Combined Input-Crosspoint Buffered Packet Switch with Shared Crosspoint Buffers[1]

Roberto Rojas-Cessa and Ziqian Dong
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
University Heights
Newark, New Jersey 07102
{rrojas, zd2}@njit.edu

Abstract — **The amount of memory in buffered crossbars in combined input-crosspoint buffered switches is proportional to the number of crosspoints, or $O(N^2)$, where N is the number of ports, and to the crosspoint buffer size, which is defined by the distance between the line cards and the buffered crossbar, to achieve 100% throughput under high-speed data flows. A long distance between these two components can make a buffered crossbar costly to implement. In this paper, we propose a combined input-shared-crosspoint buffered packet switch that uses small crosspoint buffers to support long round-trip times, which consider the distance between the line card and the buffered crossbar. The proposed switch reduces the required buffer memory of the buffered crossbar by 50% or more, and uses no speedup.**

I. Introduction

As optical technology spreads quickly and ubiquitously, it is becoming feasible to transmit single flows with increasingly high data rates. High-performance switches and routers are required to handle such flows and, therefore, to provide high-speed ports.

Combined input-crosspoint buffered (CICB) 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 [3]. These packet switches use time efficiently as input and output port arbitrations are performed independently [2]-[14]. Incoming variable-size packets are segmented into fixed-length packets, called cells, at the ingress side of a switch and re-assembled at the egress side, before the packets depart from the switch. This paper considers the use of cells.

The amount of memory in a buffered crossbar is

$$N^2 \times k \times L, \tag{1}$$

where N is the number of input/output ports, k is the crosspoint buffer size in number of cells, and L is the cell size in bytes. The value of k is defined by the length of the round-trip time (RTT), which is defined in [7] as the sum of the delays of: 1) the input arbitration IA, 2) the transmission of a cell from an input to the crossbar $d1$, 3) the output arbitration OA, and 4) the transmission of the flow-control information back from the crossbar to the input, $d2$. Cell and bit alignments are included in the transmission times. For example, the switch proposed in [7] requires the size of k be equal to or larger than the round-trip time to avoid throughput degradation or crosspoint-buffer underflow for flows (here defined as the data arriving at input i and destined to output j, where $0 \leq i, j \leq N-1$) with high data rates.

In a CICB switch, the crosspoint-buffer size to avoid underflow by flows of data rate C b/s, where C is the port speed, is

$$RTT = d1 + OA + d2 + IA \leq k, \tag{2}$$

such that cells are transmitted continuously every time slot [7].

Furthermore, as the buffered crossbar can be physically located far from the input ports in a multi-rack router implementation, actual RTTs can be long. To support long RTTs in a buffered-crossbar switch, the crosspoint-buffer size needs to be increased [9], such that up to RTT cells can be buffered. However, the memory amount that can be allocated in a chip is limited, and therefore, it can make the implementation costly or infeasible when the distance between line cards and the buffered crossbar is long, or else to provide k such that $k < RTT$ without supporting high data rates. An interesting scheme using limited memory is presented in [10] for a switch with p traffic classes, where the crosspoint buffer size is larger than RTT for a single class, and smaller than $p \times RTT$.

A solution to keep the crosspoint buffer small while supporting long RTTs and high data rates is needed. In this paper, we study a CICB switch that uses round-robin arbitration and credit-based flow control, named CIXB switch, under long round-trip times and high data-rate flows. We show the throughput degradation as a function of the round-trip time and the crosspoint buffer size.

To reduce the memory amount or support longer RTT values, we propose a CICB switch with shared memory (CICB-SM) at the crosspoints. The crosspoint buffer is shared by two input ports. The shared memory in the crosspoints reduces the overall memory of a switch with dedicated crosspoint buffers required to handle flows with high data rates. We show a basic CICB-SM architecture, as proposed here, supports a given round-trip time with half or less memory than a a buffered crossbar with dedicated crosspoint buffers. Furthermore, we show that no speedup is needed when using the shared-memory approach.

