

# Asymptotic Performance Limits of Switches with Buffered Crossbars supporting Multicast Traffic

Paolo Giaccone, Emilio Leonardi  
Politecnico di Torino, Italy

email: paolo.giaccone@polito.it, emilio.leonardi@polito.it

**Abstract**—Input queued switches exploiting buffered crossbars (CICQ switches) are widely considered very promising architectures that outperform input queued (IQ) switches with bufferless switching fabrics both in terms of architectural scalability and performance.

Indeed the problem of scheduling packets for transfer through the switching fabric is significantly simplified by the presence of internal buffers in the crossbar, which makes possible the adoption of efficient, simple and fully distributed scheduling algorithms.

In this paper we study the throughput performance of CICQ switches supporting multicast traffic, showing that, similarly to IQ architectures, also CICQ switches with arbitrarily large number of ports may suffer of significant throughput degradation under “pathological” multicast traffic patterns. Despite of the asymptotic nature of our results, we believe that they can contribute to a deeper understanding of the behavior of CICQ architectures supporting multicast traffic.

## I. INTRODUCTION AND PREVIOUS WORK

Input Queued (IQ) switches have been extensively studied in the last decade [1], since they can achieve similar performance of pure output queued (OQ) switches, while guaranteeing a much better scalability in terms of ports number and line data rate. Under unicast traffic, IQ switches achieve 100% throughput (i.e., the same throughput achieved by OQ) without speedup [2], [3]; moreover IQ switches with speedup 2 can perfectly emulate OQ switches, i.e. they can behave identically to OQ switches when observed at the output ports [4], [5]. On the contrary, very large size IQ switches can experience throughput degradation with respect to OQ under multicast traffic [6]; indeed, for number of ports  $N \rightarrow \infty$ , the required speedup to achieve 100% throughput may grow to infinity, showing a fundamental and intrinsic limitation of IQ switches supporting multicast traffic.

In a switching architecture built around buffered crossbars (“CICQ switch”) the bufferless switching fabric of an IQ architecture is replaced by a crossbar provided with small buffers at each crosspoint (see Fig. 1). A survey on CICQ switch architectures and their scheduling algorithms is available in [7]; some interesting proposals (also for multicast traffic) have been described in [8], [9], [10], [11], [12], [13], [14], [15].

The problem of scheduling packets through a CICQ architecture is significantly simplified with respect to an IQ architecture by the presence of internal buffers. In IQ switches a centralized scheduler is required to control the access to the bufferless switching fabric in a globally coordinated fashion;

on the contrary, in CICQ switches access to the switching fabric can be managed by uncoordinated schedulers residing at every input and output ports.

For these reasons, CICQ architectures are widely recognized as very promising solutions which potentially outperform IQ switches both in terms of scalability and performance. This belief seems confirmed by both theoretical and simulative performance analysis of CICQ architectures. Recently it has been shown that under unicast traffic, CICQ architectures with speedup 2 and minimal internal buffers operating under very simple uncoordinated schedulers, achieve the maximum throughput and in some cases may perfectly emulate an OQ switch [16], [17]. The minimum speedup required to achieve 100% throughput can be further lowered by increasing the internal buffers size, as shown in [18]. For any speedup greater than 1, indeed, it has been proved that there exists an internal buffer size such that the CICQ architecture achieves 100% throughput under uncoordinated scheduling algorithms. Finally, simulative investigations have shown that performance under unicast traffic are always very close to 100% throughput when no speedup is provided, even if no theoretical evidence has been provided.

In this paper we show that, unfortunately, similarly to what happens for IQ switch, heavy performance degradations may be experienced by CICQ architectures with  $N \rightarrow \infty$  supporting multicast traffic, for any finite speed-up, regardless of scheduling complexity and internal buffer size. Indeed, we have identified “pathological” traffic patterns which create frequent contentions for output ports, thus leading to hot-spot congestion of internal buffers and, consequently, limiting the switch throughput. We recognize that our findings have mainly a pure theoretical relevance due their asymptotic nature. Nevertheless we believe that they can contribute to a deeper comprehension of the behavior of CICQ architectures supporting multicast traffic.

The paper is organized as follows: in Section II we describe the problem of scheduling multicast traffic in CICQ switches; in Section III we introduce a class of “worst-case” traffic patterns, i.e. traffic patterns that lead to a minimization of the switch throughput, and we define traffic admissibility conditions, in Section IV, we analytically prove that any scheduling algorithm leads to poor performance when CICQ switches are loaded with traffic defined in Section III. Finally Section V concludes the paper. To ease a first reading of the paper, most analytical derivations and theorem proofs were

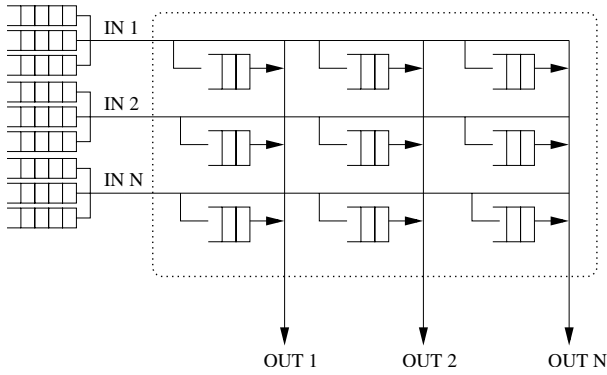


Fig. 1. The  $N \times N$  CICQ architecture with VOQ and buffered crosspoints

moved to the Appendices.

## II. SYSTEM DYNAMICS

We consider an  $N \times N$  CICQ architecture, where  $I = \{i_j\}_{j=0}^{N-1}$  is the set of switch input ports ( $|I| = N$ )<sup>1</sup> and  $O = \{o_j\}_{j=0}^{N-1}$  is the set of switch output ports ( $|O| = N$ ). Fig. 1 shows a basic model for it. Each crosspoint of the crossbar is provided with an internal buffer of size  $B$  packets; thus, internal buffers are in one-to-one correspondence with input-output pairs. An instantaneous flow control mechanism from each crosspoint to the corresponding input port prevents the input port from overflowing the internal buffers. We assume time to be slotted, and packets to be of fixed size (denoted as “cells”). Cells arrive at the input ports according to a discrete time random process. At every slot, some cells enqueued at input ports are moved in the internal buffers while cells in internal buffers are moved toward output ports.

Each multicast cell  $r$  is associated with a pair  $(i, D)$ , where  $i \in I$  is the input port at which it arrives and  $D \subseteq O$  is the set of its destination outputs (denoted as the cell “fanout-set”). Let the cell “fanout” be the number of destination ports. Unicast cells correspond to the subset of multicast cells whose fanout-set comprises just one output port. We partition the set of cells arriving at the switch in *flows* according to their attributes  $(i, D)$ . Thus, a different flow corresponds to every possible attribute  $(i, D)$ .

For each flow  $f$ , function  $J(f)$  returns the input port  $i$  at which flow cells arrive. Function  $U(f)$  returns the fanout-set associated to the flow cells. Function  $J^{-1}(i)$  returns the set of flows which arrive at input port  $i$ . Function  $U^{-1}(o)$  returns the set of multicast flows whose fanout-set comprises  $o$ , i.e.  $o \in U(f)$ .

In order to reach a particular output port  $o$  belonging to the fanout-set, a copy of the multicast cell must be enqueued in the internal buffer leading to output port  $o$ . As a consequence, each multicast cell origins several unicast copies, one for destination, called “fragments”, which are enqueued in the corresponding internal buffers. Fragments of a multicast cell,

after being copied in the internal buffers, are independently delivered to the output ports.

Two possible basic strategies can be devised for the transfer of multicast cells from inputs to internal buffers.

- *No fanout splitting policies*: all the fragments originated by a multicast cell are transferred to internal buffers simultaneously. If any of the internal buffers corresponding to fanout destinations is full, no fragments of the cell can be transferred to the internal buffers.
- *Fanout splitting policies*: different fragments originated by the same cell may be transferred to buffers in different internal timeslots.

