On the Stability of Input-Buffer Cell Switches with Speed-up

M. Ajmone Marsan, E. Leonardi, M. Mellia, F. Neri
Dipartimento di Elettronica
Politecnico di Torino
Torino, Italy
email: {ajmone,leonardi,mellia,neri}@polito.it

Abstract—We consider cell-based switch 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 towards output ports. The stability properties of several classes of scheduling algorithms are studied in the paper, using analytical techniques mainly based upon Lyapunov functions. Original stability conditions are derived for some scheduling algorithms that are being used today in high-performance switch architectures.

I. INTRODUCTION

Cell-based switch architectures originated from the design of ATM network nodes, and are today popular also in IP networks, where the core of high-performance routers is often provided by a fast cell-based switching fabric (for example, the CISCO 12000 [1] and the Lucent Cajun [2] and PacketStar [3] router families adopt a cell-based internal switching fabric).

Many of the recent designs of high-performance cell-based switches and routers do not adopt the classical output queueing (OQ) architecture (where cells are stored at the output of the switching fabric), preferring either input queuing (IQ) or combined input/output queuing (CIOQ) structures. The reason is that, in OQ, both the switching fabric and the input/output queues in line cards must operate at a speed equal to the sum of the rates of all input/output lines. When the number of ports is large or the line rate is high, this makes OQ impractical. With respect to IQ schemes (where cells are stored at the input of the switching fabric), OQ has the advantage that delays through the switch can be more easily controlled, and the implementation of fair queuing algorithms at the output is relatively easy and well-understood [4].

The main advantage of IQ schemes is that they permit all the components of the switch (input interfaces, switching fabric, output interfaces) to operate at a speed which is compatible with the data rate of input and output lines. Performance reductions due to head-of-the-line blocking in the case of a single queue per input interface [5] can be overcome by Virtual Output Queuing (VOQ) (also called Destination Queuing) schemes, which largely reduce this problem [6]: they organize input buffers in each line card into a set of queues where cells awaiting access to the switching fabric are stored.

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, 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 must be devoted to the scheduling algorithm, either at a centralized scheduler, or at the line cards, in a distributed manner.

Several scheduling algorithms for IQ cell switches were proposed and compared in the recent literature [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. They provide performances very close to those of OQ architectures.

Scheduling algorithms for IQ architectures are always relatively demanding in terms of computing power and control bandwidth. It has been shown [20] that this complexity can be partly reduced 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.

As soon as an IQ switch takes advantage of an internal speed-up, 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 become useless.

A speed-up equal to 2 in CIOQ switches, independent of the number of switch ports, can be shown [20], [21] to be sufficient to exactly emulate an OQ architecture, at the expense of quite complex scheduling algorithms, whose implementation appears problematic. In this paper we prove that simpler scheduling algorithms, whose implementation is surely feasible, provide the same throughput performance of OQ with speed-up equal to 2 (even if they do not exactly emulate the behavior of OQ switches).

The paper is organized as follows. Section II introduces our notation and terminology on CIOQ switches. Section III recalls known results on the stability of queueing systems and extends them to cope with the systems studied in this paper. Sections IV, V, and VI provide novel stability conditions for different classes of scheduling algorithms. Section IV focuses on scheduling algorithms that aim at guaranteeing probabilistically known transmission rates for all input/output port pairs. Section V focuses on scheduling algorithms that select the set of cells to be transferred from input ports to output ports in each time slot according to the length of input queues. While the algorithms of Sections IV and V select the set of non-contending cells with random 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 QoS requirements. In this paper the term scheduling algorithm is only used to refer to the first class of algorithms.
II. NOTATION AND MODELING ASSUMPTIONS

We consider CIOQ cell-based switches with $N$ input lines and $N$ output lines, all at the same cell rate (and we call them $N \times N$ CIOQs).

At each input line card, cells are stored according to a Virtual Output Queue (VOQ) policy: one separate queue is maintained for each output. Let $q_{ij}$ be the queue at input $i$ storing cells directed to output $j$.

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 $N \times N$ CIOQS achieves speed-up $S$ when the cell transfer rate through the internal 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 also 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 1:** The traffic pattern loading an $N \times N$ CIOQS is admissible if the total arrival rates in cells/external slot are less than 1 for each input and each output, that is

$$
\begin{align*}
   r_i &= \sum_{j=1}^{N} r_{ij} < 1 & i = 1, 2, \ldots, N \\
   r^T_j &= \sum_{i=1}^{N} r_{ij} < 1 & j = 1, 2, \ldots, N 
\end{align*}
$$

During each internal time slot, some cells may be transferred from input queues to output queues. 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 to each output.

**Definition 2:** A set of cells extracted from queues $q_{ij}$ is a set $B = \{b_{ij}\}$ of non-contending cells (also called a switching matrix) if

$$
\begin{align*}
   \sum_{j} b_{ij} \leq 1 & \quad \text{for all } i \\
   \sum_{i} b_{ij} \leq 1 & \quad \text{for all } j
\end{align*}
$$

In an $N \times N$ CIOQS with speed-up $S$, a set of non-contending cells can be transferred from input ports to output ports during each internal time slot, so that $S$ sets of non-contending cells can be transferred during each external time slot.

III. DEFINITIONS AND PRELIMINARY RESULTS

Let $A$ be a finite set of real numbers $a_i$, such that each element $a_i$ is greater than zero and 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., $A = \{a_i \in \mathbb{R}^+, \sum_{i=1}^{N} a_i \leq 1\}$. $\mathbb{R}^+$ denotes the set of non-negative real numbers.