This paper is organized as follows. Section II describes the CIXB switch, which uses dedicated crossspoint buffers. Section III discusses the effect of long round-trip times in the

[1] This work was supported in part by the National Science Foundation under Grants 0435250 and 0423305, and in part by NJIT under Grant 421070.

CIXB switch. Section IV introduces the proposed CICB-SM switch. Section V presents the throughput performance of the CICB-SM switch under different traffic patterns. Section VI presents the conclusions.

II. Combined Input-Crosspoint Buffered (CIXB) Switch

A buffered crossbar has N inputs and outputs. A crosspoint (XP) element in the buffered crossbar that connects input port i to output port j is denoted as $XP(i,j)$.

There are N VOQs at each input. A VOQ at input i that stores cells for output j is denoted as $VOQ(i,j)$. The XP Buffer of $XP(i,j)$ is denoted as $XPB(i,j)$. The size of $XPB(i,j)$ is k cells, where $k \geq 1$.

A credit-based flow control mechanism indicates input i whether $XPB(i,j)$ has room available for a cell or not. Each VOQ has a credit counter, where the maximum count is the number of cells that $XPB(i,j)$ can hold. When the number of cells sent by $VOQ(i,j)$ reaches the maximum count, $VOQ(i,j)$ is considered not eligible for input arbitration and overflow on $XPB(i,j)$ is avoided. The counter is increased by one each time a cell is sent to $XPB(i,j)$ and decreased by one each time that $XPB(i,j)$ forwards a cell to output j. If $XPB(i,j)$ can receive at least one cell, then $VOQ(i,j)$ is considered eligible by the input arbiter. In this paper, we consider credit-based flow control mechanism.

Round-robin arbitration is used at the input and output ports. An input arbiter at input i selects $VOQ(i,j)$ among the eligible VOQs, to send a cell to XPB for output j at buffered crossbar. An output arbiter at output port j in the buffered crossbar selects a $XPB(i,j)$, among occupied XPBs from input i, to send a cell to output j. The eligibility of VOQs is determined by the flow control mechanism.

III. Effects of Long Round-Trip Time and Limited k

To keep up with high data rates, switch ports must be able to handle flows of up to C b/s,[1] where C is the data-rate capacity of a port in a switch or router. In a CICB switch (e.g., the CIXB switch presented in [7]), the maximum flow rate that the switch can handle is $C\frac{k}{RTT}$. Note that when $r_{f(i,j)} = C$, where $r_{f(i,j)}$ is the rate of $f(i,j)$, the maximum flow rate that the CIXB switch can transfer from inputs to outputs is equivalent to its achievable throughput.

We simulated the CIXB switch to observe the throughput obtained under different k and RTT values in a 32×32 switch, and to validate the traffic model to test the proposed architecture. Different from [7], we consider $RTT > 0$ in this paper. Here, we assume that the distances between input ports and the buffered crossbar are identical.[2] To model flows with different rates, we use the unbalanced traffic model [7].

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,

$$\rho(s,d) = \begin{cases} \rho\left(w + \frac{1-w}{N}\right) & \text{if } s = d \\ \rho\frac{1-w}{N} & \text{otherwise.} \end{cases} \tag{3}$$

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 $s = d$.

Therefore, the fraction of C that $f(i,j)$ uses is

$$r_{f(i,j)} = w + \frac{1-w}{N}. \tag{4}$$

The maximum data rate of $f(i,j)$ is represented by setting $w = 1$ or

$$r^{max}_{f(i,j)} = C,$$

and the minimum data rate is represented when $w = 0$ or

$$r^{min}_{f(i,j)} = \frac{1}{N}.$$

We emphasize our observations in these two w values of the unbalanced traffic model.

Figure 1 shows that when flows have a rate $r_{f(i,j)} = r^{min}_{f(i,j)}$ (i.e., w=0) for different k values, such that $RTT - k < N$, the throughput is 100%, as shown by curves 1) and 5), where $RTT - k = 0$, and as shown by curves 4) and 6), where $RTT - k = 31$. The uniform distribution of traffic relaxes the demand for buffer space, resulting in high throughput. The figure also shows that when $RTT - k \geq N$, the throughput is less than 100%, as shown by curve 2), where $RTT - k = 32$.