In the latter case, the set of destinations in the fanout-set corresponding to the internal buffers into which a fragment has still to be transferred, is called “residual fanout-set” of a cell. In both cases a cell is cancelled at inputs when every internal buffer corresponding to fanout destinations has been reached by a cell fragment, i.e., its residual fanout-set is null. For the sake of readability, we will adopt the following convention: a fragment is “transferred” when it is moved from the input queue to the internal buffer, and a fragment is “delivered” when it is moved from the internal buffer to the output port.

In IQ and CICQ switches supporting unicast traffic only, to avoid head-of-line (HoL) blocking [19], one separate queue for cells directed to each output is necessary at inputs; this queueing architecture is called *Virtual Output Queueing* (VOQ) [20].

As shown in [6], one queue is required at every input for each possible fanout-set (i.e.  $2^N - 1$  queues per input) to avoid HoL blocking in switches supporting multicast traffic; this queueing scheme is denoted as MC-VOQ or “per multicast-flow queueing”. In a “per multicast-flow queueing” structure, upon its arrival at the switch, each cell is enqueued in the MC-VOQ that corresponds to the flow fanout-set and it is removed only when its residual fanout-set is empty. Only cells at the HoL of MC-VOQs are eligible for transfer to the internal buffers<sup>2</sup>.

We are aware that this queueing structure is hardly implementable in large switches, due to the large number of queues needed at every input. However, since in this paper we are mostly interested in providing a theoretical investigation of intrinsic throughput limits of CICQ structures, by considering this idealized queueing structure we are able to find general bounds on the system performance, which do not depend from the particular queueing structure implemented at inputs.

We say that a speedup  $S$  (with  $S$  integer) is available at the switch, when the internal data-paths between inputs and output ports, through the buffered crossbar, run at a speed  $S$  times faster than the external input/output links. Note that all the following rates are accelerated by the same factor  $S$  with

<sup>2</sup>We notice that, in order to maximize the switch throughput, partially transferred cells should be re-enqueued into the queue corresponding to their residual fanout-set [6]; the adoption of such an optimal throughput queueing scheme will lead to out-of-sequence cells delivery at output ports, which is not tolerable in many applicational contexts. For these reasons, in analogy to what done in [6] for IQ and CIOQ, we limit our performance investigation to “per multicast-flow queueing” schemes.

<sup>1</sup>We denote with  $|A|$  the size of set  $A$ .

respect to the link speed: read rate from the input queues, read/write rates from/to the internal buffers, write rate to the output queues.

When  $S > 1$ , it is necessary to distinguish between external and internal timeslot; the former is the transmission time of a cell on the external input/output links; the latter is the transmission time of a cell on the internal data path (between input and output ports). In this case, queues are necessary at output ports, to store cells that cannot be immediately transmitted on output links. Each external timeslot corresponds to  $S$  consecutive internal timeslots. We denote with  $n$  the  $n$ -th external timeslot from a conventional time origin; in a similar way we denote with  $m$  the  $m$ -th internal timeslot from the same conventional time origin, or equivalently we denote with  $(n, h)$  the  $h$ -th internal timeslot with  $1 \leq h \leq S$  corresponding to external timeslot  $n$ , being  $m = (n - 1)S + h$ .

Considering internal timeslot  $m$ , we define the following variables:

- $e_{i,o}(m)$ : the number of fragments from input  $i$  to output  $o$  (denoted as  $i \rightarrow o$ ) transferred to the internal buffer;
- $e_{i,*}(m)$ : the number of cells partially or fully transferred from input  $i$ ;
- $\tilde{e}_{i,*}(m)$ : the number of cells whose transfer to the internal buffers is completed;
- $u_{i,o}(m)$ : the number of fragments  $i \rightarrow o$  delivered from the internal buffer to the output port;
- $u_{*,o}(m)$ : the number of fragments delivered from internal buffers to output  $o$ ; note that  $u_{*,o}(m) = \sum_i u_{i,o}(m)$ ;
- $B_{i,o}(m)$ : the number of fragments  $i \rightarrow o$  stored in the internal buffer at the beginning of the timeslot;
- $B_{i,*}(m) = \sum_o B_{i,o}(m)$ : the total number of fragments stored in the internal buffers fed by input  $i$  at the beginning of the timeslot;
- $B_{*,o}(m) = \sum_i B_{i,o}(m)$ : be the total number of fragments stored in internal buffers leading to output  $o$  at the beginning of the timeslot.

The dynamics of internal buffer occupancy  $B_{i,o}(m)$  is given by:

$$B_{i,o}(m+1) = B_{i,o}(m) + e_{i,o}(m) - u_{i,o}(m)$$

The following constraints must be satisfied by the cells switched across CICQ architectures:

- in each timeslot  $m$ , at most one cell per input can be transferred to the internal buffers:

$$e_{i,*}(m) \leq 1 \quad \forall i \quad (1)$$

- in each timeslot  $m$ , at most one fragment is delivered to each output port:

$$u_{*,o}(m) \leq 1 \quad \forall o \quad (2)$$

- due to flow control, no fragment can be transferred to an internal buffer which is full:

$$e_{i,o}(m) = 0 \quad \text{if } B_{i,o}(m) = B \quad (3)$$

TABLE I

EXAMPLE OF A (2,2)-COMPLEX FLOW SET FOR A  $6 \times 6$  SWITCH

Input	Flow and fanout-sets	
$i_0$	$f_0 = \{o_0, o_1, o_2\}$	$f_1 = \{o_0, o_3, o_4\}$
$i_1$	$f_2 = \{o_1, o_3, o_5\}$	$f_3 = \{o_2, o_4, o_5\}$

### III. TRAFFIC AND PERFORMANCE DESCRIPTION

#### A. $(k, N_a)$ -complex traffic

In this section we introduce the ‘‘pathological’’ multicast traffic patterns under which throughput degradation is experienced.

Consider an  $N \times N$  switch, with  $N_a$  inputs receiving cells from at least a flow (inputs receiving cells are called ‘‘active inputs’’ through this paper).

*Definition 1:* A set  $F$  of flows with  $|F| = kN_a$  is said  **$(k, N_a)$ -complex**, with  $k \in \mathbb{N}$ ,  $k > 1$ , if:

- 1)  $k$  flows in  $F$  arrive at active input port  $i$ , i.e.  $|J^{-1}(i)| = k$ , for any active input port  $i$ ;
- 2)  $k$  flows in  $F$  are directed to output port  $o$ , i.e.  $|U^{-1}(o)| = k$ , for any output port  $o$ ;
- 3) for each sub-set of  $F$  comprising  $k$  flows, a destination exists to which all the flows in the subset are directed.

Table I reports an example of a (2,2)-complex flow set for a  $6 \times 6$  switch, where only inputs  $i_0$  and  $i_1$  are active. In an  $N \times N$  switch where  $N_a$  input ports are active,  $N = \binom{kN_a}{k}$  and  $N_a \geq 2$ , a  $(k, N_a)$ -complex flow set  $F$  of size  $kN_a$  can be generated with the following algorithm:

Step 1. Assign the first  $k$  flows in  $F$  to the first input, the second  $k$  flows to the second input, and so on, until the last  $k$  flows have been assigned to input  $N_a$ . With reference to Table I,  $F = \{f_0, f_1, f_2, f_3\}$ , and  $J^{-1}(i_0) = \{f_0, f_1\}$  and  $J^{-1}(i_1) = \{f_2, f_3\}$ .

Step 2. Form all the possible  $N = \binom{kN_a}{k}$  different subsets of  $R$  whose size is  $k$ , and create an arbitrary injective correspondence from subsets of cells to destinations. With reference to Table I,  $U^{-1}(o_0) = \{f_0, f_1\}$ ,  $U^{-1}(o_1) = \{f_0, f_2\}$ ,  $U^{-1}(o_2) = \{f_0, f_3\}$ ,  $U^{-1}(o_3) = \{f_1, f_2\}$ ,  $U^{-1}(o_4) = \{f_1, f_3\}$ , and  $U^{-1}(o_5) = \{f_2, f_3\}$ .

