Cell selection algorithm for the multiple input-queued ATM switch: Chessboard and Random cell selections

Hakyong Kim, Kiseon Kim, Hyunho Yoon, Yongtak Lee, and Changhwan Oh

ComNet Lab, Dept. Information & Communications,
Kwang-Ju Institute of Science and Technology (KJIST)
1, Oryong-dong, Puk-ku, Kwangju, 500-712, S. Korea
URL: http://charly.kjist.ac.kr/~hykim/

Abstract

Proposed is a simple and efficient cell selection algorithm for the multiple input-queued ATM switch, named *chessboard cell selection algorithm*. The proposed algorithm selects one of the transmission requests for the output port with the lowest value of transmission request sum. By doing so, we can reduce a newly introduced *Front-Of-Line (FOL) blocking* so as to achieve an enhancement in the throughput for the uniform arrival traffic. Besides the enhanced throughput, the proposed algorithm can reduce mean cell delay by 50% or more and cell loss probability by 90% or more than the random selection scheme. Time complexity is $O(N^2)$ in the worst case, where $N$is the switch size.

1 Introduction

To improve the limited throughput $(2-\sqrt{2})$ of the nonblocking ATM switch with a single FIFO in each input port, multiple or bifurcated input-queued ATM switches were proposed where each input port manages m separate FIFOs, called bifurcated queues, respectively for an output group as shown in Figure 1 [1, 2, 3]. It was reported that the multiple input-queued ATM switch attains a significant enhancement in terms of the switch throughput as the number of bifurcated queues or bifurcation parameter m increases. Moreover, the switch does not require either internal speedup in the switch fabric or expansion in the size of the switch fabric. Only problem is that the switch should control multiple bifurcated queues in every time slot, and thus, it is a key issue to engineer a simple and efficient cell selection algorithm for the multiple input-queued switch. Especially, the engineering issue has a significance in developing future high-speed switching system.

Until now, however, there have been limited number of researches on the cell selection algorithm for the multiple input-queued ATM switch, while much studies on the performance evaluation of the switch [4]-[8]. In references [4] and [5], the authors classified the arbitration rule for the multiple input-queued ATM switch into two cartegories according to the number of cells switched from an input port in a time slot: the free contention/arbitration rule and the restricted contention/arbitration rule. With the former rule, multiple cells can be switched from an input port conditioned that no more than one cell is switched from each bifurcated queue. With the latter rule, on the other hand, no more than one cell is switched from an input port even though there are multiple bifurcated queues in an input port. As the result of restriction in the number of switched cells from an input port, the switch operates at the same speed as the external link rate.

One remarkable and detailed work in the cell selection algorithm for the multiple input-queued switch is the two-dimensional round-robin scheduling proposed by LaMaire and Serpanos, which makes an arbitration map first and then performs arbitration [9]. In this approach, the arbitration itself is very fast since it is performed based on the arbitration map. However, making the arbitration map is very complicated and thus it takes much processing time.

In this paper, we propose a novel cell selection algorithm for the multiple input-queued ATM switch, named *chessboard cell selection algorithm*, which also uses a simply-made arbitration map. The algorithm selects a non-contending transmission request first or, if not, one of the transmission requests for the output port with the lowest value of total transmission requests. By doing so, the proposed algorithm can minimize the *Front-Of-Line (FOL) blocking* which can be resolved potentially by intelligent selection of one transmission

