A High-Speed ATM Switch Architecture Using Random Access Input Buffers and Multi-Cell-Time Arbitration

Hakyong Kim*, Changhwan Oh**, Kiseon Kim* and Aftab Ahmad*

*Dept. of Information and Communications, KJIST
**Electronics Telecommunication Research Institute
E-mail: hykim@charly.kjist.ac.kr

Abstract— **In this paper, we introduce a new high-speed ATM switch using Random Access Input Buffers (RAIB) and Multi-Cell-Time Arbitration (MCTA), and evaluate its performance for the uniform traffic both in numerical way and in computer simulations. The switch has N same input modules each of which is similar to the common shared buffer switch. N Address Buffers (ABs) in the input module are used for N output and the ABs for a certain output in different input modules are controlled by an external arbitrator. MCTA arbitration is employed in order to reduce the required arbitration rate as well as to provide the guard time when the switch is operated at very high-speed. MCTA arbitration means that the service order of two or more cells destined for a same output is determined by one arbitration, but the cells are transmitted one by one in each time slot.**

1 Introduction

An input-buffered ATM switch can transmit cells at high speed because it can operate at the same speed as the external link speed. However, the switch has been turned away by the switch designers, since it suffers from the *Head-of-Line* (HOL) blocking so that its maximum throughput is limited to 58.6% as the switch size increases [1].

And thus, much research has been suggested so far to keep the high-speed property and to improve the limited throughput of the input-buffered switch. Some of the suggestions only consider the high-speed property [2], while the others are dedicated to the switching performances [3, 4]. For the former, Kim et al. surveyed and compared the high-speed property and the requirements for the high-speed property of four input-buffered switches [5]. The study shows that the control speed of the switch is the very important issue to keep the switching speed and to improve the limited throughput. For the latters, it is usually used that the common or shared buffer(s) with diverse arbitration methods instead of the simple FIFO. However, this type of switches have to operate at higher speed internally to keep up with the increase in the memory access speed, caused by several cells transmitted from one input port in a unit time slot, and to execute all complicated control algorithms in a unit time slot. As a result, the switch can not support high-speed external links.

In this paper, we propose a new high-speed ATM switch architecture using Random Access Input Buffers (RAIB) and Multi-Cell-Time Arbitration (MCTA). The proposed switch makes use of the common buffers called CIB in each input module in order to prevent the cells from being blocked. This Common Input Buffer (CIB) in each input module is managed by N Address Buffers (ABs) corresponding to N output ports, where N is the switch size. Basically, at most one AB out of N ABs for a same output can be selected to send a cell to a corresponding output in every time slot and at most one AB out of N ABs for N different outputs in an input module can be selected. Arbiter Status Bit Map (ASBM), a kind of register, is used for this purpose. That is, ASBM records whether the input port is selected or not. Also, the switch employs MCTA arbitration to reduce the required arbitration rate as well as to provide enough guard time to efficient cell transmission time [6]. MCTA arbitration means that the arbitration is executed only once in every m time slots for m cells in a same AB inside an input module, where $1 \leq m \leq N$. However, the cells whose service order is determined by MCTA arbitration are transmitted one by one in each time slot.

The configuration of this paper is as follows. In Section 2, the proposed switch architecture is illustrated and the functions of each block are described. Also, the operation of the switch including MCTA arbitration is explained. In Section 3, we present a queueing model for the proposed scheme and analyze it. Finally, Section 4 is devoted to evaluate performance measures such as the cell loss probability and the mean waiting time by com-

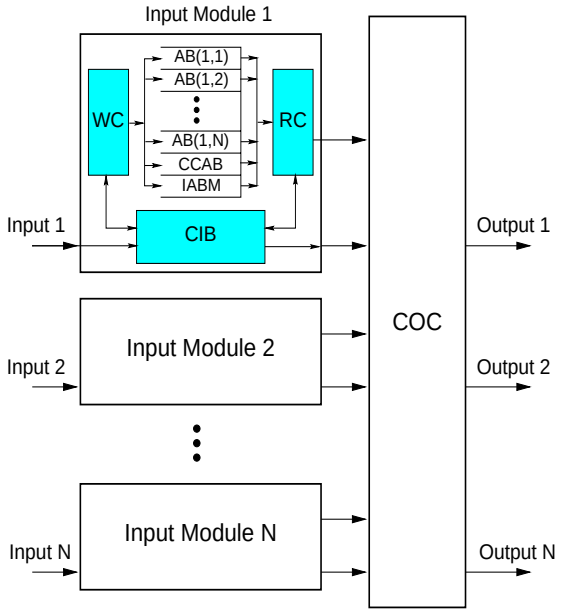

Figure 1: The Schematic diagram of the RAIB switch.

