PAPER

Round-Robin Selection with Adaptable Frame-Size for Combined Input-Crosspoint Buffered Packet Switches

Roberto ROJAS-CESSA[†], ***Member and*** **Zhen GUO**[†], ***Student Member***

SUMMARY Combined input-crosspoint buffered (CICB) 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 CICB packet switch that provides 100% throughput for several admissible traffic patterns, including those with uniform and nonuniform distributions, using one-cell crosspoint buffers and no speedup. The presented scheme uses adaptable-size frames, where the frame size is determined by the traffic load.

key words: *buffered crossbar, stability, adaptable frame, admissible traffic, crosspoint buffer*

1. Introduction

The deployment of higher-speed interconnection technologies and the advances in digital compression techniques are resulting in an increased volume of traffic on the Internet. This growth motivates the search for high-capacity and high-speed switches.

The performance of a switch can be analyzed according to the adopted buffering strategy. A switch with buffers* at the inputs is named input-buffered (IB) switch. In an IB switch, the input buffers store packets that cannot be forwarded to the outputs because of output contention. An IB switch has better scalability that an output-buffered (OB) switch as the switch fabric and input buffers in an IB switch work at the same speed as the external lines (no speedup), while an OB switch needs to speedup the switch and buffers N times, where N is the number of input and output ports. However, IB switches need to resolve input and output contention before cells are forwarded to the outputs. Arbiters at input and outputs perform contention resolution by means of a parallel matching process. Furthermore, the switching performance of an IB switch requires complex matching schemes to provide high-switching performance. This high complexity limits the switch port speeds. The requirements for arbiters to be feasible and to provide a high performance are: (a) low complexity, (b) fast contention resolution, (c) fairness, and, (d) high matching efficiency. As an example, the matching scheme must perform input or output arbitration within 6.4 ns in an IB switch with 40 Gbps (OC-768) ports and 64-byte cells, assuming that input and output arbitrations may use up to half of a time slot and that the transmission delays are decreased to negligible amounts (e.g., the arbiters are implemented in the same chip, in a centralized way).

Crosspoint buffered (CB) packet switches are an alternative to IB switches to relax arbitration timing and to provide high-performance switching for packet switches with high-speed ports. The arbitration in a CB switch is only performed for input selection at each output of the buffered crossbar, where packets stored in the crosspoint buffers are considered. However, the number of buffers in a crossbar grows in the same order as the number of crosspoints, $O(N^2)$. 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.

An example of a CB switch was proposed in [1], where a 2×2 crossbar chip with a crosspoint memory of 16 Kbytes was implemented to provide an acceptable cell loss. In addition to the crosspoint buffers, placement of input buffers can be used to reduce the memory amount at the crosspoints. A variety of combined input-crosspoint buffered switch (CICB) switches were presented in [2]-[6]. CICB switches with a single-cell buffer were proposed in [2], [3]. These switches used first-in first output (FIFO) input buffers at the input ports, or FIFO-CICB switches. The switches provide a throughput of 91%, where the head-of-line (HOL) blocking [5] was still present. The FIFO buffers at the inputs limit the maximum throughput in that switch. As in IB switches, the HOL blocking problem for FIFO buffers can be overcome in CICB switches by using virtual output queues (VOQs), or VOQ-CICB switches. For the sake of brevity, we refer to VOQ-CICB switches as CICB switches in the remainder of this paper.

CICB switches use time efficiently as input and output port selections are performed separately. Back to the example of the stringent timing, a CICB switch with 40-Gbps and 64-byte packets can perform input (or output) arbitration within 12.8 ns, therefore, the timing for arbitration is extended. 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 VOQs to avoid HOL blocking. 3) Use of crossbar fabrics for implementation of packet

Manuscript received February 18, 2005.
Manuscript revised September 06, 2005.
Final manuscript received 00, 2005.

[†]New Jersey Institute of Technology, University Heights, Newark, NJ 07102, Email: rrojas@njit.edu.

*This paper uses the terms queue and buffer interchangeably.

switches because of their non-blocking capability, simplicity, and market availability. This paper follows these practices.

In CICB switches, high matching efficiency is achieved with simpler arbitration schemes than those used in bufferless crossbars (i.e., IB switches) at the expense of having to accommodate buffers in the crosspoints. These features have been shown to be attractive in several switches [6]-[15].

A CICB 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 [7]. Furthermore, it has been shown that a CICB switch using one-cell crosspoint buffers, a simple round-robin arbitration (RR) scheme for input and output arbitration, and a credit-based flow control provide 100% throughput for uniform traffic [10]. 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 as: $\sum_i \lambda_{i,j} \leq 1$, and $\sum_j \lambda_{i,j} \leq 1$, where $\lambda_{i,j}$ is the cell arrival rate at input i for output j.

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. It has been shown that weight-based [13] and priority-based [14] schemes in buffered crossbars can provide high throughput under various traffic patterns. Two schemes were presented in [13]: 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, thus increasing the implementation complexity. Moreover, weight-based schemes (e.g., queue-occupancy based) may starve some queues for very long time to provide more service to the congested ones, presenting unfairness. On the other hand, RR algorithms have been shown to provide fairness and implementation simplicity, as no comparisons are needed among queues, and high-performance under uniform traffic [16]. 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. For example, it has been shown that a switch using RR needs a large crosspoint buffer to provide high throughput under admissible unbalanced traffic [17], where the unbalanced traffic model is a nonuniform traffic pattern [10]. 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?

