On the Stability of Input-Queued Switches with Speed-Up

Emilio Leonardi, *Member, IEEE*, Marco Mellia, *Student Member, IEEE*, Fabio Neri, *Member, IEEE*, and Marco Ajmone Marsan, *Fellow, IEEE*

***Abstract*—We consider cell-based switch and router architectures whose internal switching matrix does not provide enough speed to avoid input buffering. These architectures require a scheduling algorithm to select at each slot a subset of input buffered cells which can be transferred toward output ports. In this paper, we propose several classes of scheduling algorithms whose stability properties are studied using analytical techniques mainly based upon Lyapunov functions. Original stability conditions are also derived for scheduling algorithms that are being used today in high-performance switch and router architectures.**

***Index Terms*—Input buffered switches, Lyapunov methods, scheduling algorithm, stability.**

I. Introduction

A NUMBER of high-performance IP routers (for example, the CISCO 12 000 [1], the Lucent Cajun [2] family, or the Nortel Versalar TSR45000 [3]) are built around fast cell-based switching fabrics. The design of these high-performance routers generally does not adopt the classical output queueing (OQ) architecture (where cells are stored at the output of the switching fabric), preferring either input queueing (IQ) or combined input/output queueing (CIOQ) structures. The reason is that, in OQ, both the switching fabric and the output (and possibly input) queues in line cards must operate at a speed equal to the sum of the rates of all input lines; since this speed grows linearly with the number of switch ports, the OQ approach is impractical for large switches. Instead, in IQ schemes, all the components of the switch (input interfaces, switching fabric, output interfaces) can operate at a data rate which is compatible with the data rate of input and output lines, and does not grow with the switch size. The traditional performance penalty of IQ architectures is due to head-of-the-line blocking in the case of a single queue per input interface [4], but can be largely reduced by virtual output queueing (VOQ) (also called destination queueing) schemes [5], which organize input buffers in each line card into a set of queues where cells awaiting access to the switching fabric are stored according to their destination output cards.

A major issue in the design of IQ switches is that the access to the switching fabric must be controlled by some form of scheduling algorithm,[1] which operates on a (possibly partial) knowledge of the state of input queues. This means that control information must be exchanged among line cards, either through an additional data path or through the switching fabric itself, and that intelligence and computational complexity must be devoted to the scheduling algorithm, either at a centralized scheduler, or at the line cards, in a distributed manner.

The problem faced by the scheduling algorithm can be formalized as the classical graph theory problem of maximum size or maximum weight matching on the bipartite graph in which nodes represent input and output ports, and edges represent cells to be switched. The optimal solution to this problem is known, but complexity is too large for practical implementations [8]. Several scheduling algorithms for IQ cell switches were proposed and compared in the recent literature [5], [9]–[12], [14]–[18]. They usually aim at *maximal* size or weight matching, which are sub-optimal solution of the maximum size/weight matching at lower complexity, but were shown (using simulation) to provide performances very close to those of OQ architectures at reduced complexity. They are however still relatively demanding in terms of computing power and control bandwidth in switches with a large number of input/output ports.

The complexity of the scheduling algorithm can be partly reduced [19] when the switching fabric, as well as the input and output memories, operate with a moderate speed-up with respect to the data rate of input/output lines. In this case, buffering is required at outputs as well as inputs, and the term "combined input/output queueing" (CIOQ) is used. Obviously, when the speed-up is such that the internal switch bandwidth equals the sum of the data rates on input lines, input buffers are useless.

Moreover, in [19], a speed-up equal to 2 in CIOQ switches, independent of the number of switch ports, was shown to be sufficient to exactly emulate an OQ architecture, at the expense of quite complex scheduling algorithms, whose implementation appears to be problematic. A similar result was proved in [20],[2] while in [21], [22] the authors showed that a limited speed-up is sufficient to emulate work-conserving switches.

Manuscript received February 22, 2000; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Chao. This work was supported in part by a contract with CSELT, and in part by the Italian Ministry for University and Scientific Research. Preliminary versions of this paper were presented at INFOCOM 2000 and at ICC 2000.

The authors are with the Dipartimento di Elettronica, Politecnico di Torino, Torino 10129, Italy (e-mail: leonardi@mail.tlc.polito.it; mellia@mail.tlc.polito.it; neri@mail.tlc.polito.it; ajmone@mail.tlc.polito.it).

Publisher Item Identifier S 1063-6692(01)01313-9.

[1]The term "scheduling algorithm" for switching architectures is used in the literature for two different types of schedulers: switching matrix schedulers and flow-level schedulers [6], [7]. *Switching matrix schedulers* decide which input port is enabled to transmit in a non-purely-output-queueing switch; they avoid blocking and solve contentions within the switching fabric. *Flow-level schedulers* decide which cell flows must be served in accordance to quality-of-service (QoS) requirements. In this paper the term scheduling algorithm is only used to refer to the first class of algorithms.

[2]An error in the algorithm presented in the paper was pointed out and a correction is reported on the web page of the first author.

In previous papers [23], [24], we proved that simpler scheduling algorithms, whose implementation is surely feasible, provide the same throughput performance of OQ with speed-up equal to 2 (although the behavior of an OQ architecture is not exactly emulated), and that a wide class of Maximal Size Matching (MSM) scheduling algorithms (comprising well-known scheduling algorithms, such as i-SLIP [9], [11] and 2DRR [16]), whose implementation is quite simple, also provides the same throughput performance of OQ with speed-up equal to 2. These results provide a solid theoretical background to manufacturers of high-speed switches and routers that will be a major ingredient of future telecommunication infrastructures.

In this paper we present an extended and generalized version of these stability results, proposing and studying simple classes of novel scheduling algorithms as well as proving the stability of well known scheduling algorithms that are being used today in high-performance switch and router architectures. After introducing some definitions and preliminary results in Section II, and our notation and modeling assumptions in Section III, we provide stability results for rate-driven scheduling algorithms in Section IV, for queue-length-driven scheduling algorithms in Section V, for deterministic weighted scheduling algorithms in Section VI, and for maximal size matching scheduling algorithms in Section VII. Finally, we conclude the paper with Section VIII.

In order to simplify the task of the reader, sections are divided into two parts: in the first part we state our definitions, theorems and corollaries; in the second part we provide proofs of theorems and corollaries.

II. Definitions and Preliminary Results

In this section we define three different criteria for the stability of systems of discrete-time queues, we recall some basic results that are useful to prove stability, and we somewhat extend and generalize those, so as to be able to compare two different systems of queues, and to derive the conditions that allow the stability of one system to be inferred from the stability of the other.

Given a system of N discrete-time queues of infinite capacities, let X_n be the row vector of queue lengths at time n, i.e., $X_n = (x_n^1, x_n^2, \ldots, x_n^N)$, where x_n^i is the number of customers in queue i at time n.

Each queue-length evolution is described by $x_{n+1}^i = x_n^i + a_n^i - d_n^i$, where a_n^i represents the number of customers arrived at queue i in time interval $(n, n+1]$, and d_n^i represents the number of customers departed from queue i in time interval $(n, n+1]$. Let $x_0^i = 0$. Let $A_n = (a_n^1, a_n^2, \ldots, a_n^N)$ be the vector of the numbers of arrivals at the N queues, and $D_n = (d_n^1, d_n^2, \ldots, d_n^N)$ be the vector of the numbers of departures from the N queues. With this notation, the equation that describes the evolution of the system of queues is

$$X_{n+1} = X_n + A_n - D_n. \tag{1}$$

We assume that vectors A_n are independent and identically distributed, although this constraint can be relaxed in part.

Given a vector $X = (x_1, x_2, \ldots, x_K)$, we indicate with $\|X\|$ its Euclidean norm: $\|X\| = \sqrt{\sum_{i=1}^K x_i^2}$.

Definition 1: A system of queues achieves **100% throughput** if $\lim_{n\to\infty}(X_n/n) = \lim_{n\to\infty}(1/n)\sum_{i=0}^n (A_i - D_i) = 0$ with probability 1.

Definition 2: A system of queues is **weakly stable** if, for every $\epsilon > 0$, there exists $B > 0$ such that $\lim_{n\to\infty} \Pr\{\|X_n\| > B\} < \epsilon$ (where $\Pr\{E\}$ is the probability of event E).

Definition 3: A system of queues is **strongly stable** if $\lim_{n\to\infty} \sup E[\|X_n\|] < \infty$.

Note that strong stability implies weak stability, and that weak stability implies 100% throughput. Indeed, the 100% throughput property allows queue lengths to indefinitely grow with sub-linear rate, while the weak stability property entails that the servers in the system of queues are able to process the whole offered load, but the delay experienced by customers can be unbounded. Strong stability implies, in addition, the boundedness of the average delay of customers.

We assume that the process describing the evolution of the system of queues is an irreducible discrete-time Markov chain (DTMC), whose state vector at time n is[3] $Y_n = (X_n, K_n)$, $Y_n \in \mathbb{N}^M$, $X_n \in \mathbb{N}^N$, $K_n \in \mathbb{N}^{N'}$, and $M = N + N'$. Y_n is the combination of vector X_n and a vector K_n of N' integer parameters. Let H be the state space of the DTMC, obtained as a subset of the Cartesian product of the state space H_X of X_n and the state space H_K of K_n.

From definition 2 we can immediately see that if all states Y_n are positive recurrent, the system of queues is weakly stable; however, the converse is generally not true, since queue lengths can remain finite even if the states of the DTMC are not positive recurrent due to instability in the sequence of parameter $\{K_n\}$.

Note that most systems of discrete-time queues of interest can be described with models that fall in the DTMC class.

The following general criterion for the (weak) stability of systems that can be described with a DTMC is therefore useful in the design of scheduling algorithms. This theorem is a straightforward extension of Foster's Criterion; see [25]–[27].

Theorem 1: Given a system of queues whose evolution is described by a DTMC with state vector $Y_n \in \mathbb{N}^M$, if a lower bounded function $V(Y_n)$, called Lyapunov function, $V: \mathbb{N}^M \to \mathbb{R}$ can be found such that $E[V(Y_{n+1})|Y_n] < \infty\ \forall Y_n$ and there exist $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$ such that $\forall\, \|Y_n\| > B$

$$E[V(Y_{n+1}) - V(Y_n)|Y_n] < -\epsilon \tag{2}$$

then all states of the DTMC are positive recurrent and the system of queues is weakly stable.

Note that an explicit dependence of the Lyapunov function on the time index n is allowed, so that it is possible to explicitly write $V(Y_n) = V(Y_n, n)$.

If the state space H of the DTMC is a subset of the Cartesian product of the denumerable state space H_X and a *finite* state space H_K, the stability criterion can be slightly modified, since the stability of the system can be inferred only from the queue-length state vector X_n.

Corollary 1: Given a system of queues whose evolution is described by a DTMC with state vector $Y_n \in \mathbb{N}^M$, and whose state space H is a subset of the Cartesian product of a denumer-

[3]In this paper $\mathbb{N}$ denotes the set of nonnegative integers, $\mathbb{R}$ denotes the set of real numbers, and $\mathbb{R}^+$ denotes the set of nonnegative real numbers.

able state space H_X and a finite state space H_K, then, if a lower bounded function $V(X_n)$, called Lyapunov function, $V\colon \mathbb{N}^N \to \mathbb{R}$ can be found such that $E[V(X_{n+1})|Y_n] < \infty\ \forall Y_n$ and there exist $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$ such that $\forall Y_n\colon \|X_n\| > B$

$$E[V(X_{n+1}) - V(X_n)|Y_n] < -\epsilon \tag{3}$$

then all states of the DTMC are positive recurrent.

In this case, the system of discrete-time queues is weakly stable iff all states of the DTMC are positive recurrent.

In the remainder of this paper we restrict our analysis to the class of systems of queues for which Corollary 1 applies.