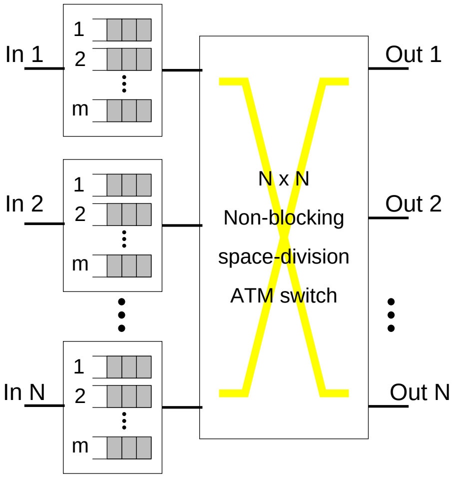


Figure 1: *Multiple input-queued ATM switch.*

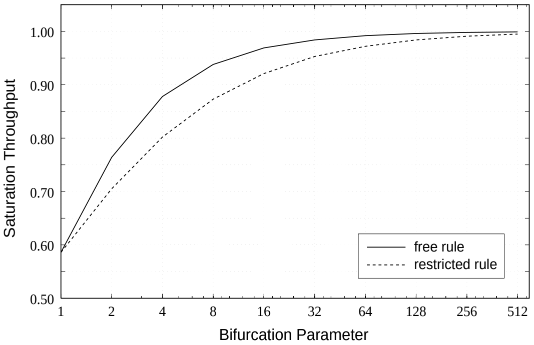


Figure 2: *Saturation throughput for the multiple input-queued ATM switch* [5].

request from multiple bifurcated queues in an input port. The complexity of the proposed algorithm is $O(N^2)$ in the worst case, where N is the switch size. Even though the proposed cell selection algorithm is simple comparatively, it guarantees a significant enhancement in the switch throughput as well as the decrease in the mean cell delay and cell loss probability.

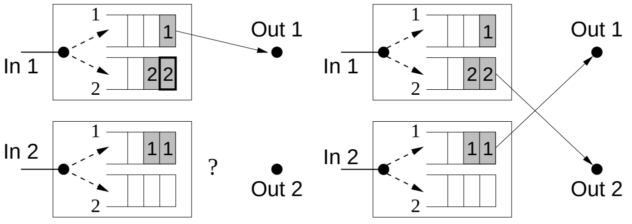
The paper is organized as follows: In Section 2, described are the concept of the multiple input-queued ATM switch and the cell selection approach for the switch. In this section, we also introduce the FOL blocking which is a new concept of blocking phenomenon occurred in the multiple input-queued switch. In Section 3, we elaborates on the chessboard cell selection algorithm with an arbitration example. In Section 4, we present simulation results on the switch throughput, mean cell delay, and cell loss probability for the homogeneous arrival traffic.

2 Cell selection in the multiple input-queued ATM switch and Front-Of-Line blocking

It was proved that the multiple input-queueing approach in the non-blocking space-division ATM switch raises the saturation throughput up to 100% as the switch size increases [4]-[8]. However, since it should manage multiple queues simultaneously the control logic generally becomes more complicated than other input-queueing schemes such as window policy or cell discarding. Moreover, since the multiple input-queued switch has plural memory blocks in each input port it can switch more than one cell from an input port in a time slot, causing the internal speedup in the switch fabric or in the interfaces between the switch fabric and input modules. Therefore, it is essential to device a simple and efficient control algorithm requiring neither internal speedup nor expansion of the switch fabric.

In references [4] and [5], the authors classified the contention/arbitration rule for the multiple input-queued ATM switch into two categories according to the number of switched cells from an input port: the free contention/arbitration rule and the restricted contention/arbitration rule. As the names imply, the former can switch multiple cells from an input port while the latter switches no more than one cell in a time slot. The authors also pointed out that the multiple input-queued switch adopting the free rule has slightly better performance characteristics than the switch employing the restricted rule. Figure 2 shows the saturation throughput of the multiple input-queued ATM switch both for the free and restricted rules, where saturation throughputs of both rules approach to 1.0 as m increases [5]. However, the restricted rule operates at the same speed as the external link speed while the free rule should operate at the speed of m times the external speed. Hereafter, we focus our attentions on the multiple input-queued ATM switch with the restricted rule since the switch is suitable for the future high-speed switching system.

In the multiple input-queued switch employing the restricted rule, a challenge is how to select and switch one among multiple transmission requests from each input port. Moreover, the selected cell or transmission request should not be duplicated for an output port. If not, the switch may require either internal speedup or



(a) (b)

Figure 3: *Cell selection in the multiple input-queued approach and the Front-Of-Line (FOL) blocking.*

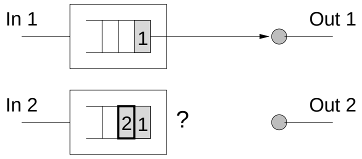


Figure 4: *Head-of-line (HOL) blocking.*

switch fabric expansion as well as the output buffer. Intuitively, the random selection might be the most easy way to come up with, even though its hardware implementation is not much simpler than that of other types of selection schemes. However, in the schemes where each input manages multiple queues, a cell selection among multiple queues could have a considerable influence on the switching performance, as will be described below. For the reason, it is necessary to engineer a more intelligent cell selection scheme than those mentioned above. A simple example shown in Figure 3 corroborates this consideration, which depicts the cell selection in the multiple input-queued ATM switch and the *Front-Of-Line (FOL) blocking*.