Frame-based matching have been shown to have improved switching performance under different traffic scenarios [18]. However, how to set the frame size is a complex issue. 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 and by the arrival traffic load. This paper shows that this arbitration scheme can achieve nearly 100% throughput under several nonuniform traffic patterns with one-cell crosspoint buffers. This paper also proves that this switch retains the high performance, 100% throughput, of simple round-robin arbitration under uniform traffic.

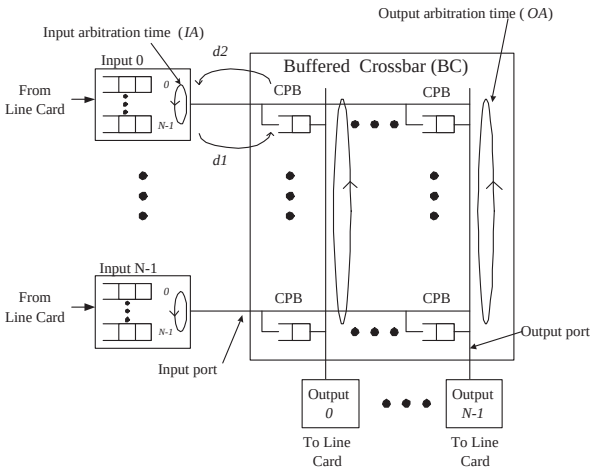

Fig. 1 $N \times N$ buffered crossbar with VOQs.

This paper is organized as follows. Section 2 presents the switch model under study. Section 3 introduces the proposed arbitration scheme. Section 4 presents a stability analysis of the proposed arbitration scheme. Section 5 presents a simulation study of the throughput and delay performance of the resulting switch under uniform and nonuniform traffic patterns. Section 6 discusses the properties of the proposed arbitration scheme. Section 7 presents the conclusions.

2. Combined Input-Crosspoint Buffered Switch Model

Figure 1 shows a buffered crossbar (BC) switch with N inputs and outputs. In this switch model, there are N VOQs at each input. A VOQ at input i, where $0 \leq i \leq N-1$, that stores cells for output j, where $0 \leq j \leq N-1$, is denoted as $VOQ_{i,j}$. A crosspoint (CP) element in the BC that connects input port i to output port j is denoted as $CP_{i,j}$. The buffer at $CP_{i,j}$ is denoted as $CPB_{i,j}$. The size of $CPB_{i,j}$, k, is indicated by the number of cells that can be stored. A credit-based flow-control mechanism indicates to input i whether $CPB_{i,j}$ has room available for a cell or not, as described in [10]. For this flow-control mechanism, there is a credit counter in each VOQ that counts the number of outstanding cells (i.e., cells sent to CPB). The credit counter increases by one each time a cell is sent to the CPB. When a cell is forwarded from the CPB to the output, the crossbar sends a release bit

to the credit counter, and the credit counter is decreased by one. To avoid overflow, once the credit counter reaches the value of k, then the VOQ is inhibited of sending a cell to the CPB. $VOQ_{i,j}$ is said to be eligible for selection if the VOQ is not empty and the corresponding $CPB_{i,j}$, at BC, has room to store a cell.

The round trip (RT) time, as in [10], 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 ($d2$). Figure 1 shows an example of RT for input 0 by showing the transmission delays for $d1$ and $d2$, and arbitration times, IA and OA. Cell and bit alignments are included in the transmission times. The condition for this switch to avoid underflow, 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.

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

The proposed arbitration scheme is round-robin based. Each time a VOQ (or a CPB 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 CPB), 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 $VOQ_{i,j}$ 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).[†] $CSC_{i,j}(t)$ 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.[††] 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

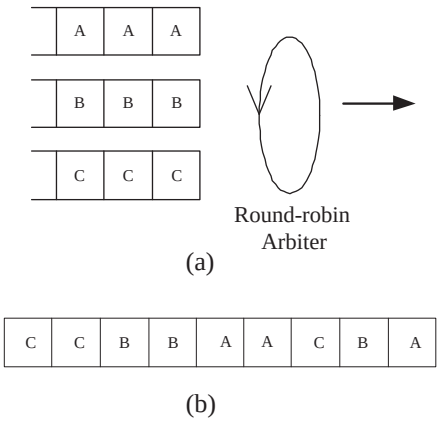

Fig. 2 Example of RR-AF among three queues.

$VOQ_{i,j'}$, if $|CSC_{i,j'}(t)| > 1$, $|CSC_{i,j'}(t+1)| = |CSC_{i,j'}(t)| - 1$, 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 $|CSC_{i,j'}(t)| = 1$, the input pointer is updated to $(j'+1) \mod N$, $|FSC_{i,j'}(t)|$ is increased by f cells, and $|CSC_{i,j'}(t)| = |FSC_{i,j'}(t)|$. 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 $VOQ_{i,j}$ and the selected $VOQ_{i,j'}$: if $|FSC_{i,h}(t)| > 1$, $|FSC_{i,h}(t+1)| = |FSC_{i,h}(t)| - 1$. 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. The increment of the frame size, done by f cells, is performed each time the previous complete frame of a VOQ has been serviced. The value of f has to be chosen as discussed in the following section.