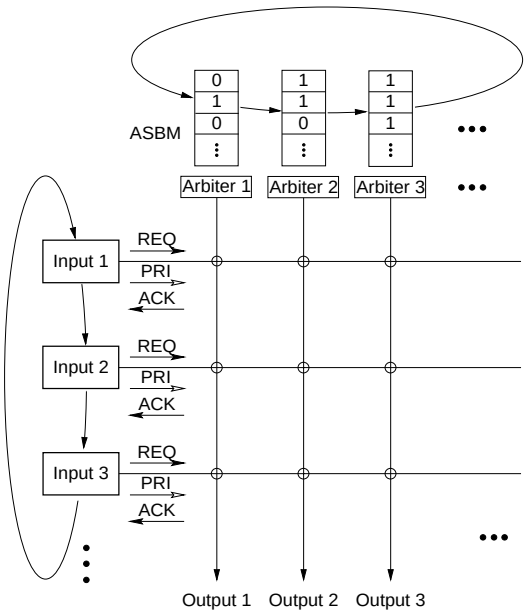

Figure 2: Cell scheduling process and the operation of the Arbiter Status Bit Map (ASBM).

puter simulations. Also, we discuss the impact of MCTA arbitration on some performance results.

2 Switch Architecture

2.1 Configuration and Operation of the RAIB switch

The basic configuration of the proposed Random Access Input-Buffered (RAIB) ATM switch is shown in Figure 1. The entire switch is constructed with N same input modules and a Common Output Controller (COC). Each input module is built with N Address Buffers for N outputs, a Cell Copy Address Buffer (CCAB), an Idle Address Bit Map (IABM), Write/Read Controllers (WC, RC) and a Common Input Buffer (CIB). The COC is constructed with a space switch fabric such as a crossbar switch, which has no internal blocking, and an arbitration logic including an Arbiter Status Bit Map (ASBM).

In order to describe the operation of each functional block, we assume that a cell is arriving in a certain input module. Then, the WC looks up the IABM to find out the empty location in the CIB first, and the arriving cell is stored at one of the empty spaces. When there is no free space, the cell is dropped. Provided that the arriving cell is stored in the CIB, the IABM is updated. At the same time, the WC investigates the header of the cell to find out the destination address. An AB for the destination address records the address of the CIB where the arriving cell is stored. If the cell is for multicasting, the address of CIB is stored in the CCAB. The CCAB has to contain an extra information about the destinations for the multicast cell to be copied. By using CCAB, the multicast cell is stored once in CIB irrespective of the number of destinations so that the required buffer size can be reduced. As previously stated, the CIB is a common buffer and is fully shared by all ABs in a same input module so that it relieves the HOL blocking effect. On the other hand, the ABs and the CCAB are managed by first-come first-serve rule in order to preserve the cell sequence. At each boundary of time slots, the RC checks all ABs or CCAB and communicates with the COC which provides the cell scheduling function. According to the scheduling result, the corresponding RC transmits a cell from the CIB and resets the related AB or CCAB and the IABM.

Figure 2 shows the cell scheduling process of the arbiters in the COC and the operation of the ASBM. The procedure follows the four phases:

Step 1. REQ Phase: In REQ phase, the arbiters receive REQ signal from RCs in every input module at each boundary of time slots. REQ signal is composed of N bits and each bit is used to stand for whether there is at least a cell in ABs destined for each output port or not. An arbiter for a certain output port compares bits of REQ signals from all input modules for the corresponding output port, and then selects an REQ signal. After an arbiter selects an REQ signal, the corresponding flag of the ASBM is set to 1. All these processes are controlled by round-robin.

Step 2. PRI Phase: PRI signal is sent to the arbiters after REQ signal is sent from RC. This signal is also N-bit long and each bit is used to stand for whether the corresponding cell has higher priority or not. Each bit of PRI signal is set to 1 when an arriving cell has higher priority initially and may be set by somewhat complex queue-length management algorithm.

Step 3. ACK Phase: As soon as the cell-selecting operation is completed, the arbiters send ACK signals, including the arbitration results, to every RC. ACK signal is composed of N bits and each bit is corresponding to ABs in an input module. The signal can include at most one bit which is set to '1' and the bit means that the corresponding AB is selected to be serviced.

Step 4. Transmission Phase: In Transmission phase, the RC received positive ACK signal from an arbiter in ACK phase transmits a cell to the corresponding output port through the space switch fabric. And then, the cell leaves the switching system.