As shown in Figure 3, each input port manages two seperate FIFOs, one for each output port. We assume that the switch size is $N \times N$ where N is 2 for easy understanding, and the random cell selection is employed as the cell selection method. We further assume that the restricted rule is adopted to allow each input to switch no more than one cell in a time slot. Note that each queue stores cells for a specific output port in order to implement the restricted rule easily. In Figure 3, when input port In 1 is in its turn to designate one HOL cell (or transmission request), there are two possibilities: one is to select the HOL cell for output port Out 1 as in Figure 3(a) and the other is to select the HOL cell for output port Out 2 as in Figure 3(b). In the case of Figure 3(a), input port In 2 can not select the HOL cell for output port Out 1 since the output port Out 1 has already been reserved for input port In 1. In the case of Figure 3(b), on the other hand, input port In 2 can select the HOL cell for output port Out 1 since the output port Out 1 has not been reserved yet. For two HOL cells for output port Out 1, it is a natural result that one of them is selected and the other is blocked as the result of the output contention. In Figure 3(b), however, the cell destined for Out 2 is selected and switched additionally. We name this kind of blocking occurred in the case of Figure 3(a) *the Front-Of-Line blocking* or simply *FOL blocking*, since the blocking occurs at the front line of bifurcated queues. The FOL blocking, however, could be resolved potentially by intelligent selection of one of HOL cells as in Figure 3(b). The HOL cell of the queue for output port Out 2 in input port In 1 in Figure 3(a) corresponds to the FOL-blocked cell since the cell can be selected as in Figure 3(b) in terms of output port Out 2.

The FOL blocking is comparable to the HOL blocking in the single input-queued switch. The difference between them is that the FOL blocking occurs at the neighboring HOL positions while the HOL blocking occurs at the position behind the HOL. Figures 3 and 4 make the difference clear. Note again that the FOL blocking is a blocking which can be potentially resolved by selecting another HOL cells intelligently.

3 Chessboard cell selection algorithm

There is no doubt that the switch in the case of Figure 3(b) provides better switching performance characteristics than the other. This fact remains us how to find out such case (that is, Figure 3(b)) intelligently and, in the previous example, we have found out that the selection of FOL-blocked cells (HOL cells for Out 2 in In 1) could be one method. The idea of selecting the FOL-blocked cells first is the main idea behind the chessboard cell selection algorithm proposed in this paper.

Before describing the chessboard cell selection algorithm in detail, let us make some assumptions. Time is slotted equally and they are called the time slot. In the beginning of every time slot, each input port sends the arbiter cell transmission requests r_{ij} where i and j are input and output port numbers respectively. The request information designates that input port i has a HOL cell destining for output port j. On receiving the request information, the arbiter makes a request map in a 2-dimensional array using the request information r_{ij} as shown in Figure 6, where one axis represents the input ports and the other for the output ports. The