Let $\alpha^{[k]}$ be a subset of $A$ such that $|\alpha^{[k]}| = k$, with $0 \leq k \leq N$. Let $\alpha^{[0]} = \emptyset$.

Let $A_k$ be the set of all possible subsets of $\alpha^{[k]}$ 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., $2^A = \{\alpha_k\}_{k=0}^{N}$.

**Definition 3:** Given $\alpha \subseteq A$, let $f(\alpha)$ be a function $2^A \rightarrow \mathbb{R}$ such that $f(\alpha) = \prod_{a \in \alpha} a$; let $f(\emptyset) = 1$.

**Definition 4:** Given $\alpha \subseteq A$, let $f(\alpha)$ be a function $2^A \rightarrow \mathbb{R}$, such that $f(\alpha) = \prod_{a \in \alpha} (1 - a)$; let $f(\emptyset) = 1$.

**Definition 5:** Let $F_k(A) = \sum_{\alpha \subseteq A} f(\alpha^{[k]})$.

**Proposition 1:** For each set $A = \{a_i \in \mathbb{R}^+, \sum_{i=1}^{N} a_i \leq 1\}$,

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

The proof is given in Appendix A.

**Definition 6:** An $N \times N$ matrix $Q$ is positive definite if $XQX^T > 0$ $\forall X \in \mathbb{R}^N$. An $N \times N$ matrix $Q$ is positive semidefinite if $XQX^T \geq 0$ $\forall X \in \mathbb{R}^N$.

**Definition 7:** An $N \times N$ matrix $Q$ is copositive if $XQX^T \geq 0$ $\forall X \in \mathbb{R}^{N+}$

Note that every positive semidefinite matrix is also copositive.

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., $X_n = (x_{1,n}, x_{2,n}, \ldots, x_{N,n})$, where $x_{i,n}$ is the number of customers in queue $i$ at time $n$.

The queue length evolution is described by $x_{i,n+1} = x_{i,n} + a_{i,n} - d_{i,n}$, where $a_{i,n}$ represents the number of customers arrived at queue $i$ in time interval $(n,n+1)$, and $d_{i,n}$ represents the number of customers departed from queue $i$ in time interval $(n,n+1)$.

Let $A_n = (a_{1,n}, a_{2,n}, \ldots, a_{N,n})$ be the vector of the numbers of arrivals at the queues, and $D_n = (d_{1,n}, d_{2,n}, \ldots, d_{N,n})$ be the vector of the numbers of departures from the queues. With this notation, the system evolution equation can be written as

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

We assume in the paper that vectors $A_n$ are independent and identically distributed, although this constraint could be in part relaxed.

In the following definitions, we denote with $||Y||$ the Euclidean norm of vector $Y = (y_1, y_2, \ldots, y_K)$: $||Y|| = \sqrt{\sum_{k=1}^{K} y_k^2}$.

**Definition 8:** A system of queues is said to be weakly stable if, for every $\epsilon > 0$, there exists $B \in \mathbb{R}^+$ such that

$$
\lim_{n \to \infty} \text{Prob}(||X_n|| > B) < \epsilon \quad \text{W.P.1}
$$

**Definition 9:** A system of queues is said to be strongly stable if it is weakly stable, and

$$
\lim_{n \to \infty} \sup E||X_n|| < \infty \quad \text{W.P.1}
$$

Note that strong stability is the property required for most applications.

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 $Y_n = (X_n, K_n)$, $Y_n \in$
N^M, X_n \in N^N, K_n \in N^N and M = N + N'. Y_n is the combination of vector X_n and a vector K_n of 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.

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 vectors \{K_n\}.

Note that most systems of discrete-time queues of practical interest can be described with models that fall in the DTMC class. A general criterion for the (strong) stability of systems falling into this class can be useful.

**Theorem 1:** Given a system of queues whose evolution is described by a DTMC with state vector Y_n \in N^M, if a lower bounded function \( V(Y_n) \), called Lyapunov function, \( V : N^M \to R \) can be found such that

\[
E[V(Y_{n+1}) | Y_n] < \infty \quad \forall Y_n
\]

and there exist \( \varepsilon \in R^+ \) and \( B \in R^+ \) such that

\[
E[V(Y_{n+1}) - V(Y_n) | Y_n] < -\varepsilon \quad \forall \|Y_n\| > B
\]  

then all states of the DTMC are positive recurrent.

**Proof:** This theorem is a straightforward extension of Foster’s Criterion; see [23], [24], [25].

Note that an explicit dependence of the Lyapunov function on the time index \( n \) is allowed, so that it is possible to have \( V(Y_n) = V(Y_{n}, n) \).

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.

**Corollary 1:** Given a system of queues whose evolution is described by a DTMC with state vector \( Y_n \in N^M \), 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, \( V : N^N \to R \) can be found such that

\[
E[V(X_{n+1}) | Y_n] < \infty \quad \forall Y_n
\]

and there exist \( \varepsilon \in R^+ \) and \( B \in R^+ \) such that

\[
E[V(X_{n+1}) - V(X_n) | Y_n] < -\varepsilon \quad \forall Y_n : \|X_n\| > B
\]  

then all states of the DTMC are positive recurrent.