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 $CPB_{i,j'}$ 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'.
-else $|FSC_{i,j'}(t+1)| = |FSC_{i,j'}(t)| + f$,
$|CSC_{i,j'}(t+1)| = |FSC_{i,j'}(t+1)|$,
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 $j > j'$:
$FSC_{i,h}(t+1) = FSC_{i,h}(t) - 1$.[†††]
- Go to the next time slot.

Note that f may be equal to a constant or a variable value. In general, f assumes the finite value of N, unless otherwise stated. This assumption is justified in Section 5. The value of f affects the performance of RR-AF in different traffic scenarios. Note that when $f = 0$, RR-AF becomes RR.

The output arbitration works in a similar way to the input arbitration, considering $CPB_{i,j}$ and the corresponding counters in each crosspoint. Figure 2 shows an example of

[†]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.

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

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

RR-AF at an input. Assume that the queues shown in the figure are the VOQs of input i in a 3×3 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.

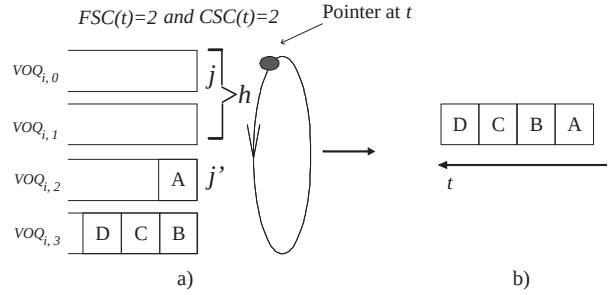

Fig. 3 Example of VOQs missing opportunities for cell forwarding.

Figure 3 shows an example of the adjustment of $FSC_{i,j}$ in an input of a 4×4 switch. In this example, $VOQ_{i,2}$ and $VOQ_{i,3}$ have cells (as Figure 3.a shows), one and three, respectively, and no VOQ is inhibited by the flow-control mechanism. At time slot t, the pointer of RR-AF points to $VOQ_{i,0}$. During this time slot, the input arbiter selects $VOQ_{i,2}$ to send a cell to the buffered crossbar. Then $VOQ_{i,0}$ and $VOQ_{i,1}$ miss an opportunity to send cells as they are empty and their FSCs are decreased by one at the end of the time slot. Note that $VOQ_{i,0}$ and $VOQ_{i,1}$ are considered $VOQ_{i,h}$ for this time slot as defined in the description of RR-AF. Table 3 shows the evolution of the FSC values for each VOQ during 6 time slots. In the next time slot, $t+1$, $VOQ_{i,2}$ is served, and it becomes empty. As the pointer points to this VOQ, $FSC_{i,2}$ is decreased to 1 in the next time slot. Therefore, the arbiter selects $VOQ_{i,3}$ at time slot $t+2$ as the next VOQ to receive service. Then, the pointer is moved to $VOQ_{i,3}$. At time slot $t+3$, $VOQ_{i,3}$ is again selected. Since the last frame cell of $VOQ_{i,3}$ is selected, $FSC_{i,3}$ is updated to $2+N = 2+4 = 6$. However, since there are no more cells in this VOQ, $FSC_{i,3}$ decreases by one in the subsequent time slots. In this table, a dash in time slots $t+4$ and $t+5$ means that no j is selected. Figure 3.b shows the order in which cells are served.

	Time slot					
FSC	t	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$
$FSC_{i,0}$	2	1	1	1	1	1
$FSC_{i,1}$	2	1	1	1	1	1
$FSC_{i,2}$	2	2	1	1	1	1
$FSC_{i,3}$	2	2	2	6	5	4
Selected j	2	3	3	3	-	-

Table 1 Evolution of FSC of example in Figure 3.

4. Stability Study

RR-AF arbitration is based on round-robin and it aims to improve the throughput under non-uniform traffic. The attractiveness of RR-AF lies on keeping the property of round-robin based schemes to deliver stability, and therefore, 100% throughput under uniform traffic. We define the stability of a switch as having the occupancy of $VOQ_{i,j}$ finite as time increases.

In this section, we prove that RR-AF, with f in a general sense, provides 100% throughput under admissible traffic, despite the use of the adaptable-size frame concept. We focus this proof on the input arbitration and VOQs. However, the results apply to the output arbitration and crosspoint buffers.
In our analysis, we use the following definitions.

Definition 1: A cycle is a service opportunity given to a VOQ where the number of cells that can be sent in consecutive time slots to the crosspoint can be up to the frame size. The cycle length is given in the number of time slots that the VOQ receives service. The start of a cycle is determined when a VOQ is selected to receive service at time slot t if that VOQ received service at time $t-1$.

Definition 2: The completion service rate $R^c_{i,j}$ is the rate at which $VOQ_{i,j}$ finishes frame service per cycle.

Definition 3: The miss service rate $R^m_{i,j}$ is the rate at what $VOQ_{i,j}$ misses service per cycle, including the following two reasons of the service miss: *i)* when the number of cells in a VOQ is smaller than the frame size, and *ii)* when a VOQ cannot send cells to the crosspoint for lacking of room in the crosspoint buffer. Therefore, $R^m_{i,j} = 1 - R^c_{i,j}$.

