Throughput Region of Finite-Buffered Networks

Paolo Giaccone, *Member*, *IEEE*, Emilio Leonardi, *Member*, *IEEE*, and Devavrat Shah, *Member*, *IEEE*

Abstract—Most of the current communication networks, including the Internet, are packet switched networks. One of the main reasons behind the success of packet switched networks is the possibility of performance gain due to multiplexing of network bandwidth. The multiplexing gain crucially depends on the size of the buffers available at the nodes of the network to store packets at the congested links. However, most of the previous work assumes the availability of infinite buffer-size. In this paper, we study the effect of finite buffer-size on the performance of networks of interacting queues. In particular, we study the throughput of flow-controlled loss-less networks with finite buffers. The main result of this paper is the characterization of a dynamic scheduling policy that achieves the maximal throughput with a minimal finite buffer at the internal nodes of the network under memory-less (e.g., Bernoulli IID) exogenous arrival process. However, this ideal performance policy is rather complex and, hence, difficult to implement. This leads us to the design of a simpler and possibly implementable policy. We obtain a natural trade-off between throughput and buffer-size for such implementable policy. Finally, we apply our results to packet switches with buffered crossbar architecture.

Index Terms—Queuing theory, flow-controlled networks, scheduling, packet switching, buffered crossbars.

1 Introduction

Most of the current communication networks are packet switched networks. A prominent feature of packet networks is the performance gain that can be obtained due to multiplexing of bandwidth. However, this requires some form of scheduling policy to coordinate the transfer of packets at the congested resources. As a consequence, the performance of such networks, in terms of throughput, depends on the scheduling policy.

The seminal work of Tassiulas and Ephremides [33] pioneered the research for studying the maximal throughput of *networks of interacting queues* (also called *constrained queueing systems*). Their scheduling policy is based just on the actual queues state without requiring any knowledge on the traffic pattern. The methods of [33] have been utilized in the context of packet switching [1], [8], [17], [22], [36], satellite and wireless networks [26], [27], etc. Although those results are quite general, they assume the availability of infinite buffers at all the nodes in the network. However, in the practical setup, buffer-sizes are always finite. This is a major limitation of the previous results.

In this paper, to overcome this limitation, we study the maximal achievable throughput in flow-controlled loss-less networks of interacting queues, with finite buffers at the internal nodes of the network; the flow control mechanism prevents the queue from overflowing. To be able to define formally the throughput region in such loss-less networks, we allow only the ingress nodes to have infinite buffer sizes. Our work can be seen as an extension of the results of [33] in the sense that it gets rid of the assumption of infinite buffers inside the network.

As an application of our results, we evaluate the maximal achievable throughput in packet switch architectures built around a crossbar with buffered crosspoints. Such switches have become architecturally appealing due to recent advances in the technology [38]. They allow for the possibility of simpler scheduling algorithms. Hence, they have received a great deal of attention recently. However, a little progress has been made in the context of designing simple throughput maximal scheduling algorithms. Based on our results for general networks, we propose a novel distributed scheduling policy, called DMWF. Under admissible Bernoulli IID traffic, we show that DMWF is stable if enough internal buffer is available. In particular, we evaluate the natural trade-off between throughput and buffer-size at crosspoints. Finally, noticing that in switching architectures an internal speedup is usually adopted to compensate throughput penalties, we highlight that our results allow also to evaluate the trade-off between speedup and buffer-size at crosspoints.

The contribution of this paper is rather theoretical in nature and may not be useful in practice, but we believe that these results will provide useful guidelines to design practical algorithms and to size buffers.

1.1 Organization

The rest of the paper is organized as follows: In Section 2, we introduce setup and notation of this paper. We then recall known results about the maximal throughput policies for networks with infinite buffer in Section 2.3. We present

- *P. Giaccone and E. Leonardi are with the Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi 24, 10129 Torino, Italy. E-mail: {paolo.giaccone, emilio.leonardi}@polito.it.*
- *D. Shah is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307. E-mail: devavrat@mit.edu.*

Manuscript received 20 Apr. 2005; revised 21 Oct. 2005; accepted 27 Oct. 2005; published online 27 Dec. 2006.
Recommended for acceptance by A. Pietracaprina.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0234-0405.

1045-9219/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

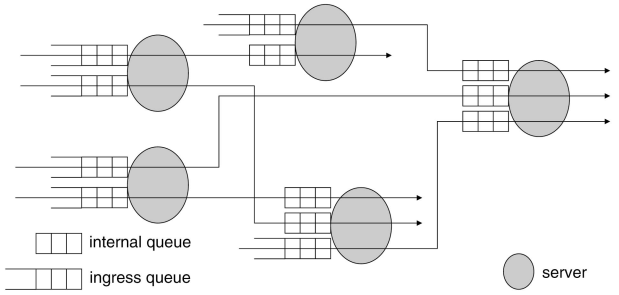

Fig. 1. Example of flow-controlled network of interacting queues with five servers and six flows.

our main results in Section 3. In Section 4, we introduce the buffered crossbar switch architecture and apply our results to obtain a distributed scheduling policy for buffered crossbar. For the ease of exposition, we present the proofs in the appendix of the paper. Finally, Section 5 provides the key ideas to compute the bounds on achievable throughput when also the ingress buffers are finite and losses are allowed.

