High-Performance Round-Robin Arbitration Schemes for Input-Crosspoint Buffered Switches

Roberto Rojas-Cessa

***Abstract*— Combined input-crosspoint buffered switches provide high-performance switching and relax arbitration timing for packet switches with high-speed ports. It has been shown that a switch with one-cell crosspoint buffers and round-robin selection at input and output ports provides 100% throughput under uniform traffic. However, under admissible traffic patterns with nonuniform distributions, only weight-based arbitration schemes are reported to provide high throughput by these switches. This paper introduces two arbitration schemes based on round-robin arbitration for combined input-crosspoint buffered packet switches. The proposed schemes provide nearly 100% throughput for several admissible traffic patterns, including uniform and unbalanced traffic, using one-cell crosspoint buffers.**

***Index Terms*— buffered crossbar, virtual output queue, adaptable-size frame, captured frame, service frame**

I. Introduction

Combined input-crosspoint buffered packet switches are an alternative to input-buffered switches to provide high-performance switching and to relax arbitration timing for packet switches with high-speed ports. These switches use time efficiently because input and output port selections are performed independently.

It is common to find the following practices in packet switch design. 1) Segmentation of incoming variable-size packets at the ingress side of a switch to perform internal switching with fixed-size packets, or cells, and reassembling of the packets at the egress side before they depart from the switch. 2) Use of separate queues at the inputs, one for each output, known as virtual output queues (VOQs) to avoid head-of-line (HOL) blocking [1]. 3) Use of crossbar fabrics for implementation of packet switches because of their non-blocking capability, simplicity, and market availability. This paper follows these practices.

As an example of the stringent timing with increasing data rates, a switch with 40-Gbps (OC-768) ports transferring a 64-byte packet must perform input (or output) arbitration within 12.8 ns. An input-crosspoint buffered switch can perform selection of a cell at inputs and outputs within this time. However, the number of buffers[1] in a crossbar grows in the same order as the number of crosspoints, $O(N^2)$, where N is the number of input/output ports. This makes implementation costly for a large crosspoint-buffer size or a large number of ports. One way to keep the buffer complexity feasible is to use small crosspoint buffers.

In general, arbitration schemes are required to provide: a) low complexity, b) fast contention resolution, c) fairness, and, d) high matching efficiency. Arbitration schemes for buffered crossbars require low complexity. These schemes provide fast contention resolution as the selection is simplified. High matching efficiency is achieved with simpler arbitration schemes than those used in bufferless crossbars (e.g., input-buffered switches) at the expense of having to allocate buffers at the crosspoints. These features have been shown to be attractive in several switches [2]-[9].

A buffered crossbar switch with timestamp-based arbitration and VOQs at the input ports showed that the crosspoint buffer size can be small if the VOQs are provided with enough storing capacity [4]. Furthermore, it has been shown that a switch using one-cell crosspoint buffers in a buffered crossbar with VOQs at the inputs, a simple round-robin arbitration scheme for input and output arbitration, and a credit-based flow control provides 100% throughput for uniform traffic [6]. However, as actual traffic may present nonuniform distributions, it is necessary to provide arbitration schemes that provide 100% throughput for admissible traffic. Admissible traffic is defined in [10].

One way to provide 100% throughput under nonuniform traffic patterns is by using weight-based arbitration schemes, where weights are assigned to input queues proportionally to their occupancy or HOL cell age [10]. It has been shown that weight-based [7] and priority-based [9] schemes in buffered crossbars can provide high throughput under various traffic patterns. Two schemes were presented in [7]: one is based on the selection of the longest VOQ occupancy at inputs and round-robin selection at the outputs; the other scheme is based on the selection of the oldest cell first (OCF) instead of VOQ occupancy. However, weight-based schemes need to perform comparisons among all contending queues, which can be a large number. Furthermore, as the queuing structures tend to be flow-based, the number of comparisons is expected to increase. Moreover, weight-based schemes may starve some queues to provide more service to the congested ones, presenting unfairness [11]. On the other hand, round-robin algorithms have been shown to provide fairness and implementation simplicity, because no comparisons are needed among queues, and high-performance under uniform traffic. However, schemes based on round-robin selection have not been shown to provide near 100% throughput under nonuniform traffic patterns with a

Roberto Rojas-Cessa is with the Department of Computer and Electrical Engineering, New Jersey Institute of Technology, University Heights, Newark NJ 07102 USA. Email: rojasces@njit.edu.