To extend the previous result, we obtain the following criterion for *strong* stability:

Theorem 2: Given a system of queues whose evolution is described by a DTMC with state vector $Y_n \in \mathbb{N}^M$, and whose state space H is a subset of the Cartesian product of a denumerable state space H_X and a finite state space H_K, then, if a lower bounded function $V(X_n)$, called Lyapunov function, $V\colon \mathbb{N}^N \to \mathbb{R}$ can be found such that $E[V(X_{n+1})|Y_n] < \infty\ \forall Y_n$ and there exist $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$ such that $\forall Y_n\colon \|X_n\| > B$

$$E[V(X_{n+1}) - V(X_n)|Y_n] < -\epsilon\|X_n\| \tag{4}$$

then the system of queues is strongly stable.

A class of Lyapunov functions is of particular interest:

Corollary 2: Given a system of queues as in Theorem 2, then, if there exist a symmetric copositive[4] matrix $W \in \mathbb{R}^{N\times N}$, and two positive real numbers $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$, such that, given the function $V(X_n) = X_n W X_n^T$, $\forall Y_n\colon \|X_n\| > B$ it holds

$$E[V(X_{n+1}) - V(X_n)|Y_n] < -\epsilon\|X_n\| \tag{5}$$

then the system of queues is strongly stable. In addition, all the polynomial moments of the queue-length distribution are finite.

This is a rephrasing of the results presented in [28, Sect. IV].

In particular, the identity matrix I is a symmetric positive semidefinite matrix, hence a copositive matrix; thus it is possible to state the following

Corollary 3: Given a system of queues as in Theorem 2, then, if there exists $\epsilon \in \mathbb{R}^+$, $B \in \mathbb{R}^+$ such that $\forall Y_n\colon \|X_n\| > B$

$$E[X_{n+1}X_{n+1}^T - X_n X_n^T|Y_n] < -\epsilon\|X_n\| \tag{6}$$

then the system of queues is strongly stable, and all the polynomial moments of the queue-length distribution are finite.

A system of discrete-time queues is stable if all its queues are stable; the standard approach to prove stability in queueing systems is based on checking that the average number of arrivals when the server is busy is smaller than the average number of departures. This formulation is provided by the next theorem. Its proof is given in terms of the Lyapunov function because it is convenient for the extension to more complex setups that we shall consider later in this paper.

Theorem 3: Consider a systems of queues composed of N queues. If there exists $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$ such that

$$E[(a_n^i - d_n^i)|x_n^i > 0] < -\epsilon \tag{7}$$

$\forall i = 1, \ldots, N$ then the system of queues is strongly stable.

[4]An $N \times N$ matrix Q is copositive if $XQX^T \geq 0\ \forall X \in \mathbb{R}^{+N}$.

The following Theorem 4 is particularly important for the rest of the paper, since it will allow us to compare different scheduling policies from the point of view of stability.

Theorem 4: Consider two systems of queues S_1 and S_2, each one comprising N queues. Let the arrival processes at each queue for both systems be statistically identical. Let $X_{S1,n}$, $D_{S1,n}$, and $X_{S2,n}$, $D_{S2,n}$, be the queue-length and departure vectors of S_1 and S_2 at time n, respectively. Assume that (6) holds for $X_{S1,n}$, and there exist $\epsilon \in \mathbb{R}^+$, $B \in \mathbb{R}^+$ such that for $\|X_{S1,n}\| > B$ and $\|X_{S2,n}\| > B$

$$E[D_{S1,n}X_{S1,n}^T - D_{S2,n}X_{S2,n}^T|Y_{S1,n} = Y_{S2,n}] < -\epsilon \tag{8}$$

then system S_2 is strongly stable and all the polynomial moments of its queue-length distribution are finite.

III. Proofs for Section II

Proof of Theorem 2: Since the assumptions of Theorem 1 are satisfied, every state of the DTMC is positive recurrent and the DTMC is weakly stable. In addition, to prove that the system is strongly stable, we shall show that $\lim_{n\to\infty} \sup E[\|X_n\|] < \infty$.

Let H_B be the set of values taken by Y_n for which $\|X_n\| \leq B$ [where (4) does not apply]. It is easy to prove that H_B is a compact set. Outside this compact set, (4) holds, i.e.

$$E[V(X_{n+1}) - V(X_n)|Y_n] < -\epsilon\|X_n\|.$$

Considering all Y_n's that do not belong to H_B, we obtain

$$E[V(X_{n+1}) - V(X_n)|Y_n \notin H_B] < -\epsilon E[\|X_n\||Y_n \notin H_B].$$

Instead, for $Y_n \in H_B$, being H_B a compact set

$$E[V(X_{n+1})|Y_n \in H_B] \leq M < \infty$$

where M is the maximum value taken by $E[V(X_{n+1})|Y_n]$ for Y_n in H_B.

By combining the two previous expressions, we obtain

$$\begin{aligned} &E[V(X_{n+1})] \\ &\quad < M \Pr\{Y_n \in H_B\} + \Pr\{Y_n \notin H_B\} \\ &\qquad \cdot \{E[V(X_n)|Y_n \notin H_B] - \epsilon E[\|X_n\||Y_n \notin H_B]\} \\ &\quad < M + E[V(X_n)] - \epsilon E[\|X_n\|] + M_0. \end{aligned}$$

M_0 is a constant such that $M_0 \geq \{-E[V(X_n)|Y_n \in H_B] + \epsilon E[\|X_n\||Y_n \in H_B]\} \Pr\{Y_n \in H_B\}$. Note that M_0 is finite, being H_B a compact set.

By summing over all n from 0 to $N_0 - 1$, we obtain

$$E[V(X_{N_0})] < N_0 M + E[V(X_0)] - \epsilon \sum_{n=0}^{N_0-1} E[\|X_n\|] + N_0 M_0.$$

Thus, for any N_0, we can write

$$\begin{aligned} &\frac{\epsilon}{N_0} \sum_{n=0}^{N_0-1} E[\|X_n\|] \\ &\quad < M + \frac{1}{N_0} E[V(X_0)] - \frac{1}{N_0} E[V(X_{N_0})] + M_0. \end{aligned}$$

$E[V(X_{N_0})]$ is lower bounded by definition; assume $E[V(X_{N_0})] > K_0$. Hence

$$\frac{\epsilon}{N_0} \sum_{n=0}^{N_0-1} E[\|X_n\|] < M + \frac{1}{N_0} E[V(X_0)] - \frac{K_0}{N_0} + M_0.$$

For $N_0 \to \infty$, being $E[V(X_0)]$ and K_0 finite, we can write

$$\frac{\epsilon}{N_0} \sum_{n=0}^{N_0-1} E[\|X_n\|] < M + M_0.$$

Hence $\lim_{N_0 \to \infty} (1/N_0) \sum_{n=0}^{N_0-1} E[\|X_n\|]$ is bounded. Since the DTMC Y_n has positive recurrent states, there exists $\lim_{n\to\infty} E[\|X_n\|]$. Furthermore, if the sequence $E[\|X_n\|]$ is convergent, the sequence $(1/n) \sum_{i=0}^{n-1} E[\|X_i\|]$ converges to the same limit (being the Cesaro sum)

$$\lim_{n\to\infty} E[\|X_n\|] = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} E[\|X_i\|].$$

But the right-hand side was seen to be bounded; hence $\lim_{n\to\infty} E[\|X_n\|] < \infty$. ■

Proof of Theorem 3: Starting from (6) we can write

$$\begin{aligned} &E[X_{n+1}X_{n+1}^T - X_n X_n^T | Y_n] \\ &\quad = E[2(A_n - D_n)X_n^T + (A_n - D_n)(A_n - D_n)^T | Y_n]. \end{aligned}$$

For $\|Y_n\|$ (and $\|X_n\|$) growing to infinity, since the number of arrivals and departures in time interval n is bounded, we have

$$\lim_{\|X_n\|\to\infty} \frac{E[(A_n - D_n)(A_n - D_n)^T | Y_n]}{\|X_n\|} = 0.$$

As a consequence

$$\begin{aligned} &\lim_{\|X_n\|\to\infty} \frac{E[X_{n+1}X_{n+1}^T - X_n X_n^T | Y_n]}{\|X_n\|} \\ &\qquad\qquad = \lim_{\|X_n\|\to\infty} \frac{2E[(A_n - D_n)X_n^T | Y_n]}{\|X_n\|} \end{aligned}$$

and from (7) we have

$$\lim_{\|X_n\|\to\infty} \frac{E[X_{n+1}X_{n+1}^T - X_n X_n^T | Y_n]}{\|X_n\|} < -\epsilon' \frac{\max_i x_n^i}{\|X_n\|} < -\epsilon''.$$

Thus, for some $B \in \mathbb{R}^+$, $\epsilon \in \mathbb{R}^+$, $\|X_n\| > B$

$$E[X_{n+1}X_{n+1}^T - X_n X_n^T | Y_n] < -\epsilon \|X_n\|.$$

■

Proof of Theorem 4: If (6) holds for $X_{S1,\,n}$, then for some $B_1 \in \mathbb{R}^+$, $\epsilon \in \mathbb{R}^+$

$$\begin{aligned} &E[X_{S1,\,n+1}X_{S1,\,n+1}^T - X_{S1,\,n}X_{S1,\,n}^T | Y_{S1,\,n}] \\ &\quad < -\epsilon \|X_{S1,\,n}\| \qquad \forall \|X_{S1,\,n}\| > B_1. \end{aligned}$$

But, as shown in the proof of Theorem 3,

$$\begin{aligned} &\lim_{\|X_{S1,n}\|\to\infty} \frac{E[X_{S1,n+1}X_{S1,n+1}^T - X_{S1,n}X_{S1,n}^T | Y_{S1,n}]}{\|X_{S1,n}\|} \\ &\quad = \lim_{\|X_{S1,n}\|\to\infty} \frac{2E[(A_{S1,n} - D_{S1,n})X_{S1,n}^T] | Y_{S1,n}]}{\|X_{S1,n}\|} < -\epsilon. \end{aligned}$$

For system S_2 being $Y_{S1} = Y_{S2}$, hence $X_{S1} = X_{S2}$, and being $E[A_{S1,\,n}] = E[A_{S2,\,n}]$, from (8) for $\|X_{S2,\,n}\| > B$ we get

$$\begin{aligned} &\lim_{\|X_{S2,n}\|\to\infty} \frac{E[X_{S2,n+1}X_{S2,n+1}^T - X_{S2,n}X_{S2,n}^T | Y_{S2,n}]}{\|X_{S2,n}\|} \\ &\quad < \lim_{\|X_{S1,n}\|\to\infty} \frac{2E[(A_{S1,n} - D_{S1,n})X_{S1,n}^T] | Y_{S1,n}]}{\|X_{S1,n}\|} \\ &\quad < -\epsilon. \end{aligned}$$

Corollary 3 applies to system $S2$. ■

IV. Notation and Modeling Assumptions

We consider CIOQ cell-based switches with P input ports and P output ports, all at the same cell rate (and we call them $P \times P$ CIOQS). The switching fabric is assumed to be nonblocking and bufferless, i.e., cells can only be stored at the switch input and/or output ports. At each input port, cells are stored according to a VOQ policy: one separate queue is maintained for each output port. Thus, the total number of queues in the switch is $N = P^2$. Let q_{ij} be the queue at input port i storing cells directed to output port j.

Although the internal switch speedup can in general be obtained in several domains (time, space, wavelength, etc.), we assume to operate in the time domain, and we say that the CIOQS achieves speed-up S when the cell transfer rate through the switching fabric is S times faster with respect to the rate of external input/output lines. Note that this requires the rate *out of input queues* as well as the rate *into output queues* to be S times the external input/output lines rate.

We call *external time slot* the time needed to transmit a cell at the data rate of the input/output lines. The *internal time slot* is, instead, the time needed to transmit a cell at the data rate of the switching fabric. The external time slot is S times longer than the internal time slot. Let r_{ij} be the average arrival rate of cells at queue q_{ij} in cells/external slot.

