Combining Distributed and Centralized Arbitration Schemes for Combined Input-Crosspoint Buffered Packet Switches

Zhen Guo, Roberto Rojas-Cessa, and Nirwan Ansari

***Abstract*— Combined input-crosspoint buffered (CICB) packet switches perform input and output arbitrations separately. This arbitration separation permits distributed selection for both inputs and outputs. While input arbitrations are implemented in a distributed manner because inputs are located in different physical locations, output arbitrations can be implemented in either distributed or centralized manner since output arbiters are placed in the buffered crossbar. An advantage of using a centralized output arbitration approach is that more complex schemes can be executed without completely sacrificing the timing efficiency that CICB switches are known to have. In this paper, we introduce a hybrid arbitration approach that combines a distributed selection scheme at the inputs and a centralized selection scheme at the outputs. By using this hybrid approach, we show that a CICB switch with the minimum crosspoint-buffer size provides 100% throughput under admissible traffic that follows the strong law of large numbers, without using speedup.**

***Index Terms*— Buffered crossbar, crosspoint queued, stability, fluid model, throughput.**

I. Introduction

Combined input-crosspoint buffered (CICB) switches are known to be a practical alternative to provide high-performance switching and to relax arbitration timing for packet switches with high-speed ports [1]-[3]. CICB switches use time efficiently because input and output port selections are performed separately, and the working speed of the crosspoint buffers in a CICB switch is as relaxed as that of the buffers in input buffered (IB) switches.

Because of the arbitration separation, buffered crossbar switches have simpler scheduling algorithms than bufferless crossbar switches (e.g., IB switches must find a maximum weight match between inputs and outputs) to provide high performance.

CICB switches with input buffers, which follow the first-in first-out (FIFO) service policy, or FIFO-CICB switches, have been used to reduce the crosspoint-buffer size and to reduce packet loss ratio. However, a CICB switch with input FIFOs may have the throughput limited by the head-of-line blocking phenomenon. CICB switches with virtual output queues (VOQs), where a queue at the input stores packets for a specific output, or VOQ-CICB switches, can provide 100% throughput under admissible traffic with uniform distribution [4]-[7]. In [8], it has been shown that with a weighted round-robin scheduler, a buffered crossbar can achieve 100% throughput with a speedup of two for admissible traffic for any distribution. However, considering speedup in crosspoint buffers is not practical as memory speed hardly keeps up with the ever increasing line rate. We refer to a VOQ-CICB switch as a CICB switch for the sake of brevity in the remainder of this paper.

In this paper, we propose the longest column occupancy (LCO) first selection as output arbitration that uses a centralized selection approach. We use the longest queue first (LQF) selection scheme as input arbitration. Contrary to LCO, LQF uses a distributed selection approach. We show that the combination of LQF and LCO provides 100% throughput under admissible traffic that follows the strong law of large numbers (SLLN), for any distribution. To support our claim, we prove that CICB switches with one-cell crosspoint buffers and no speedup can provide 100% throughput under admissible traffic that follows SLLN. In this paper, we refer cells to as fixed-size packets, which are the product of segmenting variable-size packet at the inputs. Cells in this paper are not necessarily those of Asynchronous Transfer Mode (ATM). Variable-size packets are reassembled at the outputs before leaving the switch. The intuition of this analytical result lies on the knowledge that CICB switches provide higher performance than IB switches, and that IB switches can provide 100% throughput under admissible traffic with no speedup [9], with a high-complexity matching scheme.

In our analysis, we use the CICB switch's property of performing input and output arbitrations separately. We show that input and output arbitrations can provide sufficient conditions for 100% throughput if: 1) the buffered crossbar has a crosspoint available for any input at any time slot, and 2) every input is able to send a cell of backlogged traffic to an available crosspoint at any time slot.

This paper is organized as follows. Section II describes the switch model. Section III describes the fluid model and some preliminary definitions. Section IV presents the throughput analysis of a CICB switch. Section V introduces the input and output arbitration schemes used to achieve high throughput. Section VI presents our conclusions.