2.2 Multi-Cell-Time Arbitration

As illustrated in previous subsection, each input can transmit at most one cell in a time slot and all the cells to be transmitted are destined for different output ports. As the result, the switch can operate at the same speed as the external link speed so that it is possible to switch cells at higher speed. However, the proposed switch adopts rather complicated arbitration algorithm to prevent the internal speed-up and it takes considerable control time as other switches using common-type buffer with complex control logics.

In order to reduce the required arbitration rate, we use Multi-Cell-Time (MCT) arbitration. MCT arbitration means that the arbitration is executed only once in every m time slots for m cells in a same AB, where $1 \leq m \leq N$. However, each arbitration decides the service order of m cells in a same AB and the cells are transmitted one by one in each time slot. In fact, the arbitration is done for only one cell and the following cell in the same AB is reserved for next service. Also, $m = 1$ corresponds to the RAIB switch without MCT arbitration.

For the explanation purpose, we assume that the MCT arbitration factor $m = 2$. Then, in every two time slots, service order of two cells in a same AB is decided by just one arbitration. Since the arbitration is done for two cells, one cell in the AB is considered as if there is no cell. For example, assuming that 0.5 μsec can be available for an arbitration when the external link speed is 155Mbps and $m = 1$, 1 μsec is available when $m = 2$.

By applying MCT arbitration, we can reduce the required arbitration rate needed in a time slot as stated previously. However, since the cells less than the MCT arbitration factor is ignored as none even though it exists in effect, switching performances such as cell loss probability and mean cell delay will be aggravated. Section 4 will describe this expectation.

3 Switch Modelling and Performance Analysis

For the ABs for a same output port, the proposed switch may be modelled as a multi-queue single-server discrete-time queueing system with deterministic service rate since there are N ABs for the single output port in N different input modules. Provided that the cell arrivals on the N input ports are governed by independent and identical Bernoulli process with rate $N\lambda$ and each cell has equal probability $1/N$ of being addressed to any given output independently of the next cells, the system seems to be modelled as N same single-queue single-server discrete-time queueing system with arrival rate of λ. In effect, however, the selection of any one AB from N ABs for a certain output are correlated with each other and the other ABs for the different output ports as described in the previous section. Therefore, the system can not be modelled as N same single-queue single server queueing system.

To solve this problem, we can use the Aggregated Virtual Address Buffer (AVAB) which is a large buffer with the same buffer size to the N aggregated address buffer sizes for a same output [7, 8]. For example, AVAB(7) means a logical buffer made by aggregating all AB(7)s for output port 7 in each input module and the buffer size of AVAB(7) is equal to the sum of all AB(7)'s buffer size in every input module. By arbitration rule, at most one cell can be served from the AVAB with the deterministic service rate 1 cell/time slot. Then, the system can be remodelled as a single-queue single-server system with batch arrival, that is, $Geo^{[X]}/D/1$, for the entering traffic stream following the i.i.d. Bernoulli process. The analysis for this queueing model is well documented in [9] and the solutions for the mean queue size and the mean waiting time of the system can also be obtained in the same reference directly. We do not consider MCT arbitration, priority, and multicast functions in analyzing this model.

Let $L^{(0)}$ be the number of cells in the system (queue size) at the beginning of the first slot, and let $L^{(n)}$ be the queue size immediately after the nth time slot, where $n = 1, 2, \cdots$. In the cell model, the sequence $L^{(n)}$; $n = 0, 1, 2, \cdots$ is a Markov chain. If $\Lambda^{(n)}$ denotes the number of cells that arrive in the nth time slot, we have the relation

$$p_{ij} = \begin{cases} \Lambda^{(n+1)} & L^{(n)} = 0 \\ L^{(n)} + \Lambda^{(n+1)} - 1 & L^{(n)} \geq 1. \end{cases} \quad (1)$$

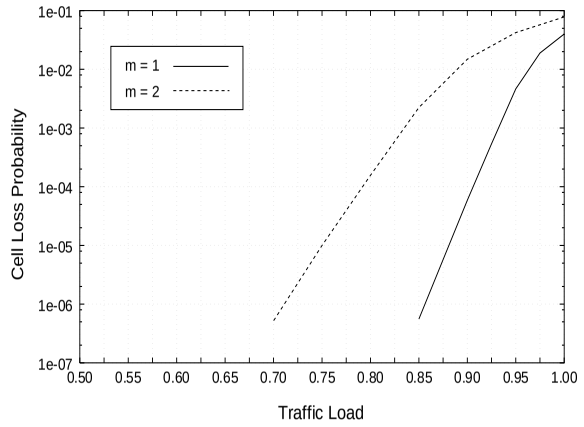

Figure 3: Cell loss probabilities of the RAIB switch. (switch size : 8×8, buffer size : 16 cells)