[1] This paper uses the terms queue and buffer interchangeably.

buffered crossbar that has small crosspoint buffers. It has been shown that a switch with round-robin arbitration needs a large crosspoint buffer to provide high throughput under admissible unbalanced traffic [8], where the unbalanced traffic model is a nonuniform traffic pattern [6]. This large buffer can make the implementation of a switch costly.

Considering the advantages round-robin arbitration presents, it is of interest to determine if an arbitration scheme based on round-robin selection for buffered crossbars can deliver nearly 100% throughput under admissible traffic with uniform and nonuniform distributions, such as unbalanced traffic, with one-cell crosspoint buffers.

This paper proposes two arbitration schemes for buffered crossbars, based on round-robin selection. The proposed schemes use the concept of adaptive frame sizes and frame sizes based on VOQ occupancies. This paper shows that these arbitration schemes can achieve nearly 100% throughput under the unbalanced traffic pattern with one-cell crosspoint buffers. This paper also shows that a switch using either of these arbitration schemes retains the high performance of simple round-robin arbitration under uniform traffic.

This paper is organized as follows. Section II presents the combined input-crosspoint buffered switch model. Sections III and IV introduce the proposed arbitration schemes. Section V presents a simulation study of the throughput and delay performance of these schemes under uniform and nonuniform traffic patterns. Section VI discusses the properties of the proposed arbitration schemes. Section VII presents the conclusions.

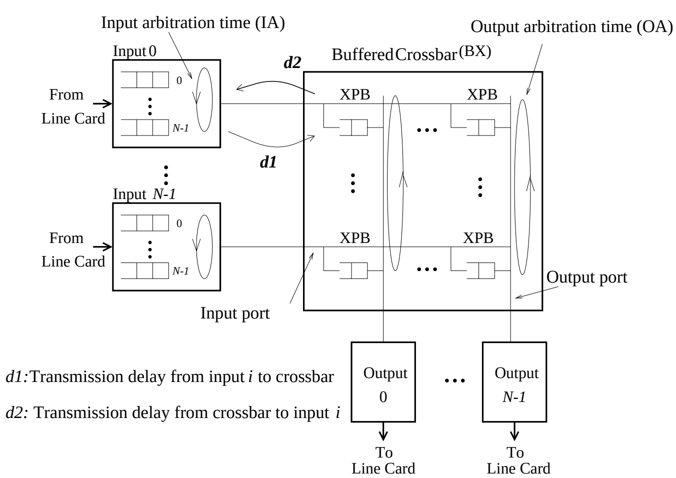

Fig. 1. $N \times N$ buffered crossbar with VOQ structure

II. Combined Input-Crosspoint Buffered Switch Model

Figure 1 shows a buffered crossbar (BX) switch with N inputs and outputs. In this switch model, there are N VOQs at each input. A VOQ at input i that stores cells for output j is denoted as $VOQ(i,j)$. A crosspoint (XP) element in the BX that connects input port i, where $0 \leq i \leq N-1$, to output port j, where $0 \leq j \leq N-1$, is denoted as $XP(i,j)$. The buffer at $XP(i,j)$ is denoted as $XPB(i,j)$. The size of $XPB(i,j)$, k, is given as the number of cells that can be stored. A credit-based flow-control mechanism indicates to input i whether $XPB(i,j)$ has room available for a cell or not, as described in [6]. $VOQ(i,j)$ is said to be eligible for arbitration if the VOQ is not empty and the corresponding $XPB(i,j)$, at BX, has room to store a cell.

