

# Combined Input-Crosspoint Buffered Packet Switch with Flexible Access to Crosspoints Buffers

Roberto Rojas-Cessa and Ziqian Dong

**Abstract**—The performance of Internet routers is greatly defined by the adopted switch architecture. Combined input-crosspoint buffered (CICB) packet switches are being considered of research interest because of their high switching performance. One of the main requirements in these switches is that the amount of memory needed to achieve 100% throughput under flows with high data rates must be proportional to the number of ports and crosspoint buffer size, which is set by the distance between the line cards and the buffered crossbar. Therefore, long distances between the line cards and the buffered crossbar can make a CICB switch costly to implement or infeasible. In this paper, we propose and discuss two CICB packet switches with flexible access to crosspoint buffers. The proposed switches allow an input to use any available crosspoint buffer at a given output, instead of having rigid access where an input can only access a dedicated crosspoint buffer at a given output, as is the case on previous existing architectures. The proposed switches provide high switching performance and support long distances between the buffered crossbar and the line cards, while using crosspoint buffers of small size. Our switches reduce the required crosspoint buffer size by a factor of  $N$ , where  $N$  is the number of ports, keep service of cells in sequence, and use no speedup.

**Index Terms**—Buffered crossbar, round-trip time, memory access, Birkhoff-Von-Neumann, crosspoint buffer

## 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 be capable of handling such flows and, therefore, to provide high-speed ports.

Combined input-crosspoint buffered (CICB) switches provide flexible arbitration timing and high-performance switching for packet switches with high-speed ports [1]-[7]. These packet switches use time efficiently as input and output port arbitrations can be performed independently.

In this paper, we consider that incoming variable-size packets can be 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.

The memory amount in a buffered crossbar can make it costly to implement as the memory amount is  $N^2 \times k \times L$ , 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 duration of the round-trip time. For example, a CICB switch with dedicated allocation of crosspoint buffers (i.e., a set of crosspoint buffers that can only be accessed by a given input) requires the size of  $k$  to 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. The round trip time  $RTT$ , as defined in [5], is 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,  $d2$ .

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

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

such that cells are transmitted continuously every time slot [5].

Furthermore, as the buffered crossbar can be physically located far from the input ports in a real implementation, actual  $RTT$ s can be long. To support long  $RTT$ s in a buffered-crossbar switch, the crosspoint-buffer size needs to be increased [8], such that up to  $RTT$  cells can be buffered. However, the memory amount that can be allocated in a chip is limited, specially because of the use of advanced high-speed interconnection technology with large area requirements. This can make the implementation costly or infeasible when the distance between line cards and the buffered crossbar is long, while achieving high throughput. A solution to keep the crosspoint buffer small while supporting long  $RTT$ s is needed.

In this paper, we propose two CICB switches with flexible access to crosspoint-buffer. In these switches, an input can send a cell to any crosspoint buffer at a given output, contrary to CICB switches where inputs can only access to their dedicated crosspoint buffer per output. Herein, the switches without flexible access are called CICB switches with rigid access (CICB-RA). Note that most CICB switches in the literature [1]-[7] belong to this category, except those switches with shared memory [9]. We start our discussion by showing the throughput degradation of an CICB switch with rigid access in function of the round-trip time and the crosspoint buffer size. We introduce a general architecture of a switch with flexible access, where an input can send a cell to any crosspoint buffer independently of other inputs. We call this switch CICB with

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

The authors are with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark NJ 07102. Roberto Rojas-Cessa is the correspondence author. Email: rrojas@njit.edu. Phone: (973)-596-3508, Fax: (973)-596-5680.

full access (CICB-FA) to crosspoint-buffers. To avoid speedup in the crosspoint buffer, the queues at the inputs are matched to the crosspoint buffers for different outputs. In addition, we introduce a simplified switch, where the interconnecting stage follows a predetermined connectivity similar to that of the Birkhoff-Von Neumann switch [10], allowing one set of crosspoints of different outputs to be accessed by an input at a given time slot. We call this switch CICB switch with single access (CICB-SA) to crosspoint buffers.<sup>1</sup> These two switches support flows with high data rates while using  $k < RTT$ . We discuss the pros and cons of these two switches. As a result, we show that a CICB switch with flexible crosspoint-buffer access requires  $\frac{1}{N}$  of the buffer amount in a switch with rigid crosspoint buffer access to achieve similar or better performance, without using speedup.