Step 3. To each flow in  $F$  assign all the destinations that correspond to sets containing the cell itself. With reference to Table I,  $U(f_0) = \{o_0, o_1, o_2\}$ ,  $U(f_1) = \{o_0, o_3, o_4\}$ ,  $U(f_2) = \{o_1, o_3, o_5\}$ , and  $U(f_3) = \{o_2, o_4, o_5\}$ .

For each flow, the resulting fanout is equal to  $\binom{kN_a-1}{k-1} = N/N_a$ .

*Definition 2:* An arrival process is **uniformly  $(k, N_a)$ -complex** with normalized rate  $\lambda$  (with  $0 \leq \lambda \leq 1$ ) if:

- 1) the associated flow set  $F$  is  $(k, N_a)$ -complex;
- 2) for every flow  $f \in F$ , cells arrive at the switch according to a general discrete time stationary arrival process satisfying the strong law of large numbers;
- 3) for each flow  $f \in F$  the average number of cell arrivals per timeslot  $\lambda_f$  is  $\lambda/k$ .

Note that  $\sum_f \lambda_f = \lambda N_a$ .

**Definition 3:** A set of cells  $R$  of size  $kN_a$  is a **perfect**  $(k, N_a)$ -**complex cell set** if each cell in  $R$  belongs to a different flow  $f \in F$ , being  $F$  a  $(k, N_a)$ -complex flow set  $F$ .

**Definition 4:** A set of cells  $R$  of size  $kN_a$  is a **generalized**  $(k, N_a)$ -**complex cell set** if each cell in  $R$  belongs to a flow  $f \in F$ , being  $F$  a  $(k, N_a)$ -complex flow set  $F$  (note that in this case more cells of the same flow can belong to  $R$ ).

### B. Traffic admissibility, stability and maximum throughput

We introduce the basic notion of traffic *admissibility*:

**Definition 5:** A stationary arrival process satisfying the strong law of large numbers is admissible if, either no input or output ports are overloaded, i.e.:

$$\begin{aligned} \sum_{f \in \mathbf{J}^{-1}(i)} \lambda_f &< 1 \quad \forall i \\ \sum_{f \in \mathbf{U}^{-1}(o)} \lambda_f &< 1 \quad \forall o \end{aligned}$$

We notice that uniformly  $(k, N_a)$ -complex arrival processes are admissible when  $\lambda < 1$ .

**Definition 6:** A switch architecture is *stable* under an admissible arrival process if, with probability 1, the observed average departure cell rate from the switch equals the average cell arrival rate at the switch input ports.

A necessary condition for the switch to be stable is that the rate at which cells are moved toward output ports equals the average cell arrival rate at inputs:

$$\begin{aligned} \liminf_{t \rightarrow \infty} \frac{1}{t} \sum_{n=1}^t \sum_{h=1}^S u_{i,o}(n, h) &= \sum_{f \in \mathbf{J}^{-1}(i) \cap \mathbf{U}^{-1}(o)} \lambda_f \quad \forall i, o \quad \text{w.p.1} \end{aligned}$$

A different necessary condition for switch stability is that the rate at which cells are transferred to the internal buffers equals the average cell arrival rate at inputs:

$$\liminf_{t \rightarrow \infty} \frac{1}{t} \sum_{n=1}^t \sum_{h=1}^S \tilde{e}_i(n, h) = \sum_{f \in \mathbf{J}^{-1}(i)} \lambda_f \quad \forall i \quad \text{w.p.1}$$

which is equivalent to:

$$\liminf_{t \rightarrow \infty} \frac{1}{t} \sum_i \sum_{n=1}^t \sum_{h=1}^S \tilde{e}_i(n, h) = \sum_f \lambda_f \quad \text{w.p.1} \quad (4)$$

**Definition 7:** A switch achieves 100% throughput if it is stable under every admissible arrival process.

Note that an OQ switch, by construction, achieves 100% throughput.

## IV. MAIN RESULTS

In this section we prove our results regarding the maximum throughput achievable by CICQ architectures under uniformly  $(k, N_a)$ -complex arrival process.

In order to prove our main results, we need first to introduce some partial results on the performance of CICQ architectures evaluated over a finite temporal horizon. We assume that at the beginning of timeslot  $m = 1$  a generalized  $(k, N_a)$ -complex cells set is waiting for transfer at inputs; we further assume that all internal buffers are empty, and no further cells arrive at inputs port. We evaluate the ‘‘input queues clearance time’’, defined as the minimum number of timeslots necessary to completely transfer all the cells residing at inputs toward the internal buffers, thus clearing the input ports memories.

These results constitute fundamental building blocks for the main theorems presented in the next subsection; however, since their proofs are quite long and articulated, to ease the first reading of the paper, we moved all the proofs in the appendices.

### A. Input queues clearance time

#### 1) No fanout splitting policies:

**Theorem 1:** Consider a CICQ switch with  $B \geq 1$ , loaded by a generalized  $(k, N_a)$ -complex cells set  $R$ . The input queues clearance time  $L$  satisfies:

$$L \geq \left\lceil \frac{N_a k (k - BN_a - 1)}{k - 1} \right\rceil$$

if fanout splitting is not allowed at input ports <sup>3</sup>.

**Corollary 1:** Consider a CICQ switch with  $B \geq 1$ , loaded by a generalized  $(k, N_a)$ -complex cells set  $R$ . Under a no fanout splitting transfer policy, if  $k > BN_a$ , at most  $k - 1$  cells can be transferred from the active inputs to the internal buffers in  $k - BN_a - 1$  consecutive internal timeslots.

#### 2) Fanout splitting policies:

**Theorem 2:** Consider a CICQ switch with  $B \geq 1$ , loaded by a generalized  $(k, N_a)$ -complex cells set  $R$ . For any finite  $S$ , there exist two integers  $N_0$  and  $k_0$ , such that  $\forall N_a > N_0$  and  $k > k_0$  the switch input queues clearance time  $L$  is greater than  $Sk$ , under any fanout-splitting transfer policy.

### B. Switch throughput results

#### 1) No-fanout-splitting scheduling policy case:

**Theorem 3:** Consider a CICQ switch with  $B \geq 1$  and finite  $S$ , in which cells arrive according to an admissible uniformly  $(k, N_a)$ -complex traffic pattern at rate  $\lambda$ . For sufficiently large values of  $k \gg BN_a$ , and  $N_a > S/\lambda$ , the switch implementing any no fanout splitting transfer policy is unstable.

**Proof:** This proof easily follows from Corollary 1. Indeed, since at most  $k - 1$  cells can be transferred to the internal buffers in  $k - BN_a - 1$  internal timeslots, the average

<sup>3</sup>Let  $\lceil x \rceil$  be the minimum integer  $\geq x$ .

number of cells transferred to the internal buffers per external timeslot satisfy the following relation:

$$E\left[\sum_i \sum_{h=1}^S \tilde{e}_i(n, h)\right] \leq S \frac{k-1}{k - BN_a - 1}$$

from which:

$$\liminf_{t \rightarrow \infty} \frac{1}{t} \sum_i \sum_{n=1}^t \sum_{h=1}^S \tilde{e}_i(n, h) \leq S \frac{k-1}{k - BN_a - 1}$$

Now, thanks to (4), if

$$\lambda N_a > S \frac{k-1}{k - BN_a - 1} \Rightarrow \lambda > \frac{S(k-1)}{N_a(k - BN_a - 1)}$$

the system is unstable. Now for  $k \gg BN_a$  the system becomes unstable for any  $\lambda > S/N_a$ . ■

## 2) Fanout-splitting scheduling policy case:

*Theorem 4:* Consider a  $N \times N$  CICQ switch with  $B \geq 1$  and finite speedup  $S$  implementing a fanout splitting scheduling policy. Under an admissible uniformly  $(k, N_a)$ -complex traffic at rate  $\lambda < 1$ , for sufficiently large  $k$  and  $N_a$ , the switch is unstable.