Definition 4: The traffic pattern loading a CIOQS is admissible if for each input port and each output port the total arrival rates in cells/external slot are less than 1, that is

$$\begin{aligned} r_i^{\text{in}} &= \sum_{j=1}^{P} r_{ij} < 1 \qquad i = 1, 2, \ldots, P \\ r_j^{\text{out}} &= \sum_{i=1}^{P} r_{ij} < 1 \qquad j = 1, 2, \ldots, P. \end{aligned} \tag{9}$$

During each internal time slot, some cells may be transferred from input queues to output ports. The set of cells transferred during one internal time slot must satisfy two constraints: i) at each internal time slot, at each input, at most one cell can be extracted from the VOQ structure, and ii) at each internal time slot, at most one cell can be transferred toward each output.

Definition 5: A set of cells extracted from queues q_{ij} is a set $B = \{b_{ij}\}$ of noncontending cells (also called a switching matrix) if

$$\sum_{j=1}^{P} b_{ij} \le 1 \quad \forall i \qquad \text{and} \qquad \sum_{i=1}^{P} b_{ij} \le 1 \quad \forall j \tag{10}$$

where b_{ij} is the number of cells extracted from q_{ij}.

In a CIOQS with speed-up S, a set of noncontending cells can be transferred from input queues to output ports during each internal time slot, so that S sets of noncontending cells can be transferred during each external time slot.

V. Rate-Driven Scheduling Algorithms

In this section we consider very simple scheduling algorithms, which determine the set of noncontending cells that are transferred from input queues to output ports in each internal time slot with a random selection based on the values of the average arrival rates r_{ij} of cells at queues q_{ij}, measured in cells/external slot. Let

$$p_{ij} = \begin{cases} 0 & \text{if } \max\left(\sum_j r_{ij}, \sum_i r_{ij}\right) = 0 \\ \dfrac{r_{ij}}{\max\left(\sum_j r_{ij}, \sum_i r_{ij}\right)} & \text{otherwise.} \end{cases}$$

From (9), we obtain $p_{ij} \geq r_{ij}$, and

$$p_i = \sum_{j=1}^{P} p_{ij} \leq 1 \qquad i = 1, \ldots, P.$$

Definition 6: A CIOQS adopts a **random rate-driven** (RRD) scheduling algorithm (SA) if the selection of the set of noncontending cells to be transferred from inputs to outputs at each internal time slot is performed according to the following algorithm:

1. At each internal time slot, the ith input, within its own VOQ structure, chooses queue q_{ij} with probability p_{ij}; with probability $(1 - p_i) \geq 0$, no queue in the VOQ is chosen for cell transfer. Queue q_{ij} is the "candidate" of input i to attempt a cell transfer (toward output j).
2. Among the contending candidate input queues storing cells directed to the same output, only one is enabled to transfer its cell. The choice among contending candidate input queues is performed at random, according to a uniform distribution; i.e., if there are k candidates for the same output j, only one input receives a transfer grant, and the probability of receiving the grant is $1/k$ for each contending candidate input queue.

Theorem 5: Under any admissible load, a CIOQS adopting a RRD-SA is strongly stable for any speed-up $S \geq 2$.

Considering a CIOQS under a uniform traffic load, we have:

Corollary 4: Under any admissible *uniform* load, a CIOQS adopting a RRD-SA is strongly stable for any speed-up $S \geq 1/(1-(1-(1/P))^P)$, and for $P \to \infty$, i.e., for switches with very large number of ports, a speed-up $S \geq e/(e-1)$ guarantees the strong stability of the system.

If we use queue lengths (instead of probabilities) to break the ties among the contending candidates, we obtain a different scheduling algorithm.

Definition 7: A CIOQS adopts a **longest queue rate-driven** (LQRD) scheduling algorithm if the selection of cells to be transferred from inputs to outputs is performed according to the following algorithm:

1. (As in Definition 6)
2. Among the contending candidate input queues storing cells directed to the same output, only one among the longest queues is enabled to transfer its cell. Ties are broken with a uniform random choice.

Corollary 5: Under any admissible load, a CIOQS adopting a LQRD-SA is strongly stable for any speed-up $S \geq 2$.

To improve the performance of the scheduling algorithm, we can consider only the set of not-empty queues:

Definition 8: A CIOQS adopts an **enhanced longest queue rate-driven** (ELQRD) scheduling algorithm if the selection of cells to be transferred from inputs to outputs is performed according to the following algorithm:

1. At each internal time slot, the ith input, within its own VOQ structure, chooses a *nonempty* queue q_{ij} with a probability proportional to r_{ij}. Queue q_{ij} is the "candidate" of input i to attempt a cell transfer (toward output j).
2. (As in Definition 7).

Corollary 6: Under any admissible load, a CIOQS adopting a ELQRD-SA is strongly stable for any speed-up $S \geq 2$.

VI. Proofs for Section V

To prove the theorems presented in Section V, we first have to derive some preliminary results. Let A be a finite set of nonnegative real numbers a_i, such that the sum of all elements in A is not greater than one; let also $N = |A|$ be the number of elements of A, i.e., $A = \{a_i \in \mathbb{R}^+, \sum_{j=1}^N a_j \leq 1\}_{i=1}^N$.

Let $\alpha^{[k]}$ be a subset of A such that $|\alpha^{[k]}| = k$, $0 \leq k \leq N$. Let $\alpha^{[0]} = \emptyset$. Let A_k be the set of all possible subsets $\alpha^{[k]}$ of A , i.e., $A_k = \{\alpha^{[k]} \subseteq A, |\alpha^{[k]}| = k, 0 \leq k \leq N\}$. It is easy to see that $|A_k| = \binom{N}{k}$. Let 2^A denote the power set of A, i.e., $2^A = \{A_k\}_{k=0}^N$.

Definition 9: Given $\alpha \subseteq A$, let $f(\alpha)$ be a function $2^A \to \mathbb{R}$ such that $f(\alpha) = \prod_{a \in \alpha} a$; let $f(\emptyset) = 1$.

Definition 10: Given $\alpha \subseteq A$, let $\overline{f}(\alpha)$ be a function $2^A \to \mathbb{R}$, such that $\overline{f}(\alpha) = \prod_{a \in \alpha}(1-a)$; let $\overline{f}(\emptyset) = 1$.

Definition 11: Let $F_k(A) = \sum_{\alpha^{[k]} \in A_k} f(\alpha^{[k]})$.

Proposition 1: For each set A

$$F_{k+1}(A) < \frac{1}{k+1} F_k(A) \qquad \forall k < N.$$

Proof: By definition

$$\begin{aligned} F_{k+1}(A) &= \sum_{\alpha^{[k+1]} \in A_{k+1}} f(\alpha^{[k+1]}) \\ &= \sum_{i=0}^{k} \frac{1}{k+1} \sum_{\alpha^{[k+1]} \in A_{k+1}} f(\alpha^{[k+1]}) \\ &= \frac{1}{k+1} \sum_{\alpha^{[k+1]} \in A_{k+1}} \sum_{i=0}^{k} f\left(\alpha^{[k+1]}\right). \end{aligned} \tag{11}$$

We can group all the $(k+1)\binom{N}{k+1}$ terms in $(k+1)/(N-k)\binom{N}{k+1} = \binom{N}{k}$ subsums, each one comprising $N-k$ different terms. A bijective correspondence between sets $\alpha^{[k]} \in A_k$ and subsums is established according to the following rule: each subsum comprises the $N-k$ terms of (11) associated with the $N-k$ different sets $\alpha^{[k+1]}$ so that $\alpha^{[k]} \subset \alpha^{[k+1]}$. It is thus possible to write

$$F_{k+1}(A) = \sum_{\alpha^{[k]} \in A_k} \sum_{\substack{\alpha^{[k+1]} \in A_{k+1} \\ \alpha^{[k]} \subset \alpha^{[k+1]}}} \frac{1}{k+1} f\left(\alpha^{[k+1]}\right). \tag{12}$$

Since all the elements $a_i \in A$ are nonnegative and their sum is less or equal to 1, for each set $\alpha^{[k]}$

$$\sum_{\alpha^{[k+1]} \supset \alpha^{[k]}} f\left(\alpha^{[k+1]}\right) < f\left(\alpha^{[k]}\right). \tag{13}$$

As a consequence, by substituting (13) in (12), we obtain

$$F_{k+1}(A) < \sum_{\alpha^{[k]} \in A_k} \frac{1}{k+1} f\left(\alpha^{[k]}\right) = \frac{1}{k+1} F_k(A). \tag{14}$$

■

Proof of Theorem 5: Denote by a_n^{ij} and d_n^{ij} the numbers of arrivals and departures, respectively, during time slot n at queue q_{ij}. The proof proceeds from the fact that, for all nonempty queues q_{ij}, it is possible to find $\epsilon \in \mathbb{R}^+$ such that $E[a_n^{ij} - d_n^{ij} | x_n^{ij} > 0] < -\epsilon$, i.e., $E[a_n^{ij} | x_n^{ij} > 0] < E[d_n^{ij} | x_n^{ij} > 0] - \epsilon$; this is sufficient to state that the system of queues is strongly stable for Theorem 3.

Since with a RRD-SA the selection of the set of noncontending cells is state-independent and memoryless, the evaluation of d_n^{ij} is easy. In each internal time slot, the number of cells leaving a queue can be either 0 or 1. Thus, $E[d_n^{ij}]$ equals the probability Q_{ij} that a cell from queue q_{ij} is selected for the transfer. Consider a particular output r; the probability that the tth input queue leading to r is selected by the input is p_{tr}.

Since all choices are state-independent, p_{tr} is the probability that queue q_{tr} is selected in any slot. Once selected, q_{tr} is granted the transfer with probability 1 if no other queue storing cells directed to output r is selected by other inputs; the queue is granted the transfer with probability 1/2 if only one other queue storing cells directed to output r is selected, and so on. Since all choices performed at each input are statistically independent from each other, joint probabilities can be easily evaluated as the product of marginal probabilities.

Let A be the set of all p_{ir} except p_{tr}, i.e., $A = \{p_{ir}, i = 1, \ldots, P, i \neq t\}$, $|A| = P-1$. Then, recalling Definitions 9 and 10, $Q_{tr} = p_{tr} L_{tr}$ where

$$L_{tr} = \sum_{k=0}^{P-1} \frac{1}{(k+1)} \sum_{\alpha^{[k]} \in A_k} f\left(\alpha^{[k]}\right) \overline{f}\left(A \backslash \alpha^{[k]}\right)$$

is the probability of serving queue q_{tr} once selected.

Using a speed-up factor equal to S, the average number of times a cell leaves any given input queue in an external time slot is equal to S times Q_{tr}.

In order to prove that $E[a_n^{tr}] < E[d_n^{tr}] - \epsilon < E[d_n^{tr}]$ (note that $E[a_n^{tr}]$ and $E[d_n^{tr}]$ are numbers: they do not depend on n) for speed-up equal to or greater than 2, it is sufficient to show that $Q_{tr} > (1/2)E[a_n^{tr}] = (1/2)r_{tr}$, i.e., $Q_{tr}/E[a_n^{tr}] > 1/2$, $\forall t, r$.

Since $p_{tr} \geq r_{tr}$

$$\frac{Q_{tr}}{E[a_n^{tr}]} = \frac{Q_{tr}}{r_{tr}} \geq \frac{Q_{tr}}{p_{tr}} = L_{tr}. \tag{15}$$

Moreover

$$L_{tr} \geq \sum_{k=0}^{1} \frac{1}{k+1} \sum_{\alpha^{[k]} \in A_k} f\left(\alpha^{[k]}\right) \overline{f}\left(A \backslash \alpha^{[k]}\right) \tag{16}$$

since all terms in L_{tr} are nonnegative. By explicitly writing the sums in (16) and by grouping all products of i elements of set A, after algebraic manipulations, it is possible to show that

$$\sum_{k=0}^{1} \frac{1}{k+1} \sum_{\alpha^{[k]} \in A_k} f\left(\alpha^{[k]}\right) \overline{f}\left(A \backslash \alpha^{[k]}\right) = 1 + \sum_{i=1}^{P-1} (-1)^i \left(1 - \frac{i}{2}\right) F_i(A) \tag{17}$$

where Definition 11 is used.

