# Performance Comparison of High-Speed Input-Buffered ATM Switches

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

*Dept. of Information and Communications, KJIST, Korea
**Electronics Telecommunication Research Institute, Korea
Phone: +82-62-970-2265, Fax: +82-62-970-2204
E-mail: hykim@charly.kjist.ac.kr

*Abstract*— **In this paper, we compare several input-buffered ATM switch architectures. Before comparison, the reasons underlying the application of input-buffered ATM switches are discussed. Then, classification of the input-buffered architectures is given. Next, structures and operation of basic switching elements are presented, including the ordinary input-buffered switch, the Odd-Even switch, the Grouped Input-Queued (GIQ) switch, and the Random Access Input-Buffered (RAIB) switch which we presented before. Finally, comparisons of switching performances and such issues as speed of operation, memory utilization efficiency and multicasting functions are presented.**

*Keywords*—**ATM Switch, Input Buffering, Non-blocking, Crossbar.**

## 1. Introduction

For reasons of flexibility and economy, Broadband Integrated Services Digital Network (B-ISDN) has been replacing the existing application-oriented communication networks. Asynchronous Transfer Mode (ATM) is a high-speed packet-switching technique that has emerged as the most promising technology for B-ISDN. In B-ISDN, the performance bottleneck of the network, which was once the channel transmission speed, is shifted to the processing speed at the switching nodes and the propagation delay of the channel [1, 2]. The latter is much alleviated by substitution of the optical fiber for the copper, while the former is being researched diversely and extensively in the industry and academia.

In fact, most research in the field of the ATM switches have been dedicated either to the performance improvement [3, 4] or to the switching speed enhancement[5]. In terms of switching performance, the output-buffered switches have been preferred to the input-buffered switches. Studies on the performance of the ATM switches easily reveal why the output-buffered switches are prevalent commercially [6, 7]. Most of them, however, have limitations such as high-speed switching and extension in the switch size. For these reasons, the input-buffered switch draws much attentions of the switch designers since it can provide the high-speed switching property [8]-[11].

In general, ATM switch architectures are classified according to their buffering methods (input-buffered and output-buffered), according to their internal switching (blocking and non-blocking), and according to their switch fabric (crossbar, Batcher-Banyan, and Starlite) [12]. In this paper, we focus on the input-buffered non-blocking crossbar ATM switch with $N$ input ports and $N$ output ports. First, we select four types of input-buffered switch architectures with non-blocking crossbar fabric. Two of them (ordinary input-buffered switch and the Odd-Even switch) use FIFO queues and the others (the Grouped Input-Queued (GIQ) switch and the Random Access Input-Buffered (RAIB) switch) use non-FIFO queues as the input buffer.

This paper is organized as follows. Section 2 specifies the input-buffered ATM switches. The Head-of-Line (HOL) blocking and the methods to tackle this problem are described. Also, the usefulness and advantages of the input-queued switch are discussed. In Section 3, we select four input-buffered switch architectures with different arbitration methods, and the switch architecture and the operation of each suggestion are discussed. In Section 4, we compare three performance evaluation parameters: maximum achievable throughput for varying switch sizes, mean cell delay, and cell loss probability. In Section 5, the characteristics of these switches are compared in terms of arbitration methods, memory access speed (or switch operation speed), memory utilization efficiency, multicasting and priority classes capabilities. Also, we discuss the factors affecting to the switching speed.

## 2. The Input-Buffered Switch Architecture

By input buffering we mean an arrangement where the cells are first queued and then switched. Since the cells are queued as soon as arriving and at most one cell can be served per time slot, the switch fabric can operate at the same speed as the input or output link speed. However, when two or more cells at different inputs are destined for a same output, only one cell is delivered to that output and the other cells have to wait until they are finally released during the future time slots. This is known as the *Head-of-Line* (HOL) blocking phenomenon.

For the output-buffered switch, on the other hand, the arriving cells are switched first and then queued at the output buffer. All the cells from the different inputs are switched as soon as arriving and go for their destined output ports. When two or more cells at different inputs are to be sent to a same output, the switch must operate at as much higher rate in order to accept all the switched cells even though the switch lets only one cell leave the output port. In the worst case, since cells as much as the port number ($N$) can be destined for a same output, the switch operates at $N$ times the external speed.