The round trip (RT) time is defined as the sum of the delays of the input arbitration (IA), the transmission of a cell from an input to the crossbar ($d1$), the output arbitration (OA), and the transmission of the flow-control information back from the crossbar to the input ($d2$) [6]. Figure 1 shows an example of RT for input 0 by showing the transmission delays for $d1$ and $d2$, and the arbitration the times, IA and OA. Cell and bit alignments are included in the transmission times. The condition for this switch to avoid underflow, as in [6], is such that:

$$RT = d1 + OA + d2 + IA \leq k \quad (1)$$

where k is the crosspoint buffer size, in time slots, which is equivalent to the number of cells that can be stored. In other words, the crosspoint buffer must be able to store a number of cells to keep the buffer busy (i.e., transmitting cells) during at least one RT time.

III. Round-robin with Frame-size Occupancy-based Arbitration (RR-FO) Scheme

This arbitration scheme is round-robin based. A frame is related to a VOQ, and it is the number of one or more cells in a VOQ that are eligible for arbitration (as allowed by the flow-control mechanism). A new frame gets assigned a size, in number of cells, each time the previous frame is serviced. The frame size is determined by the cell occupancy of the VOQ at time of the frame service completion. This is called *captured-frame size*. A VOQ is said to be in on-service status if the VOQ has a frame size of two or more cells and the first cell of the frame has been transmitted. An input is said to be on-service if there is at least one on-service VOQ. A VOQ is said to be off-service if the last cell of the VOQ's frame has been sent, or no cell of the frame has been sent to BX. Note that for frame sizes of one cell, the associated VOQ is off-service during the matching of its one-cell frame. An input is said to be off-service if all VOQs are in off-service status.

At the time t_c of selecting the last cell of a frame of $VOQ(i,j)$, the next frame is assigned a size equal to the occupancy of $VOQ(i,j)$. Cells arriving at $VOQ(i,j)$ at any time t_d, where $t_d > t_c$, are considered for selection until the current frame is totally served and they are included in a new captured frame.

For each VOQ, there is a captured frame-size counter, $CF_{i,j}(t)$.[2] The value of $CF_{i,j}(t)$, $|CF_{i,j}(t)|$, indicates the frame size, that is, the maximum number of cells that a $VOQ(i,j)$ has as candidates in the current and future time

[2]We called captured frame size as it is the equivalent of having a snapshot of the occupancy of a VOQ at a given time t. The occupancy is then considered for determining the frame size.

slots, one per time slot. $|CF_{i,j}(t)|$ takes a new value when the last cell of the current frame of $VOQ(i,j)$ is selected. $|CF_{i,j}(t)|$ decreases its count each time a cell is selected, other than the last.

The input arbitration process is as follows: an input arbiter selects an eligible on-service VOQ in a round-robin order, starting from its pointer position. If no on-service VOQ is present, an off-service VOQ is selected in a round-robin order. If the input arbiter selects $VOQ(i,j)$ and

i) If $|CF_{i,j}(t)| > 1$: $|CF_{i,j}(t+1)| = |CF_{i,j}(t)| - 1$ and this VOQ is set as on-service.

ii) Else ($|CF_{i,j}(t)| = 1$): $|CF_{i,j}(t+1)|$ is assigned the occupancy of $VOQ(i,j)$, and $VOQ(i,j)$ is set as off-service.

The pointer at the input arbiter then moves to one position beyond the accepted $VOQ(i,j)$: to the next output or $(j+1) \mod N$.

The output arbitration process is round-robin selection with pointer persistency. After $XPB(i,j)$ has been selected by the output arbiter, the output pointer keeps pointing to input i.

IV. Round-robin with Adaptable-size Frame (RR-AF) Arbitration Scheme

This scheme uses an adaptive frame size instead of queue occupancy. The frame size is determined by the serviced and unserviced traffic, such that no intervention is needed to select the frame size. Each time that a VOQ (or a XPB at an output) is selected by the arbiter, the VOQ gets the right to forward a frame, where a frame is formed by one or more cells. One cell of a frame is dispatched in one time slot. The amount of serviced (and unserviced) traffic depends on the load experienced by the queues.