2 Setup

2.1 Notation and Model

We consider a network of discrete-time stations (or servers or nodes), handling J packet (or customer) flows. To describe the network model and dynamics, we mainly need to specify:

1. arrival process of flows,
2. flow routing,
3. queuing at nodes, and
4. service (or scheduling) policy.

Before we do so, we present some notations that will be useful for the rest of the paper.

Let time be denoted by $n \in \mathbb{N}$. Packets of any given flow enter the network at a specific station, follows a flow dependent acyclic path through the network, and finally leave the network. The number of stations (hops) traversed by packets of flow $j, 1 \leq j \leq J,$ is h_j. We assume that the routing is deterministic. An example of such a network is depicted in Fig. 1. At each station, a separate virtual queue is maintained for each flow passing through it; the set of all virtual queues residing at the station forms the physical queue. Let the total number of virtual queues in the network be Q. Let $q(j,h)$, $1 \leq h \leq h_j$, be the virtual queue traversed by flow j on its hth hop. Queue $q(j,1)$ is called the "ingress queue" for flow j. The set of all ingress queues $\{q(j,1),\ 1 \leq j \leq J\}$ is denoted with Φ_I. The remaining virtual queues are called "internal queues" and the set of these queues is denoted by Φ_M.

Next, we define map f, where $f(q(j,h)) = q(j,1)$; thus, f maps $q(j,h)$ to its ingress queue. Let $u(q)$ be the upstream queue to q, that is, $u(q(j,h)) = q(j,h-1),\ 2 \leq h \leq h_j$. Similarly, let $p(q)$ be the downstream queue to q, i.e., $p(q(j,h)) = q(j,h+1),\ 1 \leq h \leq h_j - 1$. Finally, $j(q)$ returns the index of the flow traversing queue q. Since we assume deterministic routing, let the $Q \times Q$ matrix $\mathbf{R}$ be a 0-1 *routing matrix*, with binary elements $\mathbf{R}_{q_1q_2} = 1$ iff $p(q_1) = q_2$, and $\mathbf{R}_{q_1q_2} = 0$ otherwise.

We assume that packets are of fixed length, all servers have the same capacity and each server takes unit time to serve a packet.

Now, we are ready to describe the discrete time network dynamics. Let $X(n) = [x_q(n)]_{q=1}^{Q}$ be the vector[1] whose qth component, $x_q(n)$, represents the number of packets (or size of the queue) in the qth queue at the beginning of time n.[2] For ease of exposition, we will abuse notation by using $x_{j,h}(n)$ for $x_{q(j,h)}(n)$. We suppose the buffer size of all the ingress virtual queues (belonging to Φ_I) to be infinite, whereas we assume all the other virtual queues along the flow paths (belonging to Φ_M) to be of *finite buffer-size*. All the virtual queues traversed by flow j have a buffer size of l_j packets. The queues evolve as follows: For $1 \leq q \leq Q$,

$$x_q(n+1) = x_q(n) + e_q(n) - d_q(n),$$

where $e_q(n)$ represents the number of packets entering the queue and $d_q(n)$ represents the number of packets departing the queue in time $(n, n+1]$. Let $E(n) = [e_q(n)]_{q=1}^{Q}$ and $D(n) = [d_q(n)]_{q=1}^{Q}$. Again, we abuse notation for by using $d_{j,h}(n)$ in place of $d_{q(j,h)}(n)$.

We first note that, neither an empty queue can be serviced nor a full queue can be sent a packet (thus, restricting service of upstream queue, thanks to the flow control mechanism). This can be expressed as follows:

$$D(n) \leq X(n) \quad \text{and} \quad D(n)\mathbf{R} \leq L - X(n), \qquad (1)$$

where $L = [l_{j(q)}]_{q=1}^{Q}$ (assuming $l_{j(q)} = +\infty$ for $q \in \Phi_I$). Now, the arrivals to ingress queues depend only on exogenous process. However, arrivals to internal queues depend on the departure from other queues. Let $A(n) = [a_q(n)]_{q=1}^{Q}$, denote the exogenous arrival process. Note that $a_q(n) = 0$ if $q \in \Phi_M$ (i.e., q is not an ingress queue). The internal arrivals can be represented by $D(n)\mathbf{R}$. In summary, the evolution of queues can be rewritten as:

$$X(n+1) = X(n) + A(n) - D(n)(\mathbf{I} - \mathbf{R}), \qquad (2)$$

where $\mathbf{I}$ denotes the identity matrix.

We assume that the external arrival process $\{A(n) : n \in \mathbb{N}_+\}$ is a stationary memoryless process, i.e., $A(n)$ are IID random vectors with average $E(A(n)) = \Lambda = [\lambda_q]_{q=1}^{Q}$. In addition, we assume all the polynomial moments of $A(n)$ to be finite. Note that, since external arrivals are directed only to ingress queues, $\lambda_q = 0$ if $q \in \Phi_M$.

At time slot n, the scheduling policy selects the service vector $S(n)$, whose element $s_q(n)$ represents the amount of work provided to the qth queue during time slot n; again, $s_{j,h}(n) = s_{q(j,h)}(n)$. The departure vector $D(n)$ is related to $S(n)$ according to the following equation:

$$\sum_{t=0}^{n} D(t) = \left\lfloor \sum_{t=0}^{n} S(t) \right\rfloor \Rightarrow D(n) = \left\lfloor \sum_{t=0}^{n} S(t) \right\rfloor - \sum_{t=0}^{n-1} D(t).$$

[1] All vectors in this paper are row vectors, unless specified otherwise.