In this case the system 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 queueing systems for which Corollary 1 applies. By restricting the general results to a particular class of Lyapunov functions, 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 \( Y_n \in N^M \), and whose state space \( H \) is a subset of the Cartesian product of the state space \( H_X \) and a finite state space \( H_K \), then, if a lower bounded function \( V(X_n) \), called Lyapunov function, \( V : N^N \to R \) can be found such that

\[
E[V(X_{n+1}) | Y_n] < \infty \quad \forall Y_n
\]

and there exist \( \varepsilon \in R^+ \) and \( B \in R^+ \) such that

\[
E[V(X_{n+1}) - V(X_n) | Y_n] < -\varepsilon \|X_n\| \quad \forall Y_n : \|X_n\| > B
\]  

then the system of queues is strongly stable.

This theorem is an extension of the results presented in [26, Sect.IV]. The proof can be found in [22], and is not reported here for the sake of brevity.

A class of Lyapunov functions is of particular interest:

**Corollary 2:** If there exist a symmetric copositive matrix \( W \), and two positive real numbers \( \varepsilon \in R^+ \) and \( B \in R^+ \), such that, given the function \( V(X_n) = x_n^T W x_n \), it holds

\[
E[V(X_{n+1}) - V(X_n) | Y_n] < -\varepsilon \|X_n\| \quad \forall Y_n : \|X_n\| > B
\]  

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

This is a re- phrasing of the results presented in [26, Sect.IV]. Readers are again referred to such publication for a proof.

Being the identity matrix \( I \) a symmetric positive semidefinite matrix, hence a copositive matrix, it is possible to conclude that:

**Corollary 3:** If there exists \( \varepsilon \in R^+ \) and \( B \in R^+ \) such that

\[
E[(x_n^I - d_n^I) | x_n^I > 0] < -\varepsilon \quad \forall i = 1, \ldots , N
\]  

then the system of queues is strongly stable.

**Proof:** Starting from (8) we can write

\[
E[X_{n+1} X_{n+1}^T - X_n X_n^T | Y_n] =
\]

\[
= E[(X_n + A_n - D_n)(X_n + A_n - D_n)^T - X_n X_n^T | Y_n]
\]

\[
= E[(A_n - D_n)X_n^T(An - D_n)A_n - D_n)^T | Y_n]
\]

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

\[
\lim_{\|X_n\| \to \infty} \frac{E[(A_n - D_n)(A_n - D_n)^T | Y_n]}{\|X_n\|} = o(1)
\]

As a conclusion

\[
\lim_{\|X_n\| \to \infty} \frac{E[X_{n+1} X_{n+1}^T - X_n X_n^T | Y_n]}{\|X_n\|} =
\]

\[
= \lim_{\|X_n\| \to \infty} \frac{2E[(A_n - D_n)X_n^T | Y_n]}{\|X_n\|}
\]

and from (9) we have

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

\[
< -\varepsilon \max x_n^I \quad \text{such that} \quad \|X_n\| > B
\]

< -\varepsilon
Thus, for some \( B \in \mathbb{R}^+, \varepsilon \in \mathbb{R}^+, ||X_n|| > B \)
\[
E[X_{n+1}X_{n+1}^T - X_nX_n^T | Y_n] < -\varepsilon ||X_n||
\]

Note that Theorem 3 concerns just one queue, and states that such queue is stable provided that the average number of arrivals when the server is busy is less than the average number of departures. This result is standard in queuing theory, but a formulation in terms of the Lyapunov function is convenient for the extension to more complex setups that we shall consider in this paper. One such generalization is given by the following theorem.

**Theorem 4:** Consider two queueing systems \( S_1 \) and \( S_2 \), each one composed of \( N \) queues. Let the arrival processes at each queue for both systems be statistically identical. Let \( X_{S_1,n} \), \( D_{S_1,n} \), and \( X_{S_2,n} \), \( D_{S_2,n} \), be the queue length and departure vectors of \( S_1 \) and \( S_2 \), respectively. Assume that (8) holds for \( X_{S_1,n} \), and there exist \( \varepsilon \in \mathbb{R}^+ \), \( B \in \mathbb{R}^+ \) such that for \( ||X_{S_1,n}|| > B \) and \( ||X_{S_2,n}|| > B \)
\[
E[D_{S_1,n}X_{S_1,n}^T - D_{S_2,n}X_{S_2,n}^T | Y_{S_1,n} = Y_{S_2,n}] < -\varepsilon
\]
then system \( S_2 \) is strongly stable (and all the polynomial moments of the queue length distribution are finite).

**Proof:** If (8) holds for \( X_{S_1,n} \), then for some \( B_1 > 0 \), \( \varepsilon > 0 \)
\[
E[X_{S_1,n+1}X_{S_1,n+1}^T - X_{S_1,n}X_{S_1,n}^T | Y_{S_1,n}] < -\varepsilon ||X_{S_1,n}|| < B_1
\]
but, as shown in the previous proof,
\[
\lim_{||X_{S_1,n}|| \to \infty} \frac{E[X_{S_1,n+1}X_{S_1,n+1}^T - X_{S_1,n}X_{S_1,n}^T | Y_{S_1,n}]}{||X_{S_1,n}||} = -\varepsilon
\]

For system \( S_2 \)
\[
\lim_{||X_{S_2,n}|| \to \infty} \frac{E[X_{S_2,n+1}X_{S_2,n+1}^T - X_{S_2,n}X_{S_2,n}^T | Y_{S_2,n}]}{||X_{S_2,n}||} = -\varepsilon
\]

As a consequence
\[
\lim_{||X_{S_2,n}|| \to \infty} \frac{E[X_{S_2,n+1}X_{S_2,n+1}^T - X_{S_2,n}X_{S_2,n}^T | Y_{S_2,n}]}{||X_{S_2,n}||} < -\varepsilon
\]

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.

