

# A Combined Input - Output Queuing ATM Switch With $m$ Internal Links For Cell Transfer

Carlos Roberto dos Santos

Shusaburo Motoyama

DEE-INATEL, Santa Rita do Sapucaí MG, Brazil  
Phone: +55-3471-9221 Fax: +55-3471-9314

[Carlos@inatel.br](mailto:Carlos@inatel.br)

DT-FEEC-UNICAMP, Campinas SP, Brazil  
Phone: +55-19-3788-3765 Fax: +55-19-3289-1935

[motoyama@dt.fee.unicamp.br](mailto:motoyama@dt.fee.unicamp.br)

*Abstract* – In this paper a combined input-output queuing ATM switch with  $m$  internal links for cell transfer is proposed. A single buffer is used at each input, whereas each output needs  $m$  buffers, where  $m$  is the number of internal links (or channels) that connect each input to  $m$  output buffers. The proposed structure uses physical internal links instead of time division, thus no speedup is required. Since until  $m$  cells can be transmitted to the same output line from different inputs the head of line blocking (HOLB) probability can be reduced and a very high throughput can be obtained. Simulation results show that for  $m=3$ , only a 10-cell size buffer at each input port is required to achieve a cell loss probability better than  $10^{-8}$ , independent of the switch size.

*Key words:* computer networks, ATM, switching.

## I. Introduction

Since the ITU-T (international telecommunication union – telecommunication sector) adopted ATM to be the transfer mode of the BISDN, many structures for high performance ATM switch have been proposed in the literature, based on three basic buffering schemes: input buffered switch, output buffered switch and shared buffer switch.

Shared buffer switch consist of a single memory where arriving packets, or cells, at inputs are stored while they wait for their turn to be transmitted to their addressed outputs. This scheme has the advantage of optimizing buffer usage, but technical issues such as the access time to write and read a cell in a memory limit the maximum throughput of these structures.

The output-buffering switch has many desirable characteristics such as maximum theoretical throughput equal to 100%. However, the output buffer size may be too large to achieve the desired throughput.

The input-buffering switch presents the HOLB (Head Of Line Blocking) problem that limits the throughput to 58%. Many algorithms were proposed to increase the throughput [1], [2], [3], [4], [5], [6]. The main advantage of the input-buffering switch is that the buffer size to store the cells needs to be only for a link. Furthermore, in an input-buffering switch the cells can be kept at input until the switching network is free from congestion or thrown out before they go through the switching network, so avoiding increasing the congestion.

Recently two new schemes have received attention: Virtual Output Queuing (VOQ) and Combined Input-Output Queuing (CIOQ). VOQ is a queuing scheme used to solve the HOL blocking problem associated with FIFO input queuing or to provide facility to satisfy quality of service (QoS) in input queuing. In this technique each input port maintains a virtual separate queue for each output port. CIOQ is a queuing scheme that combines input and output queuing. It is a good compromise between the performance and scalability of both output-queued and input-queued switches. For input-queued switches, at most one cell can be delivered to an output port in one unit of time. For an output-queued switch, up to  $N$  cells can be delivered to an output port in one unit of time. Using CIOQ, instead of choosing these two extreme choices, we can choose a reasonable value in between.

A CIOQ structure named High-speed Statistical Retry Switch (HSR switch) is presented in [8]. This structure uses an internal speedup factor of  $m$ , i.e., cells are transmitted between input and output buffers  $m$  times the input line speed ( $1 < m < N$ , where  $N$  is the switch size). With ever-increasing line rates, this kind of scheme are becoming increasingly difficulty to implement. The algorithm used in the HSR switch, is not fair, because first  $m$  input ports have always priorities over the rest. Thus the last input-buffer will have more cells waiting than the first  $m$  input-buffers. The advantage of the HSR switch is the simplicity of the arbitration circuit at the crosspoint.

Another structure presented in [9], uses a combination of VOQ and CIOQ. At input port  $N^2$  buffers are provided and a two-stage approach for scheduling purpose is implemented. The arbiters at the input side perform input contention resolution, while the output-buffered switch element performs classical output contention resolution. The two main bottlenecks in this switch structure are the shared memory interconnection complexity (each input and each output must be able to reach each memory location) and the output queue bandwidth (up to  $N+1$  accesses per queue per packet cycle).

By using the combined input-output queued structure, an efficient way to transfer the cells from input buffers to

output buffers, without the need of internal speedup, is proposed in this paper. The proposed switch is based on crossbar type of structure, but vertical lines of this structure have each one a bunch of  $m$  lines instead only one line. This simple provisioning permits to the switch a parallel transfer of cells, thus giving a very high throughput at the output port.