In each VOQ (and XPB), there are two counters: a frame-size counter, $FSC_{i,j}(t)$, and a current service counter, $CSC_{i,j}(t)$. The value of $FSC_{i,j}(t)$, $|FSC_{i,j}(t)|$, indicates the frame size, that is, the maximum number of cells that a $VOQ(i,j)$ can send in back-to-back time slots to the buffered crossbar, one cell per time slot. The initial value of $|FSC_{i,j}(t)|$ is one cell (i.e., its minimum value).[3] $CSC_{i,j}(t)$ counts the number of serviced cells at time slot t in a frame corresponding to a VOQ in a regressive fashion, starting with an initial value equal to FSC.[4] The initial value of $CSC_{i,j}(t)$, $|CSC_{i,j}(t)|$, is one cell (i.e., its minimum value).

The input arbitration process is as follows. An input arbiter selects an eligible $VOQ(i,j')$ in round-robin fashion, starting from the pointer position, j. For the selected $VOQ(i,j')$, if $|CSC_{i,j'}| > 1$, $|CSC_{i,j'}| = |CSC_{i,j'}| - 1$, and the input pointer remains at $VOQ(i,j')$, so that this VOQ has the higher priority for service in the next time slot and the frame transmission can continue. If $|CSC_{i,j'}| = 1$, the input pointer is updated to $(j'+1) \mod N$, $|FSC_{i,j'}|$ is increased by g cells, and $|CSC_{i,j'}| = |FSC_{i,j'}|$. For any other $VOQ(i,h)$, where $h \neq j'$, which is empty or inhibited by the flow-control mechanism, and it is positioned between the pointed output j and the selected $VOQ(i,j')$: if $|FSC_{i,h}| > 1$, $|FSC_{i,h}| = |FSC_{i,h}| - 1$. If there exists one or more VOQs that fit the description of $VOQ(i,h)$ at a given time slot, it is said that those VOQs missed a service opportunity at that time slot.

The variable g is the increment of the frame size each time a VOQ receives a complete frame service. Note that g may be equal to a constant or a variable value. For the rest of the paper, g assumes a value of N, unless otherwise stated. This assumption is discussed in Section V. The value of g affects the performance of RR-AF in different traffic scenarios. Note that when g $= 0$, RR-AF becomes pure round-robin [6].

The output arbitration works in a similar way to that at the inputs, considering $XPB(i,j)$ and its corresponding counters instead of considering the VOQs.

[3] It is considered that $|FSC_{i,j}(t)|$ can be as large as needed, although practical results have shown that its value does not reach large numbers.

[4] A regressive-fashion count is used in CSC as CSC only considers FSC at the end of a serviced frame.

V. Performance Evaluation

This section presents the performance evaluations of three combined input-crosspoint buffered switches, each one using a distinct arbitration scheme: RR-FO, RR-AF, and pure round-robin (RR) arbitration. In addition, an output buffered (OB) switch is also considered. The performance evaluations are produced through computer simulation. The traffic models considered have destinations with uniform and nonuniform distributions, the latter called unbalanced. Both models use Bernoulli arrivals. The simulation does not consider the segmentation and reassembly delays.

A. Uniform Traffic

Figure 2 shows simulation results of 32×32 combined input-crosspoint buffered switches with RR-AF, RR-FO, and an OB switch under uniform traffic with Bernoulli arrivals ($l = 1$) and bursts with average lengths of 10 and 100 cells ($l = 10$ and $l = 100$). As the load value below 0.5 shows no measurable difference, the graph only shows input load values larger than 0.5. The burst length is exponentially distributed. The buffered crossbars have crosspoint buffers with a size of one cell each. The simulation shows that the RR-AF and RR-FO arbitration schemes provide 100% throughput under uniform traffic.