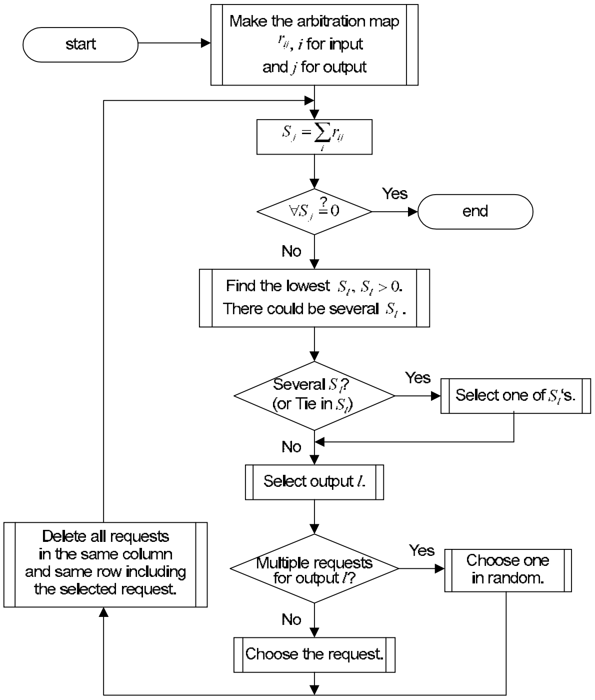


Figure 5: *Chessboard cell selection algorithm.*

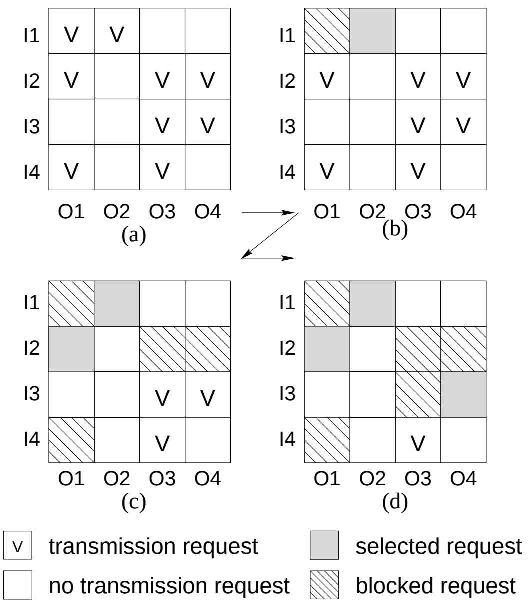


Figure 6: *Arbitration map and the procedure of the chessboard cell selection algorithm for a* 4 × 4 *switch.*

request map is usually called the arbitration map. Let S_j represent the transmission request sum for output j from all input ports, where $1 \leq j \leq N$. That is,

$$S_j = \sum_{i=1}^{N} r_{ij} .$$

Further, a set of S_j forms a request vector of $\boldsymbol{S} = \{S_j : 1 \leq j \leq N\}$. The request vector $\boldsymbol{S}$ is managed in coupled with the request map during the arbitration. After making the arbitration map and the request vector $\boldsymbol{S}$, the arbiter performs the following chessboard cell selection process:

Step 1 Repeat **Step 2** through **Step 5** until there is no transmission request left.

Step 2 Sum the number of requests for each output port and update the request vector $\boldsymbol{S}$.

Step 3 Select the output port l with the lowest value of the request sum S_l greater than zero among $\boldsymbol{S}$. If there are ties in the lowest value of S_l, then select one output port randomly.

Step 4 Choose one request for the output port selected in **Step 3**. If there are multiple requests for the output port from different input, then select one request randomly.

Step 5 Mark the selected request and deletes all other requests in the same row and column including the selected request.

The flow chart shown in Figure 5 elaborates on the chessboard cell selection algorithm described above. This procedure confirms us that the time complexity of the proposed cell selection algorithm is $O(N^2)$ in the worst case, because the sequential sorting has the complexity of $O(N)$. If a more efficient sorting algorithm is used, the complexity becomes lower than $O(N^2)$. Comparing the chessboard algorithm with ordinary cell selection schemes, only the added process in the arbitration is to sum the transmission requests and to compare the (request) sum. Since the addition process is a gate-level operation, it has no significant effect on the control time and hardware implementation.

One example for the case of 4 × 4 switch is illustrated in Figure 6 step by step. Figure 6(a) shows the arbitration map just after reception of cell transmission requests. The number of requests for each output is summed and, in this example, the request vector at first becomes $\boldsymbol{S} = \{3, 1, 3, 2\}$. Since the request vector has the lowest value for output port 2 and there is only one request for the output port, i.e., non-contending request, the request $r(1,2)$ is selected first. Other requests in the same row and column containing the selected request are deleted from the map, which is illustrated in Figure 6(b). And then, the arbiter updates the request vector into $\boldsymbol{S}=\{2,0,3,2\}$ where 0 designates the output port selected. Since there occurs, in this time, a tie between output ports 1 and 4, the arbiter selects output port in random. Let us assume that the arbiter selects the request output port 1 and the request of $r(2,1)$ in similar way. Such steps are repeated up to 4 times or until all the request sums become zero.