II. CICB Switch Model

Figure 1 shows a CICB switch with N inputs and outputs. There are N VOQs at each input. A VOQ at input i that stores cells for output j is denoted as $VOQ_{i,j}$. A crosspoint

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

The authors are with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102. Email: {rrojas, zxg1554, ansari}@njit.edu.

element in the buffered crossbar that connects input port i, where $0 \le i \le N-1$, to output port j, where $0 \le j \le N-1$, is denoted as $CP_{i,j}$. The buffer at $CP_{i,j}$ is denoted as $CPB_{i,j}$, and it is considered of one-cell size. Therefore, the transmission and arbitration delays are considered negligible, without loss of generality. A large CPB size would allow non-negligible transmission delays. $CPB^{Busy}_{i,j}$ denotes a CPB that is currently storing a cell (e.g., an occupied CPB).

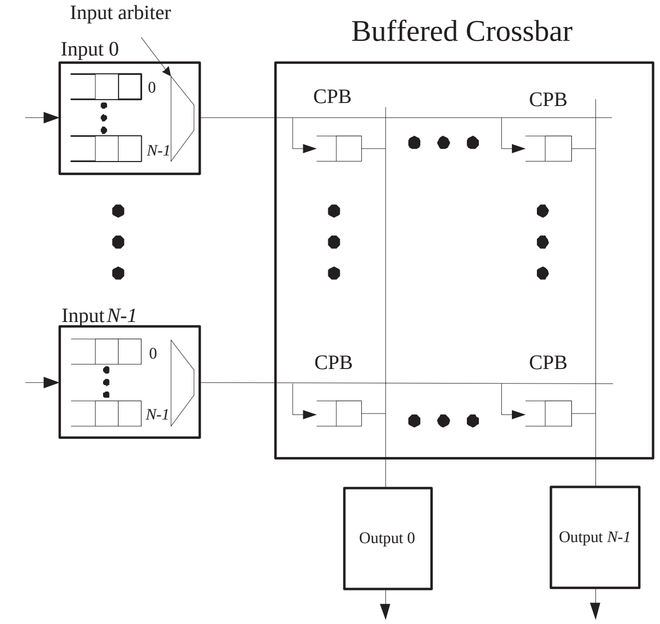

Fig. 1. Combined input-crosspoint buffered crossbar switch.

The occupancy of $VOQ_{i,j}$ at up to time slot n is denoted as $Z_{i,j}(n)$. The cumulative number of packets that have arrived at $VOQ_{i,j}$ by time slot n is denoted as $A_{i,j}(n)$, and the cumulative number of packets that have departed from $VOQ_{i,j}$ by time slot n is denoted as $D_{i,j}(n)$.

In a CICB switch, the input arbitration at input i selects a cell from a non-empty VOQ, whose corresponding CPB is available, to be forwarded to the buffered crossbar (this VOQ is said to be uninhibited). At the same time, the output arbitration selects a cell from an CPB among all those for output j to leave the buffered crossbar. We consider that the output arbitration can adopt either a distributed or a centralized approach as all output arbiters can be placed in the same chip.

III. Fluid Model

We use a fluid model [9] to analyze the properties of the VOQs in a CICB switch with no speedup and look at the stability property of this switch under a traffic model with the restrictions of being admissible and where the cell arrivals follow SLLN:

$$\lim_{n\to\infty} \frac{A_{i,j}(n)}{n} = \lambda_{i,j}, \tag{1}$$

where $\lambda_{i,j}$ is the average arrival rate at $VOQ_{i,j}$.

An input arbitration uses scheme m, such that the selected VOQs can be expressed by matrix $\pi^m_{i,j}(n) \in \Pi$ at time slot n. For π, let T^m_π be the cumulative amount of time that a combination π has been used by time slot n. Therefore, $D_{i,j}(n)$ is the number of departures from $VOQ_{i,j}(n)$ up to time slot n, where $D_{i,j}(0) = 0$ is defined.