This difference in the switch operating speed between the input- and output-buffered switches implies that the input-buffered switch can be used to support much higher external link speed if the performance degradation by the HOL blocking is minimized. There are several methods dealing with this HOL blocking. One way is to increase the switch operating speed similar to the output-buffered switch. In this case, multiple cells can be routed to one output port, but all cells except one are dropped, if there are no output buffers, when entering into the output port. As the result, the switch can support only slow external link speed. Another approach is to use several same switching fabrics in parallel. At the expense of the cost, this method also requires additional control logic. Another way is to use somehow complex arbitration and cell selection process which are employed to resolve any potential contention among the input buffers. The complex arbitration, however, requires as much complicated circuitry and relatively longer control time if there is no parallelism. Apparently, it looks that the third approach is most appropriate in implementing a high-speed input-buffered switch with high throughput.

As far as the input-buffered switch with FIFO memory is concerned, [13] has mentioned three possible cell selection policies : One of the $k$ cells is chosen at *random*, each selected with equal probability $1/k$, *longest queue* selection, in which the controller sends the cell from the longest input queue, and *fixed-priority* selection where the $N$ inputs have fixed priority levels, and of the $k$ cells, the controller sends the one with highest priority. There is another method using odd and even FIFOs, which will be presented in the next section.

For the switches using non-FIFO memory, many suggestions also have been made. The *Look-ahead* scheme [14] is very well-known strategy treating HOL blocking. In the Look-ahead strategy the arbitration and cell selection scheme employs $w$ separate contention resolution rounds, where $w$ is the window size. Two of suggestions using non-FIFO memory, the Grouped Input-Queued (GIQ) and the Random Access Input-Buffered (RAIB) scheme, suggested recently, will also be presented in next section.

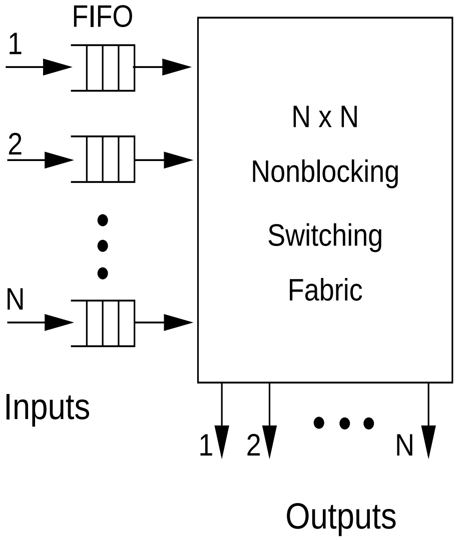

Figure 1: Ordinary input-buffered switch architecture with FIFOs.

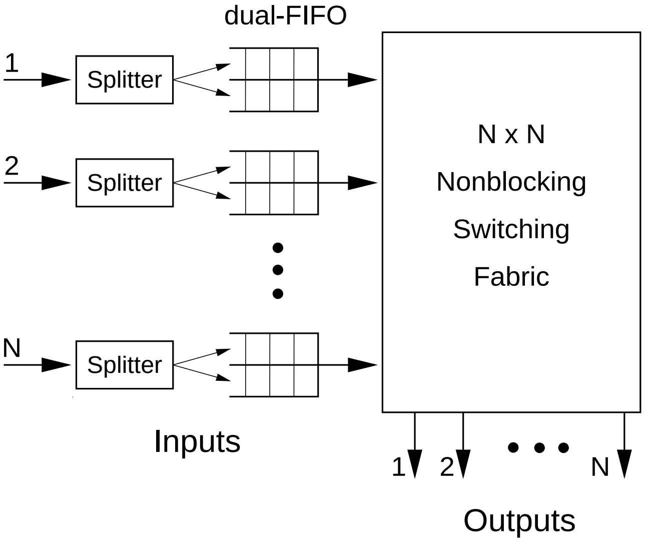

Figure 2: Odd-Even switch architecture with dual-FIFOs.

## 3.The Switch Configurations

Many arbitration and cell selection strategies have been proposed to improve the limited switching throughput of input-buffered switches. In this section, we confine our concerns to four types of the input-buffered switches with non-blocking space switching fabric although many variants or hybrid architectures have been explored. First two of them use FIFO queues and the others use non-FIFO queues as the input buffer.

