Load-Balanced Combined Input-Crosspoint Buffered Packet Switches

Roberto Rojas-Cessa *Member, IEEE* and Ziqian Dong *Member, IEEE,*

***Abstract*—Combined input-crosspoint buffered (CICB) switches can achieve high switching performance without speedup. However, the dedicated crosspoint buffers in a CICB switch may not be efficiently used, and throughput degradation may occur. This throughput degradation is especially observable under flows with high data rates and long distances between the line cards and the buffered crossbar. This paper introduces two load-balanced CICB switches: the load-balancing CICB switch with full access (LB-CICB-FA) and the load-balancing CICB switch with single access (LB-CICB-SA). The proposed switches use the crosspoint buffers efficiently and support long distances between the line cards and buffered crossbar with crosspoint buffers smaller than those in a CICB switch by a factor of N, where N is the number of ports. It is proven that the LB-CICB-FA switch with random selection of the configuration of the load-balancing stage, input queues, and crosspoint queues is weakly stable under admissible independent and identical distributed (i.i.d.) traffic. Additional simulation results support the correctness of the theoretical analysis. Furthermore, it is shown that the throughput of the LB-CICB-SA switch with the longest-queue first (LQF) and first-come first-served (FCFS) as input and output arbitrations, respectively, is 100% under admissible i.i.d. traffic. The proposed switches keep cells in sequence and use no speedup. The low implementation complexity of the load-balancing stage is discussed and shown to be small.**

***Index Terms*—Buffered crossbar, round-trip time, crosspoint buffer, Birkhoff-von Neumann, load balancing.**

I. Introduction

Combined input-crosspoint buffered (CICB) switches, also known as combined input-crosspoint queued (CICQ) switches, are an alternative to input-queued (IQ) switches to provide high-performance switching for packet switches with high-speed ports. In addition, CICB packet switches use time efficiently as input and output arbitrations are performed separately. It has been shown that crosspoint buffers can provide higher performance than IQ switches with selection schemes of smaller complexity than those used in IQ switches [1]-[19]. These advantages have raised research interest on whether CICB switches with memory speedup can emulate output-queued (OQ) switches [20]-[24]. OQ switches provide optimum switching performance (such as 100% throughput and small queuing delay), however, at the expense of adopting a large speedup. Speedup is the ratio of the speed to transmit a packet in the switch over the transmission speed of the external links, and this is usually equal to N for an OQ switch, where N is the number of ports.

This paper follows the mainstream practice of segmenting incoming variable-length packets into fixed-length packets, called cells, at the ingress side of a switch, and re-assembling the packets at the egress side, before they depart from the switch [2], [4]-[6], [8], [18], [25]-[37]. In addition, this paper considers admissible traffic, which is described as $\sum_{i=0}^{N-1} \rho_{i,j} \leq 1, \forall j$ and $\sum_{j=0}^{N-1} \rho_{i,j} \leq 1, \forall i$ for an $N \times N$ switch, where i is the input port number ($0 \leq i \leq N-1$), j is the output port number ($0 \leq j \leq N-1$), and $\rho_{i,j}$ is the traffic load from input i destined to output j [28].

The high performance of CICB switches comes at the expense of having crosspoint buffers for each input-output pair at the crossbar, where cells are stored before they are forwarded to outputs. Each of the crosspoints has a dedicated buffer that only cells from input i to output j can use. The amount of memory in the buffered crossbar is N^2kL, where N is the number of input and output ports, k is the crosspoint-buffer size in number of cells, and L is the cell size in number of bytes. The value of k is determined by the length of the round-trip time (RTT), which is defined 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 used to avoid buffer overflow, back from the crossbar to the input $d2$ [8]. This relationship is expressed as

$$RTT = d1 + OA + d2 + IA. \qquad (1)$$

In a CICB switch, the required crosspoint-buffer size to avoid underflow by flows of data rate R_c b/s (or cells/time slot), where R_c is the port speed, is determined by:

$$RTT \leq \frac{kL}{R_c}. \qquad (2)$$

For example, if k=1 cell, L=64 bytes, and R_c=155 Mb/s, then the supported RTT is 5.12 μs or 1 time slot. The terms *cell* and *time slot* are indistinguishably used in this paper as the units of R_c because cells have a fixed length.

The crosspoint-buffer size of a CICB switch, k, with dedicated crosspoint buffers [6]-[9], is required to hold $k \geq RTT$ cells to avoid crosspoint-buffer underflow, which causes throughput degradation for flows with high data rates. Here, a data flow$f(i,j)$ is defined as the set of cells arriving in input i and destined to output j where the sequence of cells at arrival must be kept at their departure from the switch. As the

Preliminary results were presented in [13], [14], and [31].

Roberto Rojas-Cessa is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark NJ 07102. E-mail: rojas@njit.edu.

Ziqian Dong is with the Department of Electrical and Computer Engineering, New York Institute of Technology, New York, NY 10023. She was with New Jersey Institute of Technology. E-mail: ziqian.dong@nyit.edu.

line cards (input ports) can be located far from the buffered crossbar, actual RTTs can be long. A long RTT is defined as the number of transmitted cells during a period of time larger than the number of cells a crosspoint buffer can store. If $\frac{L}{R}$=1 time slot, a long RTT is when $RTT > k$. For example, if RTT has a length of two time slots and a crosspoint buffer has a capacity of one cell, RTT is said to be long.

The performance degradation of a 32×32 CICB switch (N=32) with dedicated crosspoint buffers under long RTTs was shown [12]. In that example, different k and RTT values were considered, where $RTT \geq k$. The distances of the input ports and the crossbar were considered equal. The rates of different flows were modeled with the unbalanced traffic model [8], which defines $w + \frac{1-w}{N}$ as the fraction of the input load directed from input i to output j, for 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, for $j \neq i$ (with a uniform distribution). The combination of these two conditions defines the rate of $f(i,j)$ as $r_{f(i,j)}$=$R_c(w + \frac{1-w}{N})$. The maximum data rate of $f(i,j)$ is represented by w=1 or $r^{max}_{f(i,j)}$=R_c, and the minimum data rate is represented by w=0 or $r^{min}_{f(i,j)}$=$\frac{R_c}{N}$. The following observations are based on $r^{max}_{f(i,j)}$ and $r^{min}_{f(i,j)}$ under an input load of 1.0. The throughput of the switch approaches 100% for flows with the minimum rate $r^{min}_{f(i,j)} = \frac{R_c}{N}$ if $RTT < k + N$, including RTT=k. The low rate of the flows reduces the demand for buffer space under long RTTs and that amortizes throughput degradation under long RTTs. The throughput of the CICB switch decreases if RTT further increases to $RTT \geq k + N$. In addition, if RTT is long and constant, the throughput of the switch decreases as the flow rate increases. The throughput of the CICB switch approaches $\frac{k}{RTT}$ when $r_{f(i,j)}$=$r^{max}_{f(i,j)}$=R_c b/s. Ironically, although $r_{f(i,j)} = r^{max}_{f(i,j)}$ is the simplest switching scenario for a switch, it can compromise the performance of a CICB switch. This case can occur as the amount of memory that can be implemented on a chip is limited. Therefore, a CICB switch that supports long RTTs with a small memory is needed.

This paper proposes two buffered-crossbar switches that allow inputs to flexibly access the buffers of different crosspoints to increase buffer utilization without using memory speedup. The increased utilization of the crosspoint buffers benefits the switching performance and supports flows with high data rates when RTT>k. One of the switches allows each input to access any crosspoint buffer. This is called the load-balancing CICB switch with full access to crosspoint buffers or LB-CICB-FA. The other switch allows an input to access a set of crosspoint buffers (one per output) by using a pre-defined configuration of the load-balancing stage. This is called the load-balancing CICB switch with single access to crosspoint buffers or LB-CICB-SA. The LB-CICB-FA switch using a random selection scheme is proven to be weakly stable under admissible independent and identical distributed (i.i.d.) traffic. Simulation results of the analyzed switch model support the correctness of the analysis.

Because the LB-CICB switches are two-stage switches, they provide multiple paths between inputs and outputs. It is shown that these switches transmit cells in sequence to the outputs when the first-come first-serve (FCFS) policy is used as the output arbitration scheme. Furthermore, the switching performance of the proposed switches with the longest-queue first (LQF) and FCFS as input and output arbitration schemes, respectively, is studied under admissible i.i.d. traffic with uniform and nonuniform distributions. The simulation study also includes long RTTs, and the results show that the flexibility to access crosspoint buffers supports N times longer RTTs than those supported by a CICB switch with the same crosspoint-buffer size. The results show that the LB-CICB-SA switch can achieve a comparable performance to that of the LB-CICB-FA switch. A design of the load-balancing stage for the LB-CICB-SA is presented, and it is shown that this stage has low complexity.

The remainder of this paper is organized as follows. Section II discusses related work. Section III introduces the proposed load-balancing CICB switches. Section IV presents the stability analysis of the LB-CICB-FA switch with random selection at the arbiters and the load-balancing stage under i.i.d. traffic. Section V proves that a load-balancing CICB switch with FCFS policy as output arbitration serves cells in sequence. Section VI presents a performance study of the proposed switches under uniform and nonuniform traffic models. Section VII presents the design and complexity analysis of the data and control paths of the load-balancing stage for the LB-CICB-SA switch. Section VIII presents concluding remarks.