[2] Note that queue-length or queue-size is time-variable queue-occupancy, whereas buffer-size of a queue denotes the maximum number of packets that can be stored in that queue.

In other words, the number of packets served from a queue is given by the amount (approximated to the integer part) of cumulative service provided to the queue. We define the difference between the two quantities by:

$$\Delta(n) = \sum_{t=0}^{n} S(t) - \sum_{t=0}^{n} D(t),$$

whose $q\text{th}$ element $\delta_q(n) \in [0, 1)$ represents the amount of work provided by the scheduling policy to the head-of-the-line packets of the $q\text{th}$ queue at the end of time slot n. In this paper, we restrict our investigation to the class of *dynamic* scheduling policies, i.e., those scheduling policies which select $S(n)$ on the only basis of the actual queues state without requiring any knowledge on the traffic pattern.

In general, we assume that the set of possible service vectors $S(n)$ is constrained by a system of linear equations representing the topological interference among services at queues (*blocking constraints*):

$$S(n)\mathbf{K} \leq T. \tag{3}$$

Matrix $\mathbf{K}$ and vector T describe the blocking constraints in the services. For example, simple topological constraints are those expressing the fact that the sum of services provided to all the virtual queues residing at the same physical queue is limited by the server capacity to one packet per slot. However, we do not exclude additional constraints which relate the behavior of queues residing at different stations. Let $\widehat{\mathcal{D}}$ be the set of nonnegative $S(n)$ which satisfy the blocking constraints. We notice that $\widehat{\mathcal{D}}$ defines a polyhedral convex region. Let $\mathcal{D}$ the set of all vertices of $\widehat{\mathcal{D}}$. We assume that T is integer valued and $\mathbf{K}$ is totally unimodular (i.e., the determinant of all square submatrices are ±1); hence, all vectors in $\mathcal{D}$ are integer valued. Finally, we notice that all vectors $\gamma^{(q)}$, with $1 \leq q \leq Q$, whose elements are all null except the $q\text{th}$, which is unitary, belongs to $\widehat{\mathcal{D}}$, i.e., $\gamma^{(q)} = [0, 0, 0, \cdots 1, 0, 0, 0] \in \widehat{\mathcal{D}}$. We remind that $S(n)$ must be chosen in such a way that the service constraints defined for $D(n)$ in (1) are not violated.

If the scheduling policy is *atomic*, i.e., packets are transmitted by servers in an "atomic" fashion, without interleaving their transmission with packets residing in other queues, then $S(n)$ is integer valued, and $D(n) = S(n)$ for any n. In this case, $X(n)$ is a DTMC (Discrete Time Markov Chain). In the more general case, $(X(n), \Delta(n))$ is a discrete time Markov process defined on a general state space [23]. In the latter case, let us define the workload vector $Y(n) = [y_q(n)]_{q=1}^{Q}$:

$$Y(n) = X(n) - \Delta(n)(\mathbf{I} - \mathbf{R}).$$

We notice that $(X(n), \Delta(n)) \Rightarrow Y(n)$ is a one to one correspondence. Furthermore, it is easy to verify that $Y(n)$ satisfies the following system evolution equation, derived by (2):

$$Y(n+1) = Y(n) + A(n) - S(n)(\mathbf{I} - \mathbf{R}). \tag{4}$$

Note that if the scheduling policy is atomic, then $X(n) = Y(n)$ for any n and (4) coincides with (2). For the sake of easier notation, we define $x_{j,h_j+1} = y_{j,h_j+1} = 0$.

Finally, let us introduce the following useful positive convex functional:

Definition 1. *Given a vector* $Z \in \mathbb{R}_+^Q$, $Z = (z^{(q)}, 1 \leq q \leq Q)$, *the positive convex functional* $\|Z\|$ *is defined as:*[3]

$$\|Z\| = \inf\left\{\alpha \in \mathbb{R}_+ : \frac{1}{\alpha} Z \in \widehat{\mathcal{D}}\right\}.$$

In the rest of the paper, we will refer to it with the improper term of "norm."

Remark. The functional defined above is equivalent to the well-known Minkowski convex functional associated to $\widehat{\mathcal{D}}$.

From the definition of $\|Z\|$, it immediately follows that for any $S(n) \in \widehat{\mathcal{D}}$, $\|S(n)\| \leq 1$. Under atomic policy, $S(n) = D(n) \in \mathcal{D}$. Thanks to the fact that $\|\gamma^{(q)}\| = 1$, we can claim:

$$\|Z\| = \left\|\sum_{q=1}^{Q} z_q \gamma^{(q)}\right\| \leq \sum_{q=1}^{Q} z_q \|\gamma^{(q)}\| \leq \sum_{q=1}^{Q} z_q. \tag{5}$$

A generic norm on $\mathbb{R}_+^Q$ is represented by $\|\cdot\|_*$.

2.2 Stability: Definitions and Known Results

We present definitions and known results regarding the system stability in the context of stochastic network.

Definition 2. *A stationary traffic pattern is admissible if* $\|\Lambda(\mathbf{I} - \mathbf{R})^{-1}\| < 1$.

Let $\rho = \|\Lambda(\mathbf{I} - \mathbf{R})^{-1}\|$. For the simplest case in which virtual queues residing at different servers are not topologically interacting, traffic is admissible iff no servers are overloaded; in addition, ρ represents the load of the heaviest loaded server in the network.