The paper is organized as follows: in section II, the proposed switch structure is presented. The estimation of number of parallel vertical lines, details of simulation and the comparison with the theoretical model are presented in section III. Finally the conclusions are presented in section IV.

## II. Proposed Structure

Fig. 2.1 shows the proposed structure. Each input port has a single buffer, whereas at each output port a set of  $m$  buffers are provided. Each input buffer is connected to the  $m$  output buffers through the  $m$  physical links.

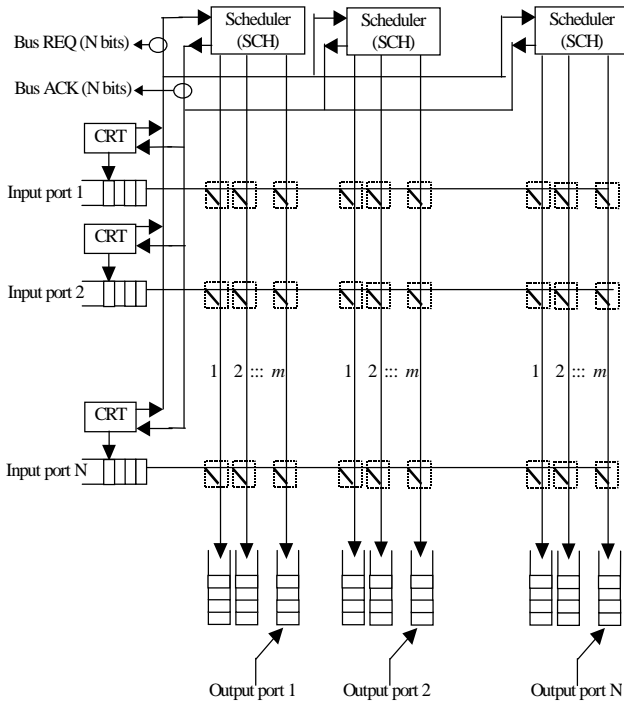


Figure 2.1 - Crossbar switch with single input buffer and  $m$  output buffers

Each control unit (CTR) first examines if the input buffer has any cell to transmit. If any cell is waiting in the queue, each CTR sends a REQUEST signal (only a bit) to the scheduler (SCH) using a reserved line. Each CTR has a separate reserved line to SCH for request purpose. Only one REQ signal can be sent during one time slot from the each input. If there are more than  $m$  requests in the same time slot, each SCH selects  $m$  inputs on a round-robin basis and sends ACK signals (only a bit for each ACK) to

the chosen CTRs through the reserved lines, signaling each CTR that a cell was chosen to transmit in that slot. In the next cycle, the scheduler will begin to serve from the last input that wasn't served in the previous cycle. Thus, the scheduler ensures fair arbitration among all inputs.

The scheduler has also the responsibility to manage the output buffers to avoid that a cell has arrived after be transmitted before a cell that has arrived first, from the same input, i.e., to avoid that cells are transmitted out of sequence. It can be easily implemented using a round-robin scheme that stores the cells in sequential form at the buffers of this output. Let suppose that in the first cycle the inputs  $i_1, i_2, i_3, i_4$  e  $i_5$  have cells to transmit to the same output port address  $j$  (Fig. 2.2), and the number of internal links is 3 ( $m=3$ ). After these inputs have sent the REQ signals to the scheduler  $j$ , only the inputs  $i_1, i_2$  and  $i_3$  will receive the ACK signals. Simultaneously, the scheduler  $j$  will choose internal links 1, 2 and 3, and will enable the respective gates at crosspoint. The inputs  $i_4$  and  $i_5$  will be served in the next time slot (cycle), using internal links 1 and 2, respectively. Note that only one cell is transmitted from output buffer to output line every cycle. Fig. 2.2 shows the basic operation described above.