Since $\sum_{r=1}^{P} p_{ir} \leq 1$ for construction, Proposition 1 applies, and it is possible to see that the second term of the sum at the right-hand side of (17) is larger than the third, and that the fourth is larger than the fifth, and so on. This means that it is possible to retain only the first term of the summation, and to write

$$L_{tr} \geq 1 + \sum_{i=1}^{P-1} (-1^i) \left(1 - \frac{i}{2}\right) F_i(A) \geq 1 - \frac{1}{2} F_1(A) \geq \frac{1}{2}. \tag{18}$$

We can now combine (15) and (18) to obtain $Q_{tr}/p_{tr} = L_{tr} \geq 1/2$. ■

Fig. 1 plots L_{3j} for the third input and a generic output j in a switch with $P = 4$ input and output ports, versus different values of arrival rates r_{ij}, when output j is at maximum admissible load: $\sum_{i=1}^{4} r_{ij} = 1$. Since $L_{3j} S > 1$ guarantees stability, $1/L_{3j}$ indicates a lower bound to the value of speedup that guarantees stability. Note that $1/L_{3j}$ is always smaller than 2, and that it is minimum for the most balanced traffic condition, i.e., for $r_{1j} = r_{2j} = r_{4j} = (1 - r_{3j})/3 \approx 0.33$.

Proof of Corollary 5: The algorithm for choosing at any internal time slot the set C_n of candidate queues, given the state of the system of queues, X_n, is the same as in RRD-SA. As a consequence, the probability $P(C_n | X_n)$ that a particular set C_n of candidate queues is selected by the inputs is the same

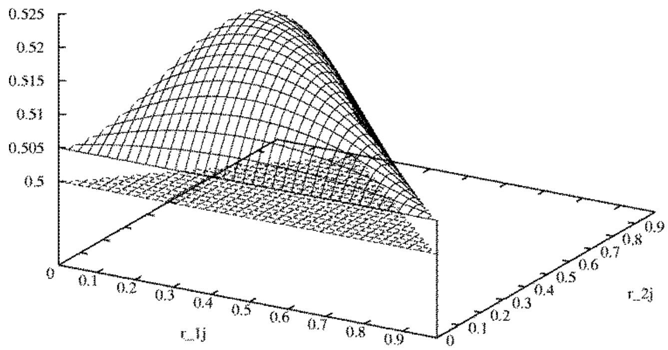

Fig. 1. 4×4 switch using the RRD scheduling algorithm under uniform traffic pattern: probability L_{3j} that VOQ q_{3j} is served in an internal time slot once selected versus possible values of rates r_{1j}, r_{2j}; $r_{3j}=0.01$ and $r_{4j}=1-r_{1j}-r_{2j}-r_{3j}$. Dashed lines show the admissible region for rates r_{1j} and r_{2j}.

for both policies. Given a set C_n of candidate queues selected by inputs, the output contention resolution policy implemented by LQRD-SA guarantees that $E[D_n(\mathrm{LQRD})X_n^T|X_n, C_n]=\max_{D_n} E[D_nX_n^T|X_n, C_n]$. As a consequence

$$\begin{aligned}&E[D_n(RRD)X_n^T-D_n(\mathrm{LQRD})X_n^T|X_n]\\&\quad=\sum_{C_n}E[D_n(RRD)X_n^T-D_n(\mathrm{LQRD})X_n^T|X_n, C_n]\\&\qquad\cdot P(C_n|X_n)\le 0\qquad\forall X_n.\end{aligned}$$

Hence, for Theorem 4, the system is strongly stable. ■

Proof of Corollary 6: Note that, given X_n, ELQRD-SA guarantees that the size of the sets of *nonempty* candidate queues is maximal (i.e., the size of all the sets of candidate queues is equal to the number of inputs with at least one nonempty queue). For each nonempty queue, the probability of being selected as candidate under ELQRD-SA is therefore not smaller than under LQRD-SA.

Indeed, it is possible to perfectly emulate the statistical distribution of candidate sets of ELQRD-SA starting from the candidate sets generated with LQRD-SA and completing each nonmaximal set to obtain a maximal one. To prove it, it is sufficient to build the relation $R(X_n)$ between the sets of candidate queues obtained with policies LQRD-SA and ELQRD-SA for each queue configuration X_n:

- each maximal candidate set $C_n(\mathrm{LQRD})=C_1$ is put in correspondence with set $C_n(\mathrm{ELQRD})=C_2$, so that $C_1=C_2$;
- each nonmaximal candidate set C_1 is put in correspondence with all sets C_2, such that $C_1\subseteq C_2$.

With each pair of sets (C_1, C_2), we associate the probability $P_{\mathrm{ELQRD}}(C_2|C_1)$ that C_2 is obtained according to ELQRD-SA, given that some inputs have already chosen their candidate queue (C_1 is the nonempty set of candidate queues that have been already chosen by some inputs). Note that $\sum_{C_2}P_{\mathrm{ELQRD}}(C_2|C_1)=1$.

It is possible to see that, starting from the candidate sets C_1 generated with LQRD-SA, and completing each nonmaximal set by applying $R(X_n)$ (i.e., choosing a set C_2 in correspondence with C_1 according to the associated probability distribution), the statistical distribution of the ELQRD-SA candidate sets is perfectly emulated.

Given a set C_n of nonempty candidate queues selected by inputs, the output contention resolution algorithm implemented at the outputs for both policies guarantees that $E[D_nX_n^T|X_n, C_n]=\max_{D_n}E[D_nX_n^T|X_n, C_n]$. Thus, given two sets of candidate queues C_1 and C_2 such that $C_1\subset C_2$, then $E[D_nX_n^T|X_n, C_1]\le E[D_nX_n^T|X_n, C_2]$.

As a consequence, Theorem 4 applies and the system of queues is proven to be strongly stable. ■

VII. Queue-Length-Driven Scheduling Algorithms

In this section we prove that a simple scheduling algorithm that determines the set of noncontending cells to be transferred from inputs to outputs in each internal time slot with a random selection based on queue lengths is stable for any speed-up value greater than 2. Before reaching this point, however, we need to introduce some definitions and to derive some preliminary results.

Definition 12: Let U be the set of $V\in\mathbb{R}^{+N}$ such that

$$\begin{aligned}&\sum_{i=1}^{P}V_{i+jP}\le 1\qquad j=0,\ldots,P-1\\&\sum_{j=0}^{P-1}V_{i+jP}\le 1\qquad i=1,\ldots,P.\end{aligned}\tag{19}$$

Definition 13: Given a vector $V\neq 0$, $V\in\mathbb{R}^{+N}$, let $\tilde{V}$ be the maximal vector parallel to V in U, i.e., $\tilde{V}\in U$, $k\in\mathbb{R}$

$$\tilde{V}=\max_k kV.$$

Definition 14: Given a vector $V\neq 0$, $V\in\mathbb{R}^{+N}$, define

$$\hat{V}=\frac{V}{||V||}=\frac{\tilde{V}}{||\tilde{V}||}.$$

Definition 15: Let Γ_V be the symmetric matrix associated with the projection operator along the direction of $\hat{V}$, i.e.

$$\Gamma_V=\hat{V}^T\hat{V}.$$

Indeed, $X\Gamma_V=(X\hat{V}^T)\hat{V}$ for X, $V\in\mathbb{R}^N$ and $X\Gamma_X=X$.

The following theorem states that, if the vector of the average departure rates $E[D_n]$ from the VOQ is parallel to the queue-length vector X, and longer than $\tilde{X}$, then the CIOQS is stable.

Theorem 6: In a CIOQS with VOQ at each input, a scheduling algorithm such that $E[D_n]=\tilde{X}_n(1+\alpha)$ is strongly stable for each $\alpha\in\mathbb{R}^+$.

Moreover, if the average departure rate vector $E[D_n]$ is proportional to the maximal queue-length vector incremented by a *positive* vector D', then the CIOQS is stable:

Theorem 7: In an CIOQS with VOQ at each input, a scheduling algorithm such that $E[D_n]=\tilde{X}_n(1+\alpha)+D'$ with $D'\in\mathbb{R}^{+N}$ is strongly stable for each $\alpha\in\mathbb{R}^+$.

Note that Theorem 6 does not imply Theorem 7, since the choice of D' affects the evolution of $\tilde{X}_n$.

The following scheduling algorithm is similar to RRD, but the selection rates are now derived from queue lengths, rather than average arrival rates:

Definition 16: A CIOQS adopts a **longest-queue-driven** (LQD) scheduling algorithm if the selection of the set of noncontending cells to be transferred from inputs to outputs at each internal time slot is performed according to the following algorithm.

1. At each internal time slot n, the ith input, within its own VOQ structure, chooses queue q_{ij} with a probability proportional to the queue-length x_n^{ij}. The probability[5] of selecting queue q_{ij} is 0 if $x_n^{ij}=0$, and $p_n^{ij}=x_n^{ij}/\max(\sum_{j=1}^P x_n^{ij}, \sum_{i=1}^P x_n^{ij})$ otherwise. With probability $1 - \sum_{i=1}^P p_n^{ij}$, no queue in the VOQ is chosen for cell transfer. Queue q_{ij} is the candidate of input i to attempt a cell transfer (toward output j).
2. Among the contending candidate input queues storing cells directed to the same output, only one is enabled to transfer its cell. The choice among contending candidate input queues is performed at random, according to a uniform distribution; i.e., if there are k candidates for the same output j, only one input receives a transfer grant, and the probability of receiving the grant is $1/k$ for each input queue.

Theorem 8: Under admissible load conditions, a CIOQS adopting a LQD-SA is strongly stable for any speed-up $S \geq 2$.

VIII. Proofs for Section VII

Proof of Theorem 6: Consider the $N \times N$ positive matrix $Q = I - \gamma\Gamma_{E[A]}$, where $0 \leq \gamma \leq 1$ and $E[A] \in U$ is the vector of the average cell arrival rates r_{ij}; it is easy to prove that Q is positive (semi)definite. By defining $W = XQX^T$ as the Lyapunov function for the CIOQS, we prove that for some $B \in \mathbb{R}^+$, $\epsilon \in \mathbb{R}^+$, there exists γ such that

$$E[X_{n+1}QX_{n+1}^T - X_nQX_n^T|X_n] < -\epsilon\|X_n\| \qquad \|X_n\| > B.$$

Hence Corollary 5 applies, and the CIOQS is strongly stable.

Indeed, for $\|X_n\|$ growing to infinity, and using (1)

$$\begin{aligned}
&\lim_{\|X_n\|\to\infty} \frac{E[X_{n+1}QX_{n+1}^T - X_nQX_n^T|X_n]}{\|X_n\|}\\
&= \lim_{\|X_n\|\to\infty} \frac{2}{\|X_n\|}\\
&\quad \cdot \{E[A_n]X_n^T - E[D_n]X_n^T - \gamma E[A_n]X_n^T\\
&\quad + \gamma E[D_n]\Gamma_{E[A_n]}X_n^T\}\\
&= 2\{E[A_n]\hat{X}_n^T(1-\gamma) - (1+\alpha)\tilde{X}_n\hat{X}_n^T\\
&\quad + \gamma(1+\alpha)\tilde{X}_n\Gamma_{E[A_n]}\hat{X}_n^T\}\\
&= F(\gamma, \hat{X}_n).
\end{aligned}$$

[5]Note that this probability is directly dependent from the time instant n now.