### 3.1 Ordinary Input-Buffered Switch

It is well known that the (ordinary) input-buffered switch with FIFO suffers from the HOL blocking, so that the maximum throughput is limited to 58.6% for the uniform traffic as the switch size becomes infinite [13]. In Section 2, we discussed about the switch and the HOL phenomenon in detail, and thus we do not mention it again.

From the structural point of view, the entire switch architecture is very simple and additional control logic is not shown as shown in Figure 1. In fact, the architecture shows the concept of the input-buffered switching simply. Assuming there are no additional control logics, when two or more cells are destined for a same output, the HOL blocking occurs, so that the maximum throughput is limited. Also, it is difficult to render the multi-class and multicast services.

### 3.2 Odd-Even Switch

The Odd-Even switch is much similar to the ordinary input-buffered switch with FIFO only except using dual-FIFO and splitter. That is, each input port has two separate FIFO queues, an *odd* and an *even* queue. An incoming cell is stored at the input at either of two FIFOs according to its output port destination [8]. The full contention resolution process consists of two very short rounds. Arbitration during the first round involves the HOL cells at the even input queues. In the second round cells at the HOL of the odd input queues contend for the odd output addresses. However, those input ports whose even queues could not access an output port in the first round are allowed to participate in a contention among their odd queues in the subsequent second round.

The quantitative evaluation for the uniform traffic stream shows that the throughput of the Odd-Even switch, for large switch size ($N$), is 0.719. This scheme outperforms considerably the regular switch (0.5868) [13] by more than 20%. Also, the switch provides the maximum throughput of around 0.64 for

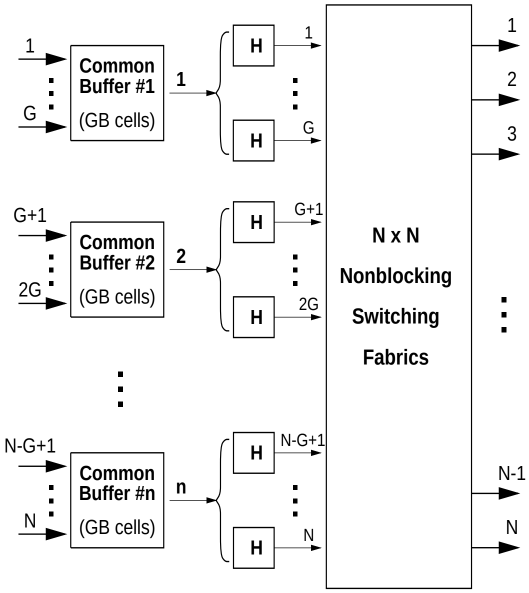
1
G
Common Buffer #1 (GB cells)
1
H
H
G
G+1
2G
Common Buffer #2 (GB cells)
2
H
H
G+1
2G
N-G+1
N
Common Buffer #n (GB cells)
n
H
H
N-G+1
N
N x N Nonblocking Switching Fabrics
1
2
3
N-1
N

Figure 3: Grouped input-buffered switch architecture.

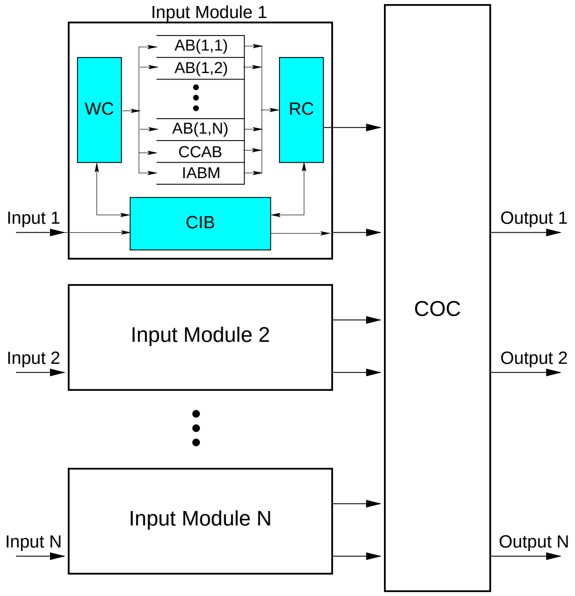

Figure 4: Random access input-buffered switch architectures.

the IBP traffic with the probability, $p = 0.05$, of being switched from the ON (busy) state to the OFF (idle) state.

