Round-robin with Adaptable-Size Frame Arbitration for Input-Crosspoint Buffered Switches

Roberto Rojas-Cessa

***Abstract*— Combined input-crosspoint buffered switches relax arbitration timing and provide high-performance switching for packet switches with high-speed ports. It has been shown that these switches, with one-cell crosspoint buffer and round-robin arbitration at input and output ports, provide 100% throughput under uniform traffic. However, under admissible traffic patterns with nonuniform distributions, only weight-based selection schemes are reported to provide high throughput. This paper proposes a round-robin based arbitration scheme for a combined input-crosspoint buffered packet switch. The presented scheme uses adaptable-size frames, where the frame size is determined by the received service. The resulting switch provides nearly 100% throughput for several admissible traffic patterns, including uniform and unbalanced traffic, using one-cell crosspoint buffers.**

***Index Terms*— Crosspoint-buffered switch, packet scheduling arbitration, virtual output queue, credit-based flow control, adaptable-size frame**

I. Introduction

Combined input-crosspoint buffered packet switches are an alternative to input-buffered switches to relax arbitration timing and to provide high-performance switching for packet switches with high-speed ports. These switches use time efficiently as input and output port selections are performed independently. As an example of the stringent timing, a switch with 40-Gbps (OC-768) ports transferring 64-byte cells 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 buffer size or large N. One way to keep the buffer complexity feasible is to use crosspoint buffers that are small in size.

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 re-assembling 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.

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.

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 in 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, as 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 nearly 100% throughput under nonuniform traffic patterns with a buffered crossbar that have crosspoint buffers of small size. 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.

A question arises: is it possible to provide an arbitration scheme based on round-robin selection for buffered crossbars such that a switch can deliver high throughput under admissible traffic with nonuniform distributions, such as unbalanced traffic, with a small crosspoint buffer size?

This paper proposes an arbitration scheme for buffered crossbars, based on round-robin selection, that uses the concept of adaptable-size frame. The frame size is called adaptable as it is determined by the amount of service that a queue receives. This paper shows that this arbitration scheme can achieve nearly 100% throughput under the unbalanced traffic pattern with one-cell crosspoint buffers. It also shows that this switch retains the high performance of simple round-robin arbitration under uniform traffic.

This paper is organized as follows. Section II presents the switch model under study. Section III introduces the proposed arbitration scheme. Section IV presents a simulation study of the throughput and delay performance of the resulting switch under uniform and nonuniform traffic patterns. Section V discusses the properties of the proposed arbitration scheme. Section VI presents the conclusions.

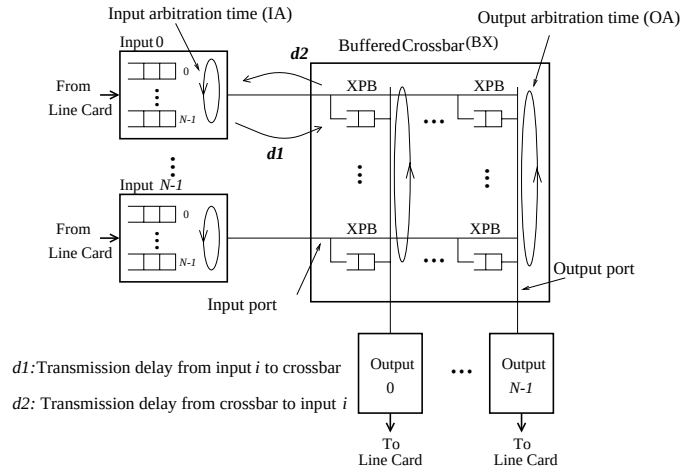

Fig. 1. $N \times N$ buffered crossbar with VOQ structure

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 selection if the VOQ is not empty and the corresponding $XPB(i,j)$, at BX, has room to store a cell.

This switch uses the round-trip time as defined in [6], and the crosspoint buffer size complies with Eq. 1 in [6]. Let us denote $\lambda(i,j)$ to the cell arrival rate at input i for output j that is received in $VOQ(i,j)$. As in [10], let us consider traffic admissible such that $\sum_i \lambda(i,j) < 1$, and $\sum_j \lambda(i,j) < 1$.