This paper is organized as follows. Section II shows the throughput degradation of a CICB with rigid access to crosspoint buffers. Section III introduces the CICB switch with full access to crosspoint buffers. Section IV introduces the CICB switch with single access to crosspoint buffers. Section V discusses the service of cells in sequence by first-come first-serve output arbitration. Section VI presents the throughput performance of the proposed switches. Section VII presents the conclusions.

## II. THE EFFECT OF LONG ROUND-TRIP TIME IN A CICB SWITCH WITH RIGID ACCESS TO CROSSPOINT BUFFERS

To keep up with high data rates, switch ports must be able to handle flows of up to  $C$  b/s,<sup>2</sup> where  $C$  is the data-rate capacity of a port in a switch or router. In a CICB switch with rigid access crosspoint buffers to each VOQ (also referred as CICB with rigid crosspoint-buffer access), the maximum flow rate that can be handled 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 is equivalent to the achievable throughput.

We simulated a CICB switch that uses longest queue first (LQF) as input arbitration and first-come first-serve (FCFS) as output arbitration scheme to observe the throughput obtained under different  $k$  and  $RTT$  values by a  $32 \times 32$  switch, and to validate the traffic model used to simulate flows with high data rates. We consider  $RTT > 0$  in this paper. We also assume that the distances between input ports and the buffered crossbar are identical. To model flows with different rates, we use the unbalanced traffic model [5], which uses  $w + \frac{1-w}{N}$  as the fraction of the input load directed from input  $i$  to output  $j = i$ , where  $w$  is the unbalanced probability. The remainder of the input load (i.e.,  $\frac{1-w}{N}$ ) is directed from input  $i$  to output  $j \neq i$  (with a uniform distribution). Therefore, the fraction of  $C$  that  $f(i,j)$  uses is  $r_{f(i,j)} = w + \frac{1-w}{N}$ . The maximum data rate of  $f(i,j)$  is represented by making  $w=1.0$  or  $r_{f(i,j)}^{max} = C$ , and the

minimum data rate is represented when  $w=0.0$  or  $r_{f(i,j)}^{min} = \frac{1}{N}$ . We emphasize our observations in these two  $w$  values of the unbalanced traffic model.

Figure 1 shows that the throughput degrades when  $r_{f(i,j)} = r_{f(i,j)}^{min}$  (i.e.,  $w=0.0$ ) in curve 2, where  $RTT = 31$  and  $k = 1$ , and curve 5, where  $RTT=61$  and  $k=2$ . In these two cases the throughput is below 99%. A preliminary conclusion is that the throughput falls under 99% when  $RTT \geq kN$ . However, the arbitration schemes may be the factor that causes the throughput loss. The figure also shows that the throughput remains close to 100% when:  $RTT \leq kN$ , as shown by curve 1, where  $k = RTT = 1$ , curve 3, where  $k=2$  and  $RTT=3$ , and curve 4, where  $k=2$  and  $RTT=33$ , all at  $w=0.0$ .

As the flow data rate increases (i.e.,  $w$ ), the throughput degradation increases. The worse-case scenario is observed when  $r_{f(i,j)} = C$  b/s (i.e.,  $w=1.0$ ) as the achieved throughput is  $\frac{k}{RTT}$ , as shown by curves with  $RTT > k$ . The case of port-speed flows, although mostly ignored, is when a flow, at input  $i$ , with a rate equal to the port capacity is being sent to output  $j$ .

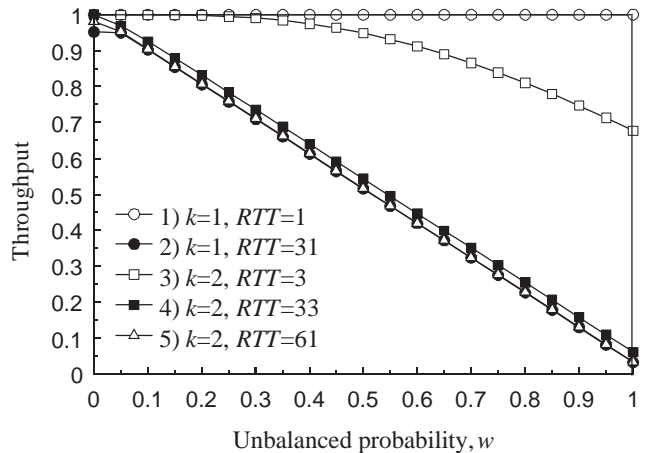


Fig. 1. Throughput performance of a  $32 \times 32$  CICB switch with  $RTT > 0$ .