This figure also shows that the average delay performances of RR-AF and RR-FO under Bernoulli arrivals are close to that of an OB switch. The adaptable frame-size condition in the RR-AF arbitration does not degrade the throughput performance, neither does it increase the average delay under this traffic model. As the RR-AF uses the history of serviced and unserviced traffic from all queues (i.e., VOQ and XPB), the switch practically adapts itself to uniform traffic. As for RR-FO, the service also adapts itself to the VOQ occupancy, and therefore, to uniform traffic. This figure also shows that RR-AF and RR-FO arbitrations offer a similar performance to

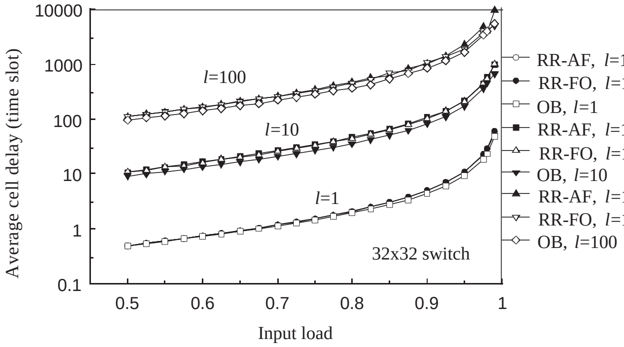

Fig. 2. Average cell delay of arbitration schemes under Bernoulli and bursty uniform traffic

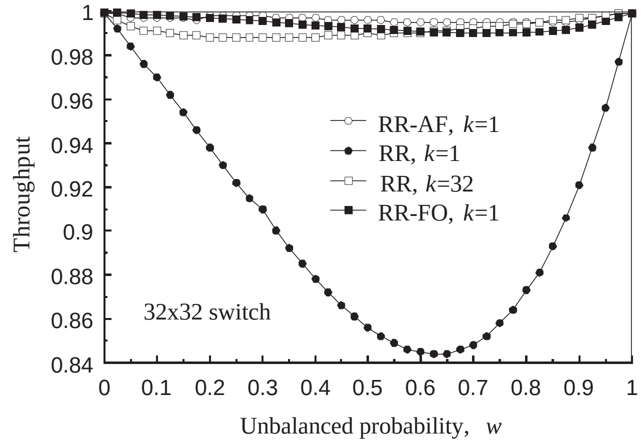

Fig. 3. Throughput performance of RR-AF, RR-OF, and RR under unbalanced traffic

that of an OB switch under bursty traffic. The average delay of both schemes is similar and proportional to the burst length, and the throughput is unaffected.

B. Nonuniform Traffic

RR-AF, RR-FO, and RR arbitrations were simulated under a nonuniform traffic model, the unbalanced traffic model [6]. The unbalanced traffic model uses a probability, w, as the fraction of input load directed to a single predetermined output, while the rest of the input load is directed to all outputs with uniform distribution. Let us consider input port s, output port d, and the offered input load for each input port ρ. The traffic load from input port s to output port d, $\rho_{s,d}$ is given by,

$$\rho_{s,d} = \begin{cases} \rho\left(w + \frac{1-w}{N}\right) & \text{if } s = d \\ \rho\frac{1-w}{N} & \text{otherwise.} \end{cases} \tag{2}$$

When $w = 0$, the offered traffic is uniform. On the other hand, when $w = 1$, it is completely directional, from input i to output j, where $i = j$. This means that all traffic of input port s is destined for only output port d, where $s = d$.

Four combined input-crosspoint buffered switches of size $N = 32$ were simulated under unbalanced traffic. The switches use RR-AF, RR-FO and, RR arbitration with $k = 1$, and the fourth switch uses RR with $k = 32$. RR-AF uses g $= N = 32$. Figure 3 shows that RR-AF and RR-FO with $k = 1$, provide above 99% throughput under the complete range of w. It is considered that such throughput is nearly 100% for practical purposes. These results show that RR-AF and RR-FO, with $k = 1$, outperform RR with $k = 32$. The use of $k = 1$ results in a feasible implementation of buffered crossbars as the size of the crosspoint buffer is reduced. In this example, RR, with $k = 32$ and a cell size of 64 bytes, would need 16Mb of memory, while RR-AF and RR-FO, with $k = 1$, would need 512kb of memory. Furthermore, a switch with either RR-AF of RR-FO can provide nearly 100% throughput under unbalanced traffic.