Definition 3. *The system of queues is* stable *if:*

$$\limsup_{n\to\infty} E(\|X(n)\|_*) < \infty$$

$$\text{or equivalently :} \quad \limsup_{n\to\infty} E\|Y(n)\|_*) < \infty,$$

i.e., the system is positive (Harris) recurrent.

Note that the admissibility of traffic pattern is a necessary condition for the system of queue to be stable as shown in [33].

We say that the system is stable at point Λ if it is stable under every stationary memoryless external arrival processes $A(n)$ with average Λ and finite polynomial moments.

Definition 4. *We define as* stability region *(or throughput region) the set of points* Λ *in correspondence of which the system of queues is stable.*

Definition 5. *We say that a system of queues is 1-efficient (or equivalently achieves 100 percent throughput), if it is stable under any admissible traffic pattern.*

[3] We notice that, according to previous assumptions on $\widehat{\mathcal{D}}$, $Z = 0$ is an interior point of $\widehat{\mathcal{D}}$ in $\mathbb{R}_+^Q$. Thus, for any $Z \in \mathbb{R}_+^Q$, $\|Z\|$ is well defined, i.e., finite.

Definition 6. *For any $0 < \rho < 1$, we say that the system of queues is ρ-efficient if it is stable under any traffic pattern such that $\|\Lambda(\mathbf{I}-\mathbf{R})^{-1}\| \leq \rho$.*

The method of Lyapunov function is a powerful tool to prove stability (i.e., positive, or Harris, recurrency) of irreducible Markovian systems. Next, we recall some of the well-known results regarding Lyapunov function methodology that will be used in the remaining paper. An interested reader can see [13], [23], [27] for a more detailed exposition.

Theorem 1. *Let $Z(n)$ be an irreducible Q-dimensional Markov chain (or, general space Markov process), whose elements $z_l(n), l = 1, 2, \dots, Q$ are nonnegative, i.e., $Z(n) \in \mathbb{N}_+^Q$ (or, $Z(n) \in \mathbb{R}_+^Q$). If there exists a nonnegative valued function $\{\mathcal{L} : \mathbb{R}_+^Q \rightarrow \mathbb{R}_+\}$ such that both:*

$$E[\mathcal{L}(Z(n+1)) - \mathcal{L}(Z(n))|Z(n)] < \infty \tag{6}$$

and

$$\limsup_{\|Z(n)\|_* \rightarrow \infty} \frac{E[\mathcal{L}(Z(n+1)) - \mathcal{L}(Z(n))|Z(n)]}{\|Z(n)\|_*} < -\epsilon \tag{7}$$

are satisfied for some $\epsilon > 0$, then $Z(n)$ is positive recurrent, and

$$\limsup_{n \rightarrow \infty} E[\|Z(n)\|_*] < \infty.$$

Inequality (6) requires that the increments of the Lyapunov function $\mathcal{L}(Z)$ are finite on average. The second inequality (7) requires that, for large values of $\|Z\|_*$, the average increment in the Lyapunov function from time n to time $n+1$ is negative. An intuitive explanation of this result can be given by interpreting the Lyapunov function $\mathcal{L}(Z(n))$ as the system energy associated to state $Z(n)$. In this case, (7) forces the system to be dissipative on average for large $Z(n)$; as a consequence, a negative feedback exists, which is able to pull the system toward the empty state, thus making it ergodic. For these reasons, inequality (7) is often referred as the Lyapunov function drift condition.

In our case, $Z(n)$ represents the number of packets in the network of queue $X(n)$ or the workload $Y(n)$, whose evolution is given by (2) or (4). For polynomial Lyapunov functions (i.e., functions $\mathcal{L}(Z)$ polynomial with respect to Z elements), it is immediate to verify that (6) can be always met when all the polynomial moments of $A(n)$ are finite.

For these reasons, since in the remainder of this paper we restrict our investigation to polynomial Lyapunov functions, the satisfaction of (7) for some (polynomial) Lyapunov function entails system stability.

2.3 Previous Work

The problem of the definition of the stability region in complex systems of interacting queues under dynamic scheduling policies, has attracted significant attention in the last decade from the research community since the pioneering work [33].

In [33], applying the Lyapunov function methodology, it has been shown that a system of interacting queues whose buffer-size is infinite achieve 100 percent throughput, if atomic max-scalar scheduling policy $\mathcal{P}_{MS}$ is applied at each node of the network. According to $\mathcal{P}_{MS}$, at each time slot n the departure vector is selected as follows:

$$D(n) = S(n) = \arg\max_{Z \in \mathcal{D}} Z(\mathbf{I}-\mathbf{R})X(n)^T. \tag{8}$$

The result in [33] has been generalized and adapted to different application contexts in the last years. As matter of example we just briefly recall some of the related works.

In the switching context, several studies have been aimed at the definition of the stability region in Input-Queued (IQ) switching architectures built around a bufferless crossbar: papers [1], [17], [22], [32], [36] have proposed different extensions of $\mathcal{P}_{MS}$, which have been shown to be 1-efficient; stability properties for simpler scheduling policies have been also studied in [8], [16]; in [2], [3], [17], finally, the problem of the definition of the stability region in networks of IQ switches has been considered. In the context of the satellite and wireless networks, generalizations of $\mathcal{P}_{MS}$ have been recently proposed and shown to be 1-efficient in [26], [27], [34]. Finally, the recent paper [7] generalizes the result in [33] under more general exogenous arrival processes applying a different analytical technique called fluid models.