## III. CICB SWITCH WITH FULL ACCESS (CICB-FA) TO CROSSPOINT BUFFERS

The  $N \times N$  CICB-FA switch has virtual output queues (VOQs) in the input ports, a fully interconnected stage that provides connectivity for input  $i$  to any of the  $N^2$  crosspoint buffers, and a buffered crossbar. Figure 2 shows this switch architecture. A VOQ at input  $i$  that stores cells for output  $j$  is denoted as  $VOQ(i,j)$ . The fully interconnecting stage is combined with the buffered crossbar. A crosspoint in the buffered crossbar is denoted as  $XP(h,j)$ , where  $0 \leq h \leq N-1$ ,<sup>3</sup> and the corresponding crosspoint buffer is denoted as  $XPB(h,j)$ . As per the fully interconnecting stage, input  $i$  is able to access any  $XPB(h,j)$ .<sup>4</sup> To ensure only one cell be

<sup>1</sup>The study of the CICB-SA switch is motivated by the high performance of a round-robin based switch [11].

<sup>2</sup>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  $r_{single} < C$ .

<sup>3</sup>Note that a XP does not have a one-to-one association with a VOQ as in CICB switches with rigid access.

<sup>4</sup>In CICB-RA, input  $i$  can only access  $XPB(h,j)$ , where  $h = i$ .

written into a crosspoint buffer, each crosspoint has a  $N - to - 1$  multiplexer, denoted as  $MUX(h, j)$ . Furthermore, each input can send one cell to the crosspoint buffer, and each crosspoint buffer can receive up to one cell at each time slot. There is an output arbiter for each output port. There is an output access scheduler (OAS) per output port and an input access scheduler (IAS) per input port, both located at the buffered crossbar. IAS and OAS perform a parallel matching to determine which  $XPB$  can be accessed by a cell (or input). There are  $N$  VOQ counters at the buffered crossbar, denoted as  $VC(i, j)$ , which counts the number of cells at  $VOQ(i, j)$ . In this paper, we consider crosspoint buffers with  $k = 1$  and with no speedup.

The way this switch works is as follows. When a cell destined to output  $j$  arrives at input  $i$  is stored in  $VOQ(i, j)$ . The input sends a request for this cell to the buffered crossbar and the corresponding VOQ counter  $VC(i, j)$  is increased by one. In the time slot after the increment of VC, a request is sent to the OAS for output  $j$ . The OAS for output  $j$  selects up to  $N$  cells for crosspoints at output  $j$  after considering all requests from non-empty VOQs and the availability of XPBs. The access scheduler notifies the IAS which requests were selected. Since an input may be granted access to XPBs at different outputs (i.e., IAS receives several grants), the IAS performs accepts one grant and notifies the OAS. The scheme used by IAS and OAS is LQF selection. After being notified by a forward signal, an input sends the cell to the crosspoint buffer one time slot after receiving the forward-signal information. After a cell arrives in the XPB, the corresponding VC decreases by one.

The output arbiter at output  $j$  (note that this is not part of the crosspoint access process) selects an occupied crosspoint buffer to forward a cell to the output in a first-come first-serve (FCFS) fashion. FCFS is used for output arbitration to keep cell in sequence as it will be discussed in Section V. This switch uses no speedup as inputs and crosspoint process one cell per time slot.

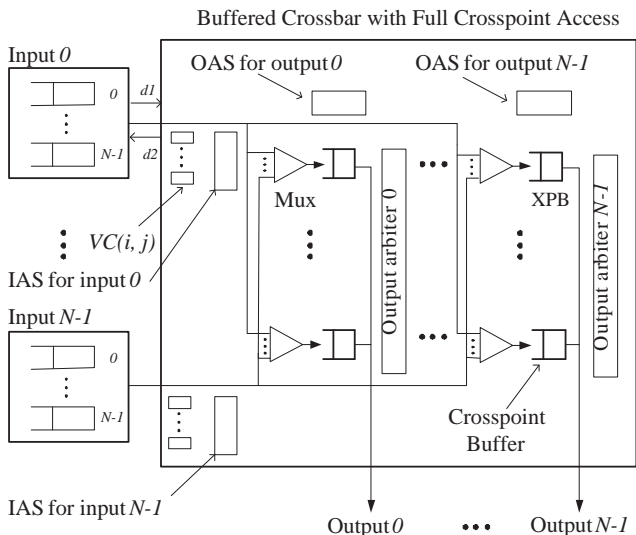


Fig. 2.  $N \times N$  CICB switch with full access (CICB-FA).