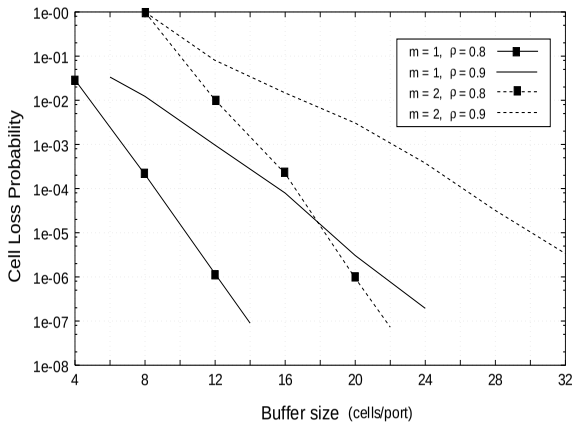

Figure 4: The cell loss probability for different buffer sizes and different traffic load. (switch size : 8×8)

The generating function of the steady-state probabilities for Markov chain $L^{(n)}$; $n = 0, 1, 2, \cdots$ is then easily given by [10]

$$P(z) = \frac{(1-\lambda)(1-z)\Lambda(z)}{\Lambda(z) - z} \tag{2}$$

where

$$\lambda^{(i)} \equiv E[\Lambda(\Lambda-1)\cdots(\Lambda-i+1)] = \Lambda^{(i)}(1)$$

and $\lambda = \Lambda^{(1)}(1)$ is the mean number of cells that arrive in a time slot. From Equation (2), we have [10]

$$E[L] = \frac{\lambda^{(2)}}{2(1-\lambda)} + \lambda. \tag{3}$$

The PGF $W(u)$ for the waiting time of an arbitrary cell is given by

$$W(u) = \frac{(1-\lambda)[1-\Lambda(u)]}{\lambda[\Lambda(u) - u]}. \tag{4}$$

From Equation (4), we have

$$E[W] = \frac{\lambda^{(2)}}{2\lambda(1-\lambda)}. \tag{5}$$

From Equations (3) and (4), we can confirm Little's theorem that

$$E[L] = \lambda(E[W] + 1). \tag{6}$$

4 Numerical Results and Discussions

For the queueing model derived in Section 3, we obtain the cell loss probability and the mean cell delay of the RAIB switch. In this evaluation, the multicasting function is not considered. For MCT arbitration, $m = 1$ and $m = 2$ cases are considered.

Figure 3 shows the cell loss probability of the 8×8 RAIB switch. The buffer size is assumed 16 cells per each input. Figure 4 also shows the cell loss probability for the various buffer sizes and different traffic load ($\rho = 0.8$ and $\rho = 0.9$). To provide 10^{-6} cell loss probability, if we employ MCT arbitration about twice of buffer space is required.

Figure 5 shows the mean cell delay of the 8×8 RAIB switch with infinite buffer space. When MCT arbitration is used, cells must stay for around 5 cell times in the buffer at the relatively low traffic load. The dashed line is the analysis result for the mean waiting time in Section 3 and is very close to the curve with $m = 1$.

Since the RAIB switch uses common buffer as the input buffer, the HOL blocking is not problematic anymore, and the maximum throughput is about 100% for the infinite buffer space irrespective of the switch size. Figure 6 shows the maximum throughput for the different buffer sizes. The switch size is assumed 8×8. To provide 0.9 throughput, twice the buffer space is required when MCT arbitration is used.

It is apparent from four figures that the switching performances deteriorate and the source utilization becomes worse when MCT arbitration is used. However, we must consider the switching arbitration time. Today's ATM or packet switch architectures are very complex and employ very complicated control logics to ensure the required performances. Even though these control methods could improve the performances, the switch itself must suffer from the heavy burden of controller complexity since it must get through all the control logics in a time slot or less than that.