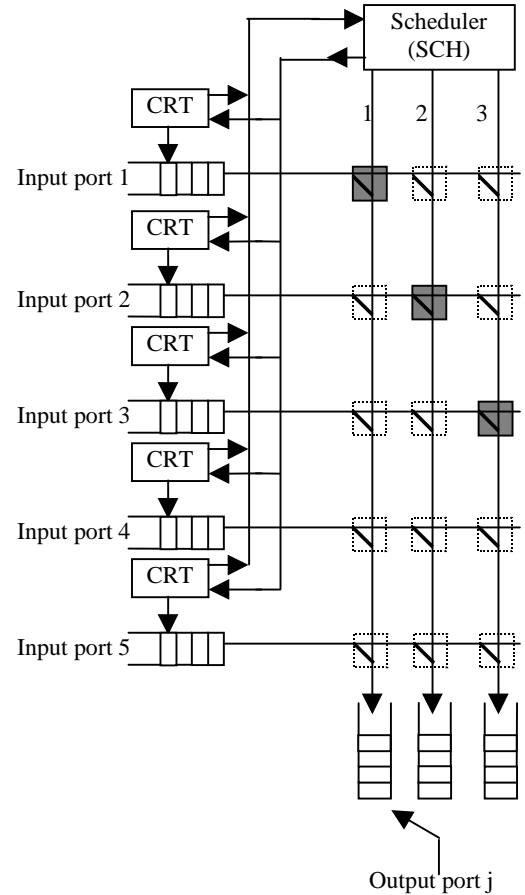


Figure 2.2 - Basic operation of the proposed structure

### III. Performance Analysis

#### A. Blocking Probability Calculation

For the analysis the following assumptions are made. The number of input or output is  $N$  (switch size  $N \times N$ ) and the cell arrivals at the inputs obey independent and identical Bernoulli process. The probability of a cell arrival in a slot is  $\rho$ , and  $m$  is the number of internal links that connect each input to  $m$  output buffers. The probability that a cell will be sent to the particular output port is  $\rho/N$ , thus the probability that  $j$  cells will be sent to the particular output is

$$\binom{N}{j} \left( \frac{\rho}{N} \right)^j \left( 1 - \frac{\rho}{N} \right)^{N-j} \quad (1)$$

The probability that up to  $m$  cells will be transmitted to the particular output is

$$\sum_{j=1}^m \binom{N}{j} \left( \frac{\rho}{N} \right)^j \left( 1 - \frac{\rho}{N} \right)^{N-j} \quad (2)$$

The blocking will occur when more than  $m$  inputs request for transmission to the same output. Thus the blocking probability is given by:

$$B = \sum_{j=m+1}^N \binom{N}{j} \left( \frac{\rho}{N} \right)^j \left( 1 - \frac{\rho}{N} \right)^{N-j} \quad (3)$$

The average blocking probability is:

$$B_{av} = \sum_{j=m+1}^N (j-m) \binom{N}{j} \left( \frac{\rho}{N} \right)^j \left( 1 - \frac{\rho}{N} \right)^{N-j} \quad (4)$$

Fig. 3.1 shows the numerical examples of Eq. 4, for a load of 0.9 at each input port and several switch sizes. It can be observed that for  $m$  greater than 5 the average blocking probability is less than  $10^{-4}$ . It is important to note that for  $m=1$  the proposed structure works as an input FIFO buffer switch and output buffer is not necessary. From Fig. 3.1, assuming the switch size 128 x 128, the average blocking probability values are  $3.0528 \times 10^{-1}$  for  $m=1$ ,  $7.7893 \times 10^{-2}$  for  $m=2$  and  $2.5822 \times 10^{-3}$  for  $m=4$ . Thus, for  $m=2$  the proposed structure has about 4 times gain compared with FIFO input buffer and about 120 times gain if  $m=4$ .

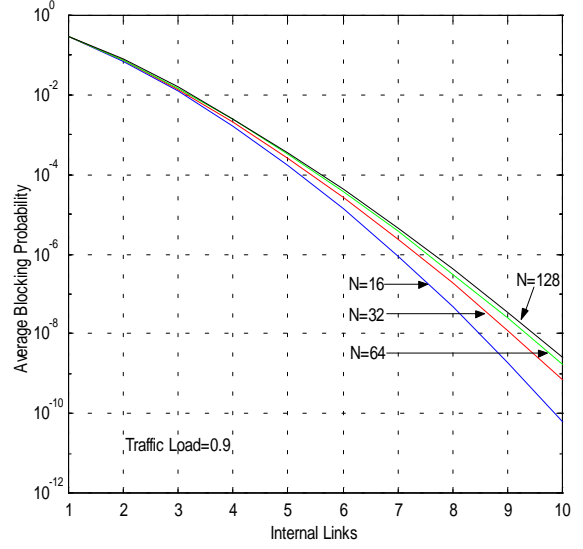


Figure 3.1 - Average blocking probability versus number of internal links

The average blocking probability performance as function of number of internal links, for several traffic loads, is shown in Fig. 3.2 assuming a 64 x 64 switch size. The analytical results show that the proposed structure behaves very well even under heavy load ( $\rho=1$ ).