#### IV. CICB SWITCH WITH SINGLE ACCESS (CICB-SA) TO CROSSPOINT BUFFERS

The switch with full access has  $N^2 N - to - 1$  multiplexers. In addition, the crosspoint access scheduler needs to perform matching between inputs and outputs. To minimize the complexity and hardware amount, we present a simpler CICB switch with flexible access to crosspoint buffers, the CICB switch with single access (CICB-SA).

This switch has VOQs in the input ports, an interconnecting stage that uses pre-determined and cyclic configurations, similar to those used in a Birkhoff-Von Neumann switch [10], and a buffered crossbar. Figure 3 shows this switch architecture. In this switch, the input ports are also called external inputs, each of which is denoted as  $EI_i$ . The outputs of the interconnecting stage are called internal outputs, each of which is denoted as  $IO_l$ , where  $0 \leq l \leq N - 1$ .  $IO$ s are physically equivalent to the inputs of the buffered crossbar, also called internal inputs, each of which is denoted as  $II_l$ . The outputs of the buffered crossbar, or output ports, are also called external outputs, each of which denoted as  $EO_j$ .

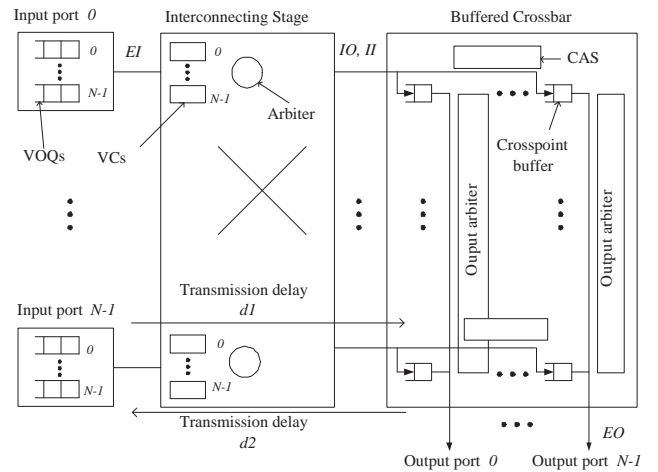


Fig. 3.  $N \times N$  CICB switch with single access (CICB-SA).

As in CICB-FA, there are  $N$  VOQ counters,  $VC(i, j)$  for each input in the interconnecting stage in CICB-SA. In each  $EI$ , there is an input arbiter. In each  $II_l$ , there is one crosspoint access scheduler, denoted as  $(CAS_l)$ , that schedules access of a cell from input  $i$  to  $XPB_l$  (via  $II_l$ ). A  $CAS$  and the input arbiter at  $EI_i$  select a crosspoint buffer and a VOQ, respectively, by using longest queue first selection in a 3-phase parallel matching process and a predetermined configuration of the interconnecting stage (e.g., similar to a load-balanced stage) that changes every time slot.

The way this switch works is as follows. At  $EI_i$ , a cell with destination to output  $j$  arrives in  $VOQ(i, j)$  and sends a request indicating the arrival to  $VC(i, j)$ . Each input arbiter sends a request to a  $CAS$  for which  $VOQ(i, j)$  has a cell for.  $CAS_l$  selects a request from non-empty  $VOQ(i, j)$  with the longest occupancy for available  $XPB$  at  $j$ . At a scheduling time  $t$ , the configuration of the interconnecting stage pairs  $EI_i$

to  $II_i$  by using  $l = (i + t)$  modulo  $N$ . A crosspoint in the buffered crossbar that connects internal input  $l$  to output port  $j$ , is denoted as  $XP(l, j)$ . The buffer at  $XP(l, j)$  is denoted as  $XPB(l, j)$ .  $CAS_l$  sends a grant to the input arbiter, and the input arbiter selects a grant, among all those received, by using LQF selection, and acknowledges  $CAS_l$ . At each time slot, a forward signal is sent to the inputs to indicate which VOQ can send a cell to the buffered crossbar. The input dispatches the selected cell to the buffered crossbar. The input dispatches the selected cell to the XPB in the next time slot. Once dispatched by the input, the cell traverses the interconnecting stage and is held at the XPB, and the corresponding VC is decremented by one. A cell going from  $EI_i$  to  $EO_j$  may enter the buffered crossbar through  $II_i$  and be stored in  $XPB(l, j)$ . Cells leave  $EO_j$  after being selected by the output arbiter. As in CICB-FA, the output arbiters in CICB-SA also use FCFS selection to keep cells of  $f(i, j)$  in order. The output arbiter considers the time when a cell arrives at the crosspoint buffer to perform FCFS among dedicated crosspoint buffers. Section V presents the proof of keeping cells in order by FCFS arbitration. Cells and flow control data experience the transmission delay from input ports to the buffered crossbar.