### 3.3 Grouped Input-Buffered Switch

The basic idea of the grouped input-queued (GIQ) switch is to divide the whole input ports into $n$ groups and let input ports in the same group share the buffers, where $n = N/G$, $N$ is the switch size and $G$ is the group size or the number of input ports in a same group [9]. The distinguishing difference from the previous two architectures is the buffer type, i.e., common buffer not just FIFO. Also, this suggestion uses look-ahead contention resolution scheme. To prevent the increase in memory access speed, at most $G$ cells can be sent to the output from a grouped queue in a single time slot.

When the group size $G$, $1< G < N$, is equal to 1, the GIQ scheme is the same as the window policy scheme with typical look-ahead contention resolution scheme suggested by Hluchyj and Karol [14]. When $G = N$, that is, all input ports completely share one large common buffer, this scheme becomes complete-shared input-buffered scheme. In this case, however, the memory access speed must be increased as $N$ times the external link speed. This method can present efficient utilization of buffers and high throughput of cells as the group size becomes small [15].

### 3.4 Random Access Input-Buffered Switch

Random Access Input Buffered (RAIB) switch, which we presented before [10, 11], is constructed with $N$ input modules and a Common Output Controller (COC) as shown in Figure 4. Each input module takes the similar form to the buffering part of the common-buffered switch. That is, there is a common input buffer (CIB) in each input module and this buffer is managed by $N$ address buffers (AB) corresponding to each output port. Also, all address buffers for a given output port are controlled by the arbiter for that output and an Arbiter Status Bit Map (ASBM) which are located in the COC module. ASBM is a register recording whether each input is used or not in every time slot so it keeps each input port send at most one cell from its own CIB. COC is built with these two control elements and non-blocking crossbar switch fabric.

The switch operates at the same speed with the external link speed by virtue of the ASBM. Even if not

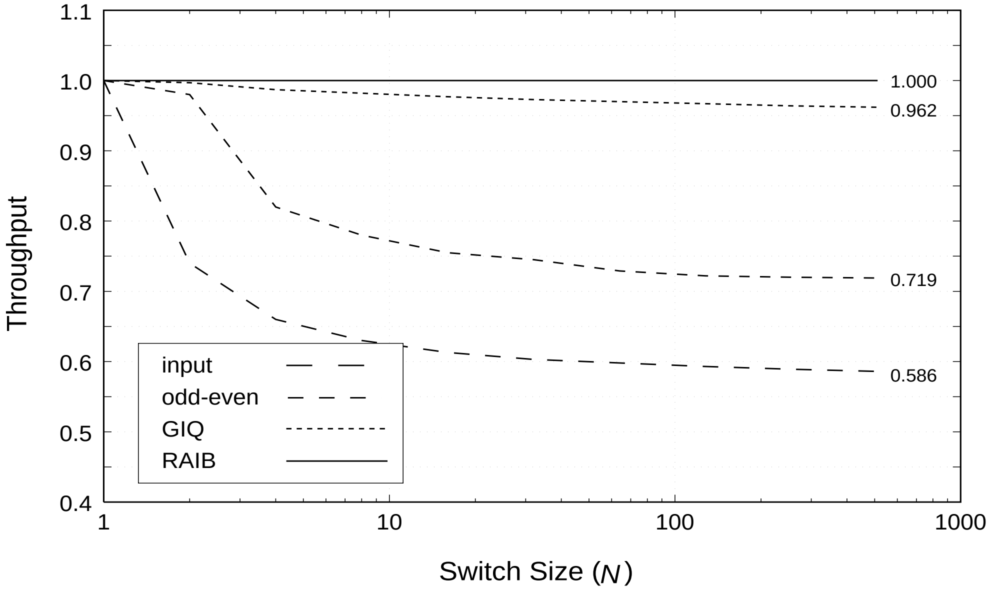

Figure 5: Throughputs of four input-buffered switches (buffer size : infinite).

illustrated in the schematic diagram of Figure 4, it also employes multi-cell-time arbitration (MCTA). MCTA can reduce the arbitration rate by arbitrating $m$ cells going for a same destination at a time slot. For the uniform traffic, this switch can provide 100% throughput for the uniform traffic stream irrespective of the switch size as shown in Figure 5. This implies the potential as the high-speed ATM switch with much improved switching performances.

Also, the use of several address buffers including Cell Copy Address Buffer (CCAB) makes it possible to provide multicasting function and multi-class services. It appears that the entire switch architecture adopts somewhat complex control logics such as ASBM and MCTA, but most of them can be implemented easily with the existing technology.