**IV. RATE-DRIVEN SELECTION POLICIES**

In this section we consider very simple scheduling algorithms, which determine the set of non-contending cells that are transferred from inputs to outputs in each internal time slot with a random selection policy based on the values of the average arrival rates \( r_{ij} \) of cells at queues \( q_{ij} \), measured in cells/external slot.

Let \( p_{ij} = r_{ij}/\max(\sum_j r_{ij}, \sum_i r_{ij}) \). From (1), we obtain \( p_{ij} \geq r_{ij} \), and
\[
p_i = \sum_{j=1}^{N} p_{ij} \leq 1 \quad i = 1, \ldots, N
\]

**Definition 10:** A CIOQS adopts a **Random Rate-Driven** (RRD) Selection Policy (SP) if the selection of the set of non-contending 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 \( i \)-th 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 input queue.

**Definition 11:** A CIOQS adopts a **Longest-Queue Rate-Driven** (LQRD) selection policy if the selection of cells to be transferred from inputs to outputs is performed according to the following algorithm:
1. (As in Definition 10)
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.

**Definition 12:** A CIOQS adopts an **Enhanced Longest-Queue Rate-Driven** (ELQRD) selection policy 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 $i$-th input, within its own VOQ structure, chooses a non-empty 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 11)

Theorem 5: Under admissible load conditions, an $N \times N$ CIOQS adopting a RRSD-SP achieves 100% throughput for any speed-up $S \geq 2$.

Proof: Denote by $d_{n}^{ij}$ 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 non-empty queues $q_{ij}$, it is possible to find $\epsilon \in \mathbb{R}^+$ such that $E[a_{n}^{ij} - d_{n}^{ij} | x_{n}^{ij} > 0] < -\epsilon$ for some $\epsilon > 0$, 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-SP the selection of the set of non-contending 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, $E[a_{n}^{ij} | x_{n}^{ij} > 0]$ is equal to $\frac{1}{2}$, for construction, Proposition 1 applies, and it is possible to see that the second term of the expression in (13) 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

$$1 + \sum_{i=1}^{N-1} (-1)^i \left(1 - \frac{i}{2}\right) F_i(A) \geq 1 - \frac{1}{2} F_1(A) \geq \frac{1}{2} \quad (14)$$

We can now combine (11) and (14) to obtain

$$P \frac{p_{tr}}{P} \geq L \geq \frac{1}{2}$$

Corollary 4: Under any admissible uniform load, an $N \times N$ CIOQS adopting a RRSD-SP achieves 100% throughput for any speed-up $S \geq \frac{1}{1-\frac{1}{n}}$, and for $N \rightarrow \infty$, i.e., for switches with very large number of ports, a speed-up $S \geq \frac{1}{1-\frac{1}{N}}$ is sufficient to achieve 100% throughput.

Proof: In this case, $r_{ij} = r_{kl} = r \forall i, j, k, l \in [1, \ldots, N]$ and $p_{ij} = p_{kl} = p = \frac{1}{N} \forall i, j, k, l \in [1, \ldots, N]$.

It is now easy to evaluate the expression for $L$

$$L = \sum_{i=0}^{N-1} \frac{1}{i+1} \left(N-1\right) p^i (1-p)^{N-1-i}$$

and

$$\lim_{N \rightarrow \infty} L(N) = 1 - \frac{1}{e}$$

Figure 1 plots $P$ for the third input and a generic output $j$ in a switch with $N = 4$ inputs and outputs, versus different values of arrival rates $r_{ij}$, when output $j$ is at maximum admissible load: $\sum_{i=1}^{4} r_{ij} = 1$. In these conditions $LS > 1$ guarantees stability, hence $1/L$ indicates a lower bound to the value of throughput that guarantees stability.

Corollary 5: Under any admissible load, an $N \times N$ CIOQS adopting a LQRD-SP achieves 100% throughput for any speed-up $S \geq 2$.

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

\[
E[D_n(\text{RRD})X_n^T - D_n(LQRD)X_n^T | X_n] = \sum_{C_n} E[D_n(\text{RRD})X_n^T - D_n(LQRD)X_n^T | X_n, C_n] \cdot P(C_n | X_n) \leq 0 \quad \forall X_n
\]

Hence, for Theorem 4, the system of queues is stable. ■

**Corollary 6:** Under any admissible load, an \( N \times N \) CIOQS adopting an ELQRD-SP achieves 100% throughput for any speed-up \( S \geq 2 \).

**Proof:** Note that, given \( X_n \), ELQRD-SP guarantees that the size of the sets of non-empty 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 non-empty queue). For each non-empty queue, the probability of being selected as candidate under ELQRD-SP is therefore not smaller than under LQRD-SP.

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

- each maximal candidate set \( C_n(LQRD) = C_1 \) is put in correspondence with set \( C_n(ELQRD) = C_2 \), so that \( C_1 \subseteq C_2 \);
- each non-maximal 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 \( P_{ELQRD}(C_2 | C_1) \) that \( C_2 \) is obtained according to ELQRD-SP, given that some inputs have already chosen their candidate queue \((C_1)\) is the non-empty set of candidate queues that have been already chosen by some inputs). Note that \( \sum_{C_2} P_{ELQRD}(C_2 | C_1) = 1 \).