*Proof:* In this proof we denote with the term “frame” a limited temporal horizon comprising a finite set of consecutive internal/external timeslots. We say that the transfer of a cell is completed within a given finite time horizon (timeslot/frame) if the transfer of the last fragment toward the internal buffer occurs within the considered finite temporal horizon. We say that a cell is fully transferred within a given finite time horizon (timeslot/frame) if all the fragments of the considered cell are transferred toward internal buffers within the considered finite time-horizon.

The theorem can be proved by contradiction; suppose that there exists a per-multicast-flow scheduler that makes the switch stable with speedup  $S$  under a uniformly  $(k, N_a)$ -complex traffic pattern at rate  $\lambda$ . Since on average  $\lambda N_a$  cells arrive at the switch in each timeslot, in a stable switch, with probability 1, it must be:

$$\liminf_{t \rightarrow \infty} \sum_i \frac{1}{t} \sum_{n=1}^t \sum_{h=1}^S \tilde{e}_i(n, h) = \sum_f \lambda/k = \lambda N_a \quad (5)$$

Consider a sample path for which the above property holds, and on it consider a sequence of time windows  $T_m = [1, t_m]$  with  $t_m = m \lceil \frac{2k}{\lambda} \rceil$ , and group the timeslots belonging to  $T_m$  in frames of length  $\lceil \frac{2k}{\lambda} \rceil$  external timeslots (which correspond to  $S \lceil \frac{2k}{\lambda} \rceil$  internal timeslots); now (5) becomes:

$$\liminf_{t_m \rightarrow \infty} \sum_i \frac{1}{t_m} \sum_{n=1}^{t_m} \sum_{h=1}^S \tilde{e}_i(n, h) = \liminf_{m \rightarrow \infty} \sum_i \frac{1}{m} \frac{1}{\lceil \frac{2k}{\lambda} \rceil} \sum_{j=0}^{m-1} \sum_{l=1}^{\lceil \frac{2k}{\lambda} \rceil} \sum_{h=1}^S \tilde{e}_i(l + j \lceil \frac{2k}{\lambda} \rceil, h) = \lambda N_a$$

Thus there exists  $j_0$  such that  $\frac{1}{\lceil \frac{2k}{\lambda} \rceil} \sum_i \sum_{l=1}^{\lceil \frac{2k}{\lambda} \rceil} \sum_{h=1}^S \tilde{e}_i(l + j_0 \lceil \frac{2k}{\lambda} \rceil, h) \geq \lambda N_a$  from which immediately follows:

$$\sum_i \sum_{l=1}^{\lceil \frac{2k}{\lambda} \rceil} \sum_{h=1}^S \tilde{e}_i(l + j_0 \lceil \frac{2k}{\lambda} \rceil, h) \geq \lambda N_a \lceil \frac{2k}{\lambda} \rceil \geq 2k N_a$$

that is: there exists a frame of  $\lceil \frac{2k}{\lambda} \rceil$  consecutive external timeslots (to which  $S \lceil \frac{2k}{\lambda} \rceil$  internal timeslots correspond), in which the transfer of at least  $2k N_a$  cells from inputs to internal buffers is completed.

Since the scheduler is per-multicast-flow, under a generalized  $(k, N_a)$ -complex traffic pattern no more than  $k N_a$  cells may be simultaneously handled by the scheduler (one for flow). Hence, when the considered frame starts, among the  $2k N_a$  cells whose transfer will be completed within the frame, no more than  $k N_a$  cells can be already partially transferred toward internal buffers. As a consequence, at least  $k N_a$  cells are necessarily fully transferred within the considered frame.

However, since any set of  $k N_a$  cells belonging to a uniformly  $(k, N_a)$ -complex arrival process forms a generalized- $(k, N_a)$ -complex cell set, making  $N_a$  and  $k$  sufficiently large, for any finite  $S$  a contradiction with what has been proved in Theorem 2 is obtained. ■

## V. CONCLUSIONS

In this paper we have proved that for asymptotically large CICQ switches with finite internal buffer size, supporting multicast traffic, no finite speedup guarantees 100% throughput under some admissible multicast traffic patterns. Even if the nature of our results are mainly theoretical, we believe that they can contribute to a deeper understanding of the behavior of CICQ architectures supporting multicast traffic.

## REFERENCES

- [1] E. Leonardi, M. Mellia, F. Neri, M. Ajmone Marsan, “On the Stability of Input-Queued Switches with Speedup”, *IEEE/ACM Trans. on Networking*, vol. 9, n. 1, Feb. 2001, pp. 104-118
- [2] J.G. Dai, B. Prabhakar, “The throughput of data switches with and without speedup”, *IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000, pp. 556-564
- [3] N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand, “Achieving 100% throughput in an input-queued switch”, *IEEE Trans. on Communications*, vol. 47, n. 8, Aug. 1999, pp. 1260-1272
- [4] S.T. Chuang, A. Goel, N. McKeown, B. Prabhakar, “Matching output queueing with a combined input/output-queued switch”, *IEEE Journal on Selected Areas in Communications*, vol. 17, n. 6, Jun. 1999, pp. 1030-39
- [5] Stoica I., Zhang H., “Exact emulation of an output queueing switch by a combined input output queueing switch”, *6<sup>th</sup> International Workshop on Quality of Service (IWQoS’98)*, Napa, CA, May 1998, p. 218-224
- [6] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi, F. Neri, “Multicast traffic in input-queued switches: Optimal scheduling and maximum throughput”, *IEEE/ACM Transactions on Networking*, Vol.3, No.11, pp.465-477, June 2003
- [7] K. Yoshigoe, K.J. Christensen, “An evolution to crossbar switches with virtual output queueing and buffered crosspoint”, *IEEE Network*, Sept. 2003, pp. 48-56
- [8] N. Chrysos, M. Katevenis, “Weighted Fairness in Buffered Crossbar Scheduling”, *IEEE HPSR 2003*, Torino, Italy, June 2003, pp. 17-22
- [9] T. Javadi, R. Magill, T. Hrabik, “A high throughput scheduling algorithm for a buffered crossbar switch fabric”, *IEEE ICC 2001*, June 2001, pp. 1581-1591

- [10] M. Katevenis, G. Passas, D. Simos, I. Papaefstathiou, N. Chrysos, "Variable Packet Size Buffered Crossbar (CICQ) Switches", *IEEE ICC 2004*, Paris, France, June 2004
- [11] L. Mhamdi, M. Hamdi, "MCBF: a high-performance scheduling algorithm for buffered crossbar switches", *IEEE Communications Letters*, vol. 7, n. 9, Sept. 2003, pp. 451-453
- [12] M. Nabeshima, "Performance evaluation of a combined input and crosspoint queued switch", *IEICE Trans. Comm.*, vol. E83-B, n. 3, Mar. 2000
- [13] R. Rojas Cessa, E. Oki, Z. Jing, H.J. Chao, "CIXB-1: combined input-one-cell-crosspoint buffered switch", *IEEE HPSR 2001*, Dallas, USA, pp. 324-329
- [14] R. Rojas Cessa, E. Oki, H.J. Chao, "CIXOB-k: combined input-crosspoint-output buffered packet switch", *IEEE GLOBECOM 2001*, Nov. 2001, pp. 2654-60
- [15] R. Rojas-Cessa, E. Oki, "Round robin selection with adaptable size frame in a combined input-crosspoint buffered switch", *IEEE Communications Letters*, vol. 7, n. 11, Nov. 2003
- [16] R.B. Magill, C.E. Rohrs, R.L. Stevenson, "Output queued switch emulation by fabrics with limited memory", *IEEE Journal on Selected Area in Communications*, vol. 21, n. 4, May 2003
- [17] S.-T. Chuang, S. Iyer, N. McKeown, "Practical algorithms for performance guarantees in buffered crossbars", *IEEE INFOCOM 2005*, Miami, FL, Mar. 2005
- [18] Paolo Giaccone, Emilio Leonardi, Devavrat Shah, "On the Maximal Throughput of Networks with Finite Buffers and its Application to Buffered Crossbars", *IEEE INFOCOM 2005*, Miami, FL, Mar. 2005
- [19] Hluchyj M.G., Karol M.J., Morgan S., "Input versus output queueing on a space division switch", *IEEE Transactions on Communications*, vol. 35, n. 12, pp. 1347-1356, Dec. 1987
- [20] Y. Tamir, G.L. Frazier, "High performance multiqueue buffers for VLSI communication switches", *ACM SIGARCH*, pp.353-354, May/June 1988

## APPENDIX I

### EVALUATION OF THE INPUT QUEUES CLEARANCE TIME

In this section we study the problem of switching multicast cells through a CICQ architecture over a finite temporal horizon. Essentially we are interested in evaluating the minimum number of internal timeslots (that we call also the "minimum frame length") needed to fully transfer (up to evacuation of input queues) a perfect or a generalized  $(k, N_a)$ -complex set of cells residing at the inputs toward internal buffers.