Note that the domain of $F(\gamma, \hat{X}_n)$, for a given γ, is the surface of the unit sphere in $\mathbb{R}^{+N}$, and that $F(\gamma, \hat{X}_n)$, for a given $\hat{X}_n$, is linear in γ, hence

$$F(\gamma, \hat{X}_n) = F(1, \hat{X}_n) + (\gamma - 1)\frac{\partial}{\partial\gamma}F(\gamma, \hat{X}_n)|_{\gamma=1}. \quad (20)$$

If $\gamma = 1$, then $F(\gamma, \hat{X}_n)$ is negative for all $\hat{X}_n$ that are not parallel to $E[A_n]$, since $\tilde{X}_n\hat{X}_n^T > \tilde{X}_n\Gamma_{E[A_n]}\hat{X}_n^T$, while it is null for $\hat{X}_n$ parallel to $E[A_n]$.

In order to prove stability, it is necessary to find a value of γ for which $F(\gamma, \hat{X}_n)$ is smaller than a finite negative constant on the whole domain of $\hat{X}_n$.

Note that $\partial F(\gamma, \hat{X}_n)/\partial\gamma$, performed for $\hat{X}_n$ parallel to $E[A_n]$, is strictly positive. As a consequence, there exists a ϵ-sphere around $\hat{X}_n = E[\hat{A}_n]$ where such derivative remains larger than a finite positive constant. This implies that, in each point inside the ϵ-sphere, for any $0 \leq \gamma < 1$, $F(\gamma, \hat{X}_n)$ is smaller than a finite negative constant. Outside the ϵ-sphere, the domain of $F(\gamma, \hat{X}_n)$ is closed, hence the maximum value exists; moreover, being away from $\hat{X}_n = E[\hat{A}_n]$, $\max_{\hat{X}_n} F(\gamma, \hat{X}_n)$ is strictly negative for $\gamma = 1$.

For continuity, $\max_{\hat{X}_n} F((1-\delta), \hat{X}_n)$ is negative for δ sufficiently small. As a consequence, for $\gamma = 1 - \delta$, $F(\gamma, \hat{X}_n)$ is smaller than a finite negative constant for all values of $\hat{X}_n$. ■

Proof of Theorem 7: Since $D' \in \mathbb{R}^{+N}$, two cases are possible.

- $E[A_n'] = E[A_n] - D' \in U$; in this case the stability of the algorithm can be easily proved by using $Q = I - \gamma\Gamma_{E[A_n']}$ in the previous proof.
- $E[A_n] - D'$ is not in U due to negative components; in this case it is possible to split $D' = D'' + D'''$, so that $E[A_n'] = E[A_n] - D'' \in U$, and D''' (containing all negative components) is orthogonal to $E[A_n] - D''$. Also in this case, the algorithm can be proved stable by using $Q = I - \gamma\Gamma_{E[A_n']}$ in the previous proof (note that $D'''\Gamma_{E[A_n']} = 0$, because of the orthogonality between D''' and $E[A_n] - D''$).

■

Proof of Theorem 8: The average number of times that a particular queue is selected as candidate in internal time slots is p_n^{ij}, and $p_n^{ij} \geq \tilde{x}_n^{ij}$ by definition. Since matrix $[p_n^{ij}]$ can be viewed as a matrix of admissible rates loading the CIOQS, from the proof of Theorem 5 the probability that a queue is served once selected in each internal time slot is not less than 1/2. As a consequence, using speed-up $S \geq 2$, we have $E[D_n(LQD)] = (1+\alpha)\tilde{X}_n + D_n'$, with $\alpha = 0$, and where D_n' accounts for the extra service due to the fact that $p_n^{ij} \geq \tilde{x}_n^{ij}$. Vector D_n' is a function of X_n, so that Theorem 7 does not directly apply. Note that D_n' is indeed a function of $\hat{X}_n$ (it does not depend on $\|X_n\|$), since p_n^{ij} is a function of $\tilde{x}_n^{ij}$.

Considering the evolution of the system we can write: $E[X_{n+1}] = E[X_n] + E[A_n] - (1+\alpha)\tilde{X}_n - D'(\hat{X}_n)$. Without loss of generality, we assume $E[A_n] - D_n' \in U$. If instead $E[A_n] - D_n' \notin U$, arguments similar to those used in the proof of Theorem 7 can be applied.

The system of queues can be proved to be strongly stable by using a time-variant Lyapunov function. Define

$$Q_n = \beta(X_n)\left(I - \gamma\Gamma_{E[A_n]-D'_n}\right)$$

where $0 < \gamma < 1$ is a real constant (as in the proof of Theorem 6), and $\beta(X_n)$ is a function of X_n that will be defined in the sequel. According to Corollary 5, we need to prove that

$$\lim_{\|X_n\|\to\infty} \frac{E[X_{n+1}Q_{n+1}X_{n+1}^T - X_nQ_nX_n^T|X_n]}{\|X_n\|} < -\epsilon.$$

By adding and subtracting $E[X_{n+1}Q_nX_{n+1}^T|X_n]$ we get

$$\begin{aligned}\lim_{\|X_n\|\to\infty} \frac{1}{\|X_n\|} \{E[X_{n+1}Q_{n+1}X_{n+1}^T - X_nQ_nX_n^T \\ + X_{n+1}Q_nX_{n+1}^T - X_{n+1}Q_nX_{n+1}^T|X_n]\} \\ < -\epsilon\end{aligned}$$

but, if $\beta(X_n)$ takes only positive (nonnull) values, from Theorem 7 follows that

$$\lim_{\|X_n\|\to\infty} \frac{E[X_{n+1}Q_nX_{n+1}^T - X_nQ_nX_n^T|X_n]}{\|X_n\|} < -\epsilon$$

since only the (symmetric positive definite) matrix Q_n appears in the last inequality. Furthermore, we can choose function $\beta(X_n)$ such that

$$\lim_{\|X_n\|\to\infty} \frac{E[X_{n+1}Q_{n+1}X_{n+1}^T - X_{n+1}Q_nX_{n+1}^T|X_n]}{\|X_n\|} = 0.$$

Indeed, if we assume that

$$\beta(X_n) = \begin{cases} \dfrac{E[X_{n+1}Q_{n+1}X_{n+1}^T|X_n]}{E[X_{n+1}C_nX_{n+1}^T|X_n]} & \forall,\ X_n\colon \|X_n\| > B \\ 1 & \forall,\ X_n\colon \|X_n\| \le B \end{cases}$$

where $C_n = (I - \gamma\Gamma_{E[A_n]-D'_n})$, we can write

$$\begin{aligned}&\lim_{\|X_n\|\to\infty} \frac{E[X_{n+1}Q_{n+1}X_{n+1}^T - X_{n+1}Q_nX_{n+1}^T|X_n]}{\|X_n\|} \\ &= \lim_{\|X_n\|\to\infty} \frac{1}{\|X_n\|} \\ &\cdot \left\{E\left[X_{n+1}Q_{n+1}X_{n+1}^T \right.\right.\\ &\left.\left. - \frac{E[X_{n+1}Q_{n+1}X_{n+1}^T]}{E[X_{n+1}C_nX_{n+1}^T]} X_{n+1}C_nX_{n+1}^T|X_n\right]\right\} \\ &= 0.\end{aligned}$$

Note that $\beta(X_n)$ is defined in recursive form as long as $\|X_n\|$ keeps greater than B:

$$\beta(X_{n+1}) = \beta(X_n)\frac{E[X_{n+1}C_nX_{n+1}^T|X_n]}{E[X_{n+1}C_{n+1}X_{n+1}^T|X_n]}.$$

Since matrices C_n are symmetric and positive definite, the fraction at the right-hand side of the equation above is always positive. Taking $\beta(X_0) = 1$, it is possible to show that $\beta(X_n)$ remains strictly positive (even for $n \to \infty$). ■

IX. Deterministic Weighted Scheduling Algorithms

Next, we apply the results obtained in the previous sections to scheduling algorithms that were proposed in the literature for input queueing switches.

Definition 17: A CIOQS adopts a **maximum weight matching** (MWM) scheduling algorithm if the selection of the set of noncontending cells to be transferred from inputs to outputs at each internal time slot is performed according to the MWM algorithm [8].

Let W_n represent a weight vector at time n, and let D_n be an admissible departure vector. The departure vector produced by an MWM-SA is such that $D_n(\mathrm{MWM})W_n^T = \max_{D_n}(D_nW_n^T)$.

If we set $W_n = X_n$, where X_n represents the state at time n of the system of queues of a CIOQS with VOQ at each input, the departure vector produced by an MWM-SA is such that $D_n(\mathrm{MWM})X_n^T = \max_{D_n}(D_nX_n^T)$.

In [5], using as Lyapunov function $V(X_n) = X_nX_n^T$, it was proved that, for any CIOQS adopting an MWM-SA with $W_n = X_n$

$$E[X_{n+1}X_{n+1}^T - X_nX_n^T|X_n] < -\epsilon\|X_n\|$$

for $\|X_n\|$ sufficiently large. As a consequence, due to Corollary 3, the following result holds true.

Theorem 9: Under any admissible traffic pattern, a CIOQS adopting an MWM-SA with weights equal to queue lengths is strongly stable for any speed-up $S \ge 1$.

The algorithmic complexity required for the computation of the MWM departure vector is quite large (algorithms are known with asymptotic complexity $O(P^3 \log P)$, see [8]). This severely limits the practical relevance of the stability result, and has encouraged researchers to look for simpler policies to approximate the MWM algorithm in input buffered switches with speed-up $S = 1$. Note that none of the many heuristic proposals that appeared in the literature was proven stable under any admissible traffic pattern for speed-up $S = 1$, or larger.

The following result indicates a way to design stable transfer policies for switches with moderate speed-up S, whose computational requirements can be arbitrarily constrained, at the expense of increased cell queueing delays and burstiness of the cell transfers.

Corollary 7: Consider a CIOQS with speed-up S. Assume that a scheduling algorithm $\mathcal{P}$ is found such that for some $\epsilon \in \mathbb{R}^+$, $B \in \mathbb{R}^+$,

$$D_n(\mathcal{P})X_n^T > \left(\frac{1}{S} + \epsilon\right) D_n(\mathrm{MWM})X_n^T, \qquad \|X_n\| > B$$

for each vector X_n. Given $\mathcal{P}$, a new scheduling algorithm $\mathcal{P}^{(S)}$ is defined, according to which $\mathcal{P}$ is executed only once in each external time slot, to select a set of noncontending cells, whose transfer is enabled S times, once in each of the S internal time slots comprised in the external time slot. Thus, up to S cells can be transferred from the selected queues in each external time slot.

A CIOQS with speed-up S adopting policy $\mathcal{P}^{(S)}$ is strongly stable under any admissible traffic pattern.

Note that the previous result can be easily extended:

Corollary 8: Consider a CIOQS with speed-up S. Assume that a scheduling algorithm $\mathcal{P}$ is found such that for some $\epsilon \in \mathbb{R}^+$, $B \in \mathbb{R}^+$

$$D_n(\mathcal{P})X_n^T > \left(\frac{1}{S} + \epsilon\right) D_n(\text{MWM})X_n^T, \qquad \|X_n\| > B$$

for each vector X_n. Given $\mathcal{P}$, and $K \in \mathbb{N}$, a new scheduling algorithm $\mathcal{P}^{(KS)}$ is defined, according to which $\mathcal{P}$ is executed only once every K external time slots, to select a set of noncontending cells, whose transfer is enabled KS times, in each one of the KS internal time slots comprised between two successive executions of the algorithm. A CIOQS with speed-up S adopting policy $\mathcal{P}^{(KS)}$ is strongly stable under any admissible traffic pattern.

Consider a CIOQS with speed-up S, and adopting a scheduling algorithm $\mathcal{P}'$, which is executed at each internal time slot to select a set of noncontending cells. Let $D_{n,i}$, $i = 1, \ldots, S$, be the departure vectors referring to the ith internal time slot corresponding to the nth external time slot. Let $X_{n,i}$ be the queue-length vectors referring to the ith internal time slot corresponding to the nth external time slot. Note that $X_{n,1} = X_n$ and that $X_{n,i+1} = X_{n,i} - D_{n,i}$, $i = 1, 2, \ldots, S-1$.