In addition, we use the following notations:

- $T_{i,j}$ denotes the accumulative total number of time slots that $VOQ_{i,j}$ receives service from t_0 to any time t, where t_0 is the starting time and t is any time slot such that $t > t_0$.
- $\sigma_{i,j}$ is the cumulative number of opportunities a VOQ receives for service from cycle n_0, the time before VOQ receives any service during the switch working time, to cycle n.
- $C^{inc}_{i,j}$ is the cumulative number of cycles where FSC increases until cycle n.
- $C^{min}_{i,j}$ is the cumulative number of cycles where FSC has no changed because it has reached the minimum of one cell.

In this section, for the sake of clarity, we denote the value of FSC at the end of cycle n as $FSC_{i,j}(n)$. Note that $FSC(n)$ is different from $FSC(t)$ in Section 3, where the first refers to a serving cycle and the second to a time slot, respectively.

In addition, let $E[FSC_{i,j}(n)]$ denote the expected frame size of any VOQ at the end of n^{th} cycle. Since the average arrival rate is $\lambda_{i,j}$, let's $\lambda_{i,j}E(x)$ be the number of cell arrivals

per cycle (based on Little's theorem), where $E(x)$ is the average number of time slots that a VOQ receives service. Also, we denote the occupancy of a VOQ at the end of cycle n as $L_{i,j}(n)$.

Under traffic with uniform distribution among all outputs, the stability of the switch is directly related to the stability of of the frame size of each queue. The stability of RR-AF is then based in the proof of the following claim:

Theorem 1: A CICB switch using RR-AF scheduling algorithm is stable under traffic with uniform distribution.

Proof 1: We assume that all inputs receive traffic independently and identically distributed. Therefore, identical service is expected in each VOQ.

Since the service that a VOQ (or CPB) receives is determined by FSC, then we define the following lemma.

Lemma 1: In a CICB packet switch using RR-AF as input arbitration, $VOQ_{i,j}$ is stable if $FSC_{i,j}$ is stable, under uniform traffic.

Proof 2: When $FSC_{ij}(n)$ is stable, $L_{i,j}(n)$ can be either cases:

(*i*) $\lim_{n\to\infty} L_{i,j}(n) = \infty$.
(*ii*) $\lim_{n\to\infty} L_{i,j}(n) = a$, where a is a finite value and $a \geq 1$.

Let's consider the case (*i*) first: in a cycle, the service to $VOQ_{i,j}$ always complete because $\lim_{n\to\infty} L_{i,j}(n) = \infty$. $FSC_{i,j}(n)$ will be increased by f each time. Therefore $FSC_{i,j}(n)$ can not be bounded by a finite value, which contradicts with the assumption that $FSC_{ij}(n)$ is stable.

Now let's consider the case (*ii*). $\lim_{n\to\infty} L_{i,j}(n) = a$ means $VOQ_{i,j}$ receives service all the time and $L_{i,j}(n)$ will never go to infinity. Since we have already proved that case (*i*) is impossible, so only case (*ii*) stands.

Summing up the cases above, if $FSC(n)$ is stable, $L(n)$ is stable. This claim is established as a sufficient condition.

□

For completeness, we state the following corollary:

Corollary 1: Under uniform traffic, if $FSC_{i,j}(n)$ is unstable, then $L_{i,j}(n)$ is unstable.

Proof 3: We prove that the following state is false: if $lim_{n\to\infty} FSC_{i,j}(n) = \infty$ then $L_{i,j}(n)$ is stable. Let's assume that the statement is true. There must be that $L_{i,j}(n)$ is bounded by a finite value b (i.e., $\lim_{n\to\infty} L(n) = b$) and therefore $FSC_{i,j}(n)$ increases its value by f each cycle until it reaches the value of b. At this point $FSC(n)$ cannot continue increasing its value at each cycle, and therefore $FSC_{i,j}(n)$ converges to a finite value b, which contradicts the initial assumption. Therefore, if $FSC_{i,j}(n)$ is unstable, and $L_{i,j}(n)$ cannot be stable.

□

To continue with the proof of Theorem 1, it remains to prove that $FSC_{i,j}(n)$ is stable. For this, let's consider the behavior of $FSC_{i,j}(n)$, and by stating the following lemma:

Lemma 2: A CICB switch using RR-AF and under traffic with uniform distribution has $R^m_{i,j} > \frac{f}{f+1}$.

Proof 4: The accumulated FSC value from cycles n_0 to n, where $n > n_0$, is

$$FSC_{i,j}(n) = FSC_{i,j}(0) + fC^{inc} - (\sigma_{i,j} - C^{inc}_{i,j} - C^{min}_{i,j}), \quad (2)$$

where $FSC_{i,j}(0)$ is the initial FSC value at n_0.

Let's assume that a frame is completely served at this cycle. The inequality involving the stationary state follows:

$$FSC_{i,j}(0) + fC^{inc}_{i,j} - (\sigma_{i,j} - C^{inc}_{i,j} - C^{min}_{i,j}) \leq \lambda_{ij}E(x) + \delta_{i,j}, \quad (3)$$

where $\delta_{i,j}$ is the discrepancy between the actual and the expected values. Then, we can express C^{inc} as:

$$C^{inc}_{i,j} \leq \frac{\lambda_{ij}\frac{T_{i,j}}{\sigma_{i,j}} + \sigma_{i,j} + \delta_{i,j} - C^{min}_{i,j} - FSC_{i,j}(0)}{f+1}. \quad (4)$$

