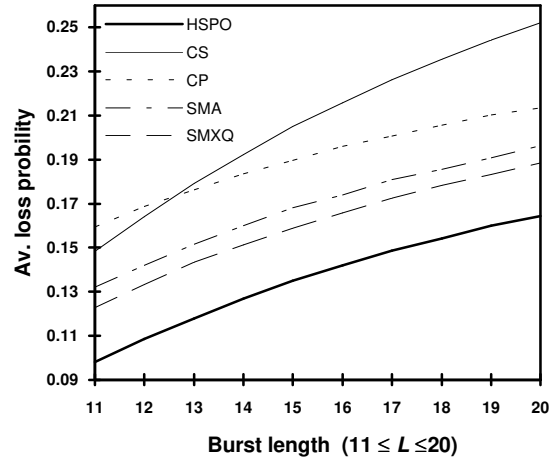


Figure7. Performance comparison of all schemes under bursty traffic with various long burst lengths - Average cell loss probability vs burst length. t_{01}=0.7, t_{00}=0.3 and 5,000,000 clock cycles in each run with 5 runs.

Switch and traffic parameters used.

schemes	configuration
HSPO	$N = 4,\ B = 20,\ p = 0.9,\ L = 2,3 \ldots 10$
CS	$N = 4, B = 20,\ p = 0.9,\ L = 2,3 \ldots 10$
CP	$N = 4, B_a = 5,\ p = 0.9,\ L = 2,3 \ldots 10$
SMA	$N = 4, B = 20,\ Q_i = 3,\ p = 0.9,\ L = 2,3 \ldots 10$
SMXQ	$N = 4, B = 20,\ X_i = 8,\ p = 0.9,\ L = 2,3 \ldots 10$

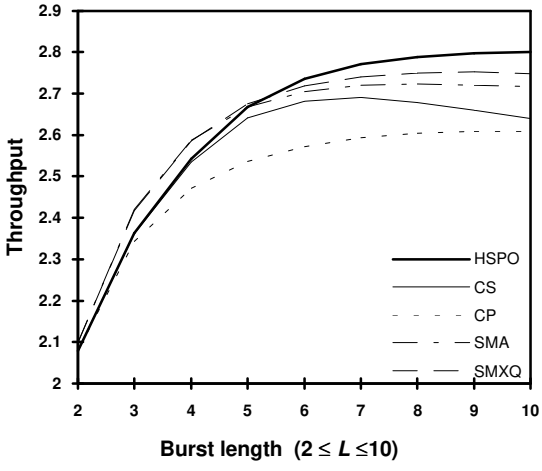


Figure 6. Performance comparison of all schemes under bursty traffic with various burst lengths - Throughput vs burst length. t_{01}=0.7, t_{00}=0.3 and 5,000,000 clock cycles in each run with 5 runs.

Switch and traffic parameters used.

schemes	configuration
HSPO	$N = 4,\ B = 20,\ p = 0.9,\ L = 11,12 \ldots 20$
CS	$N = 4, B = 20,\ p = 0.9,\ L = 11,12 \ldots 20$
CP	$N = 4, B_a = 5,\ p = 0.9,\ L = 11,12 \ldots 20$
SMA	$N = 4, B = 20,\ Q_i = 3,\ p = 0.9,\ L = 11,12 \ldots 20$
SMXQ	$N = 4, B = 20,\ X_i = 8,\ p = 0.9,\ L = 11,12 \ldots 20$

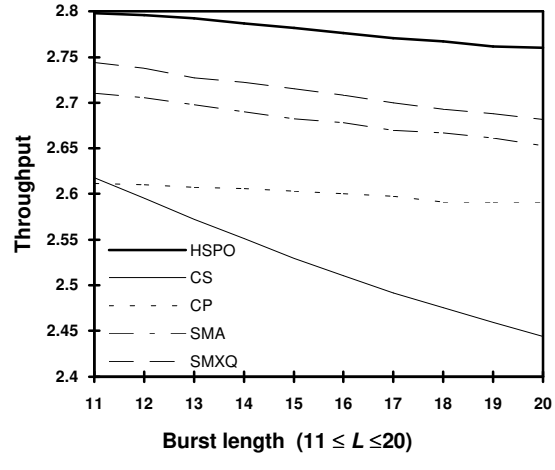


Figure 8. Performance comparison of all schemes under bursty traffic with various long burst lengths - Throughput vs burst length. t_{01}=0.7, t_{00}=0.3 and 5,000,000 clock cycles in each run with 5 runs.

Switch and traffic parameters used.

schemes	configuration
HSPO	$N = 4,\ B = 20,\ p = 0.5 \ldots 1.0,\ L = 10$
CS	$N = 4, B = 20,\ p = 0.5 \ldots 1.0,\ L = 10$
CP	$N = 4, B_n = 5,\ p = 0.5 \ldots 1.0,\ L = 10$
SMA	$N = 4, B = 20,\ Q_i = 3,\ p = 0.5 \ldots 1.0,\ L = 10$
SMXQ	$N = 4, B = 20,\ X_i = 8,\ p = 0.5 \ldots 1.0,\ L = 10$

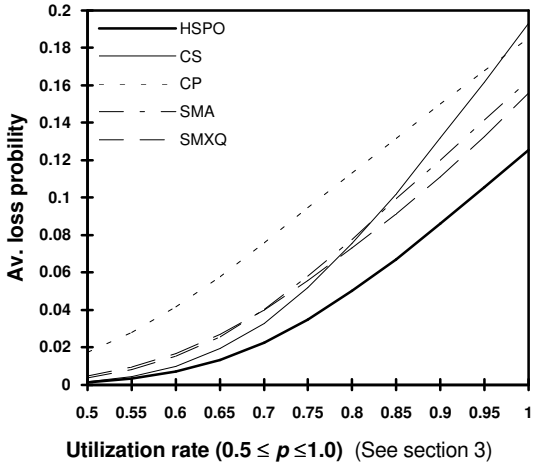


Figure 9. Performance comparison of all schemes under bursty traffic with various utilization rate - Average cell loss probability vs utilization rate. t_{01}=0.7, t_{00}=0.3 and 5,000,000 clock cycles in each run with 5 runs.