Furthermore, as the data rate of the flow increases (i.e., w), throughput degradation occurs. The worst-case scenario is observed when $r_{f(i,j)} = C$ b/s (i.e., w=1) where the achieved throughput is $\frac{k}{RTT}$ for $RTT - k > 0$, as curves 2), 3), 4), and 6) show.

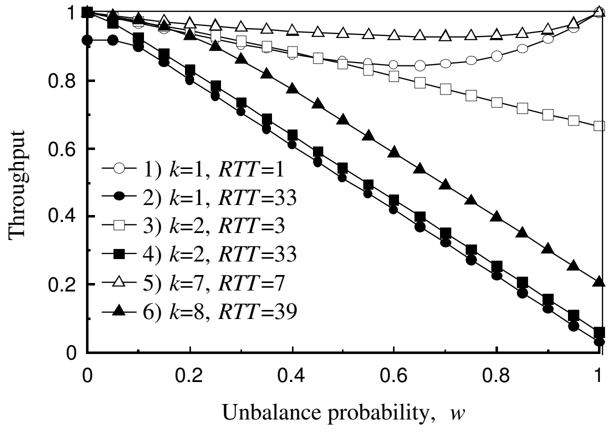

Figure 1: Throughput performance of a 32×32 CIXB switch[7] with $RTT > 0$.

IV. Combined Input-Shared-Crosspoint Buffered Switch (CICB-SM)

As discussed in Section III, the largest throughput degradation occurs when the $r_{f(i,j))} = C$ b/s, or $w = 1$ in the unbalanced traffic model. Under these conditions, all traffic

[1] In contrast, switches unable to support such flows can only handle aggregated data rates of C b/s, where each flow might have a data rate r_{single}, such that $r_{single} < C$.

[2] The results in this paper also apply for non-identical distances.

at input i goes to the crosspoint that connects to output j and the other crosspoints receive no traffic. This motivates the sharing of the crosspoint memory by two or more inputs. The resulting switch turns out as a CICB switch with shared memory (CICB-SM). In this paper, we describe a switch where the crosspoint buffer is shared by two inputs. Figure2 shows the architecture of the proposed CICB-SM switch.

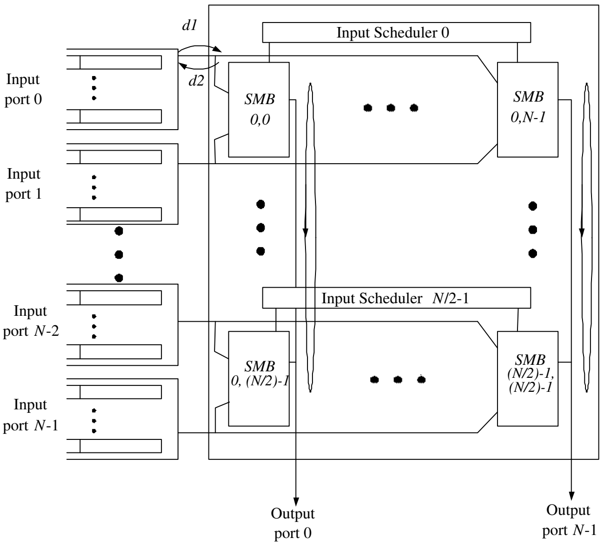

Figure 2: $N \times N$ buffered crossbar with shared crosspoints.

This switch also has N VOQs at each input. A crosspoint in the shared-memory buffered crossbar that connects input port i to output j is also denoted as $XP(i,j)$ as in the CIXB switch. The buffer for $XP(h,j)$ and $XP(h',j)$, where $0 \leq h$, $h' \leq N-1$ and $h \neq h'$,[3] stores cells for output port j and is shared by these two crosspoints (or inputs h and h') is denoted as $SMB(l,j)$, where $0 \leq l \leq \frac{N}{2}-1$. Since each crosspoint buffer is shared by two inputs at different outputs, there are $\frac{N^2}{2}$ shared-memory crosspoint buffers in the buffered crossbar.