For a very high-speed packet switch, it is practically impossible to reduce the arbitration or control time. However, MCT arbitration can reduce the control burden as much as the arbitration factor by reducing the

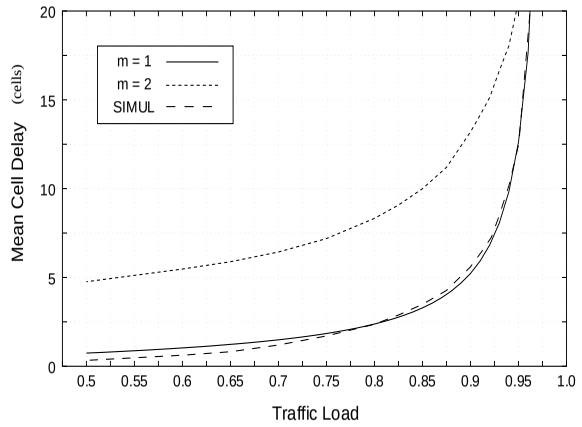

Figure 5: Mean cell delay of the RAIB switch. (switch size : 8×8, buffer size : infinite)

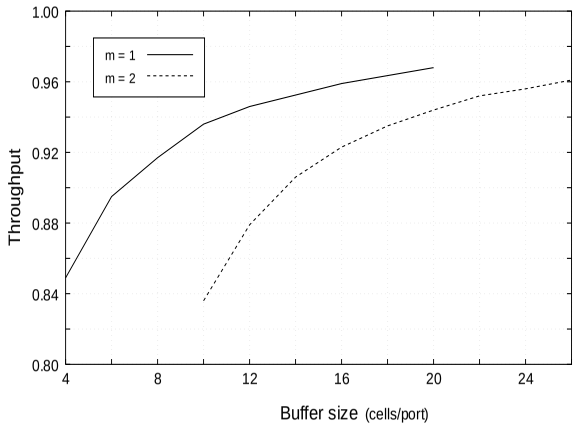

Figure 6: The maximum throughput for different buffer sizes. (switch size : 8×8)

arbitration rate even though the performance is worsened. Therefore, this algorithm can be applied to not only the high-speed packet switch also the large-scale switch whose control time increases proportional to the switch size.

5 Conclusions

In this paper, we proposed a new high-speed ATM switch architecture using random access input buffers and multi-cell-time arbitration and also studied its switching performance. By using random access buffer, we could provided much improved switching performances since the HOL blocking can be eliminated with very small buffer space. Also, by arbitrating carefully with the arbiter status bit map, we could prevent the internal speed-up needed to serve the multiple cells from one common buffer in a time slot. As the result, the RAIB switch can support very high-speed external links. Also, multi-cell-time arbitration could reduce the required arbitration rate as much as the arbitration factor. Multi-cell-time arbitration aggravates the performances, but the algorithm can be considered when a large-scale high-speed ATM switch is designed.

References

[1] M. G. Hluchyj and M. J. Karol, *"Queueing in High-Performance Packet Switching,"* IEEE J. on Selected Area in Commun., vol. 6, no. 9, pp. 1587-1597, Dec. 1988.

[2] J. Paik, Y. Jung and C. Lim, *"A High-Speed Arbitration Scheme for an Input Buffering ATM Switch Architecture,"* IEEE Asia Pacific Conference on Circuits and Systems '96, Nov. 18-21, 1996, pp. 318-321.

[3] Z. Tao and S. Cheng, *"A New Way to Share Buffer: Grouped Input Queueing in ATM Switching,"* Proc. of ICC '94, New Orleans, May 1994, pp. 475-479.

[4] C. Kolias and L. Kleinrock, *"The Odd-Even Input-Queueing ATM Switch: Performance Evaluation,"* Proc. of ICC '96, Vol. 3, June 1996, pp. 1674-1679.

[5] H. Kim, A. Ahmad, C. Oh, and K. Kim, *"Performance Comparison of High-Speed Input-Buffered ATM Switches,"* Proc. of IEEE Workshop ATM 97, Lisboa, Portugal, May 25-28, 1997, pp. 505-513.

[6] Y. B. Choi *Switching Architectures in Optical ATM Networks,* Doctoral Dissertation, Osaka University, Dec. 1995.

[7] H. Kim, *A High-Speed ATM Switch Architecture with Random Access Input Buffers,* Master's Thesis, Dept. Information and Communications, Kwang-Ju Institute of Science and Technology, 1996.

[8] C. Oh, H. Kim and K. Kim, *"A High Speed ATM Switch with Random Access Input Buffers,"* Proc. of IEICE General Conference, Osaka, Japan, March 1997.

[9] H. Takagi, *Queueing Analysis, A Foundation of Performance Evaluation, Volume 3: Discrete-Time Systems,* Elsevier Science Publishers, North-Holland, 1993.

[10] L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications,* John Wiley and Sons, New York, 1976.