As an extension of the proposed chessboard cell selection algorithm, we can think of two-dimensional chessborad cell selection scheme. The two-dimensional scheme resolves the tie problem occurred when two or more output ports have the same number of transmission requests (or same value of request sum). It compares the number of requests in terms of input port when there occur ties between output ports, and selects one request from the input with the lowest number of requests as in the one-dimentional chessboard scheme. For implementation, an additional sorting process should be placed between the second and third comparison operations of Figure 5.

The proposed chessboard cell selection algorithm can be used in other input-queueing schemes such as window policy. In this case, the window size corresponds to the number of queues (m) in an input port. Only difference is that, in the window policy, the HOL cell has the highest priority to attend arbitration among the cells in the same queue while, in the multiple input-queueing scheme, there is no priority between multiple bifurcated queues. Thus, cell order might be changed when the chessborad algorithm is applied to the window policy.

4 Results and discussion

To compare the merits of the proposed scheme with respect to the random selection algorithm, we simulated the throughput, mean cell delay, and cell loss probability for the multiple input-queued switch employing the restricted rule. In the simulation, cells are assumed to arrive at the beginning of each time slot and to leave at the end of each time slot. The arrival traffic in each input port is assumed to be *homogeneous.* That is, the arrival traffic is a Bernoulli process distributed independently and identically for each input port and uniformly for all output ports [5]. The switch size used is 16×16.

In Figure 7, the saturation throughput for the 16×16 multiple input-queued switch is plotted over different bifurcation parameters (m). When $m = 4$ the saturation throughput is over 0.9 and when $m = 16$ it is very close to 1.0. The throughput of the proposed cell selection algorithm is better than that of the random selection scheme by 10% or more.

Figures 8 and 9 show the improved performances in terms of mean cell delay and cell loss probability when we employed the chessboard cell selection algorithm. The switch size is assumed to be 16×16 and the buffer size is assumed to be infinite for the mean cell delay and to be 8 cells per each queue for the cell loss probability. Figure 8 tells that the cells experience less mean delay by 50% or more when the chessboard cell selection algorithm is employed. Figure 9 shows that the cell loss probability is reduced by 90% or more for a given traffic load and the limited buffer space.

Summerizing the performance evaluation, we could achieve remarkable enhancements by minimizing the FOL blocking effect through the use of chessboard cell selection algorithm.

5 Conclusions

In this paper we described multiple input-queueing approach in the non-blocking space-division ATM switch and proposed a novel cell selection algorithm for the switch, named *chessboard cell selection algorithm.* The proposed algorithm is mainly targeted for the restricted contention/arbitration rule in the multiple input-queued switch which allows no more than one cell from an input port to be switched in a time slot. By doing so, the switch can operate at the same speed as the external link rate. However, the restriction causes a new blocking phenomenon named *Front-Of-Line (FOL) blocking*, comparable to the HOL blocking in the single input-queued switch. However, the FOL blocking can be resolved potentially by intelligent selecting one of transmission requests from multiple queues in an input port. The chessboard algorithm improves the switching performance by minimizing the FOL blocking in a simple manner. The algorithm first selects non-contending request. If not, it selects one of request from the output ports having the lowest request sum. The performance enhancement by the proposed cell selection algorithm was proved through computer simulation.

When the chessboard algorithm was employed as the cell selection algorithm for the multiple input-queued ATM switch, the mean cell delay was decreased half compared to the case of random cell selection scheme. The cell loss probability was improved by 90% or more. Since the proposed algorithm has the time complexity less than $O(N^2)$ while guaranteeing high performance, the algorithm is expected to be a promising candidate for the high-speed high-performance ATM switch for future.

References

[1] M.M. Ali and H.T. Nguyen, "A neural network implementation of an input access scheme in a high-speed packet switch," *Proc. IEEE GLOBECOM'89*, vol.2, Dallas, TX, Nov. 1989, pp. 1192-1196.

[2] M.K.M. Ali and M. Youssefi, "The performance analysis of an input access scheme in a high-speed packet switch," *Proc. IEEE INFOCOM '91*, vol.2, Bal Harbour, FL, 1991, pp.454-161.