Recalling that $R^c_{i,j} = \frac{C^{inc}_{i,j}}{\sigma_{i,j}}$ and using (4), we have:

$$\frac{C^{inc}_{i,j}}{\sigma_{i,j}} \leq \frac{1}{f+1} + \frac{\lambda_{ij}\frac{T_{i,j}}{\sigma_{i,j}} + \delta_{i,j} - C^{min}_{i,j} - FSC_{i,j}(0)}{\sigma_{i,j}(f+1)}. \quad (5)$$

Let's consider that the switch has been functioning for a very long period of time, such that $\sigma_{i,j}$ has a very large value. Therefore, we have:

$$R^c_{i,j} \leq \frac{1}{f+1}, \quad (6)$$

or

$$R^m_{i,j} > \frac{f}{f+1}. \quad (7)$$

□

Now, with Lemma 2 proved, the dynamics of FSC are used to define the value of the frame size at time cycle $n+1$, $FSC_{i,j}(n+1)$, as:

$$E[FSC_{i,j}(n+1)] = (FSC_{i,j}(n) + f)(1 - R^m_{i,j}) + (FSC_{i,j}(n) - 1)R^m_{i,j}, \quad (8)$$

where $E[FSC_{i,j}(n+1)]$ is the expected value of FSC at cycle $n+1$. This equation considers an increment and a decrement of the FSC with probabilities $1 - R^m_{i,j}$ and $R^m_{i,j}$, respectively, at time slot n.

Considering that $FSC_{i,j}(n+2) = FSC_{i,j}(n+1+1)$:

$$E[FSC_{i,j}(n+2)] - E[FSC_{i,j}(n+1)] = f - R^m_{i,j}(f+1). \quad (9)$$

According to the definition of stability in the sense of Lyapunov [19], if $E[FSC_{i,j}(n+l+1)] - E[FSC_{i,j}(n+l)] = -\varepsilon < 0$, then FSC is stable.

Recalling R_m from Lemma 2, and substituting $R_m = \frac{f+\mu}{f+1}$ in (7), where $0 < \mu < 1$, it is clear that:

$$R_m = \frac{f+\mu}{f+1} > \frac{f}{f+1}. \quad (10)$$

Considering that $l = 1$ and n can be any service cycle, we substitute (10) in (9):

$$E[FSC_{i,j}(n+2)] - E[FSC_{i,j}(n+1)] = f - (\frac{f+\mu}{f+1})(f+1), \quad (11)$$

which is:

$$E[FSC_{i,j}(n+2)] - E[FSC_{i,j}(n+1)] = -\mu, \quad (12)$$

for any cycle n during steady state. This equality shows the stability of FSC of any VOQ. Therefore, a packet switch using RR-AF arbitration under uniform traffic is stable.

□

5. Performance Evaluation

In this section, the performance evaluations of two CICB switches, one using RR-AF arbitration and the other using RR arbitration, are presented. In addition, an 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.

5.1 Uniform Traffic

Figure 4 shows simulation results of two 32×32 CICB 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.

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 CPB), the switch practically adapts itself to uniform traffic. In addition, Figure 5 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 k. The result of the simulation shows 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 [10], the size of k needs to be determined by the RT time. As the size of k does not affect the performance of RR-AF, k is assigned the value of one cell, (i.e., $k = 1$), in the remainder of the paper, unless otherwise stated.

Another important point is to observe how the increment of the frame size, f, affects the switch performance under uniform traffic. The value of f has been assumed to be N until this point. With RR arbitration, i.e., f=0, switches deliver high throughput and an average cell delay that are independent of the switch size, under uniform traffic [10]. It is interesting to see if this property holds for RR-AF. Figure 5 shows the average delay of RR-AF under different switch sizes for $f = 1$ and $f = N$. The values of the average cell delay for all switches show no difference for input loads below 0.8, therefore, those values are not shown.

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

We also tested RR-AF under overloading conditions to observe fairness among inputs. We simulated an 8×8 switch, where inputs 0 to 6 received an input load of 0.1 and input 7 received an input load of 0.5. All this traffic was cre-

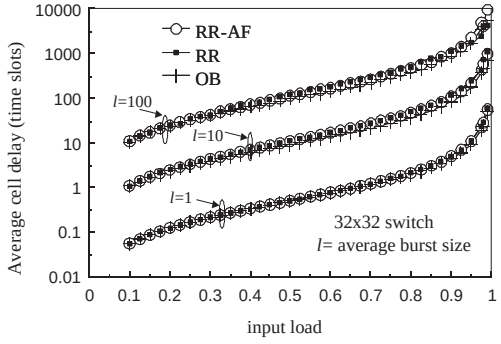

Fig. 4 Average delay of RR-AF arbitration under Bernoulli and bursty uniform traffic.

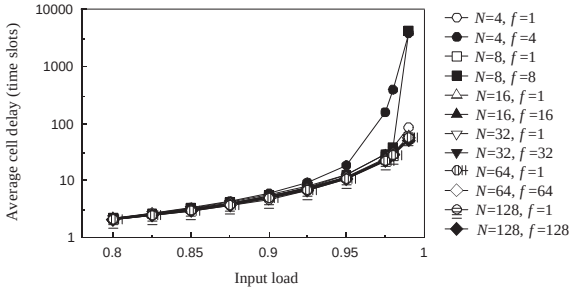

Fig. 5 Average delay of RR-AF in function of the switch size under Bernoulli uniform traffic.