For unicast traffic a solution to the problem of transferring cells over a finite horizon is provided by the well known Birkhoff-von-Neumann theorem which states that any set of unicast cells can be completely transferred through a pure IQ switch (and thus, also through a CICQ) in  $T$  timeslots, being  $T$  the maximum number of cells either stored at any input port or directed to any output port. Here we prove that an extension of Birkhoff-von-Neumann theorem to the case of multicast traffic is not possible. Before proving our results, however, we need to formalize more precisely the problem.

#### A. The scheduling problem

Recalling that  $R = \{r_i\}$  represents the set of cells to be switched at the beginning of the considered frame, we redefine functions  $J(\cdot)$ ,  $U(\cdot)$  in such a way to operate directly on cells in  $R$ , according to the following rules: for each cell  $r_i \in R$  with parameters  $(i_i, D_i)$ , function  $J(r_i)$  returns input port  $i_i$ , while function  $U(r_i)$  returns fanout-set  $D_i$ . Given a set of cells  $R' \subseteq R$ , we denote with  $J(R')$  the set of input ports corresponding to cells in  $R'$ . Function  $J^{-1}(i)$  returns the set of cells  $r_i \in R$  whose input port is  $i$ . Function  $U^{-1}(o)$  returns

the set of cells  $r_i \in R$  whose fanout-set comprises  $o$ , i.e.  $o \in U(r_i)$ .

We denote with  $\mathcal{L}$  a set comprising  $L$  consecutive (internal) timeslots which represents our finite time horizon, or frame;  $\mathcal{L} = \{1, 2, \dots, m, \dots, L\}$ .

We now introduce three functions which define the switching process of cells from input to output ports.

Function  $ITSA$  defines the correspondence between each cell  $r_i \in R$  and the set of timeslots in which the cell is selected at the input port for a (possible) full or partial transfer to the internal buffers. More formally:

*Definition 8:* An Input Time Slot Assignment  $ITSA[R]$  is defined as a function whose domain is  $R$  and whose image is the power set of  $\mathcal{L}$ , i.e.<sup>4</sup>:  $ITSA[R] : R \rightarrow 2^{\mathcal{L}}$ .

For each cell  $r_i$  the effective transfer process of different fragments to the internal buffers is fully specified by the Input Scheduling  $\mathcal{IS}$  function which, for each cell  $r_i$  and each output port  $o \in U(r_i)$ , returns the timeslot at which the fragment transfer occurs. More formally:

*Definition 9:* An Input Scheduling  $\mathcal{IS}[R, O]$  is defined as a function whose domain is the set pairs  $(r_i, o)$  with  $o \in U(r_i)$  and whose image is  $\mathcal{L} \cup \{\emptyset\}$ .

Conventionally,  $\mathcal{IS}(r_i, o) = \emptyset$  means that the fragment is not transferred in the considered frame. An input scheduling  $\mathcal{IS}[R, O]$  is said complete if  $\forall r \in R$  and  $\forall o \in U(r)$ , it results  $\mathcal{IS}(r_i, o) \neq \emptyset$ . In this case all the fragments originated by cells in  $R$  are transferred toward internal buffers within the considered frame.

Finally, function  $OTSA$  of cell  $r_i \in R$  directed to destination  $o \in U(r_i)$  returns the timeslot of the frame in which the corresponding fragment is delivered from the internal buffer to the output port; conventionally  $OTSA(r_i, o) = \emptyset$  if the fragment is not transferred in the current frame. More formally:

*Definition 10:* An Output Time Slot Assignment  $OTSA[R, O]$  is defined as a function whose domain is the set  $\{(r_i, o) \text{ with } r_i \in R \text{ and } o \in U(r_i)\} \subseteq R \times O$  and whose image is  $\mathcal{L} \cup \{\emptyset\}$ .

Note that the three functions  $ITSA$ ,  $\mathcal{IS}$  and  $OTSA$  cannot be independently defined. For every cell  $r_i$  and every output  $o \in U(r_i)$  the transfer of fragments can occur only in timeslots belonging to  $ITSA(r_i)$ , i.e., it must be  $\mathcal{IS}(r_i, o) \in ITSA(r_i)$ .

For each fragment, the delivery from the internal buffers to the output ports can occur only after that the fragment has been moved into internal buffers, i.e.,

$$\mathcal{IS}(r_i, o) \leq OTSA(r_i, o) \quad \forall r_i \in R, \forall o \in U(r_i)$$

Assuming a first-come-first-served queueing discipline at each internal buffer, fragments must be delivered from internal buffers in the same order they were moved in, i.e., if  $J(r_i) = J(r_j)$  then

$$\mathcal{IS}(r_i, o) < \mathcal{IS}(r_j, o) \Rightarrow OTSA(r_i, o) < OTSA(r_j, o)$$

<sup>4</sup>Given any set  $A$ ,  $2^A$  denotes the power set of  $A$ .

We also define the inverse relations which will be useful in the following:

- $\mathcal{ITSA}^{-1}(m)$  is the function  $\mathcal{L} \rightarrow \mathcal{R}$  returning the set of cells which, according to  $\mathcal{ITSA}[R]$ , are selected for transfer to the internal buffers in timeslot  $m$ , i.e., the set of cells  $r_l$  such that  $m \in \mathcal{ITSA}(r_l)$ .
- $\mathcal{IS}^{-1}[m, o]$  is the function in  $\mathcal{L} \times \mathcal{O}$  returning the set of fragments directed to  $o$  which are transferred to the internal buffers in timeslot  $m$ , i.e., the set of cells such that  $\mathcal{IS}(r_l, o) = m$ . Note that  $\mathcal{IS}^{-1}(m, o) \cap \mathcal{J}^{-1}(i)$  denotes the set of fragments directed to  $o$  originated by cells enqueued at  $i$  and transferred to the internal buffer in timeslot  $m$ .
- $\mathcal{OTSA}^{-1}(m, o)$  is the function in  $\mathcal{L} \times \mathcal{O}$  returning the set of cells which are transferred to output  $o$  in timeslot  $m$ , i.e., the set of cells  $r_l$  such that  $\mathcal{OTSA}(r_l, o) = m$ .

We force the Input Time Slot Assignment to satisfy the following constraint:

$$|\cup_o \mathcal{ITSA}^{-1}(m)| \cap \mathcal{J}^{-1}(i) \leq 1 \quad \forall m, i \quad (6)$$

which expresses the fact that in each timeslot  $m$ , at most one cell per input has access to the internal buffer.

The term of functions  $\mathcal{ITSA}$ ,  $\mathcal{IS}$  and  $\mathcal{OTSA}$  completely specifies the switching process of multicast cells during the considered frame, any other information on cell movements can be derived from their definition. In particular, we can express:

- $e_{i,o}(m) = |\mathcal{IS}^{-1}(m, o) \cap \mathcal{J}^{-1}(i)|$
- $e_{i,*}(m) = |\cup_o \mathcal{IS}^{-1}(m, o) \cap \mathcal{J}^{-1}(i)|$
- $\tilde{e}_i(m) = \{r \in \mathcal{R} : m \in \cup_o \mathcal{IS}^{-1}(r, o), \forall p > m, p \notin \cup_o \mathcal{IS}^{-1}(r, o)\}$
- $u_{i,o}(m) = |\mathcal{OTSA}^{-1}(m, o) \cap \mathcal{J}^{-1}(i)|$
- $u_{*,o}(m) = \sum_i u_{i,o}(m) = |\mathcal{OTSA}^{-1}(m, o)|$

As a consequence, constraints (1)-(3) can be rephrased in terms of  $\mathcal{ITSA}$ ,  $\mathcal{IS}$  and  $\mathcal{OTSA}$ .