Definition 1: If $\lim_{n\to\infty} \frac{D_{i,j}(n)}{n} = \lambda_{i,j}$, the switch is said to be rate stable. It has been proved that a switch is rate stable if the corresponding fluid model is weakly stable [9].

For $n \ge 0$, the switch dynamics are represented as:

$$Z_{i,j}(n) = Z_{i,j}(0) + A_{i,j}(n) - D_{i,j}(n), \tag{2}$$

and

$$\sum_{\pi\in\Pi} T^m_\pi(n) = n, \tag{3}$$

where $T^m_\pi(.)$ is non-decreasing.

A switch under traffic that complies with SLLN can be represented through a fluid model [9].

Definition 2: The fluid model of a switch is said to be weakly stable if for every fluid model solution (D, T, Z) with $Z(0) = 0$, $Z(t) = 0$ for almost every $t \ge 0$ [9].

The dynamics of the fluid model of the switch can be expressed as

$$Z_{i,j}(t) = Z_{i,j}(0) + \lambda_{i,j}t - D_{i,j}(t), \tag{4}$$

and

$$\dot{D}_{i,j}(t) = \sum_{\pi\in\Pi} \pi_{i,j}\dot{T}^m_\pi(t), if Z_{i,j}(t) > 0, \tag{5}$$

where $T^m_\pi(.)$ is non-decreasing and $\sum_{\pi\in\Pi} T^m_\pi(t) = t$. Here, $\dot{g}(t)$ is the derivative of a function $g(t)$ at t.

To express the switch dynamics of (2) into those of a fluid model as in (4), a limiting procedure is used to obtain the fluid limits, which are the solutions to express a time-slotted model into a continuous-time model. The fluid limit of a switch is Lipschitz continuous and therefore is absolutely continuous. For completeness, we recall the following lemma.

Fact 1: (Lemma 1 in [9]) Let $f : [0,\infty) \to [0,\infty)$ be an absolutely continuous function with $f(0) = 0$. Assume that $\dot{f}(t) \le 0$ for almost every t such that $f(t) > 0$ and f is differentiable at t. Then $f(t) = 0$ for almost every $t \ge 0$.

By the fluid behavior of the VOQs', for a weakly stable switch there must exist an $f(t)$, where $f(t) = 0$ implies $Z(t) = 0$ for every $t > 0$, and where $f(0) = 0$ implies $Z(0) = 0$.

IV. Throughput Analysis of a CICB Switch

In this section, we analyze the stability of a CICB switch using the fluid model. We present the following, by means of Theorem 1.

Theorem 1: A CICB, with a VOQ structure at the inputs and using no speedup, provides 100% throughput under admissible traffic.

Proof: 100% throughput means the switch is rate stable by Definition 1. The CICB switch is analyzed in two separated parts. The first part is concerned with the inputs, and the second part with the buffered crossbar. We start with the first part.

As in [9], let's define

$$C_{i,j}(t) = L_i(t) + M_j(t), \tag{6}$$

where

$$L_i(t) = \sum_k Z_{i,k}(t)$$

denotes the total amount of fluid buffered at the input i at time t, and

$$M_j(t) = \sum_k Z_{k,j}(t)$$

denotes the total amount of fluid destined for output j at time t. In other words, $C_{i,j}$ denotes the total amount of fluid at input i and the fluid destined to output j.

Since input and output arbitrations work separately in a CICB switch (see Figure 2, where the complete time slot is used by both input and output arbitrations), if cell c is dispatched from $VOQ_{i,j}$ and is stored at $CPB_{i,j}$, then $L_i(t)$ and $M_j(t)$ decrease by one each. Therefore $C_{i,j}(t)$ is reduced by two in a single time slot.

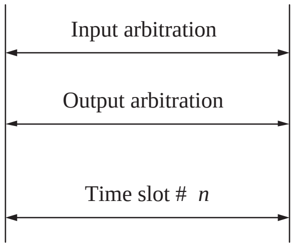

Fig. 2. Allowable arbitration time in a one-cell buffered crossbar switch.