Corollary 9: A CIOQS with speed-up S adopting policy $\mathcal{P}'$ is strongly stable under any admissible traffic pattern if

$$D_{n,i}(\mathcal{P}')X_{n,i}^T \geq \frac{1}{S} D_{n,i}(\text{MWM})X_{n,i}^T + N_{n,i}, \quad i = 1, \ldots, S$$

where N_i is the number of queues selected by both $\mathcal{P}'$ and MWM at the ith internal time slot.

We next prove the stability of MWM algorithms.

Definition 18: A CIOQS adopts a **greedy maximal weight matching** (GMWM) scheduling algorithm if the selection of the set of noncontending cells to be transferred from inputs to outputs at each internal time slot is performed according to the following algorithm.

1. All queues q_{ij} within the whole VOQ structure are initially enabled.
2. The longest enabled queue (say q_{sd}) is selected for cell transfer (ties are broken with a uniform random choice).
3. All enabled queues q_{ij} with either $i = s$ or $j = d$ are disabled.
4. If no enabled queues remain, then stop. Else return to Step 2.

Theorem 10: A CIOQS implementing a GMWM-SA is strongly stable for all $S \geq 2$.

X. Proofs for Section IX

Proof of Corollary 7: Let $D_n(\mathcal{P}^{(S)})$ be the global departure vector, referring to one whole external time slot; the ith component of $D_n(\mathcal{P}^{(S)})$ is $d_n^i(\mathcal{P}^{(S)}) = \min(Sd_n^i(\mathcal{P}), x_n^i)$, where $d_n^i(\mathcal{P})$ is the ith component of $D_n(\mathcal{P})$.

Let $D_{\delta,n} = SD_n(\mathcal{P}) - D_n(\mathcal{P}^{(S)})$. Note that, by construction, the nonnull components of $D_{\delta,n}$ correspond to components of X_n referring to selected queues with size smaller than S; as a consequence, $D_{\delta,n}X_n^T \leq P(S-1)^2$. Then

$$\begin{aligned} D_n\left(\mathcal{P}^{(S)}\right)X_n^T &= (SD_n(\mathcal{P}) - D_{\delta,n})X_n^T \\ &= SD_n(\mathcal{P})X_n^T - D_{\delta,n}X_n^T. \end{aligned}$$

From the assumptions we have thus

$$D_n\left(\mathcal{P}^{(S)}\right)X_n^T > (1 + S\epsilon)\max_{D_n}(D_nX_n^T) - P(S-1)^2.$$

For $\|X_n\|$ sufficiently large, so that $\max_{D_n}(D_nX_n^T) > P(S-1)^2/(S\epsilon)$, we have $D_n(\mathcal{P}^{(S)})X_n^T > \max_{D_n}(D_nX_n^T)$ and Theorems 4 and 9 apply. ■

The proof of Corollary 8 is a straightforward generalization of the proof above.

Proof of Corollary 9: For the sake of brevity, we report the proof only for the case $S = 2$. The extension to larger values of S is straightforward.

From the assumptions we have

$$D_{n,1}(\mathcal{P}')X_{n,1}^T \geq 1/2 D_{n,1}(\text{MWM})X_{n,1}^T + N_1 \quad (21)$$

$$D_{n,2}(\mathcal{P}')X_{n,2}^T \geq 1/2 D_{n,2}(\text{MWM})X_{n,2}^T + N_2. \quad (22)$$

By definition $D_{n,2}(\text{MWM})X_{n,2}^T = \max_{D_{n,2}}(D_{n,2}X_{n,2}^T)$. Then

$$\begin{aligned} &D_{n,2}(\text{MWM})X_{n,2}^T \\ &\quad \geq D_{n,1}(\text{MWM})X_{n,2}^T \\ &\quad = D_{n,1}(\text{MWM})X_{n,1}^T - D_{n,1}(\text{MWM})D_{n,1}^T(\mathcal{P}') \\ &\quad = D_{n,1}(\text{MWM})X_{n,1}^T - N_1. \end{aligned} \quad (23)$$

Considering that $D_{n,2}(\mathcal{P}')X_{n,1}^T \geq D_{n,2}(\mathcal{P}')X_{n,2}^T$ since $X_{n,2} = X_{n,1} - D_{n,1}$, and that from (22) and (23), we have

$$D_{n,2}(\mathcal{P}')X_{n,1}^T \geq \frac{D_{n,1}(\text{MWM})X_{n,1}^T - N_1}{2} + N_2. \quad (24)$$

Finally, combining (21) and (24)

$$[D_{n,1}(\mathcal{P}') + D_{n,2}(\mathcal{P}')]X_{n,1}^T > D_{n,1}(\text{MWM})X_{n,1}^T.$$

Comparing the departure vectors in external time slots, and being the MWM-SA stable at $S = 1$, $\mathcal{P}'$ is strongly stable due to Theorem 4. ■

Proof of Theorem 10: We show that $D_n(\text{GMWM})X_n^T > (1/2)D_n(\text{MWM})X_n^T + K$ for each X_n, where K is the number of queues from which a cell is transferred according to both $D_n(\text{GMWM})$ and $D_n(\text{MWM})$, and prove stability according to Corollary 8.

Note that the scalar product between a departure vector D_n and the vector X_n equals the sum of the queue lengths over all queues from which cells are transferred

$$D_nX_n^T = \sum_{i=1}^{N} d_n^i x_n^i = \sum_{i \mid d_n^i = 1} x_n^i.$$

Let $I_n^1(\text{MWM})$ be the set of queues selected for cell transfer with MWM; let $I_n^1(\text{GMWM})$ be the set of queues selected with GMWM. Assume that $I_n^1(\text{MWM}) \neq I_n^1(\text{GMWM})$, otherwise the proof trivially follows, since $D_n(\text{GMWM})X_n^T = D_n(\text{MWM})X_n^T$.

If the cardinality of $I_n^1(\mathrm{MWM})$ or $I_n^1(\mathrm{GMWM})$ is smaller than P, we can augment the two sets by adding some empty queues, so that the augmented sets comprise P nonconflicting queues.

Let $g^1_{i,j,n}$ be the longest queue in $I_n^1(\mathrm{GMWM})$.

If $g^1_{i,j,n} \in I_n^1(\mathrm{MWM})$, we set $I_n^2(\mathrm{MWM}) = I_n^1(\mathrm{MWM}) - \{g^1_{i,j,n}\}$ and $I_n^2(\mathrm{GMWM}) = I_n^1(\mathrm{GMWM}) - \{g^1_{i,j,n}\}$.

Otherwise, select all queues in $I_n^1(\mathrm{MWM})$ that conflict with $g^1_{i,j,n}$; the selection returns at most two queues: $m^1_{i,j^*,n}$ (conflicting with $g^1_{i,j,n}$ on input i), and $m^1_{i^*,j,n}$ (conflicting with $g^1_{i,j,n}$ on output j).

By construction, the lengths of queues $m^1_{i,j^*,n}$ and $m^1_{i^*,j,n}$ cannot exceed the length of queue $g^1_{i,j,n}$. Thus, the sum of their lengths is less or equal to twice the length of $g^1_{i,j,n}$.

Set $I_n^2(\mathrm{GMWM}) = I_n^1(\mathrm{GMWM}) - \{g^1_{i,j,n}\}$ and $I_n^2(\mathrm{MWM}) = I_n^1(\mathrm{MWM}) - \{m^1_{i,j^*,n}, m^1_{i^*,j,n}\}$. Note that $I_n^2(\mathrm{MWM})$ can not comprise queues conflicting with $g^1_{i,j,n}$.

$g^2_{i,j,n}$, the longest queue in $I_n^2(\mathrm{GMWM})$, is considered next, and the elimination of queues from $I_n^i(\mathrm{GMWM})$ continues as long as the set is not empty. If after k steps $I_n^k(\mathrm{GMWM})$ is empty, then $I_n^k(\mathrm{MWM})$ must contain only empty queues; indeed, suppose $I_n^k(\mathrm{MWM})$ contains a nonempty queue, this implies that at least one nonempty queue exists that does not conflict with any one of the queues in $I_n(\mathrm{GMWM})$. This however is not possible, since GMWM is a maximal size matching. ■

XI. Maximal Size Matching Scheduling Algorithms

In this section we consider **maximal size matching** scheduling algorithms (MSM-SA). Many scheduling algorithms proposed in the literature [13]–[16] fall in this class.

Definition 19: A CIOQS adopts an MSM-SA if the selection of the set of noncontending cells to be transferred from inputs to outputs at each internal time slot is performed according to the MSM algorithm.

Consider any queue q_{ij} in the VOQ structure, that stores cells at input i directed to output j. Recall that cells stored in q_{ij} compete for inclusion in the set of noncontending cells generated by the scheduler with cells stored in each queue q_{ik} with $k \neq j$ and q_{hj} with $h \neq i$.

If q_{ij} is nonempty, the MSM algorithm generates a set of noncontending cells that comprises at least one cell extracted from $I_{ij} = \bigcup_k \{q_{ik} \cup q_{kj}\}$ (exactly one, if the cell is extracted from q_{ij}; possibly two, if one cell is extracted from a q_{ik}, $k \neq j$, and one from a q_{kj}, $k \neq i$).

Definition 20: A CIOQS adopts a **fair** scheduling algorithm if the first two moments of the distribution of the time interval between two consecutive services of any nonempty queue in the VOQ structure are finite under any admissible traffic pattern.

In the next section we prove the following important result, which was derived with different techniques also in [30], applying the methodology presented in [29].

Theorem 11: Under any admissible traffic pattern, a CIOQS adopting an MSM-SA achieves 100% throughput for any speed-up $S \geq 2$.

If the maximal size matching scheduling policy is fair, then the queueing delay can be proved to be bounded, and then the CIOQS is strongly stable:

Theorem 12: A CIOQS with speed-up 2 implementing a fair MSM-SA is strongly stable under any admissible traffic pattern.

XII. Proofs for Section XI

The stability proof is logically subdivided in three parts:

1) In the first part of the proof, we define a queueing system S, and we prove its stability;
2) In the second part, we show how it is possible to infer, from the stability of system S, that a CIOQS implementing an MSM-SA achieves 100% throughput, thus proving Theorem 11;
3) In the third part, finally, we show how it is possible to infer, from the stability of system S, the strong stability of all input queues in a CIOQS implementing a fair MSM-SA, thus proving Theorem 12.

A. Part I

Consider a system of queues S comprising $N = P^2$ discrete-time queues s_{ij} (each queue in S corresponds to an input queue of the CIOQS).

Customers arriving at the queues in S belong to two classes, named C_a and C_b. Priority is given to the service of class C_a customers, i.e., whenever a class C_a customer is in queue s_{ij}, no customer of class C_b can be served at queue s_{ij}. However, the server of queue s_{ij} is active only when at least one customer of class C_b is in queue s_{ij}. Let x_n^a and x_n^b denote the numbers of customers of classes C_a and C_b, respectively, at time n. The vector $X_n = (x_n^a, x_n^b)$ contains the state information for the queue at time n. Similarly, the vector $D_n = (d_n^a, d_n^b)$ contains the information about the departures from the queue in time n.

Theorem 13: Each queue s_{ij} of system S is strongly stable if: i) the average number of class C_b customers at s_{ij} is greater than zero ($E[x_{ij}^b] > 0$), and ii) the average total number of arrivals per slot at s_{ij} is less than one ($E[a_{ij}^a + a_{ij}^b] < 1$), and iii) the second moment of the total number of arrivals per slot at s_{ij} is finite ($E[(a_{ij}^a + a_{ij}^b)^2] < \infty$).

Proof: Consider a generic queue s_{ij} of system S. In the following, we always refer to queue s_{ij}, and, in order to simplify the notation, we omit the queue indices.