Finally, let us extend the definition of previous functions over a generic subset  $A \subseteq \mathcal{R}$ :

- $\mathcal{I}_{\mathcal{ITSA}}(A, o)$  returns the set of timeslots  $m$  such that  $\mathcal{ITSA}(r) \leq m \leq \mathcal{ITSA}(r')$  for some  $r, r' \in A$ ;
- $\mathcal{I}_{\mathcal{IS}}(A, o)$  returns the set of timeslots  $m$  such that  $\mathcal{IS}(r, o) \leq m \leq \mathcal{IS}(r', o)$  for some  $r, r' \in A$ ;
- $\mathcal{I}_{\mathcal{OTSA}}(A, o)$  returns the set of  $m$  such that with  $\mathcal{OTSA}^{-1}(r, o) \leq m \leq \mathcal{OTSA}^{-1}(r', o)$  for some  $r, r' \in A$ .

### B. A preliminary result

*Lemma 1:* Consider a complete input scheduling  $\mathcal{IS}$ . Given  $R$ , a perfect or generalized  $(k, N_a)$ -complex cell set, for any  $A \subseteq \mathcal{R}$ , with  $|A| \geq B|\mathcal{J}(A^o)|$ , the following constraint must be satisfied:

$$|\mathcal{I}_{\mathcal{IS}}(A, o)| \geq |A^o| - B|\mathcal{J}(A^o)| \quad \forall o$$

where  $A^o = \mathcal{U}^{-1}(o) \cap A$  is the set of cells in  $A$  directed to  $o$ .

*Proof:* We focus our attention on transfer process of fragments directed to  $o$  ignoring what happens to fragments

directed toward other outputs. Conventionally we count timeslots starting from the timeslot in which the first fragment of  $A^o$  is transferred to the internal buffer (as a consequence, timeslot 1 represents the timeslot in which the first fragment from  $A^o$  is transferred). Let  $A_p$  be the number of cells belonging to  $A^o$  whose fragments destined to  $o$  are scheduled in the first  $p$  timeslots of the frame (i.e.,  $A_p = |A^o \cap [\cup_{m \leq p} \mathcal{IS}^{-1}(m, o)]|$ ). Let  $U_p = |(A^o \cap [\cup_{m \leq p} \mathcal{OTSA}^{-1}(m, o)])|$  be the number of fragments that are delivered from the internal buffers to output  $o$  during the first  $p$  timeslots of the frame. It holds:

$$B_{*,o}(p+1) = A_p - U_p + B_{*,o}(1)$$

Note that  $U_p = |(A^o \cap [\cup_{m \leq p} \mathcal{OTSA}^{-1}(m, o)])| \leq |\cup_{m \leq p} \mathcal{OTSA}^{-1}(m, o)| = \sum_{m=1}^p u_{*,o}(m) \leq \lfloor \mathcal{I}_{\mathcal{IS}}(A_p, o) \rfloor$ ; since at most one cell can be delivered to  $o$  per timeslot, it results:

$$B_{*,o}(p+1) \geq A_p - p + B_{*,o}(1)$$

now since, by construction,  $B_{*,o}(m) \leq B|\mathcal{J}(A^o)|$  for every  $m$ , thus it results:

$$B|\mathcal{J}(A^o)| \geq A_p - p + B_{*,o}(1)$$

from which:

$$A_p \leq B|\mathcal{J}(A^o)| + p - B_{*,o}(1)$$

Thus choosing  $A_p = |A^o|$ , and  $p = |\mathcal{I}_{\mathcal{IS}}(A, o)|$ , it follows:

$$|A^o| \leq B|\mathcal{J}(A^o)| + |\mathcal{I}_{\mathcal{IS}}(A, o)| - B_{*,o}(1) \leq B|\mathcal{J}(A^o)| + |\mathcal{I}_{\mathcal{IS}}(A, o)| \leq BN_a + |\mathcal{I}_{\mathcal{IS}}(A, o)|$$

### C. Input queues clearance time for non-fanout splitting policies

As already said, under non-fanout splitting policies, all the fragments originated from a cell must be simultaneously transferred to the internal buffers. We can formalize this constraint, imposing:

$$|\mathcal{ITSA}(r_l)| = |\mathcal{IS}(r_l, o)| = 1, \quad \forall r_l \in \mathcal{R} \text{ and } \forall o \in \mathcal{U}(r_l)$$

Then we obtain the following result:

*Theorem 5:* Consider a switch loaded by a generalized  $(k, N_a)$ -complex cell set. Input queues clearance time  $L$  must satisfy:

$$L \geq \left\lceil \frac{N_a k (k - BN_a - 1)}{k - 1} \right\rceil$$

*Proof:* Consider any set  $A$  (with  $|A| = k$ ) of cells extracted from a generalized  $(k, N_a)$ -complex cell set. Since, by definition, there exists an output port  $o$  to which all the cells in  $A$  are directed, the minimum number of timeslots in which the fragments directed to  $o$  are transmitted is, by Lemma 1:

$$|\mathcal{I}_{\mathcal{IS}}(A, o)| \geq k - B|\mathcal{J}(A)| \geq k - BN_a \quad (7)$$

Since no partial transfer of cells is allowed, the previous result is equivalent to say that in any set of  $k - BN_a - 1$  consecutive

timeslots at most  $k - 1$  cells can be transferred to the internal buffers of the switch. As a consequence, the minimum number of timeslots to transfer all the cells in  $R$  without violating (7) is:

$$L \geq \left\lceil \frac{N_a k(k - BN_a - 1)}{k - 1} \right\rceil$$

which for  $k \rightarrow \infty$  tends to  $kN_a$ .  $\blacksquare$

*Corollary 2:* Under a no fanout splitting scheduling policy, considering a switch loaded by a generalized  $(k, N_a)$ -complex cell set. If  $k > BN_a$ , not more  $k - 1$  cells can be transferred to the internal buffers in  $k - BN_a - 1$  consecutive timeslots.

*Proof:* This statement has been already proved in the previous proof (Theorem 5).  $\blacksquare$

### D. Input queues clearance time for fanout splitting policies

In this case to help the reader we start from a particular case and then gradually generalize our result.

*Theorem 6:* Consider a CICQ switch with generic  $B \geq 1$  loaded with a perfect  $(k, N_a)$ -complex cell set  $R$ . For sufficiently large values of  $k$  and  $N_a$  the input queues clearance time  $L$  satisfies:  $L > 2k$ , under any fanout-splitting policy.

*Proof:* The assertion of the theorem is equivalent to say that no complete  $\mathcal{IS}$  can exist in a frame of length  $L = 2k$ .

We prove this theorem by contradiction. Suppose that a complete  $\mathcal{IS}$  exists in a frame of length  $L = 2k$ . Now observe that, for any  $r \in R$ ,  $|\mathcal{ITSA}(r)|$  is the number of transmission opportunities for cell  $r$ . By construction, the average number of transmission opportunity per cell cannot exceed 2. This is due to the fact that at each input at most  $2k$  transmission opportunity can be distributed among the  $k$  enqueued cells without possibility of repetitions, as described by (6).

Consider  $R_2$ , the set of all cells for which  $|\mathcal{ITSA}(r)| \leq 2$ . Thanks to lemma 2, at least half of the cell set is in  $R_2$ , i.e.  $|R_2| \geq |R|/2 = kN_a/2$ .