In a similar way as in [9], let Q be a $N \times N$ matrix with each entry being 1. Then

$$C(t) = QZ(t) + Z(t)Q, t \geq 0 \tag{7}$$

where $C_{i,j}$ is an element of $C(t)$.

We define $f(t)$ as:

$$f(t) = \langle Z(t), C(t) \rangle = \sum_{i,j} Z_{i,j}(t) C_{i,j}(t). \tag{8}$$

It follows that $f(t) \geq 0$ for $t \geq 0$ and $f(0) = 0$. It is easy to see that $f(t) = 0$ implies $Z(t) = 0$. Next, we show that $f(t) > 0$ implies $\dot{f}(t) \leq 0$ for almost every t.

In this case, from [9], it follows that

$$\dot{f}(t) = 2 \sum_{i,j} Z_{i,j}(t) \dot{C}_{i,j}(t). \tag{9}$$

Therefore, $\dot{f}(t) \leq 0$ if and only if $\dot{C}_{i,j}(t) < 0$. As mentioned above,

$$\dot{C}_{i,j}(t) = \sum_k \lambda_{i,k} + \sum_k \lambda_{k,j} - 2 \tag{10}$$

where

$$\sum_j \lambda_{i,j} \leq 1,$$

and

$$\sum_i \lambda_{i,j} \leq 1,$$

thus making $\dot{C}_{i,j}(t) \leq 0$. Therefore, from (9) and (10), $\dot{f}(t) \leq 0$ whenever $f(t) > 0$.

Here, $f(t)$ and Fact 1 establish that the fluid model of a CICB switch with one-cell crosspoint buffers is weakly stable as long as no input is inhibited from sending a cell to the buffered crossbar in a time slot. Then, it remains to complete the proof of Theorem 1 with the following lemma.

In (5), the arbitration scheme m selects a VOQ such that an CPB at j receives one cell, as expressed by (6). We state the following lemma about the non-inhibition of an input arbiter:

Lemma 1: At any time slot, input i has at least an available $CPB_{i,j}$ under admissible traffic such that inhibition is avoided.

Proof:

Lemma 1 can be rephrased in terms of the output arbitration scheme, as follows: *There exists an output arbitration scheme such that the selection result causes $\sum_j CPB_{i,j}^{Busy} < N$ for admissible traffic, at any time slot.*

Consider the following propositions, presented in [10]. *Von Neumann proposition: if a matrix $B = (B_{i,j})$ is doubly substochastic, then there exists a doubly stochastic matrix $\bar{B}$ such that $B_{i,j} < \bar{B}_{i,j} \ \forall \ i, j$.*

Birkhoff's proposition: for a doubly stochastic matrix $\bar{B}$, there exists a set of positive numbers ϕ_k and permutation matrices P_k, where $1 \leq k \leq K$, such that $\bar{B} = \sum_k \phi_k P_k$. Let e be the column vector with all its elements being 1. As $\bar{B}$ is doubly stochastic, $e = \bar{B}e{=}\sum_k \phi_k (P_k e){=}(\sum_k \phi_k)e$, thus making $\sum_k \phi_k = 1$.

Note that the occupancy of the 1-cell crosspoint buffers in the buffered crossbar can be represented by a matrix

$$CPB^{Busy} = (CPB_{i,j}^{Busy}) \tag{11}$$

such that

$$\sum_j CPB_{i,j}^{Busy} \leq N \tag{12}$$

and

$$\sum_i CPB_{i,j}^{Busy} \leq N. \tag{13}$$

Normalizing CPB^{Busy} with respect to N, the matrix is doubly substochastic.

Therefore, CPB^{Busy} can be represented as doubly stochastic $\overline{CPB}^{Busy}$, such that there exist permutation matrices that indicate which $CPB_{i,j}^{Busy}$ is served at j in a time slot.