Input	Input Load	Service Load
0	0.1	0.1
1	0.1	0.1
2	0.1	0.1
3	0.1	0.1
4	0.1	0.1
5	0.1	0.1
6	0.1	0.1
7	0.5	0.3

Table 2 Sharing in a overloaded switch using RRAF.

ated with a uniform distribution. In this way, the switch was overloaded. Table 2 shows the simulation results. These results show that inputs 0 to 6 received a service load of 0.1, and input 7 received a service load of 0.3. In this way, RRFA presented a fair distribution of bandwidth among all inputs and provided the available bandwidth to the overloaded input 7 without affecting the service for the others.

5.2 Unbalanced Traffic

RR-AF and RR arbitrations were simulated under a nonuniform traffic model, the unbalanced traffic model [10]. 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,

$$\rho_{s,d} = \begin{cases} \rho\left(w + \frac{1-w}{N}\right) & \text{if } s = d \\ \rho\frac{1-w}{N} & \text{otherwise.} \end{cases} \quad (13)$$

When $w = 0$, the offered traffic is uniform. On the other hand, when $w = 1$, it is completely completely directional, from input s to output 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 6 shows that RR-AF, with $k = 1$ and $f = N$, 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 16 Mb of memory, while RR-AF, with $k = 1$, would need 512 Kb of memory. Furthermore, the switch with RR-AF 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 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.

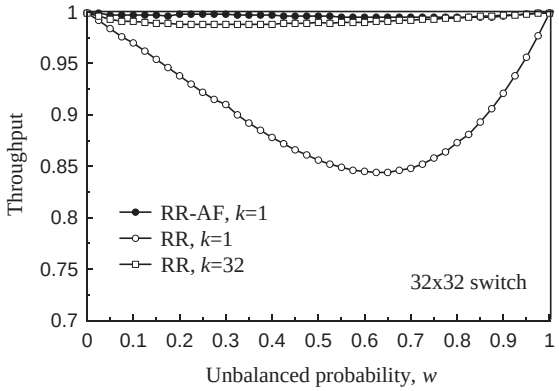

Fig. 6 Throughput performance of RR-AF under unbalanced traffic.

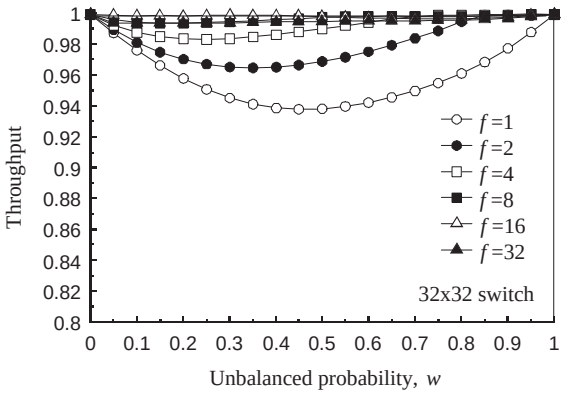

Fig. 7 Throughput performance of a 32x32 switch for different f values.

Figure 7 shows a 32×32 switch with RR-AF under unbalanced traffic. This graph shows optimal values of f to achieve a high throughput. When $f = 1$, the switch does not reach 99% throughput. The values of f to achieve over 99% throughput are $f \geq 8$ in this switch. The throughput is the nearest to 100% when $f = N/2 = 16$. 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 f, it is considered that when $f \geq 8$ the throughput performance is similar for a 32×32 switch.

To illustrate the dependency of N, Figure 8 shows the throughput of RR-AF for different switch sizes, $N = \{4, 8, 16, 32, 64\}$, with $f = 1$ and $f = N$.

As expected, RR-AF with $f = 1$ 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 $f = 1$. However, those switches have nearly 100% throughput when $f = N$. Note that contrary to the case of uniform traffic, an 8×8 switch delivers a low performance when

$f = 1$ under unbalanced traffic.[†] In general, the throughput of RR-AF improves for medium-to-large switches with large f values (e.g., $f = \{N/2, N\}$).

5.3 Chang's and asymmetric traffic models

RR-Af with $f = N$, is also tested under other nonuniform traffic models: Chang's [20] and asymmetric [21].

Chang's traffic model can be defined as $\rho = 0$ for $i = j$ and $\rho = \frac{1}{N-1}$, otherwise. The asymmetric traffic model can be defined as having different load for each input-output pair, such as $\rho_{i,(i+j) \mod N} = \rho a_j$, where $a_0 = 0$, $a_1 = (r-1)/(r^N-1)$, $a_j = a_1 r^{j-1}\ \forall j \neq 0$, and $\rho_{i,j}/\rho_{(i+1) \mod N}$, $j = r$, $\forall i \neq 0$, $(i+1) \mod N \neq 0$, and $r = (100:1)^{-1/(N-2)}$.

Figure 9 shows the average cell delay of a 32×32 switch using RR-AF under and the average cell delay of an OB switch under these traffic models. As the figure shows, the throughput of RR-AF is 100% under Chang's and Asymmetric traffic models. The average delay under Chang's traffic is larger than that of the asymmetric's traffic; however, the difference is small. RR-AF adapts the frame size to the different loads offered to each input and output. RR-AF has an average delay close to that of an OB switch under these traffic models.