A generic cell  $r \in R_2$  attempts the transfer to internal buffers at most in two timeslots  $(m_1(r), m_2(r))$  (with  $m_1(r) \leq m_2(r)$ ) and for  $h = \{1, 2\}$ ,  $m_h \in \{1, 2, \dots, 2k, \text{Null}\}$ ;  $m_h = \text{Null}$  means that  $|\mathcal{ITSA}(r)| < h$ . We further notice that since the input scheduling  $\mathcal{IS}$  is complete, it must be  $|\mathcal{ITSA}(r)| \geq 1$ .

Now consider a set  $R_u \subseteq R_2$  with  $|R_u| = k$  (this set exists since  $N_a \geq 2$  and  $|R_2| \geq kN_a/2$ ); since the considered traffic is  $(k, N_a)$ -complex, there exists an output  $u$  such that all cells in  $R_u$  are directed to  $u$ . Consider the subset  $R'_u$  of  $R_u$  comprising those cells  $r$  whose fragment directed to  $u$  is transferred in  $m_1(r)$ ; i.e.,  $\mathcal{IS}(r, u) = m_1(r) \quad \forall r \in R'_u$ . Let  $R''_u$  be the complementary set of  $R'_u$  with respect to  $R_u$ . For the assumption of completeness of the input scheduling  $\mathcal{IS}$ , for every cell  $r \in R''_u$ ,  $\mathcal{IS}(r, u) = m_2(r)$ .

Let us define  $\mathcal{L}_1 = |\mathcal{IS}(R'_u)|$  and  $\mathcal{L}_2 = |\mathcal{IS}(R''_u)|$ . Consider the cells in  $R'_u$ . Lemma 1 provides a relation between the size of  $R'_u$  and  $\mathcal{L}_1$ :

$$\mathcal{L}_1 \geq |R'_u| - B|J(R'_u)|$$

Again lemma 1 provides a relation between the size of  $R''_u$  and  $\mathcal{L}_2$ :

$$\mathcal{L}_2 \geq |R''_u| - B|J(R''_u)|$$

Thus, since  $R_u = R'_u \cup R''_u$ , summing the previous inequalities we obtain the following condition:

$$\mathcal{L}_1 + \mathcal{L}_2 \geq k - B(|J(R'_u)| + |J(R''_u)|) \geq k - 2B|J(R_u)| \quad (8)$$

which provides a necessary condition to completely schedule all the cells in  $R$ .

Now we will show that this constraint must be violated for some set  $R_u$ . We partition frame  $\mathcal{L}$  in 6 groups of contiguous timeslots (called periods)  $\{H_1, \dots, H_6\}$ , with  $m \in H_i$ , iff  $[(i - 1)L/6 + 1] \leq m \leq \lfloor iL/6 \rfloor$ . Each cell in  $R_2$  is scheduled in at most couple of periods. First we suppose  $R_2 = R$ . Consider all the  $n$  possible non null (non ordered) couples  $\{H_a, H_b\}$ , with repetitions extracted from the set  $\{H_1, \dots, H_6, \emptyset\}$  (we do not count the couple  $\{\emptyset, \emptyset\}$ ), it results  $n = 27$ .

Thanks to the pigeonhole principle, each input  $i$  can be associated with a couple  $\{H_a^i, H_b^i\}$  of periods in which the considered input has scheduled at least  $\lceil k/n \rceil$  cells, i.e.  $\{\forall i, \exists R_i \text{ and } \{H_a^i, H_b^i\} | R_i \in J^{-1}(i); |R_i| > \lceil k/n \rceil; \forall r \in R_i, \mathcal{ITSA}(r) \subset (H_a^i \cup H_b^i), |\mathcal{ITSA}(r) \cap H_a^i| = 1\}$ . If  $N_a \geq n^2$ , there exists a set of inputs  $N_H$ , with  $|N_H| = n$  and a pair of periods  $\{H_a, H_b\}$ , such that  $\{H_a^i, H_b^i\} = \{H_a, H_b\}$  for all  $i \in N_H$ . As a consequence, the total number of cell to be scheduled in  $\{H_a, H_b\}$  is  $k' = n \lceil k/n \rceil \geq k$ . But, this is in contradiction with (8) for  $k > 6Bn$ ; indeed, since  $|H_a \cup H_b| \leq 2/3k$ , according to (8):

$$\frac{2k}{3} \leq k - 2Bn \rightarrow k \leq 6Bn$$

Now we relax the assumption that  $R_2 = R$  to the more general case in which  $R_2 \subseteq R$ . Since  $|R_2| \geq kN_a/2$ , thanks to lemma 4, there exists a set  $S_a$  of inputs with  $|S_a| \geq N_a/3$  such that, each input in  $S_a$  schedules at least  $\lceil k/4 \rceil$  cells belonging to  $R_2$ . Following the same reasoning as above, each input  $i$  can be associated with a couple  $\{H_a^i, H_b^i\}$  of periods in which the considered input has scheduled at least  $\lceil k/4n \rceil$  cells. If  $N_a \geq 16n^2$ , there exists a set of inputs  $N_H$ , with  $|N_H| = 4n$  and a pair of periods  $\{H_a, H_b\}$ , such that  $\{H_a^i, H_b^i\} = \{H_a, H_b\}$  for all  $i \in N_H$ . But, also in this case, a contradiction with (8) arises for  $k > 24Bn$ ; indeed, since  $|H_a \cup H_b| \leq 2/3k$ , according to (8):

$$\frac{2k}{3} \leq k - 8Bn \rightarrow k < 24Bn$$

the previous theorem can be extended to the case of generalized  $(k, N_a)$ -complex sets.  $\blacksquare$

*Theorem 7:* Consider a CICQ switch with generic  $B \geq 1$  loaded with a generalized  $(k, N_a)$ -complex cell set  $R$ . For sufficiently large values of  $k$  and  $N_a$  no complete  $\mathcal{IS}[R]$  exists with frame length  $L \leq 2k$ , under any fanout-splitting policy.

*Proof:*

If there is an input  $i$  such that  $|J^{-1}(i)| > 2k$  the proof is trivial. Let us consider the case in which  $|J^{-1}(i)| \leq 2k$  for all  $i$ . Repeating exactly the same reasoning of the previous proof and adopting the same notation, we obtain also in this case that, for any set  $R_u \subset R_2$  with  $|R_u| = k$ , it must hold:

$$\mathcal{L}_1 + \mathcal{L}_2 \geq k - 2B|J(R_u)| \quad (9)$$

Thanks to lemma 4 there are at least  $N_a/3$  inputs which transfer at least  $\lfloor k/2 \rfloor$  each; thus repeating the similar reasoning of previous proof we obtain for  $N_a \geq (24n)^2$  with  $n = 21$  and  $k \geq 48Bn$  we obtain a contradiction. ■

The previous theorems can be extended to the case of  $L = Sk$  for any integer  $S > 2$ .

*Theorem 8:* Consider a CICQ switch with generic  $B \geq 1$  loaded with a perfect  $(k, N_a)$ -complex cell set  $R$ . For any given finite  $S$ , for sufficiently large values of  $N_a$  and  $k$  no complete  $\mathcal{IS}[R]$  exists with frame length  $L \leq Sk$ , under any fanout-splitting policy.

*Proof:* This proof has a structure similar to the proof of theorem 6; we only sketch it.

Also in this case we prove the assert by contradiction. Let us assume that a complete  $\mathcal{IS}$  exists in a frame of length  $L = Sk$ .

First notice that, by construction, the average number of per cell transmission opportunity per cell cannot exceed  $S$ .

Consider  $R_S$ , the set of all cells for which  $|\mathcal{ITSA}(r)| \leq S$ . Thanks to lemma 2, at least  $1/S$ -th of the cell set is in  $R_S$ , i.e.  $|R_S| \geq |R|/S = kN_a/S$ .

A generic cell  $r \in R_S$  attempts the transfer to internal buffers at most in  $S$  timeslots,  $(m_1(r), m_2(r), \dots, m_S(r))$  (with  $m_1(r) \leq m_2(r) \leq m_S(r)$ ) and  $m_k(r) \in \{1, 2, \dots, Sk, \text{Null}\}$  where  $m_k(r) = \text{Null}$  means that  $|\mathcal{ITSA}(r)| < k$ .