## 4. Performance Comparison

It is assumed that cell arrivals to each of the $N$ input ports are independent and identically distributed according to a Bernoulli process with parameter $\rho$. Also, we assume that each cell in the input buffer is destined to the output port with uniform distribution. That is, the cells have equal probability $1/N$ of being addressed to any given output. Each arriving cell goes into an input memory at least momentarily. The performances of the switches described in Section 3 are evaluated based on three measures: throughput, delay, and cell loss probability [16]. And the running time for the computer simulation is $10^9$.

Figure 5 shows the throughputs of four switch architectures for the varying switch sizes. Sizeable increases are observed to that of the ordinary input-buffered scheme, whose maximum throughput is known to be 0.586 [13, 14]. For the GIQ switch the group size is 2 and thus at most two cells can be switched. The authors of [9] made use of window size $w = N$ in order to guarantee the throughput. The RAIB scheme and the GIQ scheme, which adopt common buffer(s) as the input buffer, show very high throughput, and it results from the statistical use of the memory.

Figure 6 shows the mean waiting times of the switches as a function of $\rho$. It is assumed that the switch size is also $8 \times 8$ and the capacity of buffer is infinite. From this figure, we can also find that the switches using non-FIFO memories can provide much shorter waiting time even at a very high traffic load. Figure 7 shows the cell loss probability of the switches as the function of the traffic load, $\rho$. The switch size is $8 \times 8$, and the buffer space per each input port is limited to 16 cells. For the GIQ switch,

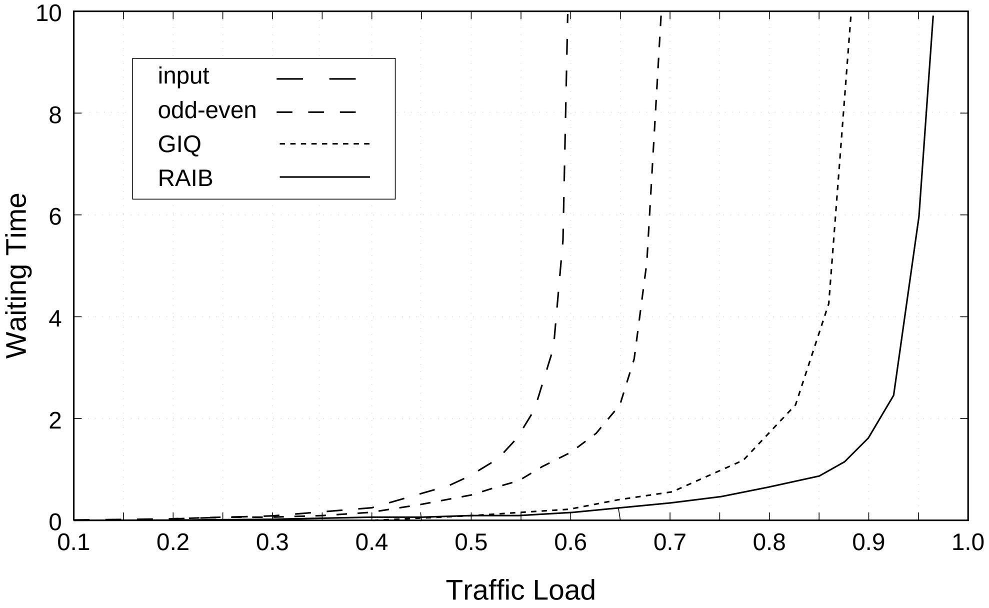

Figure 6: Mean cell delays of input-buffered switches (switch size : 8 × 8, buffer size : infinite).

the buffer space per each group is 32 cells since two input port aggregates into one group. The curves for the ordinary and grouped (group size = 2) switch have very slow slope, while the curve for the RAIB switch has very steep slope. Also, it is notable that the cell loss probability of the Odd-Even switch is worse than that of the ordinary switch below the traffic load $\rho = 0.5$.

The switches suggested recently improve the switching performances in such a way reducing the possibility of the HOL blocking, which means that the number of heads of the each input buffer increases. For the Odd-Even switch, the number of heads of line per each input port is two and for the GIQ and the RAIB switches, the numbers are $w$ and $N$, respectively. For the above simulation, $w = N = 8$.