## V. IN-SEQUENCE CELL SERVICE WITH FCFS OUTPUT ARBITRATION

An advantage of using a CICB switch is that all crosspoints buffers are located on a single chip. This makes it easy to keep the arrival time of incoming cells and to use a simple output arbitration scheme to keep cells in sequence. The FCFS selection scheme is used as the output arbitration scheme in each output. FCFS selects first the cells that come into the buffered crossbar first, independently of the cell arrival time at the input queues. The proof of in-sequence cell delivery by these switches is presented in [12], as these are based on a load-balanced CICB switch.

## VI. SWITCHING PERFORMANCE

CICB-FA and CICB-SA were tested under computer simulation, with a confidence interval of 95% for the average cell delay. We consider several admissible traffic patterns and flow data rates in the performance study of the proposed two switches. we consider Bernoulli arrivals under uniform and nonuniform distributions. We extend the traffic with uniform distributions to bursty arrivals (i.e., Markov modulated on-off traffic). We show that the performance under traffic with uniform distributions remains high as that delivered by CICB-RA switches. We also show that the proposed switches deliver higher throughput than CICB switches with rigid access under nonuniform traffic patterns, such as the unbalanced, diagonal, and power-of-two traffic patterns. We show that the throughput is 100% for  $RTT \leq k$ . Furthermore, we show that these switches using a weigh-based arbitration can deliver close to 100% throughput under admissible traffic patterns for  $RTT > k$ . This is a unique feature of these switches as CICB switches with fixed access cannot support such long  $RTT$  values.

### A. Uniform Traffic

We tested all switches under uniform traffic to study the effect of using matching processes for access to the XPBs. Therefore, we set  $k = 1$  and  $RTT = 1$ . Figure 4 shows the average cell delay of a CICB switch (or CICB-RA) using LQF as for input arbitration and FCFS as output arbitration, CICB-SA, and CICB-FA, all under uniform traffic. CICB-FA and CICB-SA also use LQF but for scheduling access to crosspoint buffers, so this is analogous to using LQF as input arbitration in CICB. The average cell delay only considers the queuing delay. For low input loads, CICB shows smaller average cell delay than the proposed switches. This is because in CICB-SA and CICB-FA, cells spend an extra time slot at the VOQs as their requests are sent to the crosspoint access scheduler and have to be granted ( $RTT = 1$ ) before forwarding the actual cells. Under larger input loads, when the average cell delay is larger than one time slot, the average delays of all switches have similar magnitude: the delay is small in any case. This indicates that the access scheduling has not measurable effect in the switching performance. The figure also shows the average delay of all switches under bursty traffic with average burst lengths  $l = \{10, 100\}$ . These results show that the delay increases in proportion to the burst length.

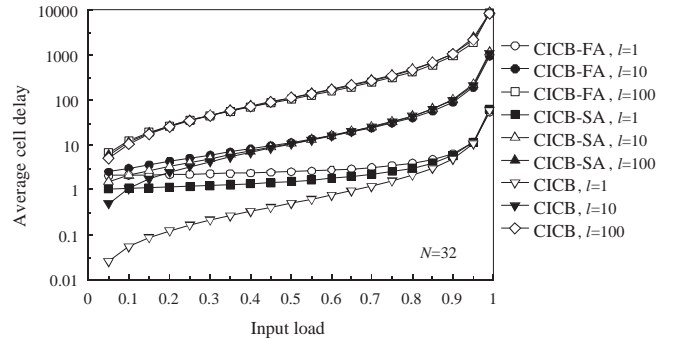


Fig. 4. Average queuing delay of a  $32 \times 32$  CICB switches under uniform traffic.