To eliminate the speedup at SMBs, only one input is allowed to access a SMB at a time. To schedule the SMB access between two inputs that are physically separated, an input-access scheduler is used among the inputs h and h' and $SMB(l,j)$. The size, in number of cells that can be stored, of a SMB is k_s. There are $\frac{N}{2}$ input-access schedulers in the buffered crossbar, each denoted as $S(l)$. An input-access scheduler matches the non-empty inputs to the SMBs that have room for storing a new cell. Figure 3 shows the input-access schedulers.

The input-access scheduler performs a matching process among the the shared-crosspoint buffers and the inputs that share them. Figure 3 shows the inputs and the shared crosspoint buffers in the matching. In this paper, the matching follows three-phase process, as performed for input-queued (IQ) switches. The matching scheme used in this switch is round-robin based [15] to have a valid comparison with the CIXB switch. However, any matching scheme can be used.

At each output j in the buffered crossbar, there is an output arbiter to select the outgoing cell from non-empty XPs. Figure 2 shows the output arbiters, where the transmission delays between ports and the crosspoint are denoted by $d1$ and $d2$. An output arbiter can consider up to two cells from a SMB, where each cell belongs to one input.

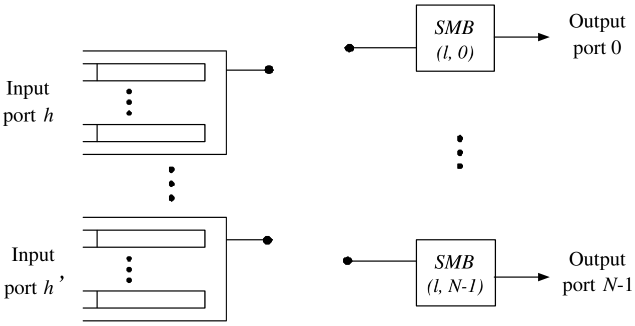

Figure 3: Bipartite matching in a input-access Scheduler

The way the switch works is as follows. Cells with destination to output j arrive at $VOQ(i,j)$ and wait for dispatching. Input i notifies the input-access scheduler about the new cell arrival. The input-access scheduler, denoted as $S(l)$, selects the next cells to be forwarded to the crossbar by matching inputs h and h' to $SMB(l,j)$. Figure 4 shows an example of a 2×4 bipartite match. The input-access scheduler uses a round-robin based matching. A cell going from input i to output j enters the buffered crossbar and is stored in $SMB(l,j)$. Cells leave output j after being selected by the output arbiter. The output arbiter uses round-robin selection.

A credit-based flow control[7] is applied by the input-access schedulers to avoid crosspoint-buffer overflow. $S(l)$ considers those eligible non-empty VOQs for which the SMBs are not full. The input-access scheduler information is sent from the SMB to the corresponding VOQ. Cells and flow-control data experience transmission delay between input ports and the buffered crossbar.

V. Throughput of the CICB-SM Switch

We observe the switching performance of two 32×32 switches, one using dedicated buffers per input and output, and the other using shared buffers by two inputs, as described in Section IV. We study the throughput and average delay under traffic with Bernoulli and bursty arrivals with uniform distribution, and the throughput under Bernoulli traffic with unbalanced distribution.

V.A Uniform Traffic

Figure 5 shows average cell delay of the CIXB and CICB-SM switch under uniform traffic. In this figure, we use $k=1$ for the CIXB switch, and $k_s=1$ for the CICB-SM switch. By using these crosspoint buffer sizes, the amount of memory used in CICB-SM switch is half of the amount of memory in the CIXB switch for a given cell size. The average delay of CICB-SM switch only considers the queuing delay. The average delay of the CICB-SM switch is similar to that of the CIXB switch without the effects of RTT (i.e, $RTT=0$). The larger average delay of the CICB-SM switch from loads of 0.1 to 0.8 are constant because the VOQs need to notify

[3] For switches with odd number of ports, one port is left with dedicated buffers of size 0.5 the capacity of a SMB.