II. Related works

To support long RTTs in a buffered-crossbar switch, the size of the crosspoint buffer must follow (2) [10]. One of the first schemes that addresses the issue of long RTTs in CICB switches was proposed to support p traffic classes, where the crosspoint-buffer size is larger than RTT for a single class, and smaller than $p \times RTT$ cells [11]. In this switch, when a single priority traffic is considered, the crosspoint buffer size is equal to or larger than RTT. A memory with crosspoint buffers shared by different inputs was proposed to support long RTTs [12]. This switch can support the RTT supported by a CICB switch with 50% of the memory amount in the buffered crossbar and without performance penalties. A CICB switch that adopts the so-called virtual crosspoint queues (VCQs) in the switch fabric that are shared by the crosspoints was proposed [15]. VCQs are equivalent to VOQs but they are placed at the buffered crossbar. This work shows that the use of VCQs may require a large amount of memory at the buffered crossbar (which is the sum of the memory of the crosspoint buffers plus that in the VCQs) to provide comparable performance to a CICB switch without VCQs. An improved version of a switch with VCQs and exhaustive service to the arbitration scheme was proposed [16]. This switch increases the throughput by 14% of that in the original version by optimizing the arbitration scheme with the same amount of the memory at the crossbar. While these studies focused on memory strategies to support long RTTs, a different approach was centered on increasing the efficiency of the flow-control mechanism [17]. The effect of long RTTs in a CICB switch was also observed in a buffered-crossbar switch with internal variable-length segments [18]. This work showed that high throughput can be achieved with a variable packet-length buffered-crossbar switch. The implementation of input arbiters placed at the buffered crossbar was then proposed [19]. Here, the latency of the information exchange between input and output arbiters is avoided as the arbiters are placed in the same chip. This approach makes the flow control independent of the RTT value. For a long (or small) RTT, this switch would require kN^2L bytes.

In addition to the support of long RTTs, a switch has to provide high throughput under admissible traffic. With this objective, a two-stage load-balanced Birkhoff-von Neumann (BVN) input-queued switch was proposed [29]. The first stage of this switch performs load balancing, and a second stage uses the BVN decomposition method of the traffic-load matrix to configure the switch fabric of an IQ switch. The load-balancing stage attempts to distribute traffic evenly in the switch. This switch provides high throughput for admissible traffic under the requirements of a preceding description of the traffic pattern and large external buffers to store cells between input ports and the crossbar. Inspired by the BVN switch, one of the proposed switches uses a pre-determined connectivity and the other a fully configurable load-balancing stage to allow an input to select any crosspoint buffer. Both switches use small internal buffers. Different from the BVN switch, the switches proposed in this paper do not require knowledge of the traffic distribution in advance.

The proposed switches resolve the forwarding of cells through the configuration of a load-balancing stage, input and output arbitration in a distributed manner, rather than in a centralized manner as previously explored [32]. In addition, the proposed switches show that distributed random arbitration can provide 100% throughput not only for traffic with uniform distribution but also for traffic with nonuniform distribution. This result shows that a weightless-based arbitration selection suffices to achieve such throughput should the crosspoint buffers be efficiently utilized.

III. Proposed Switches

A. *Load-Balancing CICB Switch with Full Access (LB-CICB-FA)*

An $N \times N$ LB-CICB-FA switch has N VOQs in each input port, a fully interconnected stage (FIS) to interconnect one input to any of the N^2 crosspoint buffers, and a buffered crossbar. Figure 1 shows the architecture of the LB-CICB-FA switch. The input ports are also called external inputs, each denoted as EI_i. The outputs of the FIS are called internal outputs, each denoted as IO_l where $0 \leq l \leq N-1$, and they are also called internal inputs of the buffered crossbar (each is also denoted as II_l). The outputs of the buffered crossbar, or output ports, are also called external outputs, each denoted as EO_j. A VOQ, denoted as $VOQ(i,j)$, stores cells from input i that are destined to output j. A crosspoint in the buffered crossbar is denoted as $CP(l,j)$ and the corresponding crosspoint buffer is denoted as $CPB(i,j)$. Here, a crosspoint is not restricted to a one-to-one association with a VOQ as in a CICB switch with dedicated CPBs because the FIS enables input i to access any $CPB(l,j)$. The FIS can be implemented with an N-to-1 multiplexer, denoted as $MUX(l,j)$, per CPB, and the selection of an input and CPB can be resolved through a matching process, such that up to one cell can be written into a CPB in a time slot. The remaining discussion in this paper considers CPBs with a size of one cell, $k=1$, and no memory speedup, unless otherwise stated.

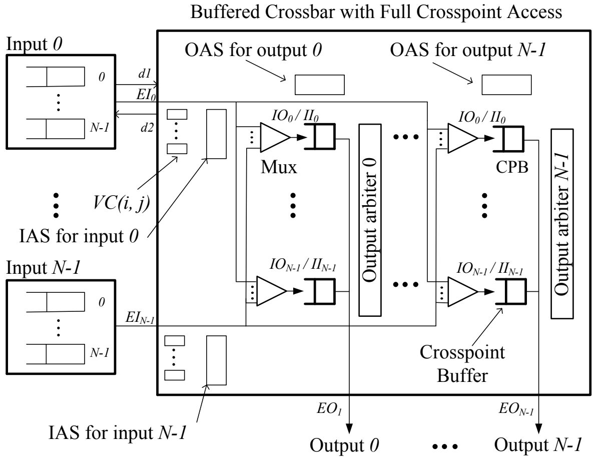


Fig. 1. $N \times N$ Load-Balancing CICB switch with full access (LB-CICB-FA).

There are N^2 VOQ counters (VCs) at the buffered crossbar, denoted as $VC(i,j)$, one per $VOQ(i,j)$. Each VC keeps a count of the number of cells in the corresponding VOQ. There is one output access scheduler (OAS) per EO, denoted as OAS_j, and one input access scheduler (IAS) per EI, denoted as IAS_i, both at the buffered crossbar. IASs and OASs perform a parallel matching to determine which $CPB(l,j)$ receives a cell by selecting a row l for each j. Parallel matching is a distributed process where each IAS sends a request to all those OASs for which it has a cell, each OAS selects a request and grants the selected IAS, and each IAS selects a grant and sends an acceptance to the selected OAS. An IAS generates requests for its associated input based on the values of the VCs and accepts a grant for the input if multiple grants are received. An OAS has a counter $RC(j)$ that counts the number of available CPBs for an output. A CPB is considered available if it has available space for one cell. The number of iterations to perform this match is equal to the minimum of either the number of requests or the value of $RC(j)$. After a matching process, $VC(i,j)$ and $RC(j)$ are updated. Each output has an (output) arbiter that selects a CPB to forward a cell to the output among those occupied.

The LB-CICB-FA switch works as follows. A cell that arrives in input i and is destined to output j is stored in $VOQ(i,j)$. The input sends a notification of the cell arrival to the buffered crossbar, and the corresponding $VC(i,j)$ is increased by one after receiving this notification. In the next time slot, a request is sent from IAS_i to OAS_j. OAS_j selects up to N cells for crosspoints at output j after considering all requests from non-zero VCs and the availability of CPBs. The OAS grants the IAS whose requests are selected. Since an input may be granted access to multiple CPBs at different outputs (i.e., an IAS may receive several grants), the IAS accepts one grant and notifies the granting OASs. The IASs and OASs use either random or LQF as selection schemes in this paper. The random selection is adopted to analyze the stability of the switch, and the LQF scheme is adopted to explore the maximum performance of the LB-CICB-FA switch under uniform and nonuniform traffic patterns using computer simulation. The IAS and OAS can use other selection schemes in addition to random and LQF. The performance of the switch would be determined by the selected scheme. LQF selection is based on the VC values. After a cell is matched to access a CPB, the input is notified by the IAS, and the input sends the selected cell to the CPB in the next time slot. After a cell arrives at the CPB, the corresponding VC is decreased by one.

The output arbiter at output j selects an occupied crosspoint buffer to forward a cell to the output. The selection schemes considered here are random for stability study (Section IV) and FCFS for output arbitration to keep cell in sequence (Section V). The switch uses no speedup.

Figure 2 shows an example of how a 3 × 3 LB-CICB-FA switch works. For simplicity, the FIS is represented as a block between the input ports and the buffered crossbar, and it is assumed that cells are selected in the same time slot they arrive in the CPBs (in the sections about performance analysis and implementation, cells can be selected in the time slot after they arrive). The selected paths that cells follow in the FIS from an input to CPBs are represented as solid lines. Figure 3 shows the matching process performed between the IAS and OAS at each time slot according to the VOQ occupancies in Figure 2.

At time slot t, as shown in Figure 2(a), there are six cells in the VOQs: A, B, C, D, E, and F. The VCs have the following values $VC(0,0)$=2, $VC(0,2)$=1, $VC(1,0)$=1, $VC(2,1)$=1, and $VC(2,2)$=1. Because all CPBs are empty, $RC(0)$=3, $RC(1)$=3, and $RC(2)$=3. The matching process is performed between IASs and OASs as shown in Figure 3(a). In the request phase, IAS_0 sends requests to OAS_0 and OAS_2, IAS_1 sends a request to OAS_0, and IAS_2 sends requests to OAS_1 and OAS_2. In the grant phase, OAS_0 sends grants to both IAS_0 and IAS_1 as it receives two requests and $RC(0)=3$, OAS_1 sends a grant to IAS_2, and OAS_2 sends grants to both IAS_0 and IAS_2. In the accept phase, IAS_0 sends an accept to OAS_0, IAS_1 sends an accept to OAS_0, and IAS_2 sends an accept to OAS_1. Cells A, C, and D are selected for forwarding in the next time slot. The corresponding VCs and RCs are updated to $VC(0,0)$=1, $VC(1,0)$=0, $VC(2,1)$=0, $RC(0)$=1, $RC(1)$=2, and $RC(2)$=3. The configuration of the interconnection stage is decided by the matching between IASs and OASs and on the selection of the CPBs. Here, available CPBs are selected randomly.