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

The proposed arbitration scheme is round-robin based. Each time 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. Each cell of a frame is dispatched in one time slot. The frame size is determined by the serviced and unserviced traffic, such that no intervention is needed to select the frame size. We call this arbitration round-robin with adaptable-size frame (RR-AF). The amount of serviced (and unserviced) traffic depends on the experienced load by 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).[2] $CSC_{i,j}(t)$ counts the number of serviced cells at time slot t in a frame corresponding to a VOQ, where the frame size is indicated by FSC, in a regressive fashion.[3] 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 exist 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.

For the sake of clarity, the following pseudo-code describes the input arbitration scheme, as seen at an input:

-At time slot t, starting from the pointer position j, find the nearest eligible $VOQ(i,j')$ in a round-robin fashion.

-Send the HOL cell from $VOQ(i,j')$ to $XPB(i,j')$ time slot $t+1$.

-If $|CSC_{i,j'}(t)| > 1$ then
$|CSC_{i,j'}(t+1)| = |CSC_{i,j'}(t)| - 1$,
the pointer points to j'.

[2]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.

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

-else $|FSC_{i,j'}(t+1)| = |FSC_{i,j'}(t)|$+g,
$|CSC_{i,j'}(t+1)| = |FSC_{i,j'}(t+1)|$,
the pointer points to (j'+1) module N.
-For VOQ(i, h), where $j \leq h < j'$ *for* $j < j'$*, or* $0 \leq h < j'$ *and* $j \leq h \leq N-1$ *for* $j > j'$*:*
$FSC_{i,h}(t+1) = FSC_{i,h}(t) - 1$.[4]
- Go to the next 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 justified in Section IV. 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.

The output arbitration works in a similar way to that at the inputs, considering $XPB(i,j)$ and its corresponding counters instead. Figure 2 shows an example of round-robin with adaptable-size frame at an input. Assume that the queues shown in the figure are the VOQs of input i in a 3×3 switch. Also assume that these VOQs are the only nonempty VOQs in this switch. Initially, all queues have three cells each, as Figure 2.a shows. Assuming that the FSC for each queue has the initial value of one, a cell from each queue is served in a round-robin fashion. Then, each frame is increased by N cells; therefore, the remaining two cells in each queue are served back-to-back. The cells leave the input as Figure 2.b shows.

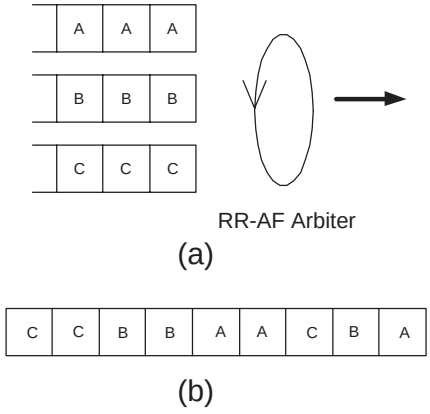

Fig. 2. Example of RR-AF among three queues

IV. Performance Evaluation

In this section, the performance evaluations of two combined input-crosspoint buffered switches, one using RR-AF arbitration and the other using RR arbitration, are presented. In addition, an output buffered (OB) switch is considered in our evaluations. 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 re-assembly delays. Simulation results are obtained with a 95% confidence interval, not greater than 5% for the average cell delay.

[4]Note that when $j' = j$, there is no $VOQ(i,h)$.

A. Uniform Traffic

Figure 3 shows simulation results of 32×32 input-crosspoint buffered switches with RR-AF, RR, 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 $l = 100$). 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 arbitration scheme provides 100% throughput under uniform traffic.

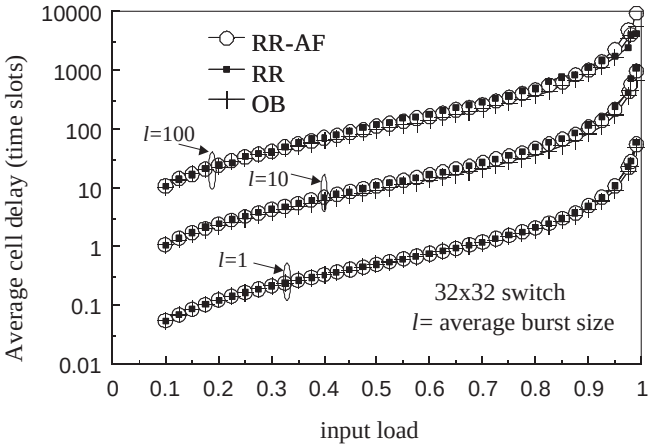

Fig. 3. Average delay of RR-AF arbitration under Bernoulli and bursty uniform traffic