We further notice that since the input scheduling  $\mathcal{IS}$  is supposed complete, it must be  $|\mathcal{ITSA}(r)| \geq 1$ .

Now consider a set  $R_u \subseteq R_S$  with  $|R_u| = k$  (this set exists for  $N_a \geq S$ ); since  $R$  is a perfect  $(k, N_a)$ -complex cell set, there exists an output  $u$  such that all cells in  $R_u$  are directed to  $u$ . Consider the subsets  $R_u^h$  of  $R_u$  comprising those cells whose fragment directed to  $u$  is transferred in  $m_h(p)$ . By construction, it results  $R_u^h \cap R_u^j = \emptyset, \forall h \neq j$  and  $\cup_h R_u^h = R_u$ .

Let us define  $\mathcal{L}_h = |\mathcal{ITSA}(R_u^h)|$ . Consider the cells in  $R_u^h$ . Lemma 1 provides a relation between the size of  $R_u^h$  and  $\mathcal{L}_h$ :

$$\mathcal{L}_h \geq |R_u^h| - B|J^{-1}(R_u^h)|$$

Thus, summing over  $h$  we get

$$\sum_{i=1}^S \mathcal{L}_h \geq \sum_{h=1}^S [|R_u^h| - B|J^{-1}(R_u^h)|] \geq k - SB|J^{-1}(R_u)| \quad (10)$$

Now we will show that this constraint must be violated for some set  $R_u$ . We partition frame  $F$  of timeslot uniformly in  $(S+1)S$  groups of contiguous timeslots (called periods)  $\{H_1, \dots, H_{(S+1)S}\}$  with  $m \in H_i$  iff  $[(i-1)L/(S+1)S +$

$1] \leq m \leq [iL/(S+1)S]$ . Each cell in  $R_S$  is scheduled in at most  $S$  periods.

First, we suppose  $R_s = R$ . Consider all the  $n$  possible non null (non ordered)  $S$ -uple  $(M_1, M_2, M_3 \dots M_S)$  with repetitions extracted from the set  $\{H_1, H_2, \dots, H_{(S+1)S}, \emptyset\}$  (we do not count the the  $S$ -uple  $(\emptyset, \emptyset, \emptyset \dots \emptyset)$ ). We assume  $M_i \leq M_{i+1}$ , under the conventional assumption that  $H_1 < H_2 < \dots < H_{(S+1)S}$ .

Thanks to the pigeonhole principle, each input  $i$  can be associated with a  $S$ -uple  $\{M_1^i, M_2^i, \dots, M_S^i\}$  of periods in which the considered input has scheduled at least  $\lceil k/n \rceil$  cells, i.e.  $\{\forall i, \exists R_i \text{ and } \{M_1^i, M_2^i, \dots, M_S^i\} \text{ such that } R_i \in J^{-1}(i); |R_i| > \lceil k/n \rceil; \forall r \in R_i, \mathcal{ITSA}(r) \subset \cup_n M_n^i\}$ .

If  $N_a \geq n^2$ , there exists a set of inputs  $N_H$ , with  $|N_H| = n$  and a  $S$ -uple of periods  $(M_1, M_2, \dots, M_S)$ , such that  $(M_1^i, M_2^i, \dots, M_S^i) = (M_1, M_2, \dots, M_S)$  for all  $i \in N_H$ . As a consequence, the total number of cells to be scheduled in  $(M_1, M_2, \dots, M_S)$  is  $k' = n \lceil k/n \rceil \geq k$ .

But, this is in contradiction with (10) for  $k > (S+1)SBn$ ; indeed, since  $|\cup M^i| \leq \frac{Sk}{S+1}$ , according to (10):

$$\frac{Sk}{S+1} \leq k - SBn \rightarrow k \leq (S+1)SBn$$

Now we relax the assumption that  $R_S = R$ . to the more general case in which  $R_S \subseteq R$ . Since  $|R_S| \geq kN_a/S$ , thanks to lemma 4, there exists a set  $S_a$  of inputs with  $|S_a| \geq N_a/S$  such that, each input in  $S_a$  schedules at least  $\lceil k/S \rceil$  cells belonging to  $R_S$ .

Following the same reasoning as above, each input  $i \in S_a$  can be associated with a  $S$ -uple of periods  $(M_1^i, M_2^i, \dots, M_S^i)$  in which the considered input has scheduled at least  $\lceil k/Sn \rceil$  cells. If  $N_a \geq (Sn)^2$ , there exists a set of inputs  $N_H$ , with  $|N_H| = Sn$  and a  $S$ -uple of periods  $(M_1, M_2, \dots, M_S)$ , such that  $(M_1^i, M_2^i, \dots, M_S^i) = (M_1, M_2, \dots, M_S)$  for all  $i \in N_H$ . But, also in this case a contradiction with (10) arises for  $k > (S+1)S^2Bn$ , since  $|\cup M^i| \leq \frac{Sk}{S+1}$ ; indeed according to (10):

$$\frac{Sk}{S+1} \leq k - S^2Bn \rightarrow k \leq (S+1)S^2Bn$$

■  
A further generalization can be obtained considering generalized  $k$ -complex cell sets:

*Theorem 9:* Consider a CICQ switch with generic  $B \geq 1$  loaded with a generalized  $(k, N_a)$ -complex cells set  $R$ . For any given finite  $S$ , for sufficiently large values of  $k$  and  $N_a$  no complete  $\mathcal{IS}[R]$  exists with frame length  $L \leq Sk$ , under any fanout-splitting policy.

*Proof:* We do not report this proof since its is clearly similar the the previous proofs. ■

## APPENDIX II

### SOME USEFUL COMBINATORIAL RESULTS

*Lemma 2:* For a strictly positive integer-valued random variable  $X$  whose average value  $E[X] = \mu$  does not exceed  $m$ ,  $\Pr\{X \leq m\} \geq 1/m$ . The proof of lemma is available in [6]. ■

The following results are extensions of the well-known combinatorics “pigeonhole” principle.

*Lemma 3:* If  $b$  balls are divided into  $z$  bins, it is always possible to find a non-empty subset of  $v$  bins ( $v \leq z$ ) that contains not less than  $\lceil bv/z \rceil$  balls in total.

The proof of this lemma is available in [6]. ■

*Lemma 4:* If  $b$  balls are distributed into  $z$  bins, assuming that each bins contains at most  $k$  balls, for any  $\alpha < 1$ , it is possible to find at least  $\frac{(1-\alpha)\lfloor b/z \rfloor}{k-\alpha\lfloor b/z \rfloor}$  bins containing more than  $\alpha\lfloor b/z \rfloor$  balls each.

*Proof:* Consider all the  $z$  bins ordered by decreasing number of contained balls. Let  $b_i$  the number of balls in bin  $i$ . By hypothesis, it must be  $\sum_{i=1}^z b_i = b$ . First observe that, at least  $b_1$  must be larger than  $\alpha\lfloor b/z \rfloor$ , otherwise  $\sum_{i=1}^z b_i \leq \sum_{i=1}^z \alpha\lfloor b/z \rfloor = \alpha z\lfloor b/z \rfloor < b$  in contradiction with the hypothesis.

Let  $j_0$  the number of bins containing more than  $\alpha\lfloor b/z \rfloor$  balls, it results:

$$\sum_{i=1}^{j_0} b_i + \sum_{i=j_0+1}^z b_i = b \leq \sum_{i=1}^{j_0} b_i + (z - j_0)\alpha\lfloor b/z \rfloor$$

Thus,

$$\sum_{i=1}^{j_0} b_i \geq b - (z - j_0)\alpha\lfloor b/z \rfloor$$

Since at most  $k$  balls can be contained by each bin, it results:

$$j_0 \geq \frac{b - (z - j_0)\alpha\lfloor b/z \rfloor}{k}$$

from which we get the assert. ■