## 5. Comparison of Input-Buffered Architectures

The input-buffered switch architecture has been turned away due to the bad throughput-delay property by the HOL blocking. However, as can be noted from the previous sections, these switches present a wide variety of queue maintenance methods so it can improve the switching throughput. In this section, we throw more light on the characteristics of each scheme described previously. These properties are summarized in Table 1.

As can be seen from Table 1, the capability of the input-buffered switches is determined by the buffer type and the arbitration method. The switches with FIFO and simple controller complexity have low memory utilization and it is difficult to provide special switching functions such as multicasting and priority classes. On the other hand, the switches with non-FIFO and complex control logic can support special switching functions and reduce the required memory size. Also, this type of switches using non-FIFO memory guarantees the high maximum throughput.

For the high-speed input-buffered switches, the arbitration time is an important parameter to determine the entire switching speed especially when the external link speed is significantly high. The arbitration time is determined by the number of arbitration round, which can be defined as the maximum number of the HOLs of an input port to be checked for an input in a time slot. Since the Odd-Even switch has two types of FIFO, it needs two rounds, the first for even FIFO and the second for odd FIFO. GIQ switch takes $w$ times arbitration irrespective of the number of group as stated in the end of Section 2 since the arbiter checks only the destination addresses of the cells inside the window. RAIB switch needs

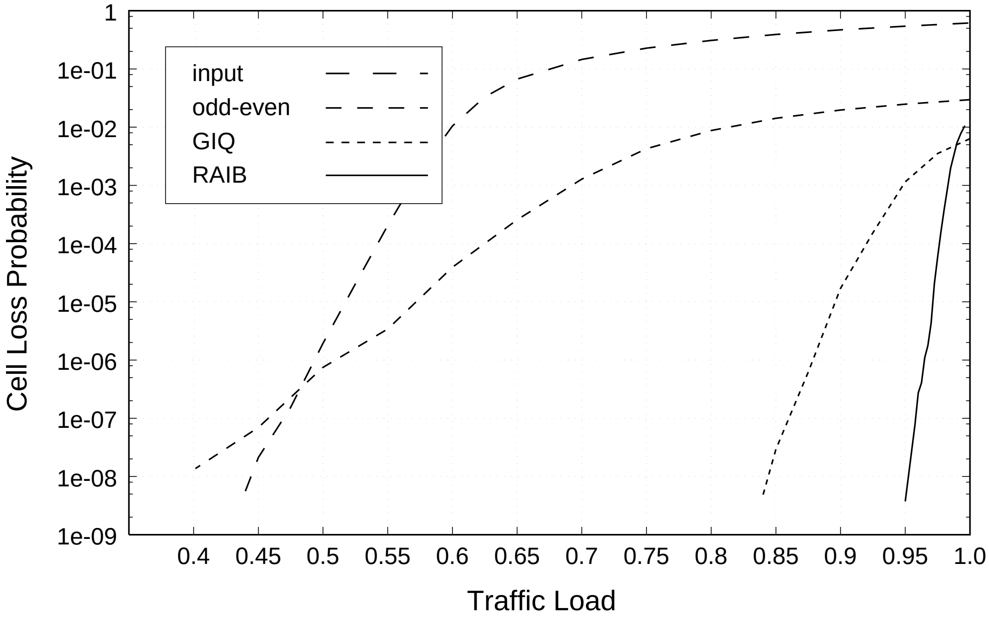

Figure 7: Cell loss probability of input-buffered switches (switch size : $8 \times 8$, buffer size : 16 cells/port).

$N$ arbitration rounds since there are $N$ ABs in every input module. However, the number of arbitration rounds for the RAIB switch can be changed depending on the use of MCTA. For example, if the MCTA factor, $m$, is 2, then the mean number of arbitration rounds is $N/2$ since an arbitration is executed in every two time slots. In general, the arbitration methods with non-FIFO memory needs larger number of arbitrations.

Besides of the number of the arbitration round, the memory access time affects the switching speed significantly. The memory access time is determined by the number of cells which can be served from an input port in a time slot. It is well documented in the researches on the common-buffered switch whose access speed is up to the $N + 1$ times the external link speed, $V$ [17, 18]. Assuming the same device property, the time of looking-up the routing table is another factor affecting to the switch speed. It may not be negligible as the switch size increases. However, up to the medium size of the switch the looking-up time is smaller than the arbitration time.