All the previous works, however, have considered system of infinite buffer size queues. As noted before, in contrast to the previous work, this paper studies the stability region (throughput region) of networks with finite buffers.

3 Performance of Network of Finite Queues

Here, we present our main results. In Section 3.1 we show that 100 percent throughput can be obtained in any network of finite, flow-controlled interacting queues, for $l_j \geq 1$. To this end, we define the optimal dynamic scheduling policy $\mathcal{P}_1$. Since policy $\mathcal{P}_1$ 1) is not atomic, i.e., servers provide fractional services to packets stored at head of the virtual queues, and 2) requires the servers to coordinate their decisions at each time slot, then its implementability results problematic in several application contexts. In Section 3.2, we propose the atomic dynamic scheduling policy $\mathcal{P}_2$ whose complexity is similar to $\mathcal{P}_{MS}$ defined for infinite queue networks. $\mathcal{P}_2$, similarly to $\mathcal{P}_{MS}$, requires a continuous exchange of state information among network servers, but it can allow servers to take local decisions in an uncoordinated fashion, when considering simple network configurations, thus resulting significantly less complex than $\mathcal{P}_1$. We show that $\mathcal{P}_2$ is ρ-efficient when enough buffer inside the network is provided, thus estimating the trade-off between network buffers and achievable throughput. Finally, in Section 3.3 we analyze the impact of imperfect, or delayed state information on the performance of policy $\mathcal{P}_2$.

3.1 Optimal Policy

3.1.1 Policy Definition

Consider the following policy, called $\mathcal{P}_1$, in vectorial format:

$$S(n) = \arg\max_{Z \in \widehat{\mathcal{D}}} Z(\mathbf{I}-\mathbf{R})\mathbf{M}(n)(2Y(n) - Z(\mathbf{I}-\mathbf{R}))^T, \tag{9}$$

where $\mathbf{M}(n)$ is a $Q \times Q$ matrix, nonnull only on its diagonal where, for $q = 1, \dots, Q$:

$$\mathbf{M}_{qq}(n) = \begin{cases} 1 & \text{if } q \in \Phi_I \\ \frac{y_{f(q)}(n)}{l_{j(q)}-1} & \text{if } q \in \Phi_M, \end{cases}$$

where we remind that given queue q traversed by flow j (i.e., $q = q(j,h)$ for some $h > 1$), $y_{f(q)}(n)$ represents the workload at the ingress queue (i.e., $f(q(j,h)) = q(j,1)$).

We now express the policy in scalar format (for the sake of easier notation, we omit (n) when not necessary). Observe that:

$$[S(\mathbf{I}-\mathbf{R})]_q = \begin{cases} s_q & \text{if } q \in \Phi_I \\ s_q - s_{u(q)} & \text{if } q \in \Phi_M, \end{cases}$$

and multiplying by $\mathbf{M}$:

$$[S(\mathbf{I}-\mathbf{R})\mathbf{M}]_q = \begin{cases} s_q & \text{if } q \in \Phi_I \\ (s_q - s_{u(q)})\frac{y_{f(q)}}{l_{j(q)}-1} & \text{if } q \in \Phi_M. \end{cases}$$

Hence, conventionally defining $y_{p(q)} = 0$ for queue $q \in \Phi_I$, the first adder in (9) becomes:

$$\begin{aligned} f_1(S) &= S(\mathbf{I}-\mathbf{R})\mathbf{M}Y^T \\ &= \sum_{q\in\Phi_I} s_q y_q + \sum_{q\in\Phi_M} (s_q - s_{u(q)}) \frac{y_{f(q)}}{l_{j(q)}-1} y_q = \\ &= \sum_{q\in\Phi_I} s_q y_q \left(1 - \frac{y_{p(q)}}{l_{j(q)}-1}\right) + \sum_{q\in\Phi_M} s_q y_{f(q)} \left(\frac{y_q - y_{p(q)}}{l_{j(q)}-1}\right) = \\ &= \sum_{j=1}^{J} \frac{y_{j,1}}{l_j - 1} \left[s_{j,1}(l_j - 1 - y_{j,2}) + \sum_{h=2}^{h_j} s_{j,h}(y_{j,h} - y_{j,h+1}) \right], \end{aligned} \tag{10}$$

whereas the second adder in (9):

$$\begin{aligned} f_2(S) &= S(\mathbf{I}-\mathbf{R})\mathbf{M}[S(\mathbf{I}-\mathbf{R})]^T \\ &= \sum_{q\in\Phi_I} s_q^2 + \sum_{q\in\Phi_M} (s_q - s_{u(q)})^2 \frac{y_{f(q)}}{l_{j(q)}-1} = \\ &= \sum_{j=1}^{J} s_{j,1}^2 + \sum_{j=1}^{J} \frac{y_{j,1}}{l_j-1} \sum_{h=2}^{h_j} \left(s_{j,h}^2 + s_{j,h-1}^2 - 2 s_{j,h} s_{j,h-1} \right) = \\ &= \sum_{j=1}^{J} s_{j,1}^2 + \sum_{j=1}^{J} \frac{y_{j,1}}{l_j-1} \\ &\left(s_{j,1}^2 + s_{j,h_j}^2 + 2\sum_{h=2}^{h_j-1} s_{j,h}^2 - 2\sum_{h=1}^{h_j-1} s_{j,h} s_{j,h+1} \right). \end{aligned} \tag{11}$$