At time slot $t+1$, as shown in Figure 2(b), Cells A, C, and D are forwarded to $CPB(0,0)$, $CPB(1,0)$, and $CPB(2,1)$, respectively, where the FIS is configured to interconnect EI_0 to IO_0, EI_1 to IO_1, and EI_2 to IO_2. Matching for this time slot is performed as shown in Figure 3(b), and Cells B and E are selected. EI_0 is interconnected to IO_2 and EI_2 is interconnected to IO_1. Output arbiters perform FCFS scheduling to select cells to be forwarded to the output ports. Here, Cells A and D are selected to be forwarded to Outputs 0 and 1, respectively. These selections empty $CPB(0,0)$ and $CPB(2,1)$. The corresponding VCs and RCs are updated, $VC(0,0)$=0, $VC(2,2)$=0, $RC(0)$=1, $RC(1)$=3, and $RC(2)$=2.

At time slot $t+2$, as shown in Figure 2(c), Cells B and E are forwarded to $CPB(2,0)$ and $CPB(1,2)$, respectively, as the FIS interconnects EI_0 to IO_2 and EI_2 to IO_1. The matching for this time slot is performed as shown in Figure 3(c), where Cell F is selected (see Figure 2(c)). Cells A and D are forwarded to the output port. Output arbiters select Cells C and E for forwarding in the next time slot. The corresponding VCs and RCs are updated, $VC(0,2)$=0, $RC(0)$=2, $RC(1)$=3, and $RC(2)$=2.

At time slot $t+3$, as shown in Figure 2(d), Cell F is forwarded to $CPB(0,2)$ as EI_0 is interconnected to IO_0. Cells C and E are forwarded to Outputs 0 and 2, respectively.

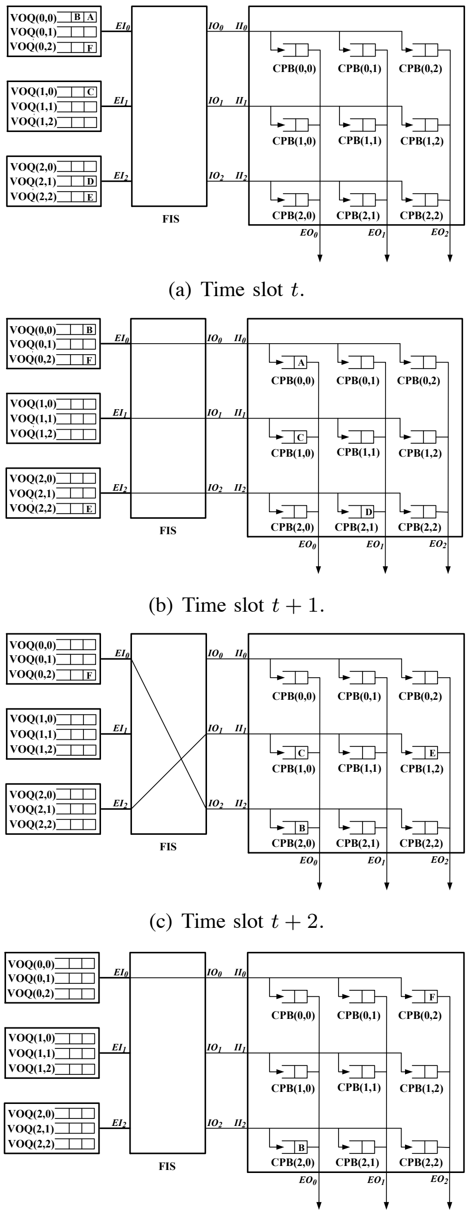
B. Load-Balancing CICB Switch with Single Access (LB-CICB-SA)

Although the LB-CICB-FA switch can fully utilize the CPBs of the buffered crossbar, the hardware complexity might be high. As an approach to a simpler switch that can also flexibly utilize the CPBs for an output, the LB-CICB-SA switch is proposed. The LB-CICB-SA switch has a simple load-balancing stage (LBS) that uses pre-determined and cyclic configurations, VOQs at the inputs, and a buffered crossbar. Figure 4 shows the LB-CICB-SA switch. As in the LB-CICB-FA, the terms used are external input (EI_i) for an input port, internal output (IO_l) for an output of LBS, where $0 \leq l \leq N-1$, internal input (II_l) for an input of the buffered crossbar, and external output (EO_j) for an output of the buffered crossbar. A crosspoint in the buffered crossbar that connects II_l to EO_j, is denoted as $CP(l,j)$ and the crosspoint buffer as $CPB(l,j)$.

As in the LB-CICB-FA switch, there are N virtual counters, denoted as $VC(i,j)$, one for each input at the LBS in the LB-CICB-SA switch. In each EI, there is an input arbiter. In II_l, there is one crosspoint-access scheduler (CAS_l) to schedule the access to CPB_l (via II_l) for input i. Here, CPB_l represents the row of CPBs at II_l. A CAS and the input arbiter at EI_i selects a CPB and a VOQ, respectively, using LQF selection.

The LB-CICB-SA switch works as follows. At time t, the configuration of the LBS interconnects EI_i to II_l by using $l = (i+t)$ modulo N. At EI_i, a cell destined to output j arrives at $VOQ(i,j)$ and sends a notification to $VC(i,j)$ indicating the arrival. At the beginning of each time slot, each input arbiter sends a request to CAS_l as assigned by the configuration of the LBS. CAS_l selects a request from the non-empty VOQ with the longest occupancy for the available $CPB(l,j)$ and the inputs are notified.

The input dispatches the selected cell to the CPB in the next time slot. After that, the cell traverses the interconnecting stage and is stored at the CPB, and the corresponding VC is decremented by one. A cell going from EI_i to EO_j may enter the buffered crossbar through II_l and be stored in $CPB(l,j)$. Cells leave EO_j after being selected by the output arbiter.



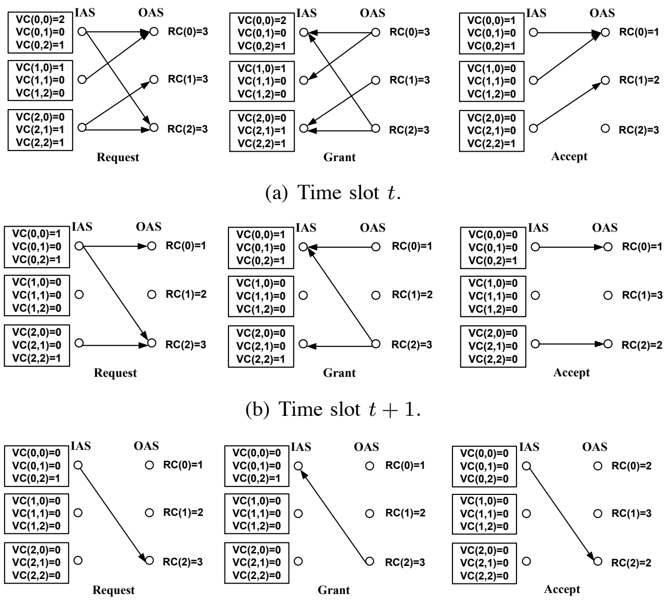
(a) Time slot t.

(b) Time slot $t+1$.

(c) Time slot $t+2$.

(d) Time slot $t+3$.

Fig. 2. Example of a 3 × 3 LB-CICB-FA switch.



(a) Time slot t.

(b) Time slot $t+1$.

(c) Time slot $t+2$.

Fig. 3. Matching process between IAS and OAS for the example of 3 × 3 LB-CICB-FA switch in Figure 2.

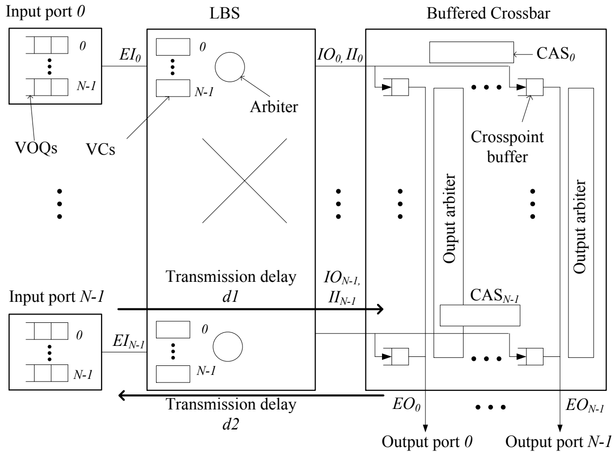


Fig. 4. $N \times N$ Load-balancing CICB switch with single access (LB-CICB-SA).