The vector $A_n = (a_n^a, a_n^b)$ contains the information about the arrivals at the queue in slot n. Since we assume that vectors A_n are statistically independent, the queue evolution process is a DTMC, whose state at time n is X_n, and whose dynamic is driven by

$$X_{n+1} = \begin{cases} (x_n^a, x_n^b) + A_n - (1, 0) & \text{if } x_n^a \neq 0 \text{ and } x_n^b \neq 0 \\ (0, x_n^b) + A_n - (0, 1) & \text{if } x_n^a = 0 \text{ and } x_n^b \neq 0 \\ (x_n^a, 0) + A_n & \text{if } x_n^b = 0. \end{cases} \tag{25}$$

Given $\alpha \in \mathbb{R}^+$, consider the Lyapunov function

$$V(X_n) = \begin{cases} \dfrac{(x_n^a)^2}{\alpha} & \text{if } x_n^b = 0 \\ (x_n^a + x_n^b)^2 & \text{otherwise.} \end{cases} \tag{26}$$

It is possible to find $\alpha \in \mathbb{R}^+$, $\alpha < 1$, $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$ such that

$$E[V(X_{n+1}) - V(X_n) | X_n] < -\epsilon \|X_n\|$$

for $\|X_n\| > B$, so that the system of queues is strongly stable for Theorem 2.

Consider first $X_n = (x_n^a, 0)$, i.e., $x_n^b = 0$, and $\|X_n\| = x_n^a$; in this case

$$\begin{aligned}\frac{E[V(X_{n+1}) - V(X_n)|X_n]}{\|X_n\|} &= \frac{E[(x_n^a + a_n^a)^2 | x_n^a, a_n^b = 0]}{\alpha x_n^a} \Pr\{a_n^b = 0\} \\ &\quad + \frac{E[(x_n^a + a_n^a + a_n^b)^2 | x_n^a, a_n^b > 0]}{x_n^a} \Pr\{a_n^b > 0\} \\ &\quad - \frac{(x_n^a)^2}{\alpha x_n^a}.\end{aligned}$$

Since the second moment of the distribution of the number of arrivals per slot is finite, taking the limit for $\|X_n\| \to \infty$, we obtain

$$\begin{aligned}\lim_{\substack{\|X_n\|\to\infty \\ x_n^b=0}} \sup \frac{E[V(X_{n+1}) - V(X_n)|X_n]}{\|X_n\|} &= \lim_{x_n^a\to\infty} \left[\frac{x_n^a}{\alpha} \Pr\{a_n^b = 0\} + x_n^a \Pr\{a_n^b > 0\} - \frac{x_n^a}{\alpha}\right] \\ &= \lim_{x_n^a\to\infty} x_n^a \left(1 - \frac{1}{\alpha}\right) \Pr\{a_n^b > 0\} \\ &= -\infty\end{aligned}$$

for any $0 < \alpha < 1$.

Instead, for $x_n^b \neq 0$,

$$\begin{aligned}\lim_{\substack{\|X_n\|\to\infty \\ x_n^b\neq 0}} \sup \frac{E[V(X_{n+1}) - V(X_n)|X_n]}{\|X_n\|} &= \lim_{\substack{\|X_n\|\to\infty \\ x_n^b\neq 0}} \sup \left[2E[a_n^a + a_n^b - 1] \frac{x_n^a + x_n^b}{\sqrt{(x_n^a)^2 + (x_n^b)^2}}\right] \\ &< 0\end{aligned}$$

since $1 \leq (x_n^a + x_n^b)/\sqrt{(x_n^a)^2 + (x_n^b)^2} \leq \sqrt{2}$, $E[(a_n^a + a_n^b - 1)^2]$ is finite, and $E[a_n^a + a_n^b] < 1$.

In addition, if the second moment of the number of arrivals per slot is bounded, there exists an M such that

$$E[V(X_{n+1})|X_n] < M \qquad \forall\, X_n\colon \|X_n\| \leq B.$$

As a consequence, queue s_{ij} is strongly stable for Theorem 2, and since the proof holds for any queue in S, the whole system of queues S is strongly stable. ■

It is possible to extend these results to arrival processes that present some form of correlation (i.e., for which vectors A_n are not independent). In this case the state definition for the Markov chain must be augmented with additional state variables. The extension is straightforward when these additional state variables take only a finite set of values, since they can be mapped onto vectors K_n of Corollary 1 and Theorem 2. For example, the arrival processes at queues can be Markov modulated Bernoulli processes. It is, however, necessary to guarantee that $E[A_n|Y_n]$ is an admissible load vector for each state Y_n.

B. Part II

Let us correlate the arrival processes of the CIOQS and of the system of queues S studied in the previous subsection. Assume that at time $n = 0$ both the system of queues and the CIOQS are empty.

If in the CIOQS a cell arrives at queue q_{ij} at time t^*, then in the system of queues at time t^*:

- a class C_b customer arrives at queue s_{ij};
- a class C_a customer arrives at each queue s_{lj}, $l = 1, \ldots, P$, $l \neq i$, and a class C_a customer arrives at each queue s_{im}, $m = 1, \ldots, P$, $m \neq j$.

The arrival of a class C_b customer at queue s_{ij} corresponds to the arrival of a cell at queue q_{ij}. The presence of a class C_a customer in queue s_{ij} corresponds to a delay, i.e., to an *internal* time slot in which no cell is transmitted from queue q_{ij}, due to the transmission from a queue that is contending with q_{ij}. Then the total number of arrivals at queue s_{ij} is

$$a_n(s_{ij}) = \sum_{ij|q_{ij}\in I_{ij}} a_n^{ij}$$

where $I_{ij} = \bigcup_k \{q_{ik} \cup q_{kj}\}$.

Let $B_n(q)$ be a function that is equal to 1 if the number of cells in queue q is greater than zero at time n; $B_n(q) = 0$ otherwise. The following results apply.

Theorem 14: Let N^* be a time in which the queue s_{ij} is empty. Then

$$\sum_{n=0}^{N^*} B_n(q_{ij}) \leq \sum_{n=0}^{N^*} B_n(s_{ij}). \tag{27}$$

Proof: First, observing that $\sum_{n=0}^{N^*} B_n(s_{ij})$ is the total busy time in the interval $[0, N^*]$, we have that

$$\sum_{n=0}^{N^*} a_n(s_{ij}) \leq \sum_{n=0}^{N^*} B_n(s_{ij}) \tag{28}$$

since no more than one customer can depart from queue s_{ij} in each time slot, but no customer can leave the queue when only class C_a customers are present.

Second, considering queue q_{ij} of the CIOQS, we have

$$\sum_{n=0}^{N} B_n(q_{ij}) \leq \sum_{n=0}^{N} a_n(s_{ij}) \qquad \forall\, N \in \mathbb{N} \tag{29}$$

since, if $B_n(q_{ij}) = 1$, at least a cell departs from queues in I_{ij}.

As a consequence, comparing (28) and (29), we derive (27).■

Corollary 10:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} B_n(q_{ij}) \leq \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} B_n(s_{ij}) < 1 \quad w.p. 1.$$

Proof: The second inequality holds because system S is described by an ergodic DTMC, so that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} B_n(s_{ij}) = 1 - \pi_0 \qquad w.p. 1$$

being π_0 the stationary probability that the queue s_{ij} is empty. For what concerns the first inequality note that, by Theorem 14, there exists a sequence of times Z_s such that

$$\frac{1}{N^*} \sum_{n=0}^{N^*} B_n(q_{ij}) < \frac{1}{N^*} \sum_{n=0}^{N^*} B_n(s_{ij}) \qquad N^* \in Z_s$$

where Z_s is the set of times in which s_{ij} is empty. Since system S, being strongly stable, is described by an ergodic DTMC, $\sup Z_s = \infty$. As a consequence

$$\lim_{N^*\to\infty} \frac{1}{N^*} \sum_{n=0}^{N^*} B_n(q_{ij}) \le \lim_{N^*\to\infty} \frac{1}{N^*} \sum_{n=0}^{N^*} B_n(s_{ij}) = 1 - \pi_0.$$

But, being sequence $(1/N)\sum_{n=0}^{N} B_n(q_{ij})$ convergent, it converges to the same limit of all its subsequences. ■

Corollary 10 implies that there exists a infinite set Z of time instants in which queue q_{ij} is empty (note that we have not yet proved stability of the CIOQS, hence of the ergodicity of the underlying DTMC), i.e.:

Corollary 11: Let $Z = \{n \in \mathbb{N} | B_n(q_{ij}) = 0\}$. Then

$$\sup Z = \infty.$$

Proof: By contradiction, let us suppose that $\sup Z = M < \infty$. Then

$$\begin{aligned}
&\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} B_n(q_{ij}) \\
&= \lim_{N\to\infty} \frac{1}{N} \left(\sum_{n=0}^{M} B_n(q_{ij}) + \sum_{n=M+1}^{N} B_n(q_{ij}) \right) \\
&= \lim_{N\to\infty} \frac{1}{N} \sum_{n=M+1}^{N} B_n(q_{ij}) \\
&= \lim_{N\to\infty} \frac{N-M}{N} \\
&= 1.
\end{aligned}$$

But this contradicts Corollary 10. ■

Finally, we can prove Theorem 11.

Proof: Let us consider a queue q_{ij}. We have to prove that

$$\lim_{N\to\infty} \frac{x_N^{ij}}{N} = 0 \qquad w.p.1.$$

Consider the subsequence $x_{N^*}^{ij}$, $N^* \in Z$. By definition, $x_{N^*}^{ij} = 0$, and

$$\lim_{N^* \in Z \to\infty} \frac{x_{N^*}^{ij}}{N^*} = 0 \qquad w.p.1.$$

The Cesaro sum $(1/N)\sum_{n=0}^{N} d_n^{ij}$ is convergent; then x_N^{ij}/N is convergent:

$$\begin{aligned}
\lim_{N\to\infty} \frac{x_N^{ij}}{N} &= \lim_{N\to\infty} \frac{\sum_{n=0}^{N} a_n^{ij} - d_n^{ij}}{N} \\
&= E[a_{ij}] - \lim_{N\to\infty} \frac{\sum_{n=0}^{N} d_n^{ij}}{N} \qquad w.p.1.
\end{aligned}$$

A converging sequence converges to the same limit of any subsequence. Thus

$$\lim_{N\to\infty} \frac{x_N^{ij}}{N} = \lim_{N^* \in Z \to\infty} \frac{x_{N^*}^{ij}}{N^*} = 0$$

and the system achieves 100% throughput. ■

C. Part III

Let us correlate in the following way the arrival processes of the CIOQS and of the system of queues S studied in the previous section. If a cell p_{ij}^* arrives at queue q_{ij} at time t^*, then at time t^*

1) A class C_b customer arrives at queue s_{ij}.
2) For each queue $s_{mn} \neq s_{ij}$, with either $m = i$ or $n = j$, a class C_a customer arrives only if cell p_{ij}^* leaves the CIOQS before any cell currently stored in the CIOQS queue q_{mn}.
3) Queue s_{ij} receives a batch of class C_a customers, whose size is equal to the number of cells stored in queues $q_{mn} \neq q_{ij}$, with either $m = i$ or $n = j$, and leaving the CIOQS before cell p_{ij}^*, but after any other cell stored in q_{ij}.

The presence of a class C_b customer in queue s_{ij} corresponds to the presence of a cell in queue q_{ij}. The presence of a class C_a customer in queue s_{ij} corresponds to a delay, i.e., to an *internal* time slot in which no cell is transmitted from queue q_{ij}. System S is work conserving, in the sense that a customer, if present, is served at each internal time slot. Note that this is true only under the particular traffic pattern that we consider, which allows class C_a customers to be in waiting lines only when also at least one class C_b customer is present. Instead, the VOQ's of the CIOQS are not work conserving, in the sense that no waiting customer may be served in a slot, due to input or output contention. Nevertheless, the flow of class C_b customers in system S exactly mimics the flow of cells in the CIOQS running a fair MSM scheduling algorithm.