By combining (10) and (11), having defined $f(Z) = 2f_1(Z) - f_2(Z)$, policy $\mathcal{P}_1$ becomes:

$$S = \arg\max_{Z\in\widehat{\mathcal{D}}} f(Z). \tag{12}$$

Policy $\mathcal{P}_1$, by construction, satisfies both service constraints. Indeed, the fact that service is never provided to empty virtual queues can be verified by observing that $\mathcal{P}_1$ can be equivalently defined as:

$$S(n) = \arg\min_{Z\in\widehat{\mathcal{D}}} \left\{ [Y(n) - Z(n)(\mathbf{I}-\mathbf{R})]\mathbf{M}(n)[Y(n) - Z(n)(\mathbf{I}-\mathbf{R})]^T \right\}.$$

Simple computations allow, indeed, to show that the minimum of this quadratic form cannot be achieved in correspondence of a point in which $[Y(n) - Z(n)(\mathbf{I}-\mathbf{R})]$ has negative components. In addition, the fact that buffer overflow can never occur (i.e., for all n and q, $y_{q(n)} \leq l_{j(q)}$) directly derives from the following two properties of policy $\mathcal{P}_1$ which can be verified by direct inspection: 1) $d_q = 0$ for every queue $q \in \Phi_I$ such that $y_{p(q)} \geq l_{j(q)} - 1$ and 2) $y_{p(q)}(n+1) \leq y_{(q)}(n)$ for each pair of queues $q \in \Phi_M$, $p(q) \in \Phi_M$.

3.1.2 Policy Performance

Now, we state our main theorem, whose proof is reported in Appendix A.

Theorem 2. *Under admissible Bernoulli traffic, policy $\mathcal{P}_1$ achieves 100 percent throughput when the buffer-size l_j of any internal queue q traversed by flow j satisfies the following relation:*

$$l_j \geq 2 \qquad \text{for } j = 1, \ldots, J.$$

3.1.3 Implementation Issue

Since $\mathcal{P}_1$ is not atomic, it selects the best service vector S in the set $\widehat{\mathcal{D}}$ and this does not guarantee that S is an integer departure vector: $s_q \in [0,1]$. As a consequence, the direct implementation of policy $\mathcal{P}_1$ requires servers to provide fractional services to packets stored at head of the virtual queues according to a weighted processor sharing policy.

Moreover, according to policy $\mathcal{P}_1$, packets are transferred through queues in a "cut-through" fashion, since servers may start the transmission of noncompletely received packets. We notice that nonatomic scheduling policies exploiting "cut-through" switching have been proposed and implemented in the contexts of wormhole networks [9], [12], [28].

At last, $\mathcal{P}_1$ must be implemented in a centralized fashion by a scheduler which has the complete view of the queues state of the network. The high implementation complexity of this policy has motivated our investigation on the performance of the following policy.

3.2 Low Complexity Policy

3.2.1 Policy Definition

Consider the following policy, called $\mathcal{P}_2$:

$$S = \arg\max_{Z\in\widehat{\mathcal{D}}} Z(\mathbf{I}-\mathbf{R})\mathbf{M}Y^T,$$

where $\mathbf{M}(n)$ is a $Q \times Q$ matrix, nonnull only on its diagonal where, for $q = 1, \ldots, Q$:

$$\mathbf{M}_{qq}(n) = \begin{cases} 1 & \text{if } q \in \Phi_I \\ \frac{y_{f(q)}(n)}{l_{j(q)}} & \text{if } q \in \Phi_M. \end{cases}$$

In other words, policy $\mathcal{P}_2$ maximizes the scalar product of the service vector Z and the weight vector $W = (\mathbf{I}-\mathbf{R})\mathbf{M}Y^T$. Due to the linearity of the scalar product, $\mathcal{P}_2$ guarantees the vector S to be an vertex of $\widehat{\mathcal{D}}$, i.e., $S(n) \in \mathcal{D}$; by assumption, all vertices of $\widehat{\mathcal{D}}$ are integer valued. Hence, $\mathcal{P}_2$ is an atomic policy, $D(n) = S(n)$ and $X(n) = Y(n)$. Formally, we can say that $\mathcal{P}_2$ can be also expressed as:

$$D = \arg\max_{Z\in\mathcal{D}} Z(\mathbf{I}-\mathbf{R})\mathbf{M}(n)X^T. \tag{13}$$

Following the same reasoning to obtain (10), a generic queue q is associated with the following weight w_q:

$$w_q = \begin{cases} x_q\left(1 - \frac{x_{p(q)}}{l_{j(q)}}\right) & \text{if } q \in \Phi_I \\ x_{f(q)}\left(\frac{x_q - x_{p(q)}}{l_{j(q)}}\right) & \text{if } q \in \Phi_M. \end{cases} \tag{14}$$

Then, policy $\mathcal{P}_2$ can be rewritten as:

$$D = \arg\max_{Z \in \mathcal{D}} \sum_{q=1}^{Q} z_q w_q. \tag{15}$$

P_2 chooses an optimal solution of the above optimization problem, according to which $d_q = 0$ in correspondence of null weights $w_q = 0$, i.e., either when $x_q = 0$ or $x_{p(q)} = l_{j(q)}$. As a consequence, $\mathcal{P}_2$ satisfies the service constraints.

3.2.2 Policy Performance

We claim the main result about $\mathcal{P}_2$, whose proof is reported in Appendix B.