### B. Nonuniform Traffic: Unbalanced

The effect of long  $RTT$ s in the proposed switch model can be studied by measuring the switch throughput under the unbalanced traffic model, as in Section II, in addition to studying the switching performance under this traffic pattern when  $RTT \leq k$ . The features of this traffic model is the nonuniform distribution of the input traffic to one output port. Figure 5 shows the throughput performance of CICB, CICB-SA, and CICB-FA when  $k = 1$  for different  $RTT$ s. When  $RTT$  is not long, say  $RTT \leq 1$ , all switches deliver close to 100% throughput under this traffic pattern. This follows the known performance for CICB switches with rigid access under weight-based arbitration schemes.

When  $RTT$  is large, say  $RTT > k$ , we can observe the following. CICB has the throughput degraded as  $w$  increases. The worst case is reached at  $w = 1.0$  as discussed in Section II. On

the other hand, CICB-SA and CICB-FA hold their throughput high despite the increase of  $RTT$  and  $w$ . However, we note that both switches have their throughput below 99% when  $RTT \geq 31$ . Furthermore, CICB-FA has higher throughput when  $RTT = N = 32$ . While the throughput of the switches with flexible access decreases for values of  $w$  between 0.3 and 0.7. The throughput remains high when  $w = 1.0$ , which is the case for high data rate flows, while a switch with rigid access has the throughput degraded to  $\frac{k}{RTT}$  for the same  $w$ . CICB-FA and CICB-SA can support a long  $RTT$  as long as the throughput is above 99%. Therefore, values of  $RTT \geq 31$  cannot be supported with  $k = 1$ . However, this is beyond the maximum  $RTT$  that can be supported by CICB switches with rigid access (i.e.  $RTT = k = 1$ ).

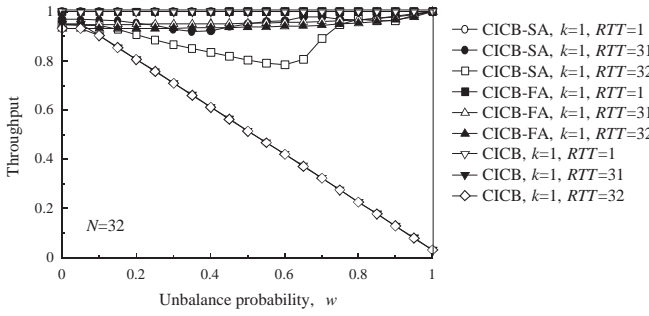


Fig. 5. Throughput of CICB switches with  $k=1$  and  $RTT = \{1, 31, 32\}$  under unbalanced traffic.

The discussion that remains is the comparison of CICB-FA and CICB-SB for the cases where the throughput is close to 99%. Figures 6 and 7 show the throughput performance of CICB-SA and CICB-FA, under different  $RTT$  values and  $k = 1$ . Figure 6 shows that CICB-FA delivers close to 100% throughput for  $RTT \leq 21$ . For larger  $RTT$  values, the throughput decreases below 99%. The throughput is the lowest when  $w = 0.0$  (i.e., uniform distribution) or for flows with low data rates.

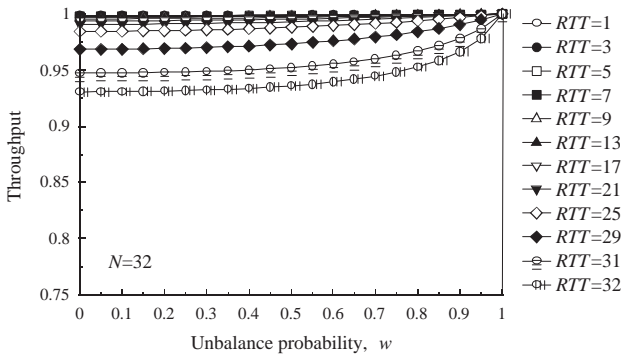


Fig. 6. Throughput of the  $32 \times 32$  CICB-FA switch with  $k=1$  under unbalanced traffic.

As Figure 7 shows, the throughput of CICB-SA is higher than that of CICB-FA when  $RTT \leq 29$ . The throughput of

CICB-FA is higher than that of CICB-SA when  $RTT \geq 31$ . For  $RTT \geq 32$ , the throughput of CICB-SA decreases rapidly. These results show that CICB-FA may have lower efficiency for access to crosspoints as matching is heavily performed. CICB-SA uses similar properties of a load-balancing process that simplifies the matching process and distribute the access to crosspoint buffers. However, for extremely long  $RTT$ s, the total flexibility to access a crosspoint buffer (i.e., the larger number of multiplexers) of CICB-FA becomes a dominant parameter.