The high throughput of RR-AF is the product of increasing or decreasing service for a queue in proportion to its received and missed service, respectively. RR-AF ensures service to the queues with high loads by increasing the frame size, and to the other queues by using round-robin selection. In addition, the decreasing policy (i.e., FSC is decremented by one unit each time the VOQ misses service) for the frame-size counter ensures that the counter does not increase infinitely, as observed experimentally. RR-FO also provides service proportionally to the queue occupancy. Therefore, heavily loaded queues are not restricted from service.

The high throughput of RR-AF for an $N \times N$ switch has some dependency on the g value. To illustrate the dependency of N of RR-AF, Figure 4 shows the throughput of RR-AF for different switch sizes, $N = \{4, 8, 16, 32, 64\}$, with g $= 1$ and g $= N$. As expected, RR-AF with g $= 1$ resembles

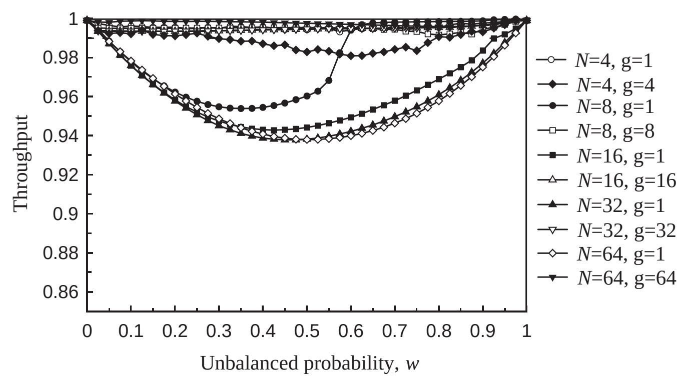

Fig. 4. Throughput performance of RR-AF for different switch sizes under unbalanced traffic

RR. Therefore, the throughput of medium-to-large switch sizes under this traffic type is higher with large g values than with g $= 1$. Switches with $N = \{8, 16, 32, 64\}$ have a throughput below 99% when g $= 1$. However, those switches have nearly 100% throughput when g $= N$. Note that contrary to the case of uniform traffic, an 8×8 switch delivers a low performance when g $= 1$ under unbalanced traffic.[5] In general, the throughput of RR-AF improves for medium-to-large switches with large g values (e.g., g $= \{N/2, N\}$).

[5]For an 8×8 switch, the performance is optimal under both uniform and unbalanced traffic patterns when g $= \{2, 4\}$.

VI. Properties of RR-AF and RR-FO

Under uniform traffic, the values of the frame counters in RR-AF are not expected to increase largely because of the cell distribution. The frame's size increase and decrease processes are balanced for all queues. This results in an arbitration that behaves as round-robin. Under unbalanced traffic, some queues are expected to have heavier loads than others. The queues with large occupancies have a higher probability of finishing servicing a complete frame in each opportunity of service, and their frame size increase consequently. The queues with low occupancy tend to have an average frame size rather small because they miss service opportunities. This different behavior of frame sizes results in higher service rates for queues with a larger number of arrivals than those for queues with a small number of arrivals. Moreover, the round-robin policy ensures that all queues receive service.

The use of a captured frame size and the service concepts used in RR-FO make this scheme deliver high performance under uniform and unbalanced traffic patterns. Note that in the case where a VOQ has no cells at the capturing time, the VOQ can still participate in a matching after a cell arrives, as long as the input is off-service. When a VOQ changes its status to on-service, that VOQ has higher priority than the others to continue sending its cells in subsequent time slots. When an input is off-service, all nonempty VOQs (independent of their CF value) become eligible for cell transmission. The use of round-robin selection at the outputs make RR-FO even simpler than RR-AF.

Note that RR-FO and RR-AF release the service for another VOQ (or XPB at the outputs) when the frame has been serviced. RR-AF even releases the service when no cell or no queue space is available. Therefore, starvation is not an issue in these schemes.