Queue s_{ij} receives customers 1) when a cell arrives at queue q_{ij}, and 2) possibly when a cell arrives at a queue q_{nm}, with either $n = i$, or $m = j$ (i.e., a cell arriving at the same input port, or directed to the same output port). In the former case a class C_b customer enters s_{ij}, possibly together with a batch of class C_a customers. In the latter case one class C_a customer may arrive at queue s_{ij}.

Note that the batch of class C_a customers is due to class C_b customers (i.e., cells in the CIOQS) stored at other queues, and that a C_b customer arrived at another queue at most generates (either upon arrival, or in a later batch) one C_a customer at queue s_{ij}.

We can therefore state the following theorem.

Theorem 15: The total average arrival rate of class C_a and class C_b customers at queue s_{ij} does not exceed the sum of the arrival rates at queues q_{mn}, with $m = i$ and/or $n = j$ (including queue q_{ij}).

Proof: By construction, at most one class C_a customer arrives at queue s_{ij} for each class C_b customer arriving or queued at queue $q_{mn} \neq q_{ij}$, with either $m = i$ or $n = j$. Since C_b customers in system S correspond to cells in the CIOQS, at most one customer (either class C_a or class C_b) arrives at queue s_{ij} for each cell arrival at a queue belonging to input i or directed to output j. ■

Obviously, the following property holds.

Proposition 2: The number of cells stored in any queue q_{ij} never exceeds the total number of customers (of classes C_a and C_b) stored in s_{ij}.

Theorem 15 implies that the average arrival rate of customers at each queue s_{ij} is less than 2 customers per external slot if the traffic loading the corresponding CIOQS system is admissible.

For what regards the second moment of the arrival process at system S, the following property holds.

Theorem 16: If the CIOQS implements a fair MSM policy (according to Definitions 20 and 19), then the second moment of the size of the batch entering queue s_{ij} is finite.

Proof: Any batch contains only as many customers as the number of cells stored in $q_{mn} \neq q_{ij}$, with either $m = i$ or $n = j$, that leave the CIOQS before p^*_{ij}, but after any other cell stored in q_{ij}, i.e., between two consecutive services of q_{ij}. Since the policy is fair, the first two moments of the time between two consecutive services is finite, and thus also the first two moments of the batch size are finite. ■

Theorem 12 holds as a consequence of the previous results.

Proof of Theorem 12: Consider the system of queues S, and assume that the rate at which customers are served in system S is equal to the internal CIOQS rate, i.e., to twice the external cell rate. Because of Theorem 15, the average total number of arrivals per internal slot (considering both classes C_a and C_b) at any queue s_{ij} is less than 1, under any admissible traffic pattern. Furthermore, because of Theorem 16, the second moment of the total number of arrivals at queue s_{ij} is finite.

As a consequence, the system of queues S is strongly stable due to Theorem 13.

Since we have seen that each queue in the VOQ structure cannot be longer than the corresponding queue in the system of queues S, also the CIOQS must be strongly stable. ■

XIII. Conclusion

In this paper we computed stability conditions for several scheduling algorithms used in combined input/output queueing switch architectures. A formal, analytical approach, mainly based upon Lyapunov functions, was used to derive the internal switch speed-up needed to grant stability to vast classes of scheduling algorithms.

Our novel results show that an internal speed-up equal to two permits strong stability to most algorithms, when virtual output queueing is implemented, and the policy to select the set of noncontending data units avoids head-of-the-line blocking phenomena.

The main results are Theorems 5 and 8, referring to a random selection of the set of noncontending cells based upon input rates or queue lengths, Theorem 10, referring to greedy maximal weight matching, and Theorems 11 and 12, referring to maximal size matchings.

These results provide interesting inputs to the implementation of the high-performance switching architectures that are necessary in the near future to support the exponentially increasing traffic of the Internet.

References

[1] Cisco Systems, Inc. (1999), San Jose, CA. [Online]. Available: http://www.cisco.com

[2] Lucent Technologies, Inc. (2001), Holmdel, NJ. [Online]. Available: http://www.lucent.com/dns/products/a500/html

[3] Avici Systems, Inc. (2001), Billerica, MA. [Online]. Available: http://www.avici.com/pdfs/Avici_TSR.pdf

[4] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queueing on a space division switch," *IEEE Trans. Commun.*, vol. 35, pp. 1347–1356, Dec. 1987.

[5] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, pp. 1260–1267, Aug. 1999.

[6] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. IEEE*, vol. 83, pp. 1374–1399, Oct. 1995.

[7] D. Stiliadis and A. Varma, "Providing bandwidth guarantees in an input-buffered crossbar switch," in *IEEE INFOCOM*, Boston, MA, Apr. 1995, pp. 960–968.

[8] R. E. Tarjan, *Data Structures and Network Algorithms*. Philadelphia, PA: Society for Industrial and Applied Mathematics, Nov. 1983.

[9] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. dissertation, Univ. of California at Berkeley, 1995.

[10] N. McKeown and A. Mekkittikul, "A pratical scheduling algorithm to achieve 100% throughput in input-queued switches," in *IEEE INFOCOM*, San Francisco, CA, Apr. 1998, pp. 792–799.

[11] N. McKeown and T. E. Anderson. A quantitative comparison of scheduling algorithms for input-queued switches. Stanford Univ., Stanford, CA. [Online]. Available: http://tiny-tera.stanford.edu/~nickm/papers.html

[12] M. Ajmone Marsan, A. Bianco, E. Leonardi, and L. Milia, "RPA: A flexible scheduling algorithm for input buffered switches," *IEEE Trans. Commun.*, vol. 47, pp. 1921–1933, Dec. 1999.

[13] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Networking*, vol. 7, pp. 188–201, Apr. 1999.

[14] H. Duan, J. W. Lockwood, and S. Mo Kang, "Matrix unit cell scheduler (MUCS) for input-buffered ATM switches," *IEEE Commun. Lett.*, vol. 2, pp. 20–23, Jan. 1998.

[15] H. Chen, J. Lambert, and A. Pitsillides, "RC-BB switch. A high performance switching network for B-ISDN," in *IEEE GLOBECOM*, Singapore, Nov. 1995, pp. 2097–2101.

[16] R. O. LaMaire and D. N. Serpanos, "Two dimensional round-robin schedulers for packet switches with multiple input queues," *IEEE/ACM Trans. Networking*, vol. 2, pp. 471–482, Oct. 1994.

[17] M. Ajmone Marsan, A. Bianco, E. Filippi, P. Giaccone, E. Leonardi, and F. Neri, "On the behavior of input queueing switch architectures," *Eur. Trans. Telecommun. (ETT)*, vol. 10, no. 2, pp. 111–124, Mar./Apr. 1999.

[18] R. Schoenen, G. Post, and G. Sander, "Weighted arbitration algorithms with priorities for input-queued switches with 100% throughput," in *3rd IEEE Int. Workshop Broadband Switching Systems (BSS'99)*, Kingston, Ontario, Canada, June 1999.

[19] S. T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with combined input and output queueing," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1030–1039, Dec. 1999.

[20] I. Stoica and H. Zhang, "Exact emulation of an output queueing switch by a combined input output queueing switch," in *6th Int. Workshop Quality of Service (IWQoS'98)*, Napa, CA, May 1998, pp. 218–224.

[21] P. Krishna, N. S. Patel, A. Charny, and R. Simcoe, "On the speedup required for work-conserving crossbar switches," in *6th Int. Workshop Quality of Service (IWQoS'98)*, Napa, CA, May 1998, pp. 225–234.

[22] A. Charny, P. Krishna, N. S. Patel, and R. Simcoe, "Algorithms for providing bandwidth and delay guarantees in input-buffered crossbars with speedup," in *6th Int. Workshop Quality of Service (IWQoS'98)*, Napa, CA, May 1998, pp. 235–244.

[23] M. Ajmone Marsan, E. Leonardi, M. Mellia, and F. Neri, "On the stability of input-buffer cell switches with speed-up," in *IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 1604–1613.

[24] ——, "Stability of maximal size matching scheduling in input-queued cell switches," in *IEEE ICC*, New Orleans, LA, June 2000, pp. 1758–1763.

[25] H. J. Kushner, *Stochastic Stability and Control*. New York: Academic, 1967.

[26] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Control*, vol. 37, pp. 1936–1948, Dec. 1992.

[27] G. Fayolle, "On random walks arising in queueing systems: Ergodicity and transience via quadratic forms as Lyapunov functions—Part I," *Queueing Systems*, vol. 5, pp. 167–184, 1989.

[28] P. R. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies," *IEEE Trans. Automat. Control*, vol. 40, pp. 251–260, Feb. 1995.

[29] J. G. Dai, “Stability of open multiclass queueing networks via fluid models,” in *Stochastic Networks*. ser. IMA Volumes in Mathematics and Its Applications, F. Kelly and R. Williams, Eds. New York: Springer-Verlag, vol. 71, pp. 71–90.

[30] J. G. Dai and B. Prabhakar, “The throughput of data switches with and without speedup,” in *IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 556–564.

Emilio Leonardi (M’99) received the Dr. Ing. degree in electronics engineering in 1991 and the Ph.D. degree in telecommunications engineering in 1995, both from the Politecnico di Torino, Torino, Italy.

He is currently an Assistant Professor in the Electronics Department of the Politecnico di Torino. In 1995, he spent one year with the Computer Science Department of the University of California, Los Angeles, where he was involved in the Supercomputer-SuperNet (SSN) project aimed at the design of a high-capacity optical network architecture. During the summer of 1999, he joined the High-Speed Networks Research Group of Lucent Technologies, Holmdel, NJ, where he worked on the design of scheduling algorithms for high-capacity switch architectures. His research interests are in the fields of all-optical networks, switching architectures, queueing theory, and wireless communications.

Marco Mellia (S’97) was born in Torino, Italy, in 1971. He received the degree in electronic engineering from the Politecnico di Torino, Torino, Italy, in 1997 where he is currently working toward the Ph.D. degree in the Dipartimento di Elettronica.

From March to October 1999, he was with the Computer Science Department at Carnegie Mellon University, Pittsburgh, PA, as a Visiting Scholar. His research interests are in the fields of all-optical networks, switching architectures, and QoS routing algorithms.

Fabio Neri (M’98) was born in Novara, Italy, in 1958. He received the Dr. Ing. and Ph.D. degrees in electrical engineering from the Politecnico di Torino, Torino, Italy, in 1981 and 1987, respectively.

He is currently a Full Professor in the Electronics Department of the Politecnico di Torino. From 1991 to 1992, he was with the Information Engineering Department at the University of Parma, Parma, Italy, as an Associate Professor.

From 1982 to 1983, he was a Visiting Scholar at the George Washington University, Washington, D.C. He has been a Visiting Researcher at the Computer Science Department of the University of California, Los Angeles (summer 1995), at the Optical Data Networks Research Department of Bell Laboratories, Lucent Technologies, Holmdel, NJ (summer 1998), and at British Telecom Advanced Communication Research, Ipswich, U.K. (summer 2000). His research interests are in the fields of performance evaluation of communication networks, high-speed and all-optical networks, packet switching architectures, discrete event simulation, and queueing theory. He has co-authored over 100 papers published in international journals and presented in leading international conferences.

Dr. Neri is a member of the IEEE Communications Society. He has served some IEEE conferences, including Globecom and Infocom, and was the Technical Program Co-Chair of the 1999 IEEE Workshop on Local and Metropolitan Area Networks.

Marco Ajmone Marsan (F’99) received degrees in electronic engineering from the Politecnico di Torino, Torino, Italy, and the University of California, Los Angeles (UCLA).

He is currently a Full Professor at the Electronics Department of the Politecnico di Torino. During the summers of 1980 and 1981 he was with the Research in Distributed Processing Group, Computer Science Department, UCLA. During the summer of 1998 he was an Erskine Fellow at the Computer Science Department of the University of Canterbury, New Zealand. He has coauthored over 200 journal and conference papers in the areas of communications and computer science, as well as two books. His current interests are in the fields of performance evaluation of communication networks and their protocols.