Therefore, the output arbitration scheme must select a set of CPBs such that, for a given P_k

$$\sum_j P_{i,j} > 0, \tag{14}$$

and therefore

$$\sum_j CPB_{i,j}^{Busy} < N \tag{15}$$

after the output arbitration. By using the permutation matrices as the set of CPBs that are selected by the output arbitration, input i has at least one CPB available at any time slot.

Furthermore, because $K \leq N^2 - 2N + 2$ [10], the smallest switch size of $N = 2$ has $K \geq 1$. Therefore, this result holds for all N values.

Since there exists an output arbitration scheme that allows inputs to be uninhibited, Lemma 1 is proved. ■

As Lemma 1 is true, then Theorem 1 is proved. ■

Figure 3 shows an example of a decomposed matrix for a 4×4 switch. Figure 3.a shows a matrix CPB^{Busy} is doubly

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad \text{Doubly sub-stochastic matrix}$$

(a)

$$\begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & 0.5 & 0.25 & 0.25 \\ 0.5 & 0 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} \quad \text{Doubly stochastic matrix}$$

(b)

After decomposition

$$0.25 \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + 0.25 \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$+ 0.25 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + 0.25 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

(c)

Fig. 3. Example of a decomposed matrix for a 4×4 switch.

substochastic according to (12) and (13). Based on the Von Neumann proposition, the normalized matrix CPB^{Busy} is converted to a doubly stochastic matrix. Figure 3.b shows the doubly stochastic matrix. With Birkhoff's proposition, the decomposed matrix is obtained, as shown in Figure 3.c. After the decomposition, four permutation matrices are obtained, where matrix elements with 1's are CPBs selected by the output arbitration. Each permutation matrix has associated the rate (e.g., 0.25) of the crosspoint selection. As shown in this figure CPBs from different inputs are served at every time slot, and therefore input inhibition is avoided.

In the following section, we present input and output arbitration schemes that comply with the conditions analyzed: under admissible traffic, we consider that no two VOQs can have the same occupancy under the same service rate under SLLN admissible traffic. Therefore, the input arbitration scheme guarantees that the $\dot{C}_{i,j}(t)$ decreases by two, and the output arbitration scheme guarantees that an effective matrix decomposition is performed to make one or more crosspoint buffers available per input.

V. Distributed Longest Queue First (LQF) and Centralized Longest Column Occupancy (LCO) First

Input arbitration: we use a distributed input arbitration scheme, longest queue first (LQF), which can differentiate among flows that require extensive service, and that can be applied independently at any input port.

The LQF scheme can be described as: an input arbiter selects the non-empty uninhibited VOQ that has the larger cell occupancy. Ties are broken arbitrarily. The selected VOQ sends a cell to the buffered crossbar in the next time slot.

Output arbitration: since the output arbiters can be placed in-chip at the buffered crossbar, we consider the LCO arbitration scheme as a centralized algorithm. The development of LCO is based on the most critical internal buffer first (MCBF) scheme [11]. However, MCBF cannot guarantee that an input can have an available crosspoint buffer for any time slot (we show an example of this claim in the Appendix). By using LCO, an output arbiter selects CPBs from different inputs and CPBs for all outputs. LCO uses two steps:

Step 1: Select an output $\{j \mid max \sum_i CPB^{Busy}_{i,j}\}$ and an input $\{i \mid max \sum_j CPB^{Busy}_{i,j}\}$. Ties are broken arbitrarily. Set output j and input i as reserved and perform Step 1 with unreserved i, j pairs until no more can be found (i.e., the number of unreserved inputs or outputs becomes zero, or else, when the remaining occupied CPBs belong to reserved inputs or outputs). Then go to step 2.

Step 2: If there are unreserved outputs where at least one CPB is occupied, select an CPB^{Busy} arbitrarily from each unreserved output. Note that a reserved input can be selected in this step.