This figure also shows that the average delay performance of RR-AF under Bernoulli arrivals is close to that of RR, and therefore, to that of an OB switch. The adaptable frame-size condition in the 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 the queues (i.e., VOQ and XPB), the switch practically adapts itself to uniform traffic. In addition, Figure 3 shows that RR-AF arbitration offers a similar performance to that of an OB switch under bursty traffic. The average delay is then proportional to the burst length and the throughput is unaffected.

RR-AF was simulated with different sizes of crosspoint buffer, k. The result of the simulation showed that there is no measurable improvement by increasing the size of k. This result is expected as the average delay of RR-AF with $k = 1$ is close to that of an OB switch. Therefore, the increasing of k negligibly affects the results. As in [6], the size of k needs to be determined by the round-trip transmission delay. As the k size does not affect the performance of RR-AF, k is assigned the value of one cell, (i.e., $k = 1$), in the rest of this paper, unless otherwise stated.

Another important point is to observe how the increment of the frame size, g, affects the switch performance under uniform traffic. The value of g has been assumed to be N until this point. With RR arbitration, i.e., g=0, switches deliver high throughput and an average cell delay independently of the

switch size under uniform traffic [6]. It is interesting to see if this property holds for RR-AF. Figure 4 shows the average delay of RR-AF under different switch sizes for g=1 and g=N. The values of the average cell delay for all switches show no difference for input loads below 0.8, so those values are not shown. This figure shows that for small switch sizes (e.g.,

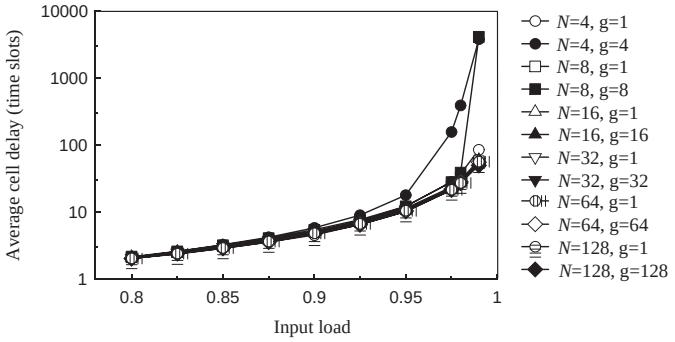

Fig. 4. Average delay of RR-AF in function of switch size, under Bernoulli uniform traffic

$N = \{4, 8\}$), it is more efficient to have a small g value, (e.g., g=1). As the switch size increases, it is more efficient to use large g values (e.g., g=N).

B. Nonuniform Traffic

RR-AF 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,

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

When $w = 0$, the offered traffic is uniform. On the other hand, when $w = 1$, it is completely 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 $s = d$.

Two combined input-crosspoint buffered switches of size $N = 32$, one with RR-AF and the other with RR, were simulated under unbalanced traffic. The switch with RR-AF uses $k = 1$ and for comparison, RR uses $k = 1$ and $k = N = 32$. Figure 5 shows that RR-AF, with $k = 1$ and g=N, provides well above 99% throughput under the complete range of w. It is considered that this throughput is nearly 100% for practical purposes. These results show that RR-AF with $k = 1$ outperforms RR with $k = 32$. This 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, with $k = 1$, would need 512Kb of memory. Furthermore, the switch with RR-AF can provide nearly 100% throughput under unbalanced traffic.