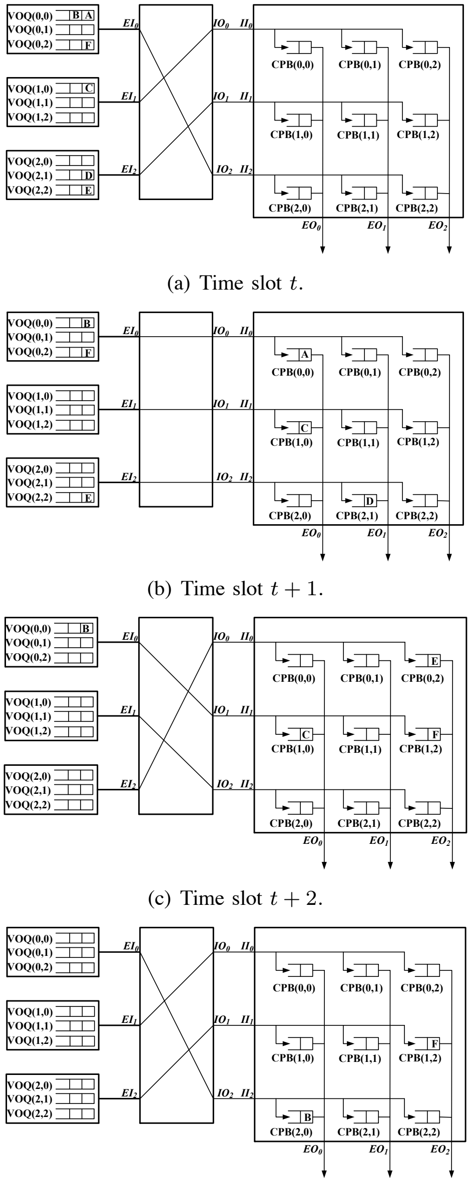
As in LB-CICB-FA, the output arbiters in LB-CICB-SA also use FCFS selection to arbitrate the forwarding of cells of flow $f(i, j)$ to output j. [1]

Figure 5 shows an example of how a 3 × 3 LB-CICB-SA switch works. The scheduling of cells takes place one time slot before the designated data path configuration is set up. At time slot t, the LBS is configured as shown in Figure 5(a). However, since this is the first time slot, no cell is scheduled to use this configuration. The scheduling scheme considers the LBS configuration of the next time slot where EI_0 is interconnected to II_0, EI_1 is interconnected to II_1, and EI_2 is interconnected to II_2, as shown in 5(b). Since all crosspoint buffers are empty, CAS_0 selects the head-of-line (HoL) cell of the longest queue, or Cell A. At the same time, CAS_1 selects Cell C (as there is no other cell in Input 1), and CAS_2 selects Cell D. This selection is arbitrary as the lengths of $VOQ(2, 1)$ and $VOQ(2, 2)$ are both equal to one cell.

At time slot $t+1$, as shown in Figure 5(b), the selected cells, A, C, and D are forwarded to $CPB(0, 0)$, $CPB(1, 0)$, and $CPB(2, 1)$, respectively. In this time slot, CAS_0 selects Cell E and CAS_1 (arbitrarily) selects Cell F to be forwarded in the next time slot. Output arbiters at Outputs 0 and 1 select Cell A and Cell D to be forwarded to Outputs 0 and 1, respectively.

At time slot $t + 2$, as shown in Figure 5(c), Cells E and F are forwarded to $CPB(0, 2)$ and $CPB(1, 2)$, respectively. In this time slot, CAS_2 selects Cell B to be forwarded in the next time slot. Cells A and D are forwarded to the output ports as scheduled. Output arbiters at Outputs 0 and 2 select Cells C and E to be forwarded to Outputs 0 and 2, respectively, in the next time slot.

[1] Section V analyzes the mechanism to keep cells in sequence.



(a) Time slot t.

(b) Time slot $t + 1$.

(c) Time slot $t + 2$.

(d) Time slot $t + 3$.

Fig. 5. Example of a 3×3 LB-CICB-SA switch.

At time slot $t + 3$, as shown in Figure 5(d), Cell B is forwarded to $CPB(0, 2)$. Cells C and E are forwarded to Outputs 0 and 2, respectively. Output arbiter at Output 2 selects Cell F to be forwarded in the next time slot.

IV. Stability Analysis

The flexibility to access any crosspoint buffer by the LB-CICB-FA switch is considered to study the maximum achievable performance of this switch using a modest policy as arbitration schemes. This section presents a stability analysis of the LB-CICB-FA. The following conditions are considered in the analysis:

- The incoming traffic at the inputs is i.i.d.
- The arrivals at each input port and crosspoint buffer are Poisson processes.
- The selection of a non-empty VOQ at an input and the selection of a non-empty CPB per output are performed using a random selection.
- A CPB where an input forwards a cell of the selected VOQ is randomly selected. This is shown in Figure 6(a).

The performance of the LB-CICB-FA switch can be stated in the following theorem:

Theorem 1: The LB-CICB-FA switch represented by the set of VOQs, where inputs under exogenous arrival processes can be assigned to a CPB, in the set $CPB(l, j)$ $\forall\, 0 \leq l, j \leq N-1$, randomly with uniform distribution among j, is weakly stable.

Proof:

Under a stationary exogenous arrival process A_n, a system of queues is weakly stable if, for every $\epsilon > 0$, there exists $B > 0$ such that $\lim_{n\to\infty} P\{||X_n|| > B\} < \epsilon$, where P_E denotes the probability of event E [33]. Weak stability implies rate stability where queue sizes are allowed to grow indefinitely with sub-linear rate.

The selection of a VOQ, a CPB, and the configuration of the LBS follow a random selection policy. This policy is selected because of its analyzable properties, despite its expected modest performance. Other selection schemes at the input and outputs can also be used to achieve higher performance than that achieved by random selection (see Section VI). In this section, the following notations are used in the analysis:

- ρ_s - input load of the switch, $0 \leq \rho_s \leq 1$.
- $\lambda_{i,j}$ - average arrival rate of flow $f(i, j)$.
- $\lambda^X_{i,j}$ - average arrival rate at $CPB(i, j)$.
- $\mu_{i,j}$ - average service rate for $CPB(i, j)$.
- P_{si} - state probability that there are i cells in the queue.
- $P_{i,j}$ - transition probability from state i to state j, in other words, the transition probability from the state where there are i cells in the queue to the state that there are j cells in the queue.
- n - number of cells in a VOQ.
- k - CPB size.

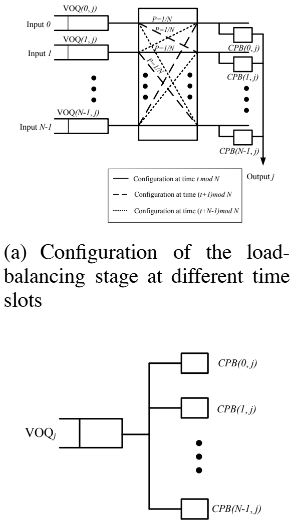
A VOQ is modeled as an $M/M/1$ queue as the arrivals are Poisson processes and the service times received is exponentially distributed. Because arrivals are i.i.d., the VOQs of all N inputs for output j can be represented as a superposed Markov process with aggregated arrival rate

$$\lambda_j = \sum_{i=0}^{N-1} \lambda_{i,j}. \tag{3}$$

This aggregated queue is represented as VOQ_j. VOQ_j can access all N CPBs for output j N times at each time slot. Therefore, it can be modeled as an $M/M/N$ queue, as Figure 6(b) shows. Here, $\rho = \frac{\lambda_j}{N\mu^I}$, where μ^I is the service rate of the aggregated $M/M/N$ queue. The steady state probability of n cells in VOQ_j is represented as P_{sn}, which is calculated using the following equations [34]:

$$P_{sn} = \begin{cases} P_{s0}\frac{(N\rho)^n}{n!}, n \leq N \\ P_{s0}\frac{\rho^n N^N}{N!}, n > N \end{cases} \tag{4}$$

$$P_{s0} = \left[\sum_{n=0}^{N-1} \frac{(N\rho)^n}{n!} + \frac{(N\rho)^N}{N!(1-\rho)}\right]^{-1}. \tag{5}$$



(a) Configuration of the load-balancing stage at different time slots

(b) $M/M/N$ queue.

Fig. 6. $M/M/N$ queuing model of the LB-CICB-FA switch.

From (4) and (5),

$$P_{sn}=\begin{cases}\left[\sum_{n=0}^{N-1}\frac{(N\rho)^n}{n!}+\frac{(N\rho)^N}{N!(1-\rho)}\right]^{-1}\frac{(N\rho)^n}{n!}, & n\le N\\ \left[\sum_{n=0}^{N-1}\frac{(N\rho)^n}{n!}+\frac{(N\rho)^N}{N!(1-\rho)}\right]^{-1}\frac{\rho^n N^N}{N!}, & n>N\end{cases}\tag{6}$$

The service rate of the $M/M/N$ queue μ^I is determined by the availability of the CPBs for output j.

The state probabilities that there are n cells in a VOQ can be written as

$$P_{sn}=\begin{cases}\left[\prod_{t=1}^{n}\frac{\rho_s\cdot\lambda_j}{t\cdot\mu^I}\right]\cdot P_{s0}, & n\le N\\ \left[\frac{1}{N!}\cdot\left(\frac{\lambda_j}{\mu^I}\right)^N\right]\left[\prod_{t=N+1}^{n}\frac{\lambda_j}{N\cdot\mu^I}\right]\cdot P_{s0}, & n>N\end{cases}\tag{7}$$

$$P_{s0}=\left[\sum_{t=1}^{N-1}\frac{1}{t!}\cdot\left(\frac{\lambda_j}{\mu^I}\right)^t+\sum_{t=N}^{n}\frac{1}{N!N^{t-N}}\left(\frac{\lambda_j}{\mu^I}\right)^t\right]^{-1}\tag{8}$$

Each crosspoint buffer $CPB(i,j)$ is modeled as an $M/M/1/k$ queue. Because one of the motivations is to use small crosspoint buffers, the LB-CICB-FA switch is set with k=1. The average arrival rate at each CPB after the LBS is:

$$\lambda_{i,j}^X=\sum_{i=0}^{N-1}\lambda_{i,j}\frac{1}{N}.\tag{9}$$

The probability that $CPB(i,j)$ is available is calculated using the $M/M/1$ queuing model. The superscript X in the following terms is used to represent the variables in regards to the CPBs. The following probabilities are then defined:

P_{ij}^X - transition probability from state i to state j.