[3] H.-C. Chi and Y. Tamir, "Decomposed arbiters for large crossbars with multi-queue input buffers," *Proc. IEEE International Conference of Computer Design (ICCD) '91*, Los Angeles, CA, Oct. 1991, pp.233-238.

[4] Hakyong. Kim, C. Oh, and K. Kim, "Throughput Analysis of the Bifurcated Input-Queued Packet Switches with Restricted Contention," *IEE Electronics Letters*, vol.34, no.17, 20th August 1998, pp.1651-1652.

[5] Hakyong Kim, C. Oh, Y. Lee, and K. Kim, "Throughput analysis of the bifurcated input-queued ATM switch," accepted by *IEICE Tr. Commun.*

[6] C. Kolias and L. Kleinrock, "Throughput analysis of multiple input-queueing in ATM switches," Broadband Communications, L. Mason & A. Casaca (eds.), Chapman & Hal London, U.K., 1996, pp.382-393.

[7] G. Thomas, "Bifurcated Queueing for Throughput Enhancement in Input-Queued Switches," *IEEE Commun., Letters*, vol.1, no.2, March 1997, pp.56-57.

[8] K.L. Yeung and S. Hai, "Throughput analysis for input-buffered ATM switches with multiple FIFO queues per input port," *IEE Electronics Letters*, vol.33, no.19, Sept. 11, 1997, pp.1604-1606.

[9] R.LaMaire and D.Serpanos, "Two-dimensional round-robin schedulers for packet switches with multiple input queues," IEEE/ACM Tr. Networking, vol.2, no.5, Oct. 1994, pp.471-482.

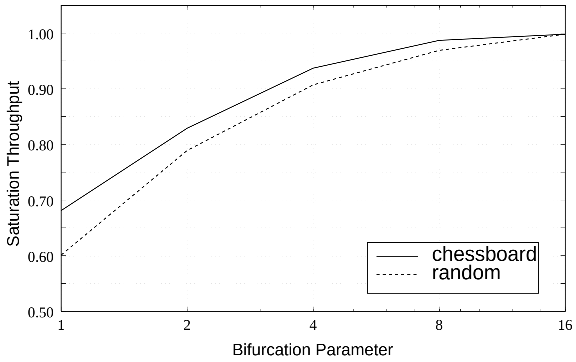


Figure 7: *Saturation throughput for a* 16×16 *switch over different values of* m.

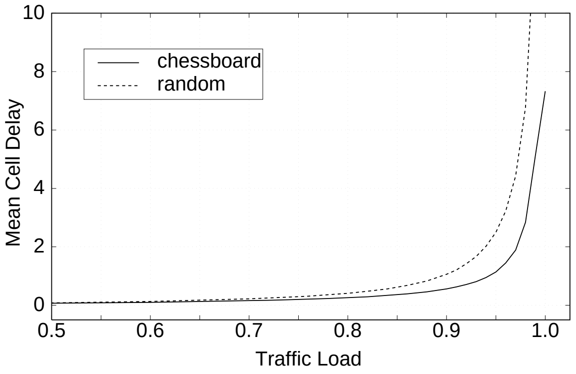


Figure 8: *Mean cell delay for a* 16×16 *switch with infinite buffer space* ($m = 8$).

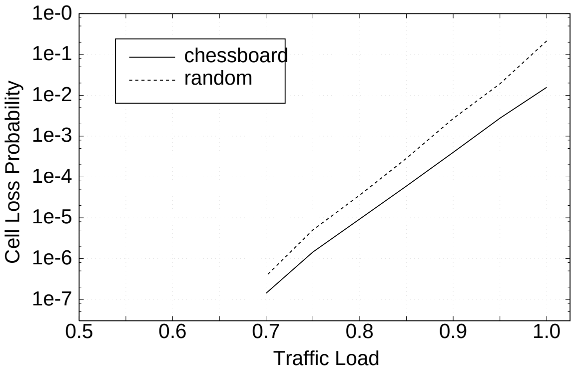


Figure 9: *Cell loss probability for a* 16×16 *switch with the buffer space of 8 cells per queue.*