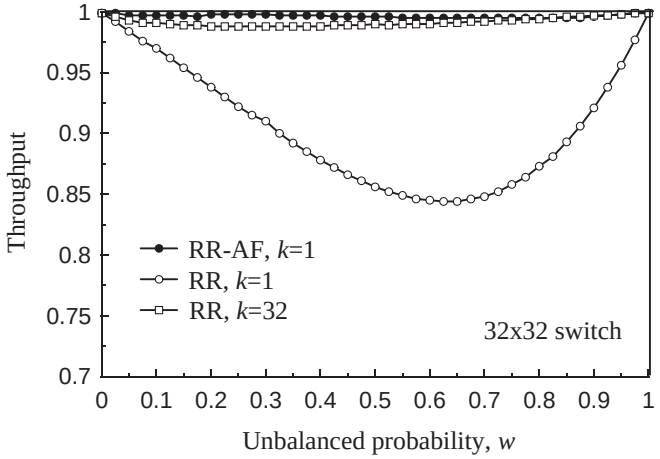

Fig. 5. Throughput performance of RR-AF under unbalanced traffic

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 load 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.

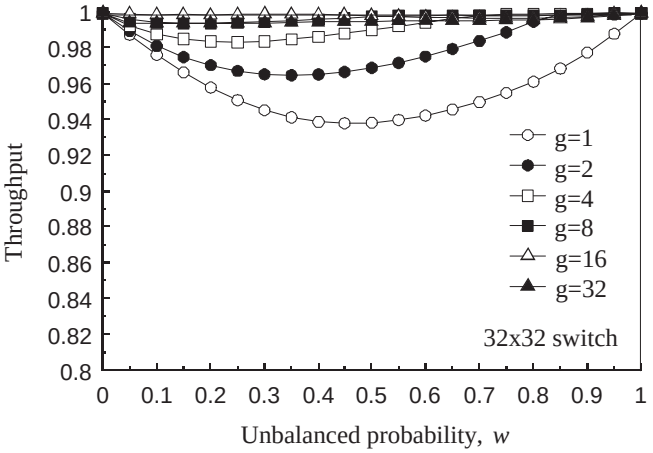

Fig. 6. Throughput performance of a 32x32 switch for different g values

Figure 6 shows a 32×32 switch with RR-AF under unbalanced traffic. This graph shows optimal values of g to achieve a high throughput. When g=1, the switch does not reach 99% throughput. The values of g to achieve over 99% throughput are g$\geq$ 8 in this switch. The throughput is the nearest to 100% when g= $N/2 = 16$. Note that the lower throughput value along the w range is the one considered. Therefore, although the graph shows some small differences in the measured throughput for some values of w with different values for g, it is considered that when g$\geq$ 8 the throughput performance is similar for a 32×32 switch.

To illustrate the dependency of N, Figure 7 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

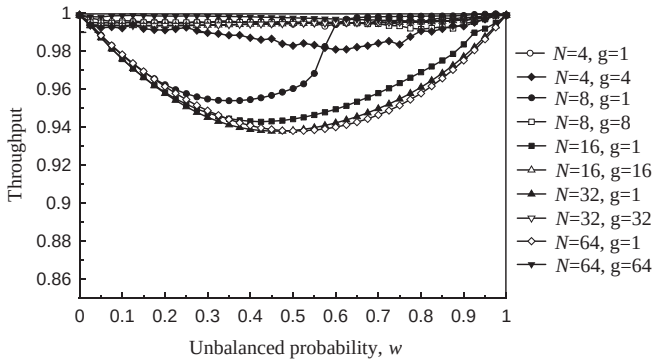

Fig. 7. Throughput performance of RR-AF for different switch sizes under unbalanced traffic

with g=1 resembles RR. Therefore, the throughput is generally low for medium-to-large switch sizes under this traffic type. 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=$\{N/2, N\}$).

V. Properties of RR-AF

Under uniform traffic, the frame counters of the queues 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 a 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 implementation complexity of RR-AF is low for 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 performs 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 compared to the implementation of RR. The FSC, CSC counters, and arbiter pointers are updated at most once in a time slot. Therefore, the timing for RR-AF is no different from the timing in RR.

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

VI. Conclusions

This paper introduced a novel arbitration scheme for input-crosspoint buffered crossbars based in round-robin selection. This scheme uses the concept of adaptable-size frame, where the frame size depends on the service received by a queue. The presented round-robin scheme with adaptable-size frame shows 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. This arbitration scheme does not need to compare the status of different queues, such as weights or priorities, as it is based in simple round-robin. In addition to high throughput, this switch provides timing relaxation that allows high-speed arbitration and scalability. This results in a simplified and scalable arbitration scheme.

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.