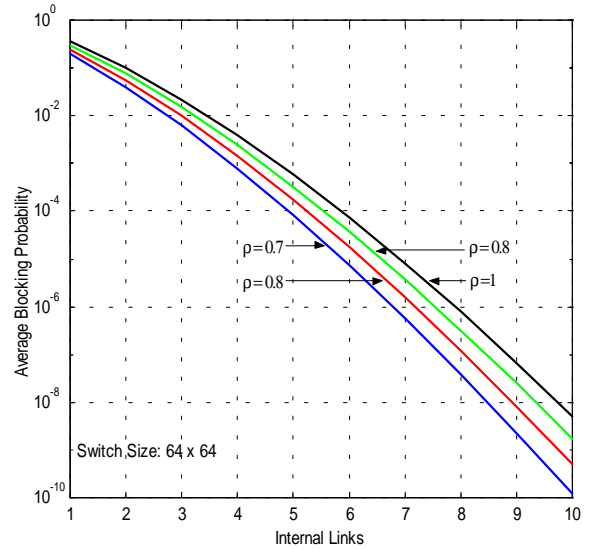


Figure 3.2 – Switch performance under several loads

The average blocking probability increases with switch size for the same number of internal links, as shown in Fig. 3.3. However, it almost saturates when switch size of 32 x 32, assuming a random load of 0.9 at each input port. This saturation is expected because for a large number of inputs  $N$ , the binomial traffic becomes Poisson traffic, thus the cell arrival rate is constant.

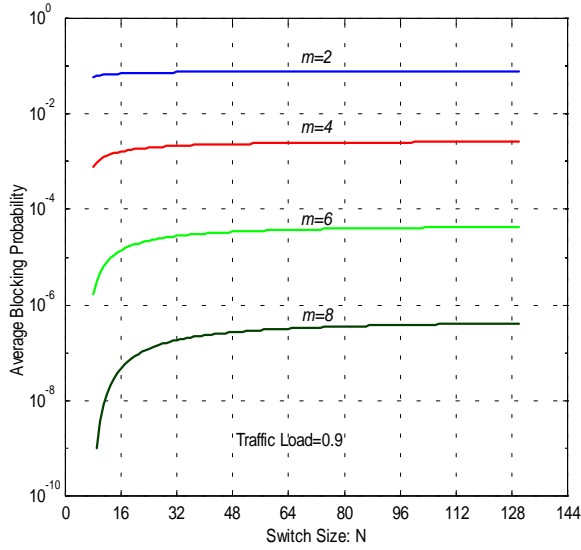


Figure 3.3 – Average blocking probability versus Switch Size N

#### B. Simulation

Simulation study was carried out aiming two main objectives: validation of proposed analytical model for blocking probability calculation and evaluation of queuing behavior in terms of average queue length and average cell loss for input buffer. To perform simulation, a computing program was written using MATLAB tool. The following variables are used: the number of input/output port (N), the number of internal links ( $1 \leq m < N$ ) and traffic load ( $0 < \rho \leq 1$ ). At each cell period, for each input, the simulation program generates a uniformly distributed random number between  $0 \leq x \leq 1$ . If the result is  $\leq \rho$ , a cell is generated with output address given by a random number between 1 and N and stored at input buffer. Then the simulation program reads the cell from input buffer to output buffer according to the number of internal links and verifies the number of cells that are not transmitted and the total number of cells that remained at the input buffer. This procedure is performed each cell period, many times, so that the average blocking probability and average queue length can be computed, assuming an infinite buffer length at each input. It can be seen in Fig. 3.4, the analytical model and simulating results are perfectly matched. Due to very small value of average blocking probability a  $8 \times 8$  switch size and load of 1 ( $\rho=1$ ) are considered for simulation

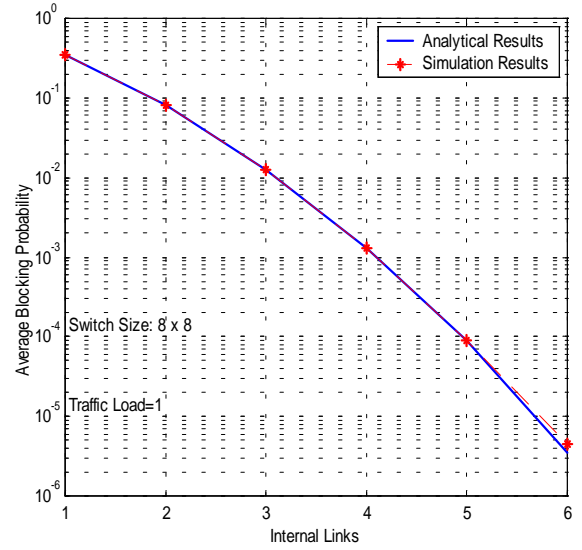


Figure 3.4 – Analytical results compared to the simulation results.