It is possible to see that, starting from the candidate sets \( C_1 \) generated with LQRD-SP, and completing each non-maximal 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-SP candidate sets is perfectly emulated.

Given a set \( C_n \) of non-empty 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 \( C_1 \subseteq C_2 \), then \( E[D_nX_n^T | X_n, C_1] \leq E[D_nX_n^T | X_n, C_2] \).

As a consequence, Theorem 4 applies, so that

\[
E[D_n(\text{LQRD})X_n^T - D_n(\text{ELQRD})X_n^T | X_n] \leq 0 \quad \forall X_n
\]

and the system of queues is proven to be stable. ■

V. QUEUE-LENGTH-DRIVEN SELECTION POLICIES

In this section we prove that a simple scheduling algorithm that determines the set of non-contending 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 derive some preliminary results.

**Definition 13:** Let \( U \) be the set of vectors \( V \in \mathbb{R}^{+N^2} \) such that

\[
\sum_{i=1}^{N} V_{i+jN} \leq 1 \quad j = 0, \ldots, N - 1
\]

(16)

**Definition 14:** Given a vector \( V \neq 0, V \in \mathbb{R}^{+N^2} \) let \( \hat{V} \) be the maximal vector parallel to \( V \) in \( U \), i.e., \( \hat{V} \in U \), \( \hat{V} = \max_{k, \sqrt{v_k}} k V \), \( k \in \mathbb{R} \).

**Definition 15:** Given a vector \( V \neq 0, V \in \mathbb{R}^{+N^2} \), define \( \hat{V} = \frac{V}{\|V\|} = \frac{\sqrt{V}}{\|V\|} \).

**Definition 16:** Let \( \Gamma_\gamma = \hat{V}^T \hat{V} \) be the symmetric matrix associated with the projection operator along the direction of \( \hat{V} \) (i.e., \( X\Gamma_\gamma = (X\hat{V}^T)\hat{V} \)). Note that \( X\Gamma_\gamma = X \).

**Definition 17:** The norm of matrix \( \Gamma_\gamma \) is defined as

\[
\|\Gamma_\gamma\| = \sup_{X \in \mathbb{R}^{N^2}, X \neq 0} \frac{\|X\Gamma_\gamma\|}{\|X\|}
\]

**Proposition 2:** \( \|\Gamma_\gamma\| \leq 1 \).

**Proof:** By Schwarz inequality

\[
\|X\Gamma_\gamma\| = \|(X\hat{V}^T)\hat{V}\| = (X\hat{V}^T)\|\hat{V}\| \leq \|X\| \|\hat{V}\| \|\hat{V}\| = \|X\|
\]

Equality is obtained for \( X \) parallel to \( \hat{V} \). ■

**Lemma 1:** For each \( V \in \mathbb{R}^{N^2} \) the matrix \( Q = I - \gamma \Gamma_\gamma \) is a symmetric positive definite matrix for each \( 0 \leq \gamma < 1 \), and it is a symmetric positive semidefinite matrix for \( \gamma = 1 \).

**Proof:** Let \( X \) be a vector in \( \mathbb{R}^{+N^2} \)

\[
XQX^T = X(I - \gamma \Gamma_\gamma)X^T = \|X\|^2 - \gamma X\hat{V}^T\hat{V}X = \|X\|^2 - \gamma \|X\|^2 (\hat{V}^T\hat{V}) \geq 0
\]

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

Proof: Consider the $N^2 \times N^2$ positive (semi)definite matrix $Q = I - \gamma E[A]$, where $0 \leq \gamma \leq 1$ and $E[A] \in U$ is the vector of the average cell arrival rates $r_{ij}$. By defining $W = XQX^T$ as the Lyapunov function for the CIOQS, we prove that for some $B \in \mathbb{R}^+, \epsilon \in \mathbb{R}^3$, there exists $\gamma$ such that

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

Hence Theorem 2 applies, and the CIOQS is strongly stable. Indeed, for $\|X_n\|$ growing to infinity, and using (3),

$$E[X_{n+1}QX_{n+1}^T - X_nQX_n^T | X_n] \approx \frac{2E[A_nQX_n^T - D_nQX_n^T | X_n]}{\|X_n\|} = 2 \left\{ E[A_n]X_n^T - E[D_n]X_n^T - \gamma E[A_n]X_n^T + \gamma E[D_n]\Gamma E[A_n]X_n^T \right\}$$

$$= \left\{ E[A_n]X_n^T(1 - \gamma) - (1 + \alpha)X_n^T + \gamma(1 + \alpha)X_n\Gamma E[A_n]X_n^T \right\} = 2 \left\{ E[A_n]X_n^T(1 - \gamma) - (1 + \alpha)X_n^T + \gamma(1 + \alpha)X_n\Gamma E[A_n]X_n^T \right\} = F(\gamma, \bar{X}_n)$$

Note that the domain of $F(\gamma, \bar{X}_n)$, for a given $\gamma$, is the surface of the unit sphere in $\mathbb{R}^{+N^2}$, and that $F(\gamma, \bar{X}_n)$, for a given $\bar{X}_n$, is linear in $\gamma$, hence

$$F(\gamma, \bar{X}_n) = F(1, \bar{X}_n) + (\gamma - 1) \frac{\partial}{\partial \gamma} F(\gamma, \bar{X}_n) |_{\gamma = 1}$$

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

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

Note that $\partial F(\gamma, \bar{X}_n)/\partial \gamma$, performed for $\bar{X}_n$ parallel to $E[A_n]$ is strictly positive. As a consequence, there exists a $\epsilon-$sphere around $\bar{X}_n = E[A_n]$ where such derivative remains larger than a finite positive constant. This implies that, in each point inside the $\epsilon$-sphere, for any $0 \leq \gamma < 1$, $F(\gamma, \bar{X}_n)$ is smaller than a finite negative constant.