## 6. Conclusions

One of the most promising solutions of ATM switches is based on the input-buffering principle. Such switches have low complexity, low cost of implementation, and high-speed switching property. The limited throughput of ordinary input-buffered switch with FIFO can be improved by substituting FIFO with non-FIFO memory and by diverse arbitration and cell selection strategies. However, the complex arbitration scheme may affect the hardware implementation. Also, when the external link speed is too high and the switch sizes get larger, the arbitration time is another important parameter determining the whole switching speed.

# References

[1] J. J. Bae and T. Suda, *"Survey of Traffic Control Schemes and Protocols in ATM Networks,"* Proc. IEEE, Vol. 79, No. 2, Feb. 1991, pp. 170-189.

|  | Ordinary input-buffered switch | Odd-Even switch | GIQ switch | RAIB switch |
|---|---|---|---|---|
| Cell storage | FIFO | dual-FIFO | complete shared buffer | complete shared buffer |
| Arbitration method | not considered | Odd-Even arbitration | look ahead | address buffers |
| Arbitration round | 1 | 2 | $w$ | N<br>affected by MCTA |
| Memory access speed | $2V$ | $2V$ | $2GV$,<br>proportional to $G$ | $2V$ |
| Memory utilization efficiency | very low | low | proportional to the group size | high |
| Multicasting, Multi-class service | not considered | not considered | not considered,<br>easily implementable | CCAB and AB |
| Hardware Implementation | very easy | easy,<br>splitter | Address Buffer,<br>look-ahead scheme | ASBM, MCTA,<br>Address Buffer |
| Maximum Throughput | 58.6% | 71.9% | 96.2%<br>(G=2) | 100% |

Table 1: Comparison of input-buffered switch architectures ($N$: switch size, $V$: external link speed, $w$: window size, $G$: group size (the number of input ports in a same group) ).

[2] Young Boo Kim, Soon Seok Lee, Chang Hwan Oh, Young Sun Kim, Chimoon Han, and Chu Hwan Yim, *"An Architecture of Scalable ATM Switching System and its Call Processing Capacity Estimation,"* ETRI Journal, Vol. 18, No. 3, Oct. 1996, pp. 107-125.

[3] M. J. Karol, K. Y. Eng, and H. Obara, *"Improving the Performance of Input-Queued ATM Packet-Switches,"* Proc. of INFOCOM '92, Vol. 1, Florence, May 1992, pp. 110-115.

[4] N. McKeown, V. Anantharam and J. Walrand, *"Achieving 100% Throughput in an Input-Queued Switch,"* INFOCOM '96, Vol. 1, 3a.4, March 1996, pp. 296-302.

[5] A. K. Gupta and N. D. Georganas, *"Analysis of a Packet Switch with Input and Output Buffers and Speed Constraints,"* Proc. of ICC '91, 1991, pp. 694-700.

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

[7] E. Del Re and R. Fantacci, *"Performance Evaluation of Input and Output Queueing Technologies in ATM Switching Systems,"* IEEE Trans. on Communications, Vol. 41, Oct. 1993, pp. 1565-1757.

[8] 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.

[9] 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.

[10] Hakyong. Kim, Changhwan. Oh and Kiseon. Kim, *"A High-Speed ATM Switch Architecture with Random Access Input Buffer,"* JCCI '96 (in Korean), Vol. 1, April 1996, pp. 240-244.

[11] 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.

[12] R. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues,* Artech House, Norwood, MA, 1994.

[13] M. Karol and M. Hluchyj, *"Input versus Output Queueing on a Space Division Switch,"* IEEE Trans. Communications, Vol. 35, No. 12, 1987, pp. 1347-1356.

[14] M. G. Hluchyj and M. J. Karol, *"Queueing in High Performance Packet Switching,"* IEEE JSAC, Vol. 6, No. 9, Dec. 1988, pp. 1587-1597.

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

[16] R. Y. Awdeh and H. T. Mouftah, *"Survey of ATM Switch Architectures,"* Computer Networks and ISDN Systems 27, 1995, pp. 1567-1613.

[17] K. Oshima, H. Yamanaka, H. Saito, et al., *"A New ATM Switch Architecture Based on STS-Type Shared Buffering and Its LSI Implementation,"* ISS '92, 1992, pp. 359-363.

[18] A. Jajszczyk, M. Roszkiewicz and J. Garcia-Haro, *"Comparison of ATM Shared-Memory Switches,"* ISS '95, 1995, pp. 409- 413.