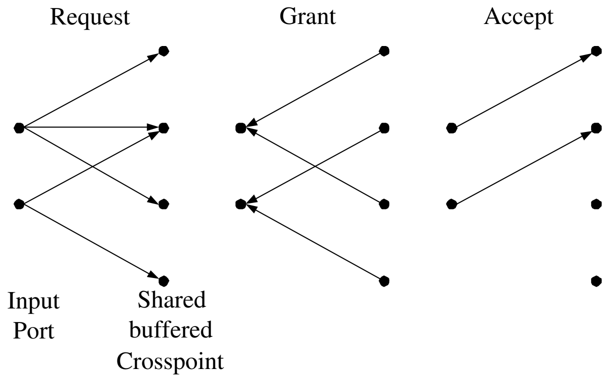

Figure 4: Example of the matching process in a 4 × 4 switch.

the input-access scheduler when a new cell arrives, which may take one time slot. However, the magnitude of one time slot is small to be considered important. For loads over 0.9, the CICB-SM switch has the average delay similar to that of the CIXB switch. The average delay of both switches under bursty traffic has similar magnitude. Therefore, the CICB-SM switch has equivalent performance under uniform traffic while using half memory amount of the CIXB switch.

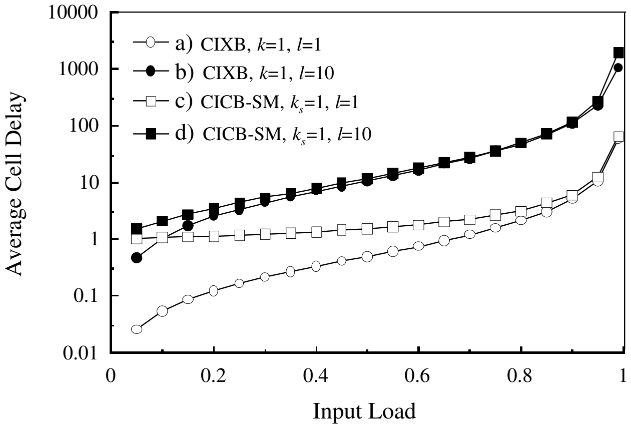

Figure 5: Average queuing delay of a 32 × 32 CICB-SM switch.

V.B Unbalanced Traffic

We observe the effect of long RTTs in the proposed switch models by measuring the switch throughput under the unbalanced traffic model, as in Section III. Figure 6 shows the throughput performance of the CICB-SM switch, with $k \geq 1$. The switch has a symmetric throughput when $w = 0$ and $w = 1$ or $r_{f(i,j)} = r^{max}_{f(i,j)} = r^{min}_{f(i,j)}$, and achieves 100% throughput for $k_s - RTT \geq 0$, as the figure shows in curve a) and c). For these values of w, the throughput can be 100% using half of the total memory used by the CIXB switch. For curves b) and d), which have $k_s - RTT < 0$, the throughput decreases when w approaches 1.

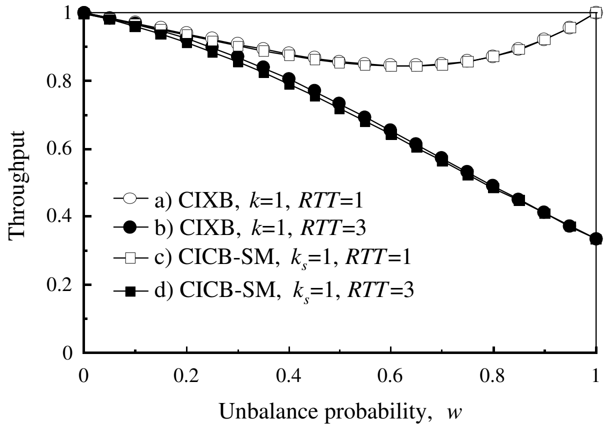

Figure 6: Throughput of a 32×32 CICB-SM switch with half the memory of the CIXB switch.