Outside the $\epsilon$-sphere, the domain of $F(\gamma, \bar{X}_n)$ is closed, hence the maximum value exists; moreover, being away from $\bar{X}_n = E[A_n], \max_{\bar{X}_n} F(\gamma, \bar{X}_n)$ is strictly negative for $\gamma = 1$. For continuity, $\max_{\bar{X}_n} F(1 - \delta, \bar{X}_n)$ keeps negative for $\delta$ sufficiently small. As a consequence, for $\gamma = 1 - \delta, F(\gamma, \bar{X}_n)$ is smaller than a finite negative constant for all possible values of $\bar{X}_n$. 

Theorem 7: In an $N \times N$ CIOQS with VOQ at each input, a scheduling algorithm such that $E[D_n] = \bar{X}_n(1 + \alpha) + D' \in \mathbb{R}^{+N^2}$ is stable for each $\alpha \in \mathbb{R}^+$. 

Proof: Since $D' \in \mathbb{R}^{+N^2}$, two cases are possible.

- $E[A_n] - D'$ is not in $U$ due to negative components; in this case it is possible to split $D' = D'' + D'''$, so that $E[A_n] - D'' \in U$, 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 $Q = I - \gamma E[A_n]$ in the previous proof (note that $D'''\Gamma E[A_n] - D' = 0$, because of the orthogonality between $D'''$ and $E[A_n] - D'$).

Definition 18: A CIOQS adopts a Longest-Queue-Driven (LQD) selection policy if the selection of the set of non-contending cells to be transferred from inputs to outputs at each internal time slot is performed according to the following algorithm.

- At each internal time slot $n$, the $i$-th input, within its own VOQ structure, chooses queue $q_{ij}$ with a probability proportional to the queue length $x_{ij}^T$. The probability of selecting queue $q_{ij}$ is $p_{ij} = x_{ij}^T / \max(\sum_{j=1}^{N} x_{ij}^T, \sum_{i=1}^{N} x_{ij}^T)$. With probability $1 - \sum_{j=1}^{N} p_{ij}$ 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 $j$).

- 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, an $N \times N$ CIOQS adopting a LQD-SP achieves 100% throughput for any speed-up $S \geq 2$.

Proof: The average number of times that a particular queue is selected as candidate in internal time slots is $p_{ij}$. But $p_{ij} \geq x_{ij}^T$ by definition. From the proof of Theorem 5, follows that the probability that each queue is selected for cell transfer in each internal time slot is no less than $1/2$. As a consequence, using speed-up $S \geq 2$ we have $E[D_n(LQD)] = (1 + \alpha)\bar{X}_n + D'_n$. 

However, vector $D'_n$ is a function of $\bar{X}_n$, so that Theorem 7 does not directly apply. $D'_n$ is a function of $\bar{X}_n$, but does not depend on $\|X_n\|$. Considering the evolution of the system we can write: $E[X_{n+1}] = E[X_n] + E[A_n] - (1 + \alpha)\bar{X}_n - D'(\bar{X}_n)$. Without loss of generality, we assume $E[A_n] = D'_n \in U$. 

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

$$Q_n = \beta(\bar{X}_n) (I - \gamma \Gamma E[A] - D'_n)$$

where $\beta(\bar{X}_n)$ is a function of $\bar{X}_n$. 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 \( E[X_{n+1}^T Q_n X_{n+1}^T] \) we get
\[
\lim_{||X_n|| \to \infty} \frac{1}{||X_n||} E[X_{n+1}^T Q_n X_{n+1}^T - X_n^T Q_n X_n^T + X_n^T Q_n X_n^T - X_n^T Q_n X_{n+1}^T | X_n] < -\epsilon
\]
but from Theorem 7 follows that
\[
\lim_{||X_n|| \to \infty} \frac{E[X_{n+1}^T Q_n X_{n+1}^T - X_n^T Q_n X_n^T | X_n]}{||X_n||} < -\epsilon
\]
In addition it can be proved that
\[
\lim_{||X_n|| \to \infty} \frac{E[X_n^T Q_n X_n^T - X_n^T Q_{n-1} X_n^T | X_n]}{||X_n||} \leq 0
\]
(for a proof see [22]).

VI. DETERMINISTIC SELECTION POLICIES

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

**Definition 19:** A CIOQS adopts a Maximum Weight Matching (MWM) selection policy if the selection of the set of non-contending cells to be transferred from inputs to outputs at each internal time slot is performed according to the Maximum Weight Matching algorithm [27].

Let \( W_n \) represent a weight vector at time \( n \), and let \( D_n \) be an admissible departure vector. The departure vector produced by a MWM-SP is such that
\[
D_n(MWM)W_n^T = \max_{D_n} D_n(W_n^T)
\]
If we set \( W_n = X_n \), where \( X_n \) represents the state at time \( n \) of the system of queues of a \( N \times N \) CIOQS with VOQ at each input, the departure vector produced by a MWM-SP is such that
\[
D_n(MWM)X_n^T = \max_{D_n} D_n(X_n^T)
\]
In [6], using the Lyapunov function \( V(X_n) = X_n X_n^T \), it was proved that, for any CIOQS adopting a MWM-SP with \( W_n = X_n \),
\[
E[X_{n+1}^T X_{n+1}^T - X_n^T X_n^T | X_n] < -\epsilon||X_n||
\]
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 \( N \times N \) CIOQS adopting a MWM-SP with weights equal to queue lengths is strongly stable for any speed-up \( S \geq 1 \).