P_{si}^X - state probability that there are i cells in the CPB. Here,

$$P_{01}^X=\sum_{i=0}^{N-1}\frac{1}{N}\cdot\frac{\lambda_{i,j}}{\sum_{j=0}^{N-1}\lambda_{i,j}}\cdot\rho\cdot\lambda_{i,j}\tag{10}$$

or

$$P_{01}^X=\frac{1}{N}\tag{11}$$

Because the output scheduler chooses a CPB with probability $\frac{1}{N}$,

$$P_{10}^X=\frac{1}{N}.\tag{12}$$

From

$$\begin{cases}P_{01}^X P_{S0}^X=P_{10}^X P_{S1}^X;\\ P_{S0}^X+P_{S1}^X=1;\end{cases}\tag{13}$$

$$P_{s0}^X=\frac{P_{10}^X}{P_{10}^X+P_{01}^X}\tag{14}$$

The arrival to each CPB after the LBS is $\lambda_{i,j}^X=\sum_{i=0}^{N-1}\lambda_{i,j}\cdot\frac{1}{N}$, as stated before. Then, (14) is $P_{S0}^X=\frac{1}{2}$.

The service rate of the aggregated queue can be approximated by the state probability when the CPB is available or there is no cell in the CPB, $\mu^I=P_{s0}^X$.

Under admissible i.i.d. traffic, $\sum_{i=0}^{N-1}\lambda_{i,j}\le 1$. Therefore, $\lambda_{i,j}^X\le\frac{1}{N}$, $\lambda_j\le 1$, $\rho_{max}=\frac{\lambda_j}{N\mu^I}=\frac{2}{N}$.

From (7):

$$P_{sn}=\frac{N^{N-n}}{N!\left[\sum_{t=0}^{N-1}\frac{2^{N-n}}{t!}+\frac{2^{N-n}N}{N!(N-2)}\right]}\tag{15}$$

As $n\to\infty$,

$$\lim_{n\to\infty}P_{sn}=0.\tag{16}$$

The VOQ length n converges to ε, where $\varepsilon<\infty$, $\lim_{n\to\infty}P\{P_{sn}>B\}<\varepsilon$. Therefore, the weakly stable condition of the system of queues is met. It is proven that the LB-CICB-FA switch, with random selection, is weakly stable under admissible i.i.d. traffic.

■

The LB-CICB-FA switch, as described in the analysis, was modeled in a C-language event-driven simulator to experimentally observe the stability of the switch in terms of throughput. The simulation results of the switches with $N=\{16,32,64\}$ show 100% throughput under i.i.d. traffic. The destinations of the simulated traffic had uniform and nonuniform distributions, including unbalanced and diagonal. These results are consistent with those of the theoretical analysis presented above. In addition, the average queuing delay and the 99.9% queuing delay, defined as $P[D<delay]=1E-3$, of this switch were evaluated under an input load of 0.99. The average queuing

delay is 50 time slots and the 99.9% delay is 77 time slots, which indicates that the worst-case delay is finite (i.e., 100% throughput) and of similar order of magnitude to that of the average queuing delay.

V. In-Sequence Cell Service

This section considers $k \geq 1$ for generality in the following analysis. An additional advantage of using a buffered crossbar is that the cells are stored in one chip (i.e., crosspoint buffers), and this permits time synchronization of cells in the crossbar. It is then possible to place a stamp of the arriving time slot to indicate the order in which cells arrive (this is similar to using a timestamp, however, without the complexity of keeping synchronization with an external clock) in the buffered crossbar. For the sake of clarity in the following discussion, the arrival time slot number is called the timestamp. Because the timestamps of all buffered cells use the same clock, a simple output arbitration scheme to keep cells in sequence can be used. This is discussed by the following theorem.

Theorem 2: Cells of a flow are served to their destined outputs in the order they come into the buffered crossbar by a FCFS output arbiter.

Proof:

The following labels are appended to a cell C for identification. Some labels may be used for actual implementation.

t	is the time at which cell C arrives in the buffered crossbar.
$C(i,l,j,t)$	is the identification of cell C that includes the input, crosspoint buffer, output indexes, and the arrival time of C.

The arrival time t is used as the sequence serving order. It is assigned to a cell at the time the cell arrives at the buffered crossbar.

With these conditions, the following facts are listed:

- **Fact 1.** One and only one cell arrives at each input each time slot.
- **Fact 2.** Each input dispatches at most one cell each time slot.
- **Fact 3.** Each internal input II assigns timestamp t to a cell at arrival.
- **Fact 4.** One and only one cell arrives in $CPB(l,j)$ each time slot.

The output arbiter at output j considers the label t of up to N HoL cells to perform FCFS selection. Ties are broken arbitrarily. Because the order in which cells depart from an output depends on the order they arrive at the HoL position in a CPB, the processes involved are the arrival of cells into the queue and the relative position of cells from the same flow in different CPBs for output j. Therefore, Theorem 2 is partitioned into the following lemmas.

Lemma 1: $C1(i,l,j,t_x)$ always arrives at $CPB(l,j)$ before $C2(i,l,j,t_y)$ if $t_x < t_y$, where $t_a < t_b$ means that t_a is an earlier time slot than t_b.

Proof: In this case, cells of the same flow are of interest. The cells departing from input i are required to arrive at the buffered crossbar in the sequence they came into the switch. The departure of cells from $VOQ(i,j)$ can only occur at the rate of up to one cell per time slot, and these cells are served in a first-in first-out manner. Because of Fact 1, $C1$ is placed closer to the HoL position in the VOQ than $C2$, thus $C1$ is served before $C2$. The arrival of two different cells into $CPB(l,j)$ can occur only in two time slots. Therefore, cell $C1$ is assigned timestamp t_x and cell $C2$ is assigned timestamp t_y, where $t_x < t_y$.

■

Because cells of a flow are allowed to traverse the buffered crossbar through multiple IIs toward their destined outputs, these cells may go through different queue lengths (i.e., different queuing delays). As cells may encounter different queue lengths, the following lemma states that the relative order of cells in a CPB from different inputs follow their arrival order. That is, cells that arrived first are placed closer to the HoL position in a CPB than the subsequent cells, independent of their originating inputs.

Lemma 2: If $C1(i,l,j,t_x)$ and $C2(i',l,j,t_y)$ are such that $t_x < t_y$, then $C1$ is always placed closer to the HoL position than $C2$ at $CPB(l,j)$.

This lemma is similar to Lemma 1, with, however, two cells that belong to different flows but concur at the same CPB. The lemma shows that the relative order in which cells arrive in the buffered crossbar is kept at the CPB, independent of the originating input.

Proof: Because of Facts 2 and 4, either $C1$ or $C2$ arrives first to $CPB(l,j)$. As $CPB(l,j)$ adopts the FCFS policy, a cell that comes first into the queue is placed closer to the HoL position than any other subsequent cell. Therefore, $C2$ can be placed ahead of $C1$ only when $C2$ arrives in the buffered crossbar before $C1$. Since, $t_y > t_x$ the initial condition contradicts this assumption.

■

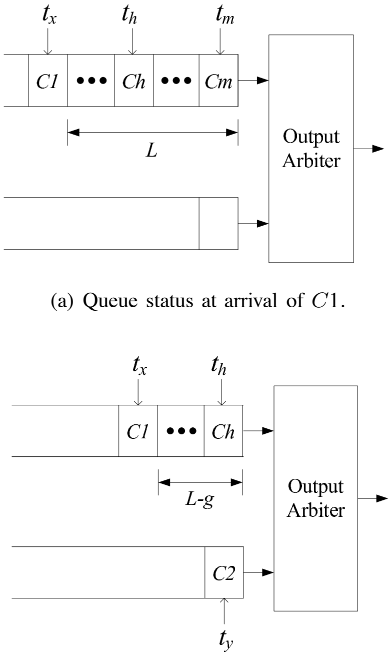
Because Lemmas 1 and 2 indicate that the relative order of the cells in one CPB is always ascending, analysis of the serving order of two different CPBs is all that remains.

Lemma 3: $C1(i,l,j,t_x)$ in $CPB(l,j)$ is served before $C2(i',l',j,t_y)$ at $CPB(l',j)$ for $i = i'$ or $i \neq i'$ and $l \neq l'$, if $t_x < t_y$ by a FCFS arbiter, independent of their positions in any queues for output j.

Proof: Let us assume that $CPB(l,j)$, which $C1$ is assigned to, has L backlogged cells at time t_x as an initial condition, and $CPB(l',j)$, which $C2$ is assigned to, is empty at t_y as an initial condition, such that $C2$ is placed at the HoL position at arrival. Figure 7 shows the position of cells $C1$ and $C2$, where (a) shows a backlog of L cells for $C1$ and (b) shows the empty queue for $C2$. In this situation, $C2$ is given the apparent advantage to depart before $C1$. Because $t_x < t_y$, $t_y - t_x = g$, such that $L - g = t_h$, where t_h is the timestamp of the HoL cell at $CPB(l,j)$ at time slot t_y, which is the time when $C2$ arrives in $CPB(l',j)$, $C1$ is in position $L - g$ from the HoL. Therefore, the timestamps of the HoL cells at time t_y are t_h and t_y. Because of the order of the cells, it is clear that $t_y = (L - g) + t_x$, and since $t_h = L - g$, therefore $t_y = t_h + t_x$, which indicates that $t_h < t_y$. The cell that arrived at t_h is then selected by the FCFS arbiter before $C2$, which arrived at t_y, $\forall t_h$. Since the arrival of $C1$ is earlier than t_y, then $C1$ is selected before $C2$ for dispatching to the output port.

■

Since Lemmas 1, 2, and 3 are proven. Theorem 1 is also proven. ■



(a) Queue status at arrival of $C1$.

(b) Queue status at arrival of $C2$.

Fig. 7. Placement of $C1$ and $C2$ as described in Lemma 3.