Theorem 3. *Under admissible Bernoulli traffic, policy $\mathcal{P}_2$ is ρ-efficient when the buffer size l_j of any internal queue q traversed by flow j, with h_j hops, satisfies the following relation:*

$$l_j > \frac{(h_j - 1)\|\mathbb{I}\|}{2(1-\rho)} \qquad \text{for } j = 1, \dots, J$$

recalling that $\rho = \|\Lambda(\mathbf{I} - \mathbf{R})^{-1}\|$, and $\mathbb{I}$ is the vector with unitary elements.

A special case applies for networks in which path lengths do not exceed two hops, as in the case of packet switches built around buffered crossbars, as discussed in Section 4. We can claim the following:

Corollary 1. *Under admissible Bernoulli traffic, a network with $h_j \leq 2$ for all j, implementing policy $\mathcal{P}_2$, is stable when $\rho < 0.5$ for any $l_j \geq 1$, being ρ the maximum offered load for a single queue in the network.*

The proof is reported in Appendix C. Hence, a network of queues implementing $\mathcal{P}_2$ is 0.5-efficient, for any choice of l_j, under the condition that no packet routes are longer than two hops.

3.2.3 Implementation Issue

Policy $\mathcal{P}_2$ is an atomic policy equivalent to $\mathcal{P}_{MS}$ of (8), but with different weights assigned to the internal queues. Indeed, $\mathcal{P}_2$ and $\mathcal{P}_{MS}$ solve the same optimization problem since they both share the same linear structure of the cost function and the same space $\mathcal{D}$ of feasible departure vectors.

Both policies require a continuous exchange of information between neighbor servers, but in addition $\mathcal{P}_2$ requires locally at each server the information about the length of the ingress queue of the corresponding flows. Note that this length should be propagated downstream from the ingress queue to all the internal queues, along the flow path: this fact can be exploited to ease the implementation.

In general, given the state of all the queues, $\mathcal{P}_2$ is executed by a central scheduler, as also observed by [33]. However, in particular (but also interesting) cases, the policy can be computed in a distributed fashion, locally on each set of queues and servers which are coupled by the blocking constraints. This fact is indeed exploited in the following section to devise a computationally efficient scheduling policy for packet switches.

3.3 Low Complexity Policy with Imperfect State Information

One of the aspects which make policy $\mathcal{P}_2$ hardly implementable in several contexts, consists in the fact that each node must have an exact information on the state of remote nodes. In this section, we study the effect on the policy performance of an imperfect state information.

Consider now policy $\mathcal{P}_\delta$ in which the state of the queues is known with a bounded error δ, i.e., the vector of queue length $X^*(n)$ used by the policy at time n differ by the actual queue length vector $X(n)$ by at most δ:

$$\left| x_q^*(n) - x_q(n) \right| < \delta \qquad \text{for all } q.$$

Hence, similarly to (13) and (15), policy $\mathcal{P}_\delta$ can be written as:

$$D^*(n) = \arg\max_{Z \in \mathcal{D}} Z(\mathbf{I} - \mathbf{R})\mathbf{M}^*(n)X^{*T}(n), \tag{16}$$

where $\mathbf{M}^*(n)$ is a $Q \times Q$ matrix, nonnull only on its diagonal where, for $q = 1, \dots, Q$:

$$\mathbf{M}_{qq}^*(n) = \begin{cases} 1 & \text{if } q \in \Phi_I \\ \frac{x_{f(q)}^*(n)}{l_{j(q)} - \delta} & \text{if } q \in \Phi_M, \end{cases}$$

or, equivalently,

$$D^*(n) = \arg\max_{Z \in \mathcal{D}} \sum_{q=1}^{Q} z_q w_q^*(n), \tag{17}$$

where w_q^* is the weight associated to queue q which is affected by the error on the queues state. We claim the following extension of Theorem 2:

Theorem 4. *Under admissible Bernoulli traffic, policy $\mathcal{P}_\delta$ is ρ-efficient when the buffer-size l_j of any internal queue q traversed by flow j, with h_j hops, satisfies the following relation:*

$$l_j > \frac{h_j(1 + 4\delta)\|\mathbb{I}\|}{2(1-\rho)} + \delta \qquad \text{for } j = 1, \dots, J.$$

The proof is reported in Appendix D.

The previous theorem can be generalized considering policies $\mathcal{P}_{\delta_M}$ in which the queues in Φ_M are known with a bounded error δ, i.e.,

$$\left| x_q^*(n) - x_q(n) \right| < \delta \qquad \text{for all } q \in \Phi_M,$$

while the error for queues in Φ_I is only bounded on average, i.e.,

$$\left| E\left[x_q^*(n) - x_q(n) \right] \right| \leq C \qquad \text{for all } q \in \Phi_I \text{ and some } C < \infty.$$

Also, in this case, we can derive:

Theorem 5. *Under admissible Bernoulli traffic, policy $\mathcal{P}_{\delta_M}$ is ρ-efficient when the buffer-size l_j of any internal queue q*

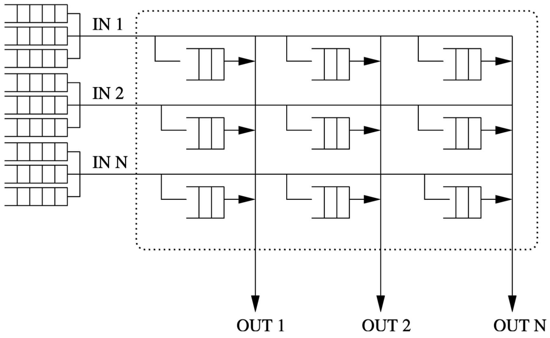

Fig. 2. The $N \times N$ CICQ architecture with VOQ and buffered crosspoints.

traversed by flow j, with h_j hops, satisfies the following relation:

$$l_j > \frac{h_j(1+4\delta)\|\mathbb{I}\|}{2(1-\rho)} + \delta \qquad \text{for } j = 1, \ldots, J.$$

The proof is reported in Appendix E.