Switch and traffic parameters used.

schemes	configuration
HSPO	$N = 4,\ B = 20,\ p = 0.9,\ L = 2,3 \ldots 10$
CS	$N = 4, B = 20,\ p = 0.9,\ L = 2,3 \ldots 10$
CP	$N = 4, B_n = 5,\ p = 0.9,\ L = 2,3 \ldots 10$
SMA	$N = 4, B = 20,\ Q_i = 3,\ p = 0.9,\ L = 2,3 \ldots 10$
SMXQ	$N = 4, B = 20,\ X_i = 8,\ p = 0.9,\ L = 2,3 \ldots 10$

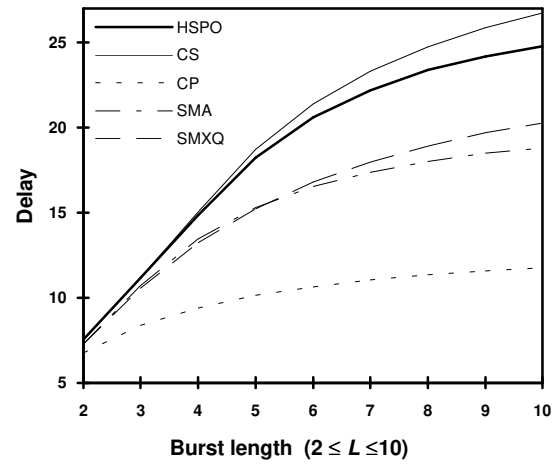


Figure 11. Performance comparison of all schemes under bursty traffic with various burst lengths - Delay vs burst length. t_{01}=0.7, t_{00}=0.3 and 5,000,000 clock cycles in each run with 5 runs.

Switch and traffic parameters used.

schemes	configuration
HSPO	$N = 4,\ B = 20,\ p = 0.5 \ldots 1.0,\ L = 10$
CS	$N = 4, B = 20,\ p = 0.5 \ldots 1.0,\ L = 10$
CP	$N = 4, B_n = 5,\ p = 0.5 \ldots 1.0,\ L = 10$
SMA	$N = 4, B = 20,\ Q_i = 3,\ p = 0.5 \ldots 1.0,\ L = 10$
SMXQ	$N = 4, B = 20,\ X_i = 8,\ p = 0.5 \ldots 1.0,\ L = 10$

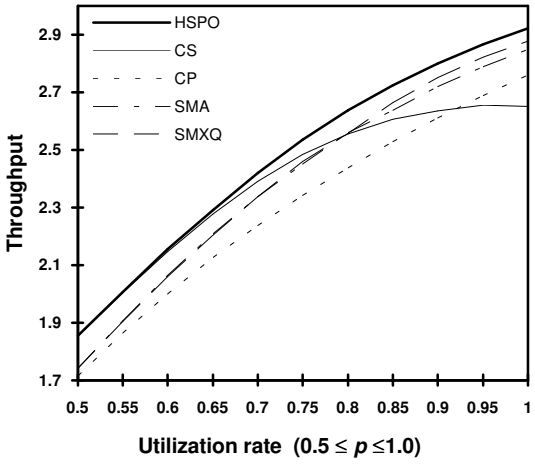


Figure 10. Performance comparison of all schemes under bursty traffic with various utilization rate - Throughput vs utilization rate. t_{01}=0.7, t_{00}=0.3 and 5,000,000 clock cycles in each run with 5 runs.

Switch and traffic parameters used.

schemes	configuration
HSPO	$N = 4,\ B = 20,\ p = 0.5 \ldots 1.0,\ L = 10$
CS	$N = 4, B = 20,\ p = 0.5 \ldots 1.0,\ L = 10$
CP	$N = 4, B_n = 5,\ p = 0.5 \ldots 1.0,\ L = 10$
SMA	$N = 4, B = 20,\ Q_i = 3,\ p = 0.5 \ldots 1.0,\ L = 10$
SMXQ	$N = 4, B = 20,\ X_i = 8,\ p = 0.5 \ldots 1.0,\ L = 10$

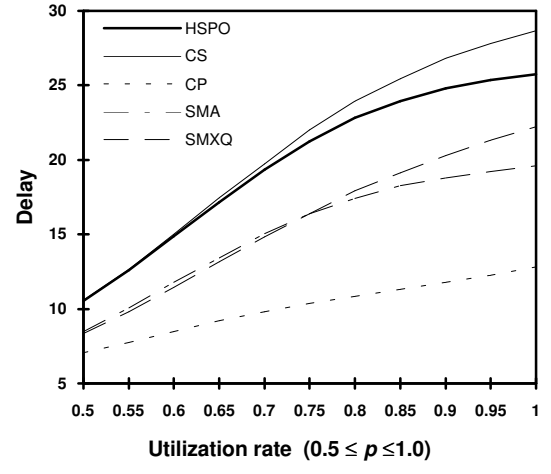


Figure 12. Performance comparison of all schemes under bursty traffic with various utilization rate - Delay vs utilization rate. t_{01}=0.7, t_{00}=0.3 and 5,000,000 clock cycles in each run with 5 runs.