VI. Performance Evaluation

The performance of LB-CICB-FA and LB-CICB-SA switches with LQF selection in the input, OASs, and CAS arbiters and FCFS selection in the output arbiters were tested using event-driven simulation, where simulators were written in C language. The evaluation is presented in terms of throughput and average cell delay (in time slots), this latter with a confidence interval of 95%. The considered traffic is uniform with Bernoulli and bursty (i.e., Markov modulated on-off traffic) arrivals and nonuniform with Bernoulli arrivals. Nonuniform traffic models include unbalanced, Power-of-Two (PO2) [35], and diagonal.

Uniform Traffic. The CICB, LB-CICB-FA, and LB-CICB-SA switches were simulated under uniform traffic with Bernoulli and bursty arrivals to observe their performance with minimum memory, or k=1, and RTT=1. An OQ switch was also simulated for comparison purposes under Bernoulli traffic. Bursty traffic is modeled as a modulated Markov On-Off model. Figure 8 shows the average cell delay of a CICB switch, the LB-CICB-SA, and LB-CICB-FA switches, all under uniform traffic. The CICB switch uses LQF policy as the input arbitration scheme and FCFS policy as the output arbitration scheme. The LB-CICB-FA and LB-CICB-SA switches also use LQF but for scheduling access to crosspoint buffers, so this is analogous to using LQF as input arbitration in the CICB switch. The average cell delay only considers the queuing delay. This figure shows that all three switches achieve 100% throughput and similar average cell delay to that of an OQ switch under large loads.

For low input loads, the CICB switch shows smaller average cell delay than the proposed switches. This is because in the LB-CICB-SA and LB-CICB-FA switches, cells spend an extra time slot at the VOQs as their requests are sent to the crosspoint-access scheduler and grants are received (i.e., RTT=1) before being forwarded. This delay is small in any case. Under larger input loads, when the average cell delay is larger than one time slot, the average delays of all switches are similar. These results indicate that the scheduling process for access has no measurable effect on the switching performance. The figure also shows that the average delay of all switches under bursty traffic with average burst lengths $l = \{10, 100\}$ increases in proportion to the average burst length. The 99.9% queuing delays of the LB-CICB-SA and LB-CICB-FA switches were evaluated under an input load of 0.99 and Bernoulli uniform traffic. The 99.9% queuing delay of the LB-CICB-SA switch is 85 time slots and that of the LB-CICB-FA is 73 time slots under this traffic model.

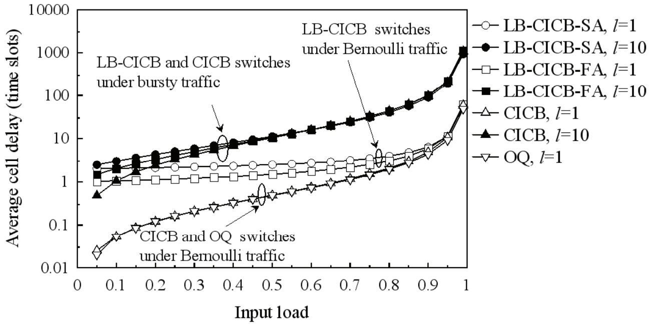


Fig. 8. Average queuing delay of a 32 × 32 LB-CICB-FA, LB-CICB-SA, and CICB switches under uniform traffic.

Nonuniform Traffic: Unbalanced. The LB-CICB switches were simulated under the unbalanced traffic model and $RTT \leq k$ to observe their switching performance (with small RTT) and the effect of long RTTs on their throughput. Figure 9 shows the throughput performance of the CICB, LB-CICB-SA, and LB-CICB-FA switches when k =1 for different RTTs. When RTT≤1, all switches achieve close to 100% throughput under this traffic pattern. This result is consistent with the throughput of CICB switches using a weight-based arbitration [9].

When RTT is large, RTT>k, the throughput of the CICB switch degrades as w increases. However, the LB-CICB-SA and LB-CICB-FA switches achieve high throughput despite the increase of RTT and w. The throughput of the proposed switches is below 99% for values of w between 0.3 and 0.7. This is produced by the combination of the traffic distribution of this model under those w values and the high flow rates, which are served using the FCFS policy at the outputs, and a small k. However, the throughput remains high when w=1, which is the case for flows with a port-speed rate. In contrast, the throughput of a CICB switch degrades to $\frac{k}{RTT}$. The throughput above 99% for flows with port-speed rates shows that the LB-CICB-FA and LB-CICB-SA switches support up to RTT=32. This RTT is N times larger than that supported by a CICB switch with k=1.

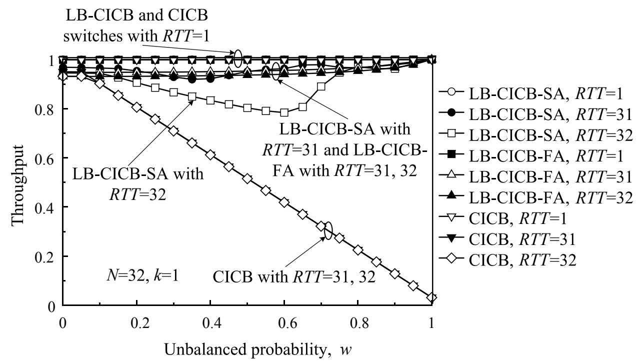


Fig. 9. Throughput of 32×32 LB-CICB-FA, LB-CICB-SA, and CICB switches with k=1 and $RTT = \{1, 31, 32\}$ under unbalanced traffic.

Figures 10 and 11 show the throughput performance of the LB-CICB-SA and LB-CICB-FA switches, respectively, with k=1 under different RTT values. Figure 10 shows that the LB-CICB-FA switch achieves close to 100% throughput for $RTT \leq 25$. The throughput is the lowest when w=0 (i.e., uniform distribution) or for flows with low data rates. For larger RTT values, the throughput falls below 99%.

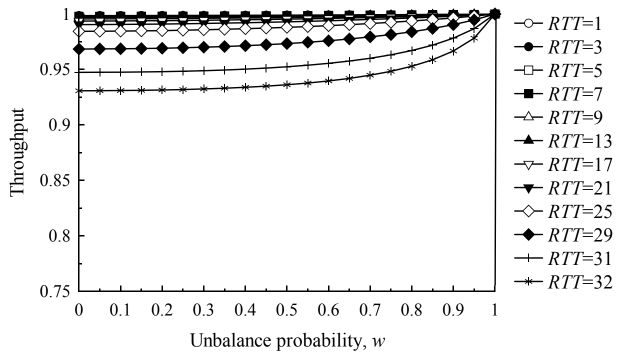


Fig. 10. Throughput of the 32×32 LB-CICB-FA switch with k=1 under unbalanced traffic.

As Figure 11 shows, the throughput of the LB-CICB-SA switch approaches 100% as $RTT \leq 30$. This throughput is higher than that achieved by the LB-CICB-FA switch, because the LB-CICB-FA switch faces more contention than the LB-CICB-SA switch and that affects the throughput. However, the throughput of the LB-CICB-FA switch deteriorates at a slower rate than that of the LB-CICB-SA switch for $RTT \geq 31$. The contention for crosspoints by the unbalanced portion of traffic (i.e., $j = i$) and the uniform portion of traffic (i.e., $j \neq i$) plus the deterministic configuration of the load-balancing stage under long RTTs (e.g., $RTT \geq 31$) causes throughput degradation, where the lowest point is at $w = 0.6$, because the uniform traffic portion is significantly large when compared to the portion of unbalanced traffic. As the portion of uniform traffic decreases as w increases, the throughput increases as contention for crosspoint buffers decreases.

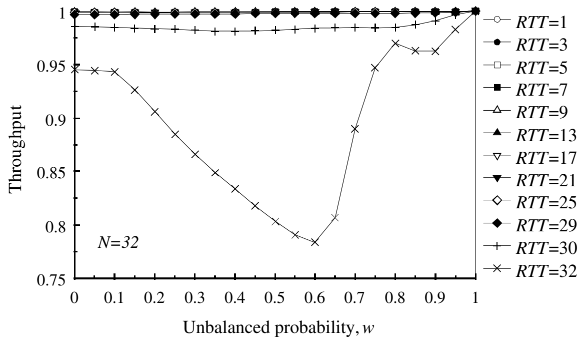


Fig. 11. Throughput of the 32×32 LB-CICB-SA switch with k=1 under unbalanced traffic.

Nonuniform Traffic: PO2. The LB-CICB-SA and LB-CICB-FA switches were simulated under PO2 traffic for 30×30 switches. The PO2 traffic model is represented as $\rho_{i,j} = 1/2^{i+j+1}\rho_i$ for $i + j < N - 1$ and $\rho_{i,j} = 1/2^{i+j-1}\rho_i$ for $i+j \leq N-1$. This traffic model presents a large nonuniformity degree of the traffic distribution among the N output ports. Although the total input load per input or output is smaller than the port capacity in this traffic model, it is difficult for a switch to achieve high throughput. Figure 12 shows that the LB-CICB-SA and LB-CICB-FA switches deliver 100% throughput under this traffic pattern for RTT=1 and k=1. This figure shows that the maximum throughput of the CICB switch is 85%. The average delay of the CICB switch increases rapidly for input loads larger than or equal to 0.825, and cell loss occurs for input loads larger than 0.85. The performance of the CICB switch decreases at this input load as the limited number of crosspoint buffers is mostly used by the flows with the largest rates because LQF selection is used as the input arbitration. The average cell delay of the LB-CICB-SA and LB-CICB-FA switches are equivalent under high input loads (under low input loads, the difference is small), and they resemble the low average cell delay achieved under uniform traffic. The load-balancing stages and flexible access to crosspoint buffers improve the performance of the proposed switches under this traffic model. Long RTTs are not considered under this traffic model because of the limited N this traffic model allows.