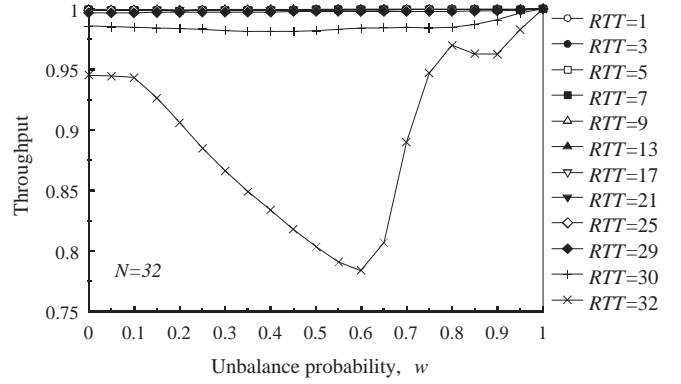


Fig. 7. Throughput of the  $32 \times 32$  CICB-SA switch with  $k = 1$  under unbalanced traffic.

### C. Nonuniform Traffic: Power of Two

In addition, CICB-SA and CICB-FA were simulated under power-of-two traffic [14] for  $30 \times 30$  switches. The power of two (PO2) traffic model can be represented by matrix  $\bar{\rho}$  as:

$$\bar{\rho} = \rho \begin{pmatrix} \frac{1}{2^1} & \cdots & \frac{1}{2^N} \\ \vdots & \ddots & \vdots \\ \frac{1}{2^N} & \cdots & \frac{1}{2^{N-1}} \end{pmatrix}$$

For example, power of two traffic of a  $4 \times 4$  switch is represented as:

$$\bar{\rho} = \rho \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{16} & \frac{1}{2} \\ \frac{1}{8} & \frac{1}{16} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{16} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \end{pmatrix}$$

This traffic model presents a large nonuniformity in the traffic distribution among  $N$  possible destinations. This traffic model, although the sum of rows and column is less than one, it has shown to be difficult for switches to achieve high throughput. Figure 8 shows that both switches deliver 100% throughput under this traffic pattern for  $RTT = 1$  and  $k = 1$ .

### D. Nonuniform Traffic: Diagonal

The diagonal traffic can be represented as  $d\rho(i, j) = d\rho_i$  for  $i = j$ ,  $(1 - d)\rho_i$  for  $j = (i + 1) \bmod N$ , where  $\rho_i$  is the load at input  $i$ , or by the matrix  $\bar{\rho}$  as:

$$\bar{\rho} = \rho \begin{pmatrix} d & (1-d) & 0 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ (1-d) & 0 & \cdots & 0 & d \end{pmatrix}$$



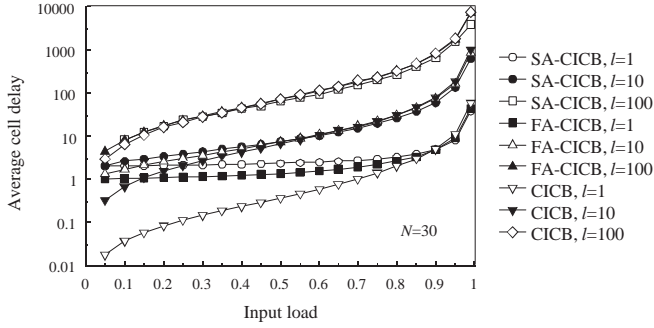


Fig. 8. Performance of  $30 \times 30$  switches with  $k = 1$  under PO2 traffic.

This traffic model presents load distributions among two outputs per each input. The distributions are given by the diagonal degree probability,  $d$ . Figure 9 shows the switching performance of CICB-FA and CICB-SA under diagonal traffic for  $0 \leq d \leq 1$ . This figure shows that these two switches can support  $RTT \leq 31$  and achieve close to 100% throughput.

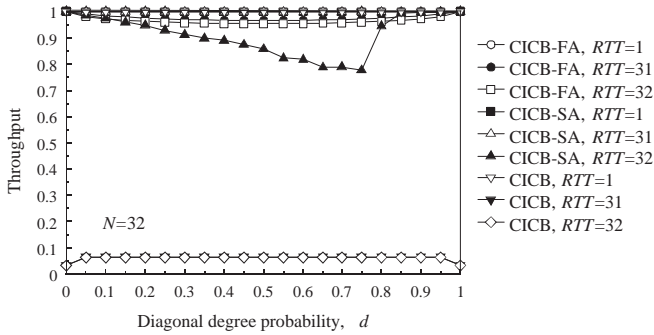


Fig. 9. Throughput of the  $32 \times 32$  switches with  $k = 1$  under diagonal traffic.