To observe the impact of f under different switch sizes, we simulated RR-AF under $N = \{4, 8, 16, 32\}$ and different f values, $f = \{0, ..., 2N\}$. Note that when $f = 0$, the RR-AF becomes equivalent to RR. This value was considered here as it makes the switch deliver lower average delay under Chang's traffic in a 4×4 switch. Figure 10 shows the simulation results with the smallest and largest values of f. These results show that switches with small N deliver higher performance when f is small. As N increases, the switch performance becomes independent of f.

Figure 11 shows the simulation of switches with different N and different f values. Similar to the results experienced under Chang's traffic, RR-AF needs a small f for small switches ($N = \{4, 8\}$) and the performance becomes less sensitive to f for larger switches. In this way, f can be chosen equal to N for a 32×32 switch to provide high switching performance under any of the traffic models considered here.

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

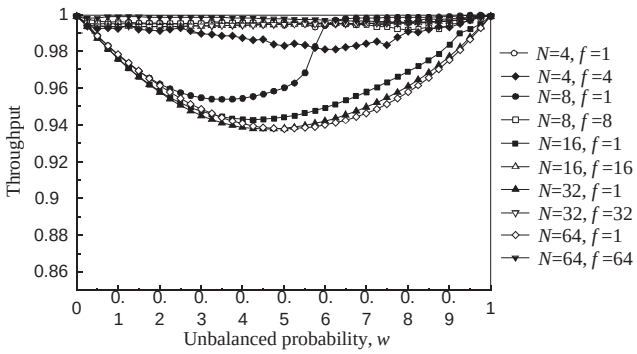

Fig. 8 Throughput performance of RR-AF for different switch sizes under unbalanced traffic.

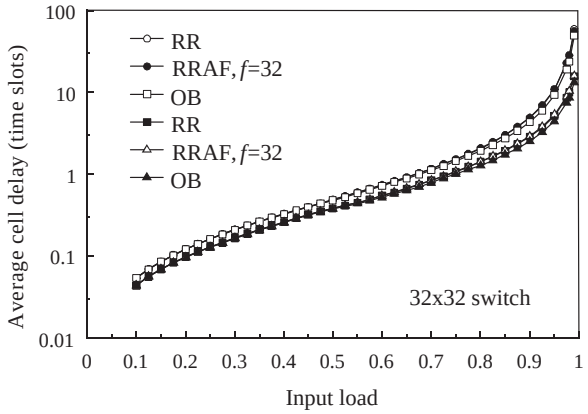

Fig. 9 Average cell delay of RR-AF under Chang's and asymmetric traffic.

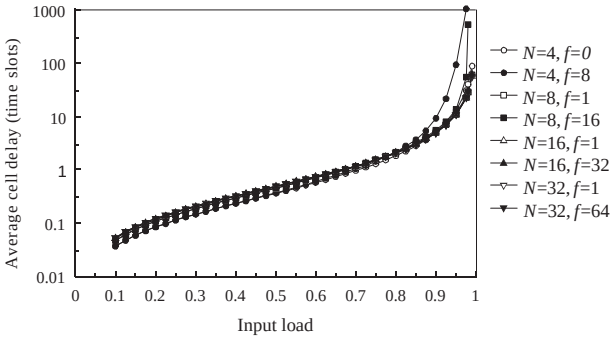

Fig. 10 Average cell delay of RR-AF under Chang's traffic for different N and f.

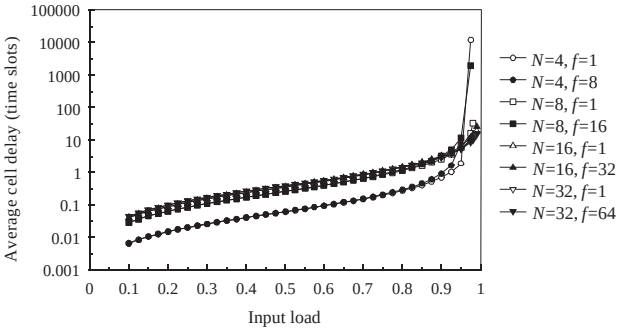

Fig. 11 Average cell delay of RR-AF under asymmetric traffic for different N and f.

6. 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 increasing and decreasing processes are balanced for all queues. This results in an arbitration that behaves as RR.

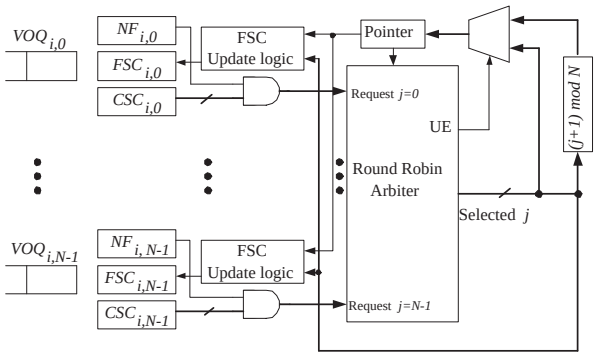

Fig. 12 Implementation of RR-AF as additions to a round-robin arbiter.

Under unbalanced traffic, some queues are expected to have heavier loads than others. The queues with large occupancies have a higher probability of servicing a complete frame in each opportunity of service, and of having their frame size increased, 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.

To give an idea of the frame size in the presented experiments, the frame size was allowed to take maximum values of 64 and 256 (when $N = 32$), and the performance was not affected under any traffic type. In this way, input-output pairs (e.g., $i - j$) that received large amounts of traffic for long period of times do not affect the starting-service latency for newly created connections (i.e., $i-j'$, where $j' \neq j$) when using a limited frame size.