The 99.9% queuing delays of the LB-CICB-SA and LB-CICB-FA switches were evaluated under an input load of 0.99 with PO2 traffic. The 99.9% queuing delay of the LB-CICB-SA switch is 130 time slots and that of the LB-CICB-FA is 65 time slots under this traffic model.

Nonuniform traffic: Diagonal. Diagonal traffic can be represented as $d\rho(i,j) = d\rho_i$ for $i = j$, $(1-d)\rho_i$ for j=$(i+1)$ modulo N, where ρ_i is the load at input i. This traffic model presents load distributions among two outputs per input. The distributions are given by the diagonal degree probability, d. Figure 13 shows the switching performance of LB-CICB-FA and LB-CICB-SA switches under diagonal traffic for $0 \leq d \leq 1$.

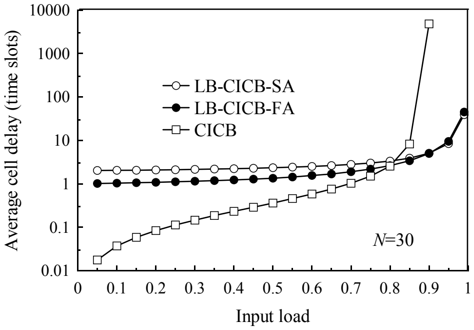


Fig. 12. Average cell delay of 30×30 switches with $k = 1$ under PO2 traffic.

For the small RTT, i.e., RTT=1, the throughput of all three switches is 100%. When RTT=31, the throughput of CICB is close to $\frac{1}{31}$, and the throughput of LB-CICB-FA and LB-CICB-SA remains at 100%. When RTT=32, the throughput of LB-CICB-FA remains close to 100%, but the throughput of LB-CICB-SA decreases to 80%. This performance degradation is related to the heavy nonuniform distribution of the traffic and the predetermined configuration of the LBS. Therefore, the LB-CICB-FA switch supports RTT=kN, and the LB-CICB-SA supports RTT<kN. Furthermore, when d=1, flows have port-speed rates and the throughput of proposed switches is 100% for RTT=32. This traffic model, together with the predetermined assignment of connections between the inputs and a row of crosspoint buffers by the load-balancing stage, may secure high utilization of a crosspoint buffer, which is accessed by the two inputs sending traffic to an output. In addition, the FCFS service for that output allows to serve cells continuously from the input that contributes the most to the load for that output. This combination avoids crosspoint buffer underflow with $RTT = 31$ and therefore, to achieve high throughput under these conditions.

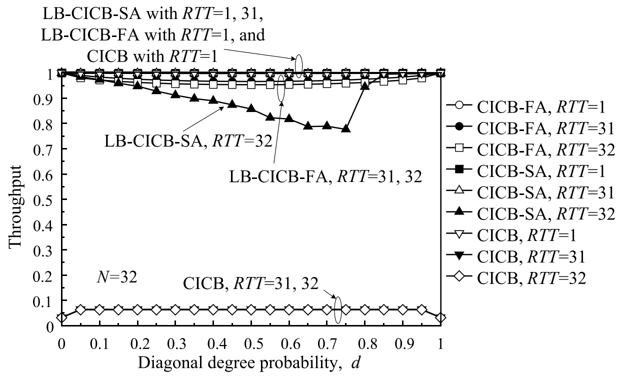


Fig. 13. Throughput of the 32×32 switches with $k = 1$ under diagonal traffic.

Performance of LB-CICB switches with $k > 1$. Under uniform traffic, the average cell delays of the LB-CICB switches with $k = 1$ is close to that of an OQ switch (as shown in Figure 8). Therefore, increasing k provides no further improvement under uniform traffic. The LB-CICB switches were simulated with $k > 1$ under unbalanced traffic. The throughput observed for $k = 2$ of the LB-CICB switches approaches 100% not only for $RTT < 32$ as observed with $k = 1$ but also for $RTT \geq 32$ (more precisely, the throughput approached 100% for $RTT \leq 63$). The performance under diagonal and PO2 traffic was also tested with $k = 2$. The throughput of the LB-CICB-SA switch under these two traffic patterns approaches 100% for $RTT = 32$ as observed under unbalanced traffic.

VII. Implementation Complexity of an LBS

Although of lower complexity, the LB-CICB-SA switch achieves high performance. Therefore, it is of interest to discuss the complexity of the LBS of the LB-CICB-SA switch. The design of the LBS is divided into control and data paths.

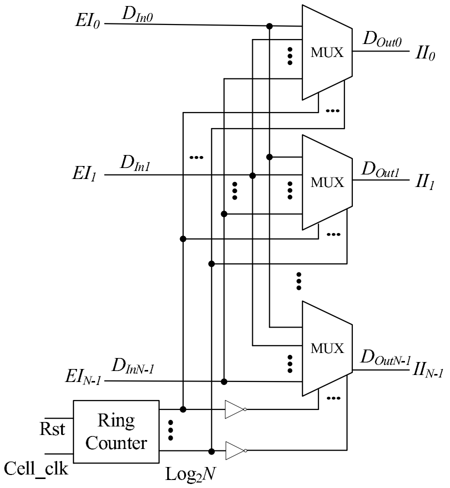


Fig. 14. Data path design of the LBS.

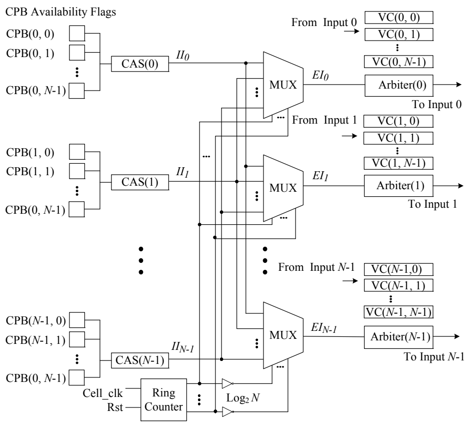


Fig. 15. Control path design of the LBS.

Figure 14 shows the data path of the LBS of an $N \times N$ LB-CICB-SA switch. The data path uses an N-count ring counter and a series of multiplexers. The ring counter is used to define the periodic configurations of the interconnection between EIs and IIs. The multiplexers implement the actual interconnection between those ports. The ring-counter value is incremented at each time slot (using the Cell_clk signal).

Each count of the ring counter is expressed in $log_2 N$ lines that feed the selection lines of the multiplexers. Each multiplexer receives a different combination in their selection lines to interconnect different EIs ($D_{In\ i}$) to IIs ($D_{Out\ l}$) each time slot. The inverters are used to provide different combinations for the multiplexers from the ring counter. The ring counter uses about

$$C_{rc} = 80 \log_2 N \tag{17}$$

3-input gates. A multiplexer uses about

$$C_{mux} = \lceil \frac{\alpha log_2 N}{3} \rceil + \lceil \frac{N}{6} \rceil \tag{18}$$

3-input gates, where α is the number of parallel bits of data transmitted. The total number of gates in this data path is

$$\begin{aligned} C_{data} &= NC_{mux} + C_{rc} \\ &= N(\lceil \frac{\alpha log_2 N}{3} \rceil + \lceil \frac{N}{6} \rceil) + (80 \log_2 N). \end{aligned}$$

For a switch with N=32 and α=32, the number of 3-input gates used is about $2,320$.

Figure 15 shows the design of the control path of the LBS of an $N \times N$ LB-CICB-SA switch. Each CAS has CPB availability flags to indicate whether the corresponding CPB is available or not. The EI-to-II interconnection is defined by a ring counter and multiplexers in a similar design to that of the data path. However, in the control path, the request-grant information travels in the opposite direction. The output of the ring counter also provides the select signal for the multiplexers, and it uses the count value previous to that used in the ring counter for the data path. Also, there are arbiters to select a VC using LQF selection based on the state of the VOQs but only for non-full $CPB(l, j)$. The gate count for the multiplexers, as in the data path, is C_{mux} and for the arbiters is

$$C_{arb}^{LQF} = \beta(1 + 2 + ... + \frac{N}{2}), \tag{19}$$

where β is the number of bits used for the queue length information. The availability flags belong to the buffered-crossbar implementation; therefore, they are not considered in this count. The CAS design is a simple combinatorial logic of the size of a multiplexer, or C_{CAS}=c_{mux}. The VCs are counters that have similar size to the ring counter or C_{VC}=C_{rc}. Then the gate count for the control path is

$$C_{cntrl} = N^2 C_{rc} + NC_{mux} + C_{rc} + NC_{mux} + \beta(1+2+...+\frac{N}{2}) \tag{20}$$

For N=32 and β=32, the control path may require about 418,192 3-input gates if the VCs are not implemented in memory, and about 8,592 3-input gates if the VCs are implemented in memory. This shows that the gate count for the LBS, including the LQF arbiters is small. The ring counter in the control path also runs at the same speed as the ring counter for the data path, or one count per time slot. The time complexity of round-robin and LQF selection schemes is low or O($\log_2 N$), and that of the LBS is O(1). Therefore, the time complexity of the LB-CICB-SA is low.

VIII. Conclusions

This paper proposed two switches that allow inputs to flexibly access the crosspoint buffers of a CICB switch. In the switch called LB-CICB-FA, an input is allowed to access any of the crosspoint buffers. In the second proposed switch, called the LB-CICB-SA switch, an input is allowed to access one set of crosspoint buffers. These two switches efficiently use the crosspoint buffers.