Figure 4 shows an example of the selection process performed by LCO in a 4×4 buffered crossbar. The rows represent inputs and the columns represent outputs. In Figure 4.a, $CPB_{3,3}$ is selected. Therefore, input 3 and output 3 are reserved. In Figure 4.b, the row selection considers only those busy crosspoint buffers from unreserved rows and columns, and therefore, $CPB_{1,0}$ is selected. Figures 4.c shows the selection of the $CPB_{2,2}$ as only row 2 and column 2 are unreserved and with busy CPBs. Figure 4.d shows that $CPB_{3,1}$ is selected by using LCO's Step 2. This example shows that LCO selects CPBs from different inputs while keeping the buffered crossbar forwarding cells to the outputs. Therefore, LCO avoids input inhibition.

LCO maximizes the number of active outputs and therefore, relaxes the requirements for an input arbitration scheme. The computation complexities (without optimization) of LQF and LCO are $O(N)$ and $O(N^2)$, respectively.

VI. Conclusions

In this paper, we introduced a hybrid approach to arbitration schemes for CICB switches: LQF with a distributed implementation at the inputs and LCO with a centralized implementation at the buffered crossbar. By using these arbitration schemes, we showed that a CICB packet switch can provide 100% throughput, with no speedup, under admissible traffic that follows the strong law of large numbers. The fact that input and output arbitrations are performed separately in a CICB switch allows us to analyze the buffers at the inputs of the CICB switch while the output arbitration at the buffered crossbar keeps inputs uninhibited, and to analyze the buffered crossbar while an input arbitration selects a VOQ that has a crosspoint buffer available. We show that LCO can select cells at crosspoint buffers from different inputs such that input inhibition is avoided.

Because all output arbiters are located in the buffered crossbar, it is possible to use a centralized scheme, as LCO,

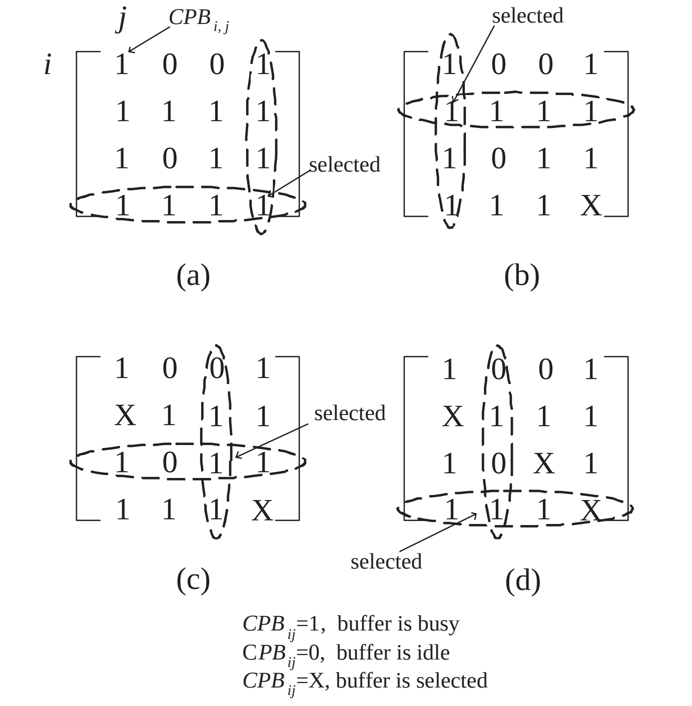

Fig. 4. Example of crosspoint-buffer selection by a centralized LCO in a 4×4 switch.

at the cost of increasing the computation complexity of the arbitration scheme. This complexity increase is considered as a sufficient condition to provide higher switching performance. Although LCO might consume time to perform a suitable selection of crosspoints, the strategic location of the output arbiters permits a short resolution time that keeps CICB switches with, nevertheless, effective timing.

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

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

[4] M. Nabeshima, “Performance Evaluation of Combined Input and Crosspoint-Queued Switch,” *IEICE Trans. on Commun.*, Vol. E83-B, No. 3, March 2000.

[5] K. Yoshigoe, K.J. Christensen, “A Parallel-Polled Virtual Output Queue with a Buffered Crossbar,” in Proc. *IEEE HPSR 2001*, pp. 271-275, May 2001.