The algorithmic complexity required for the computation of the MWM departure vector is quite large (algorithms are known with asymptotic complexity \( O(N^3 \log N) \), see[27]). 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.

**Corollary 7:** Assume that a selection policy \( P \) is found such that
\[
D_n(P)^T X_n^T > \left( \frac{1}{S} + \epsilon \right) D_n(MWM)X_n^T
\]
for some \( \epsilon \in \mathbb{R}^+ \), \( B \in \mathbb{R}^+ \), for each queue length vector \( X_n \), with \( ||X_n|| > B \). Given \( P \), a new selection policy \( P(S) \) is defined, according to which \( P \) is executed only once in each external time slot, to select a set of non-contending cells, whose transfer is enabled \( S \) times, in each one 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 \( P(S) \) is stable under any admissible traffic pattern.

**Proof:** Let \( D_n(P(S)) \) be the global departure vector, referring to one whole external time slot; the \( i \)-th component of \( D_n(P(S)) \) is \( d_n^i(P(S)) = \min(S d_n^i(P), x_n^i) \), where \( d_n^i(P) \) is the \( i \)-th component of \( D_n(P) \).

Let \( D_{\delta,n} = S D_n(P) - D_n(P(S)) \). Note that, by construction, the non-null components of \( D_{\delta,n} \) correspond to components of \( X_n \) referring to selected queues with size smaller than \( S \); as a consequence, \( D_{\delta,n} X_n < (S - 1)^2 \). Then
\[
D_n(P(S))X_n^T = (S D_n(P) - D_{\delta,n})X_n^T = S D_n(P)X_n^T - D_{\delta,n} X_n^T = (1 + S \epsilon) \max_{D_n} [D_n X_n^T] - N(S - 1)^2
\]
For \( ||X_n|| \) sufficiently large, so that
\[
\max_{D_n} [D_n X_n^T] > N(S - 1)^2 / \epsilon
\]
we have
\[
D_n(P(S))X_n^T > \max_{D_n} [D_n X_n^T]
\]
and Theorem 4 applies.

**Corollary 8:** Assume that a CIOQS adopts a selection policy \( P^* \), which is executed at each internal time slot to select a set of non-contending cells. A CIOQS with speed-up \( S = 2 \) adopting policy \( P^* \) is stable under any admissible traffic pattern if
\[
D_n(P^*)X_n^T \geq \frac{1}{2} D_n(MWM)X_n^T + N_i \quad i = 1, 2
\]
where \( N_i \) is the number of queues selected by both \( P^* \) and MWM at the \( i \)-th internal time slot.

**Proof:** Let \( D_{1,n} \) and \( D_{2,n} \) be the departure vectors referring to the first and second internal time slots of the \( n \)-th external time slot, according to policy \( P^* \). We assumed that
\[
D_{1,n}X_n^T \geq \frac{1}{2} D_n(MWM)X_n^T + N_1 \quad (18)
\]
Let \( X_n = X_n - D_{1,n} \) be the queue lengths vector at the end of the first internal time slot. Let \( D_n(MWM) \) be the departure vector produced by MWM for the queue lengths vector \( X_n \), and \( D_{1}(MWM) \) be the departure vector produced by MWM for vector \( X_n^i \), i.e., \( D_{1}(MWM)X_n^T = \max_{D_n} (D_nX_n^T) \). We
Definition 20: A CIOQS adopts a Greedy Maximal Weight Matching (GMWM) selection policy if the selection of the set of non-contending 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 \(q_{ad}\)) 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.

**Corollary 9:** A CIOQS implementing a GMWM-SP is stable for all \(S \geq 2\).

**Proof:** We show that, for each queue size vector \(X_n\), \(D_n(G M W M)X_n^T > \frac{1}{2}D_n(M W M)X_n^T + K\), where \(K\) is the number of queues from which a cell is transferred according to both \(D_n(G M W M)\) and \(D_n(M W M)\).

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

\[
D_nX_n^T = \sum_{i=1}^{N} d_n^i x_n^i = \sum_{\delta q_{ij}=1} x_n^i
\]

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

If the cardinality of \(I_n^1(M W M)\) or \(I_n^1(G M W M)\) is smaller than \(N\), we can augment the two sets by adding some empty queues, so that the augmented sets comprise \(N\) non conflicting queues.

Let \(g_{i,j,n}\) be the longest queue in \(I_n^1(G M W M)\). If \(g_{i,j,n} \in I_n^1(M W M)\), set \(I_n^2(M W M) = I_n^1(M W M) - \{g_{i,j,n}\}\) and \(I_n^2(G M W M) = I_n^1(G M W M) - \{g_{i,j,n}\}\).

Otherwise, select all queues in \(I_n^1(M W M)\) that conflict with \(g_{i,j,n}\) (i.e., all queues on input \(i\) or directed to output \(j\); the selection returns at most two queues: \(m_{i,j,n}^1\) (conflicting with \(g_{i,j,n}\) on input \(i\)), and \(m_{i,j,n}^1\) (conflicting with \(g_{i,j,n}\) on output \(j\)).