The flexibility that the LB-CICB-FA switch provides high buffer utilization that can provide high switching performance and support for long RTTs with simple selection schemes at the arbiters. The LB-CICB-FA switch with a random selection scheme at the arbiters was analyzed and shown to be weakly stable under admissible i.i.d. traffic. In other words, the switch achieves 100% throughput under i.i.d. traffic with uniform and nonuniform distributions. This is an advantage as the high throughput under uniform traffic of CICB switches is extended to nonuniform traffic, without memory speedup. Simulation results were performed under uniform, unbalanced, and diagonal traffic. The results showed 100% throughput for different switch sizes.

However, the high flexibility provided by this switch may require N^2 multiplexers and an *N-to-N* scheduler. The LB-CICB-SA switch uses a pre-determined configuration for the load-balancing stage, and the inputs are limited to access one set of CPBs, one per output. Therefore, the LB-CICB-SA switch has lower complexity than the LB-CICB-FA switch.

Because the stage that provides flexible access in the proposed switches also provides multiple paths from inputs to outputs, the transmission of cells in sequence must be provided. Therefore, it was shown that the FCFS policy used at the output ports and the use of a single clock at the buffered crossbar keep the transmission of cells in sequence.

The performance of the two proposed switches with LQF selection as input arbitration and for the configuration of crosspoint access, and FCFS as output arbitration under traffic with uniform and nonuniform traffic was investigated using computer simulation. The results show that the throughput of these two switches approaches 100% for uniform and nonuniform traffic patterns, and without using speedup.

The study also considered long RTTs and the simulation results showed that the proposed switches support about kN-time-slot RTTs. Moreover, for a given RTT size, the load-balancing CICB switches require a minimum k=$\lceil \frac{RTT}{N} \rceil$ cells while a CICB switch requires a minimum k=RTT cells. Therefore, the proposed switches require about $\frac{1}{N}$ of the amount required by a CICB switch with dedicated crosspoint buffers. Because the performance of the LB-CICB-SA switch is comparable to that of the LB-CICB-FA, but with lower hardware complexity, the implementation of the LBS stage was discussed. It was then shown that the implementation complexity of the balancing stage is low.

Acknowledgment

The authors thank the anonymous reviewers for their insightful comments and suggestions.

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] 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] D.C. Stephens and H. Zhang, “Implementing distributed packet fair queueing in a scalable switch architecture,” Proc. *IEEE Infocom*, pp. 282-290, March 1998.

[4] R. Rojas-Cessa, E. Oki, and H.J. Chao, “CIXOB-1: Combined Input-crosspoint-output Buffered Packet Switch,” Proc. *IEEE GLOBECOM 2001*, Vol. 4, pp. 2654-2660, November 2001.

[5] R. Rojas-Cessa, E. Oki, and H. Jonathan Chao, “On the Combined Input-Crosspoint Buffered Packet Switch with Round-Robin Arbitration,” *IEEE Trans. Commun.*, Vol. 53, No. 11, pp. 1945-1951, November 2005.

[6] M. Nabeshima, “Performance Evaluation of a Combined Input- and Crosspoint-Queued Switch,” *IEICE Trans. Commun.*, Vol. E83-B, No. 3, pp. 737-741, March 2000.

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

[8] 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.

[9] 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.

[10] 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,” Proc. *IEEE 10th Symposium on Hot Interconnects*, pp. 5-14, Aug. 2002.

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

[12] Z. Dong and R. Rojas-Cessa, “Long Round-Trip Time Support with Shared-Memory Crosspoint Buffered Packet Switch,” Proc. *IEEE High Performance Interconnects 2005*, pp. 138-143, Aug. 2005.

[13] R. Rojas-Cessa and Z. Dong, “Combined Input-Crossspoint Buffered Packet Switch with Flexible Access to Crosspoints Buffers: One-Cell Size Case,” Proc. *IEEE ICCDCS*, 6 pages, April 26-28, 2006.

[14] R. Rojas-Cessa, Z. Dong, and S.G. Ziavras, “Load-Balanced Combined Input-Crosspoint Buffered Packet Switch with Long Round-Trip Time Support,” Proc. *IEEE Globecom*, Vol.2, pp. 1002-1006, Nov. 2005.

[15] K. Yoshigoe, “The CICQ Switch with Virtual Crosspoint Queues for Large RTT,” Proc. *IEEE ICC 2006*, pp. 299-303, 2006.

[16] K. Yoshigoe, “Threshold-based Exhaustive Round-Robin for the CICQ Switch with Virtual Crosspoint Queues,” Proc. *IEEE ICC 2007*, pp.6325-6329, June 2007.

[17] F. Gramsamer, M. Gusat, and R Luijten, “Optimizing Flow Control for Buffered Switches,” Proc. *IEEE ICCCN 2002*, pp. 438-443, October 2002.

[18] M. Katevenis, G. Passas, D. Simos, and N. Chrysos, “Variable Packet Size Buffered Crossbar (CICQ) Switches,” Proc. *IEEE ICC 2004*, Vol. 2, pp. 1090-1096, June 2004.

[19] N. Chrysos and M. Katevenis, “Crossbar with Minimally-Sized Crosspoint Buffers,” Proc. *IEEE HPSR 2007*, 7 pages, May 2007.

[20] L. Mhamdi and M. Hamdi, “Output queued switch emulation by a one-cell-internally buffered crossbar switch,” Proc. *IEEE Globecom 2003*, vol. 7, pp. 3688-3693, Dec. 2003.

[21] S. Chuang, S. Iyer, and N. McKeown, “Practical algorithms for performance guarantees in buffered crossbars,” Proc. *IEEE Infocom*, March 2005.

[22] J. Turner, “Strong Performance Guarantees for Asynchronous Crossbar Schedulers,” Proc. *IEEE Infocom*, 11pp., April 2006.

[23] S. He, S. Sun, H. Guan, Q. Zheng, Y. Zhao, and W. Gao, “On guaranteed smooth switching for buffered crossbar switches,” *IEEE/ACM Trans. on Networking*, vol. 16, no. 3, pp. 718-731, June 2008.

[24] C. Chang, Y. Hsu, J. Cheng, and D. Lee, “A dynamic frame sizing algorithm for CICQ switches with 100% throughput,” Proc. *IEEE Infocom*, pp. 747-755, April 2009.

[25] Cisco Catalyst 6500 Series Switches, http://www.cisco.com/en/US/products/hw/switches/ps708/index.html.

[26] Juniper EX8200 Ethernet Switches, http://www.juniper.net/us/en/products-services/switching/ex-series/ex8200/.

[27] M. Karol and M. Hluchyj, “Queuing in High-performance Packet-Switching,” *IEEE J. Select. Areas Commun.*, Vol. 6, pp. 1587-1597, December 1988.

[28] N. McKeown, A. Mekkittikul, V. Anantharam, J. Warland, “Achieving 100% Throughput in an Input-Queued Switch,” *IEEE Trans. Commun.*, Vol. 47, No. 8, pp. 1260-1267, August 1999.

[29] 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.

[30] 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.

[31] R. Rojas-Cessa, Z. Dong, and Z. Guo, “Load-Balanced Combined Input-Crosspoint Buffered Packet Switch and Long Round-Trip Times,” *IEEE Commun. Letters*, Vol. 4, issue 7, pp. 661 - 663, July 2005.

[32] R. Rojas-Cessa, Z. Guo, and N. Ansari, “On the Maximum Throughput of a Combined Input-Crosspoint Buffered Packet Switch,” IEICE Trans. on Commun., vol. E89-B, no. 11, pp. 3120-3123, November 2006.

[33] I. Elhanany and M. Hamdi, “High-performance Packet Switching Architectures,” *Springer Science+Business Media*, ISBN 1-84628-273-X, Chapter 2, 2007.

[34] D. Bertsekas and R. Gallager, “Data Networks,” *Prentice Hall*, ISBN 0-13-200916-1, Chapter 3, 1992.

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

[36] M. Lin and N. McKeown, “The throughput of a buffered crossbar switch,” *IEEE Commun. Letters*, Vol. 9, Issue 5, pp. 465 - 467, May 2005.

[37] S. Sun, S. He, and W. Gao, “Throughput analysis of a buffered crossbar switch with multiple input queues under burst traffic,” *IEEE Commun. Letters*, Vol. 10, Issue 4, pp. 305 - 307, April 2006.

PLACE
PHOTO
HERE

Roberto Rojas-Cessa received the Ph.D. degree in Electrical Engineering from Polytechnic Institute of New York University, Brooklyn, NY. Currently, he is an Associate Professor in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology. He has been involved in design and implementation of application-specific integrated-circuits (ASIC) for biomedical applications and high-speed computer communications, and in the development of high-performance and scalable packet switches and reliable switches. He was part of the team designing a 40 Tb/s core router in Coree, Inc, in Tinton Falls, NJ. His research interests include high-speed switching and routing, fault tolerance, quality-of-service networks, network measurements, and distributed systems. His research has been funded by U.S. National Science Foundation and private companies. He has served in several technical committees for IEEE conferences and as a reviewer and panelist for U.S. National Science Foundation and the U.S. Department of Energy.

PLACE
PHOTO
HERE

Ziqian Dong received the B.S. degree in Electrical Engineering from Beijing University of Aeronautics and Astronautics, Beijing, China. She received the M.S. and Ph.D. degrees in Electrical Engineering from New Jersey Institute of Technology, Newark, NJ. She was with New Jersey Institute of Technology and Stevens Institute of Technology for her postdoctoral research. She is an Assistant Professor of Electrical and Computer Engineering at New York Institute of Technology. She is the recipient of Hashimoto Prize for her Ph.D. dissertation from New Jersey Institute of Technology and Graduate Student Award winner for her inventions from the New Jersey Inventors Hall of Fame. Her research interests include architecture design and analysis of practical high-speed packet switches and routers, network security and network forensics. She is a member of Institute of Electrical and Electronics Engineers (IEEE) and a member of IEEE Women in Engineering.