For the other values of w, we see that when $k_s = k$, the throughput is similar. This is because the buffered crossbar switches seems to have small sensitivity to the crosspoint buffer size. The decreased throughput around w=0.6 in curves a) and c), where k_s, $k = RTT$, is the result of having a limited and small buffer size, mixed traffic (the high data-rate flow is mixed with a large number of low data-rate flows) as described in Section III, and round-robin arbitration. In these cases, a more elaborate arbitration scheme [16] can be used to improve the throughput for small and positive $k_s - RTT$ values.

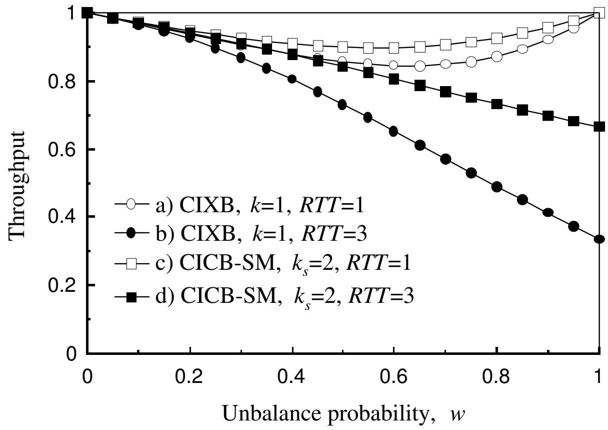

Figure 7: Throughput of the 32×32 CICB-SM and CIXB switches with same amount of memory.

Figure 7 shows the throughput of the 32×32 CICB-SM and CIXB switches under unbalanced traffic when they have the same amount of memory ($k_s = 2k$) in the buffered crossbar. The throughput of the CICB-SM switch is higher than that of the CIXB switch under the same RTT values.

VI. Conclusions

We presented the effect of long round trip times RTTs, where the crosspoint buffer size k is such that $k < RTT$. We observed that switches based on buffered crossbars with the architecture in [7] have their maximum throughput as the ratio of $\frac{k}{RTT}$, when input ports handle a single flow with a data rate equal to the port capacity. To minimize the crosspoint-buffer size, we proposed a switch where the crosspoint buffers are shared by two inputs, such that RTT can be twice as long as that supported by the CIXB switch without decreasing switching performance, and while providing 100% throughput for high data-rate flows. Therefore, the proposed switch relaxes the amount of memory to $\frac{1}{2}$ of the amount required by the CIXB switch. In addition, we showed that the shared memory used in the crosspoint buffers needs no speedup.

References

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

[2] 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.

[3] 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.

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

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

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

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

[9] F. Abel, C. Minkenberg, R. P. Luijten, M. Gusat, and I. Iliadis, "A Four-Terabit Single-Stage Packet Switch with Large Round-Trip Time Support," *Proceedings of Hot Interconnects, 2002. 10th Symposium on*, pp. 5-14, Aug. 2002.

[10] R. Luijten, C. Minkenberg, and M. Gusat, "Reducing Memory Size in Buffered Crossbars with Large Internal Flow Control Latency," *Proceedings of IEEE Globecom 2003*, Vol. 7, pp. 3683-3687, Dec. 2003

[11] M. Katevenis, G. Passas, D. Simos, I. Papaefstathiou, N. Chrysos, "Variable Packet Size Buffered Crossbar (CICQ) Switches," *Proceedings of IEEE ICC 2004*, vol. 2 , pp. 1090-1096, June 2004.

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

[13] L. Mhamdi and M. Hamdi, "MCBF: a high-performance scheduling algorithm for buffered crossbar switches," *IEEE Commun. Letters*, Vol. 7, Issue 9, pp. 451-453, September 2003.

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

[15] N. McKeown, "The *i*SLIP Scheduling Algorithm for Input-Queue Switches," *IEEE/ACM Trans. Networking*, vol. 7, no. 2, pp. 188-200, April 1999.

[16] R. Rojas-Cessa and E. Oki, "Round-Robin Selection with Adaptable-Size Frame in a Combined Input-Crosspoint Buffered Switch," *IEEE Commun. Letters*, vol. 7, issue 11, pp. 555-557, November 2003.