By construction, the lengths of queues \(m_{i,j,n}^1\) and \(m_{i,j,n}^2\) cannot exceed the length of queue \(g_{i,j,n}\). Thus, the sum of the lengths of the two queues is less or equal to twice the length of \(g_{i,j,n}\).

Set \(I_n^2(G M W M) = I_n^1(G M W M) - \{g_{i,j,n}\}\) and \(I_n^2(M W M) = I_n^1(M W M) - \{m_{i,j,n}^1, m_{i,j,n}^2\}\).

Note that \(I_n^2(M W M)\) cannot comprise queues conflicting with \(g_{i,j,n}\).

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

A variation of the GMWM-SP in accordance with Corollary 7 can be proven to be stable with similar techniques.

**VII. Conclusions**

In this paper we computed stability conditions for several scheduling algorithms used in combined input/output queuing 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 stability to most algorithms, when Virtual Output Queueing is implemented, and the policy to select the set of non-contending data units avoids head-of-the-line blocking phenomena.

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. www.cisco.com
2. www.lucent.com/dns/products/a500.html
For each set $A = \{a_i \in \mathbb{R}^+ \mid \sum_{i=1}^{N} a_i \leq 1\}_{i=1}^{N}$, it results

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

**Proof:** By definition

$$F_{k+1}(A) = \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]}) = \sum_{i=0}^{k} \frac{1}{k+1} \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]}) = \frac{1}{k+1} \sum_{i=0}^{k} \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]})$$

in (21) we can group all the $(k+1)\binom{N}{k+1}$ terms in $\frac{k+1}{N} \binom{N}{k+1} = \binom{N}{k}$ subsums, each one comprising $N-k$ different terms. A bijective correspondence between set $\alpha^{[k]} \in A_k$ and subsums is established according to the following rule: each subsum comprises the $N-k$ terms of (21) associated with the $N-k$ different sets $\alpha^{[k+1]}$ so that $\alpha^{[k]} \subset \alpha^{[k+1]}$.

It is thus possible to write

$$F_{k+1}(A) = \sum_{\alpha^{[k]} \in A_k} \sum_{\alpha^{[k+1]} \in A_{k+1}} \frac{1}{k+1} f(\alpha^{[k+1]})$$

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

$$\sum_{\alpha^{[k+1]} \supseteq \alpha^{[k]}} f(\alpha^{[k+1]}) < f(\alpha^{[k]})$$

As a consequence, by substituting (23) in (22), we obtain

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

**APPENDIX A**

**Proposition 1:** For each set $A = \{a_i \in \mathbb{R}^+ \mid \sum_{i=1}^{N} a_i \leq 1\}_{i=1}^{N}$, it results

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

**Proof:** By definition

$$F_{k+1}(A) = \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]}) = \sum_{i=0}^{k} \frac{1}{k+1} \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]}) = \frac{1}{k+1} \sum_{i=0}^{k} \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]})$$

in (21) we can group all the $(k+1)\binom{N}{k+1}$ terms in $\frac{k+1}{N} \binom{N}{k+1} = \binom{N}{k}$ subsums, each one comprising $N-k$ different terms. A bijective correspondence between set $\alpha^{[k]} \in A_k$ and subsums is established according to the following rule: each subsum comprises the $N-k$ terms of (21) associated with the $N-k$ different sets $\alpha^{[k+1]}$ so that $\alpha^{[k]} \subset \alpha^{[k+1]}$.

It is thus possible to write

$$F_{k+1}(A) = \sum_{\alpha^{[k]} \in A_k} \sum_{\alpha^{[k+1]} \in A_{k+1}} \frac{1}{k+1} f(\alpha^{[k+1]})$$

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

$$\sum_{\alpha^{[k+1]} \supseteq \alpha^{[k]}} f(\alpha^{[k+1]}) < f(\alpha^{[k]})$$

As a consequence, by substituting (23) in (22), we obtain

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

**APPENDIX A**

**Proposition 1:** For each set $A = \{a_i \in \mathbb{R}^+ \mid \sum_{i=1}^{N} a_i \leq 1\}_{i=1}^{N}$, it results

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

**Proof:** By definition

$$F_{k+1}(A) = \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]}) = \sum_{i=0}^{k} \frac{1}{k+1} \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]}) = \frac{1}{k+1} \sum_{i=0}^{k} \sum_{\alpha^{k+1} \in A_{k+1}} f(\alpha^{[k+1]})$$

in (21) we can group all the $(k+1)\binom{N}{k+1}$ terms in $\frac{k+1}{N} \binom{N}{k+1} = \binom{N}{k}$ subsums, each one comprising $N-k$ different terms. A bijective correspondence between set $\alpha^{[k]} \in A_k$ and subsums is established according to the following rule: each subsum comprises the $N-k$ terms of (21) associated with the $N-k$ different sets $\alpha^{[k+1]}$ so that $\alpha^{[k]} \subset \alpha^{[k+1]}$.

It is thus possible to write

$$F_{k+1}(A) = \sum_{\alpha^{[k]} \in A_k} \sum_{\alpha^{[k+1]} \in A_{k+1}} \frac{1}{k+1} f(\alpha^{[k+1]})$$

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

$$\sum_{\alpha^{[k+1]} \supseteq \alpha^{[k]}} f(\alpha^{[k+1]}) < f(\alpha^{[k]})$$

As a consequence, by substituting (23) in (22), we obtain

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