## VII. CONCLUSIONS

We presented the effect of long round trip times  $RTT$ 's in CICB switches with rigid access to crosspoint buffers, where the supported crosspoint buffer size is  $k \leq RTT$ . CICB switches with rigid access to crosspoint buffers 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 overcome this, we proposed two novel CICB switch architectures where inputs can access any crosspoint buffer of a given output. We call these CICB switches with flexible access to crosspoint buffers. We study the case of CICB switches with flexible access and with crosspoint buffers of one-cell size. Our proposed switches with  $k = 1$  can support an  $RTT$  close to  $N$ -time-slot long, and provide high throughput for high and low data-rate flows under a great variety of admissible traffic patterns. As a comparison, for a given  $RTT$  size, a CICB switch with flexible access requires a minimum memory amount of  $RTT \times N$  cells while a CICB switch with rigid access requires a minimum  $RTT \times N^2$  cells. Therefore, the proposed switches relax the memory requirement by a factor of  $O(N)$ . In addition, we show that these switches use the

buffered crossbar effectively to assign timestamps to cells arriving in crosspoint buffers. This simplifies the handling of cells to provide in-sequence transmissions. All these features are achieved by CICB switches with flexible access without using speedup.

## REFERENCES

- [1] 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.
- [2] E. Oki, N. Yamanaka, Y. Ohtomo, K. Okazaki, and R. Kawano, "A 10-Gb/s (1.25 Gb/s x8) 4 x 0.25- $\mu$ m CMOS/SIMOX ATM Switch Based on Scalable Distributed Arbitration," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1921-1934, December 1999.
- [3] M. Nabeshima, "Performance Evaluation of a Combined Input- and Crosspoint-Queued Switch," *IEICE Trans. Commun.*, vol. E83-B, No. 3, March 2000.
- [4] K. Yoshigoe, K.J. Christensen, "A parallel-poll Virtual Output Queue with a Buffered Crossbar," *Proc. IEEE HPSR 2001*, pp. 271-275, May 2001.
- [5] R. Rojas-Cessa, E. Oki, Z. Jing, and H. J. Chao, "CIXB-1: Combined Input-One-Cell-Crosspoint Buffered Switch," *Proc. IEEE HPSR 2001*, pp. 324-329, May 2001.
- [6] T. Javadi, R. Magill, and T. Hrabik, "A High-Throughput Algorithm for Buffered Crossbar Switch Fabric," *Proc. IEEE ICC 2001*, pp.1581-1591, June 2001.
- [7] 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.
- [8] F. Abel, C. Minkenberg, R. P. Luijten, M. Gusat, I. Iliadis, "A Four-Terabit Single-Stage Packet Switch with Large Round-Trip Time Support," *Proc. High Performance Interconnects 2002*, pp.5-14, 21-23 Aug. 2002.
- [9] R. Rojas-Cessa and Z. Dong, "Combined Input-Crosspoint Buffered Packet Switch with Shared Crosspoint Buffers," *Proc. Conference in Information Science and Systems*, Baltimore, MD, April 2005.
- [10] C-S. Chang, D-S. Lee, and Y-S. Jou, "Load Balanced Birkhoff-Von Neumann Switches," *Proc. IEEE HPSR 2001*, pp. 276-280, May 2001.
- [11] R. Rojas-Cessa, Z. Dong, and Z. Guo, "Load-balanced Combined Input-Crosspoint Buffered Packet Switch," *IEEE Commun. Letters*, Vol. 4, Issue 7, pp. 661-663, July 2005.
- [12] Roberto Rojas-Cessa, Ziqian Dong, and Sotirios G. Ziavras, "Load-Balanced Combined Input-Crosspoint Buffered Packet Switch with Long Round-Trip Time Support," *Proc. IEEE Globecom 2005*, Vol. 2, pp. 1002-1006, Saint Louis, MO, November 2005.
- [13] 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.
- [14] A. Bianco, M. Franceschinis, S. Ghisolfi, A.M. Hill, E. Leonardi, F. Neri, R. Webb, "Frame-based Matching Algorithms for Input-queued Switches," *Proc. IEEE HPSR 2002*, pp. 69-76, 2002.