#### C. Queuing Analysis

Figure 3.5 shows the simulation results for average number of cells in queue, considering load of 0.9 and a  $8 \times 8$  switch size. As it can be observed, for  $m > 2$  there is no practically cell waiting in the input buffer. Even for  $m=2$ , the number of cells in queue is small, about 4 cells in average. However, the simulation results have shown that, for  $m=1$  and load equal 0.9, the number of cells in queue is very large because of HOLB.

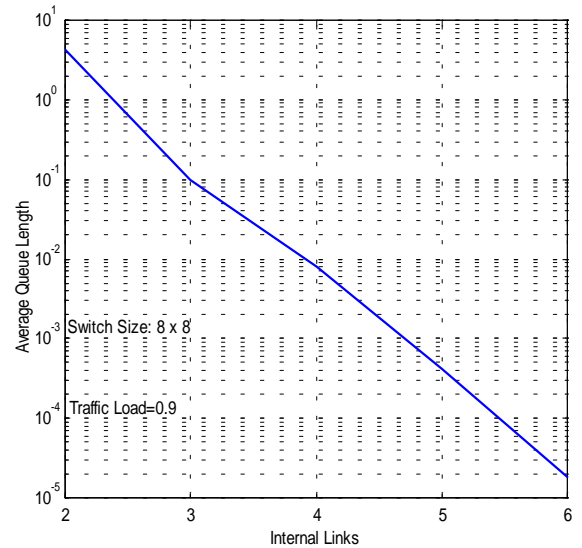


Figure 3.5 - Average Queue Length as function of internal links

For estimation of the average cell loss probability a finite buffer of size 10 was used. Simulation results show

that for  $m=3$ , cell loss probability is about  $10^{-8}$ , assuming a load of 0.9 and switch size of  $64 \times 64$ .

#### IV. Conclusion

A new high speed ATM switch with combined input-output queuing was proposed in this paper. Each input buffer is connected to the output buffer through the  $m$  internal links. Since no speedup is required the proposed structure is suited for very high-speed switches. Due to use of the  $m$  internal links, even using FIFO queues at input ports, the throughput will improve because the head-of-line-blocking probability is reduced. Simulation results have shown that for  $m=4$ , only a 10-cell size buffer at each input port is needed to achieve very low cell loss probability, assuming a load equal 0.9 and independent of the switch size. In addition, the arbitration system is not complicated.

#### References

- [1] Motoyama, S., Petr, D. W. and Frost, V. S.: "Input – queued switch based on a scheduling algorithm", Electronics Letters 6<sup>th</sup> July 1995, **31** (14) pp. 1127-1128.
- [2] McKeown, N., Varaiya, P., Walrand, J.: "Scheduling cells in an Input-Queued Switch", Electronics Letter 9<sup>th</sup> December 1993, **3**, (11). Pp 323-325.
- [3] Motoyama, S.; Ono, L. M., Mavigno, M.C.: "An iterative cell scheduling algorithm for ATM input-queued switch with service class priority", IEEE communications Letters, 1999, **3**, (11), pp. 323-325.
- [4] Anderson, T. E., Owicki, S.S., Saxe, J.B and Tacker, C. P.: "High Speed Switch Scheduling for Local Area Networks", Proc. Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, Oct. 1992, pp. 98-110.
- [5] Karol, M., Eng, K., Obara,: "Improving the performance of Input-Queued ATM Packet Switches", Proc. INFOCOM 92, 1992, pp. 110-115.
- [6] Obara, H.: "Optimum Architecture for Input Queuing ATM Switches", Electronics Letters, 28<sup>th</sup> March 1991, pp. 555-557.
- [7] Motoyama, S.: "Simple high speed ATM switch with service class priority", Electronics Letters, 16<sup>th</sup> March 2000, **36**, (6), pp. 590-591.
- [8] Genda, K., Yamanaka N., Doi, Y.: "A High-Speed ATM Switch that uses a Simple Retray Algorithm and Small Input Buffers", IEICE Trans. Commun. Vol. E76-B. No. 7, July 1993.
- [9] Minkenberg, C., Engbersen, T., "A combined Input and Output Queued Packet-Switched System Based on PRIZMA Switch-on-a-Chip Technology", IEEE Commun. Magazine, December 2000 Vol.38 No. 12.