[6] R. Rojas-Cessa, E. Oki, Z. Jing, and H.J. Chao, “CIXB-1: Combined Input-One-Cell-Crosspoint Buffered Switch,” in Proc. *IEEE HPSR 2001*, pp. 324-329, May 2001.

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

[8] S-T. Chuang, S. Iyer and N. McKeown, “Practical Algorithm for Performace Guarantees in Buffered Crossbars,” in Proc. *IEEE INFOCOM*, 2005.

[9] J.G. Dai and B. Prabhakar, “The Throughput of Data Switches with and without Speedup,” in Proc. *IEEE INFOCOM 2000*, pp.556-564, March 2000.

[10] C-S. Chang, W-J. Chen and H-Y. Huang “Birkhoff-Von Neumann Input Buffered Crossbar Switches,” in Proc. *IEEE INFOCOM 2000*, pp.1614-1623, March 2000.

[11] 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.

[12] A. Bianco, M. Franceschinis, S. Ghisolfi, A.M. Hill, E. Leonardi, F. Neri, and R. Webb, “Frame-based Matching Algorithms for Input-Queued Switches,” in Proc. *IEEE HPSR 2002*, pp. 69-76, 2002.

Appendix

Here, we show that the MCBF scheme cannot guarantee an available crosspoint buffer for any input at every time slot. In MCBF, each distributed output arbiter independently selects the crosspoint from the longest input occupancy as specified by the longest buffer first (LBF) output arbitration [11]. Therefore, if an input has high load, the associated crosspoints could be selected during the same time slot, while leaving other inputs unserved. Figure 5 shows an example of the selection process that the decentralized MCBF performs in a 4×4 buffered crossbar, which is represented as a matrix. In this matrix, rows represent the inputs and the columns represent the outputs. A CPB with a cell is represented by 1 (busy), and 0 (idle), otherwise. Figure 5.a shows the state of CPBs as busy and idle crosspoints. Figure 5.b shows the input occupancy seen by the output arbiters per CPB. For example, $CPB_{0,0}$ is 2 as there are only two cells from input 0 for all outputs. An idle CPB is indicated by a zero as it is ignored by the output arbiter. This figure also shows that this example has all CPBs from inputs 1 and 3 with the longest input occupancy, and the output arbiters, using the same selection policy, select all CPBs from input 3. Therefore, Figure 5.c shows that $CPB_{3,0}$ to $CPB_{3,3}$ are selected, marked with an X. These independent arbiters select CPBs from input 3 and leave input 1 without an available CPB in the next time slot. Therefore, the distributed MCBF scheme cannot guarantee having an available CPB at any time slot.

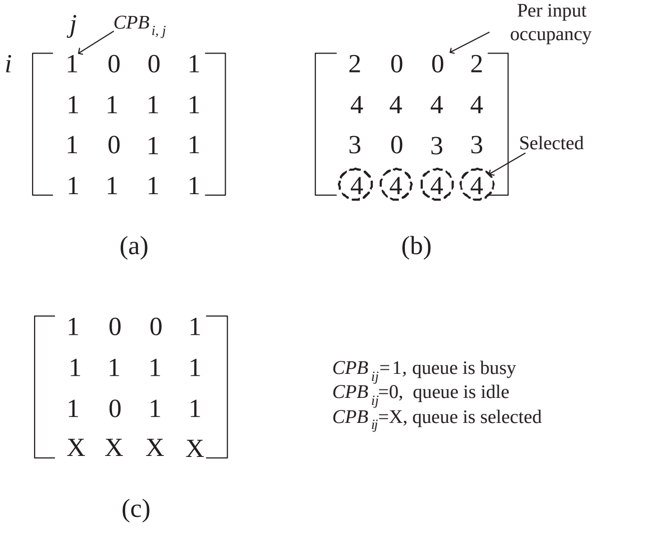

Fig. 5. A counter example of crosspoint-buffer selection by MBCF in a 4×4 switch.