The implementation complexity of both RR-AF and RR-FO is low because of the following reasons: 1) a single-cell crosspoint buffer is sufficient to make the switch deliver high performance; 2) the arbitration scheme is round-robin based. RR-AF and RR-FO perform no comparisons among different queues. Arbiters do not differentiate queues as there are no priorities or weights considered. The provision of FSC and CSC counters to a queue is the major hardware addition in RR-AF compared to the implementation of RR. In a similar way, RR-FO needs CF counters and service flags as additional hardware. The FSC, CSC, and CF counters, as well as the arbiter pointers are updated, at most, once in a time slot in both schemes. Therefore, the timing for RR-AF and RR-FO are comparably simple, and of similar complexity to RR arbitration.

The round-robin schemes presented here have lower complexity than weight-based schemes. Weight-based schemes, such as those that consider cell age or queue occupancy length, need to perform comparisons among all contending queues, which can be a large number, to select the proper queue. RR-FO and RR-AF follow a predetermined order (round-robin) and check the flags (CSC for RR-AF and on-service flag for RR-FO) which are updated in previous time slots. Hardware-wise, round-robin schemes have no magnitude comparators needed in weight-based schemes.

Acknowledgement

The author thanks the anonymous reviewers for their insightful comments.

VII. Conclusions

This paper introduced two novel arbitration schemes for combined input-crosspoint buffered switches based on round-robin selection, RR-AF and RR-FO. RR-AF uses the concept of adaptable-size frame, where the frame size depends on the service received by a queue. RR-FO uses the concept of captured frame size and frame service status. Both round-robin schemes show nearly 100% throughput under uniform and unbalanced traffic models. The simulation results show that a buffered crossbar with one-cell crosspoint buffers is sufficient to provide such throughput with round-robin based arbitration. These arbitration schemes do not compare the status among different queues, such as weights or priorities, as they are based on simple round-robin. In addition to high throughput, these switches provide timing relaxation that allows high-speed arbitration and scalability.

References

[1] M. Karol, M. Hluchyj, "Queuing in High-performance Packet-switching," *IEEE J. Select. Area Commun.*, vol. 6, pp. 1587-1597, December 1988.

[2] Y. Doi and N. Yamanaka, "A High-Speed ATM Switch with Input and Cross-Point Buffers," *IEICE Trans. Commun.*, vol. E76, no.3, pp. 310-314, March 1993.

[3] E. Oki, N. Yamanaka, Y. Ohtomo, K. Okazaki, and R. Kawano, "A 10-Gb/s (1.25 Gb/s x8) 4 x 0.25-μm CMOS/SIMOX ATM Switch Based on Scalable Distributed Arbitration," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1921-1934, December 1999.

[4] M. Nabeshima, "Performance Evaluation of a Combined Input- and Crosspoint-Queued Switch," *IEICE Trans. Commun.*, vol. E83-B, No. 3, March 2000.

[5] K. Yoshigoe, K.J. Christensen, "A parallel-polled Virtual Output Queue with a Buffered Crossbar," *IEEE HPSR 2001*, pp. 271-275, May 2001.

[6] R. Rojas-Cessa, E. Oki, Z. Jing, and H. J. Chao, "CIXB-1: Combined Input-One-cell-crosspoint Buffered Switch," *IEEE HPSR 2001*, pp. 324-329, May 2001.

[7] T. Javadi, R. Magill, and T. Hrabik, "A High-Throughput Algorithm for Buffered Crossbar Switch Fabric," *IEEE ICC 2001*, pp.1581-1591, June 2001.

[8] R. Rojas-Cessa, E. Oki, and H. J. Chao, "CIXOB-1: Combined Input-crosspoint-output Buffered Packet Switch," *IEEE GLOBECOM 2001*, vol. 4, pp. 2654-2660, November 2001.

[9] L. Mhamdi, M. Hamdi, "Practical Scheduling Algorithms For High-Performance Packet Switches," *IEEE ICC 2003*, pp. 1659-1663, vol. 3, May 2003.

[10] N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand, "Achieving 100% Throughput in an Input-queued Switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260-1267, August 1999.

[11] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California at Berkeley, Berkeley, CA, 1995.