6.1 Complexity

The implementation complexity of RR-AF 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 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. Figure 12 shows the implementation of RR-AF for an input arbiter. An output arbiter would be implemented in similar way, where the crosspoint buffers send a request to the round-robin arbiter instead of the VOQs. A request, represented as the number of outstanding cells in the frame, enters the round-robin arbiter. The request is different from zero if the non-empty flag ($NF_{i,j}$) is one; $NF_{i,j}$ is one when the VOQ is non-empty and $CPB_{i,j}$ has available room for another cell.

The round-robin arbiter performs the selection, considering the pointer value and the existing requests. The arbiter outputs the index of the VOQ (or j) that is selected and asserts the update enable (UE) signal if the CSC counter of the selected VOQ equals one. The pointer is updated with the selected VOQ index as described by the RR-AF scheme. If the UE signal is 1, the pointer updates its value to $(j + 1)$ mod N. Otherwise, the updated pointer value is j. The value of $FSC_{i,j}$ counter is updated by a logic block (i.e., FSC update logic).

The FSC, CSC counters, and arbiter pointers are updated at most once in a time slot. As the figure shows, the FSC update, if this occurs, is processed at the same time as the pointer update takes place. Therefore, the time complexity of RR-AF is equivalent to that of a simple RR.

7. 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.

This paper proved that the round-robin scheme with adaptable-size frame arbitration delivers 100% throughput under uniform traffic. The presented simulation results show that this throughput is achieved with low average cell delay and that the analytical result can be extended to nonuniform traffic patterns, including the unbalanced traffic model.

The results also show that a buffered crossbar with one-cell crosspoint buffers is sufficient to provide such throughput with the proposed round-robin based arbitration. We showed that a 32×32 CICB switch using RR-AF and 512 Kb of memory would deliver a higher performance than a CICB switch using RR and 16 Mb of memory, therefore, reducing the required memory by a factor of N (e.g., 32 in this case).

This arbitration scheme does not need to compare the status of different queues, such as weights or priorities, as it is based on simple round-robin. Furthermore, the effect of the frame increase under different values, for uniform and unbalanced traffic models, was studied with several switch sizes.

In addition to high throughput, this switch provides timing relaxation that allows high-speed arbitration and scalability.

Acknowledgement

This work is supported in part by National Science Foundation under Grant Awards 0435250 and 0423305.

References

[1] S. Nojima, E. Tsutsui, H. Fukuda, and M. Hashimmoto, "Integrated Packet Network Using Bus Matrix," *IEEE J. Select. Areas Commun.*, vol. SAC-5, no. 8, pp. 1284-1291, October 1987.

[2] A. K. Gupta, L. O. Barbosa, and N. D. Georganas, "16 x 16 Limited Intermediate Buffer Switch Module for ATM Networks," *GLOBECOM '91*, pp. 939-943, December 1991.

[3] A. K. Gupta, L. O. Barbosa, and N. D. Georganas, "Limited Inter-

mediate Buffer Switch Modules and their Interconnection Networks for B-ISDN," *ICC '92*, pp. 1646-1650, June 1992.

[4] 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.

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

[6] 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.

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

[8] F. M. Chiussi and A. Francini, "A Distributed Scheduling Architecture for Scalable Packet Switches," *IEEE J. Select. Areas Commun.*, pp. 2665-2683, December 2000.

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

[10] 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.

[11] F. Abel, C. Minkenberg, R. P. Luijten, M. Gusat, and I. Iliadis, "A four-terabit packet switch supporting long round-trip times," *IEEE Micro*, Vol. 23, Issue 1 , pp. 10-24, Jan.-Feb. 2003.

[12] R. Luijten, C. Minkenberg, and M. Gusat, "Reducing memory size in buffered crossbars with large internal flow control latency," Global Telecommunications Conference, 2003. *IEEE GLOBECOM 2003*, Vol. 7, pp. 3683-3687, Dec. 2003

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

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

[15] N. Chrysos and M. Katevenis, "Weighted fairness in buffered crossbar scheduling, High Performance Switching and Routing," *IEEE HPSR 2003*, pp. 17-22, June 2003.

[16] N. McKeown, "The iSLIP Scheduling Algorithm for Input-queued Switches," *IEEE/ACM Trans. Networking.*, vol. 7, no. 2, pp. 188-201, April 1999.

[17] 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.

[18] A. Bianco, M. Franceschinis, S. Ghisolfi, A. M. Hill, E. Leonardi, F. Neri, and R. Webb, "Frame-based Matching Algorithms for Input-queued Switches," *IEEE HPSR 2002*, pp. 69-76, May 2002.

[19] E. Leonardi, M. Mellia, M. Ajmone Marsan, and F. Neri, "Stability of Maximal Size Matching in Input-Queued Cell Switches," *IEEE ICC*, pp.1758-1763, Vol.3, June 2000.

[20] C-S. Chang, D-S. Lee, and Y-S. Jou, "Load Balanced Birkhoff-von Neumann Switches," IEEE HPSR 2001, pp. 276-280, May 2001.

[21] R. Schoenen, G. Post, and G. Sander, "Weighted Arbitration Algorithms with Priorities for Input-Queued Switches with 100% Throughput," Broadband Switching Symposium'99, 1999. http://www.schoenen-service.de/assets/papers/Schoenen99bssw.pdf