3.4 Low Complexity Policy with Delayed State Information

We consider now policy $\mathcal{P}_\tau$ adopting $X(n-\tau)$, i.e., a delayed version of $X(n)$, to schedule the services. Since during the period of τ time slots the queue length of every queues in Φ_M can change by at most τ, while for queues in Φ_I the error is bounded on average, policy $\mathcal{P}_\tau$ falls in the class of policies $\mathcal{P}_{\delta_M}$ with $\delta = \tau$. Then, as immediate corollary of Theorem 5, $\mathcal{P}_\tau$ is proved to be stable if, for all $j = 1, \ldots, J$:

$$l_j > \frac{h_j(1+4\delta)\|\mathbb{I}\|}{2(1-\rho)} + \delta \qquad \text{for } j = 1, \ldots, J.$$

4 Application to Packet Switches Based on Buffered Crossbars

Recently, switches built around crosspoint buffered crossbars have been shown to be very promising solutions for the design of fast and scalable switching architectures. A basic model for a switch with internal buffered crossbar is depicted in Fig. 2. To avoid the negative effects of the head-of-the-line blocking phenomenon, inputs cards adopts Virtual Output Queue (VOQ) scheme, according to which packets are stored at inputs in per-destination virtual queues.

Each crosspoint of the crossbar is provided with an internal buffer of size L: Internal buffers are in one-to-one correspondence with input VOQs. We refer to this architecture as Combined Input and Crossbar Queued (CICQ) switch. A flow control mechanism from each crosspoint to the corresponding VOQ avoids to overflow the internal buffer.

Assume time to be slotted, and packets to be of fixed size. With respect to pure input queued switches, the scheduling policies in CICQ switches can be simpler. The scheduling decision, indeed, can be taken in a local uncoordinated fashion by an arbiter at each input, selecting a nonfull internal buffer to which transferring a packet, and by an arbiter at each output, selecting an internal buffer from which transferring a packet. We refer in the following to this class of schedulers as "uncoordinated schedulers."

Uncoordinated schedulers can be efficiently distributed, parallelized, and pipelined making CICQ architectures so appealing. Note that, in uncoordinated schedulers, we admit that inputs and outputs can exchange some information about the state of the queues, but we assume the scheduling decision to be local. Furthermore, uncoordinated schedulers cannot be implemented in pure IQ switches, since coordination is required at inputs to avoid multiple transmissions toward the same output.

Here, we very briefly highlight the main results on CICQ switches running uncoordinated schedulers, referring to papers [38], [30], [11] for a more detailed description of the scheduling algorithms for CICQ switches, and a discussion of their properties.

There are two main families of input/output arbiters proposed and studied so far:

- Round-robin driven algorithms: The queue is selected according to a round robin (RR) mechanism [29], [30], [31], or to a weighted round robin (WRR) [5], or to a weighted fair queueing scheme (WFQ) [5], and
- Queue-state driven algorithms: A metric is associated to every queue; the queue that maximizes (minimizes) the correspondent metric is selected. As possible metrics, the queue length (LQF) [10], the waiting time of the HoL cell (OCF) [10], [25] and the queue length at internal buffer [24] were considered.

Note that the input arbiters can select only VOQs which are not inhibited by the flow control mechanism. When an internal speedup $S_P > 1$ is allowed, then up to S_P packets can be served by each arbiter during a single timeslot; hence, at each output port, a queue is necessary to compensate for the lower output link rate.

Only few theoretical results have been obtained on the performance of CICQ switches. For $S_P = 2$ a wide class of CICQ uncoordinated schedulers has been very recently proved to achieve 100 percent throughput [6], with the minimal buffer requirement $L = 1$. Moreover, $S_P = 2$ is sufficient to guarantee that CICQ architectures implementing uncoordinated schedulers can perfectly emulate output queued switches [6], [20].

For $S_P < 2$, instead, to the best of our knowledge, no general results on the maximum throughput achievable in CICQ architectures with uncoordinated schedulers have been obtained so far. Several papers have addressed the case $S_P = 1$, $L = 1$, either showing by simulation that high throughput can be obtained by CICQ architectures with uncoordinated schedulers [5], [24], [31], [38], or proving that 100 percent throughput can be achieved in CICQ, under some specific traffic scenario, such as the case of uniform traffic [10], [29]. Other papers have shown by simulation that quasioptimal performance can be obtained by CICQ with moderate $S_p < 2$ speedup [30].

Now, observe that a CICQ switch can be modeled as a network of flow-controlled interacting queues, with one server for each input (corresponding to the input arbiter) and with one server for each output (corresponding to the output arbiter). The flow control is from the internal buffers to the corresponding input arbiters. Hence, we can particularize to this context the general results obtained in the previous sections. We restrict our investigation to policy $\mathcal{P}_2$ which can be easily implemented in a CICQ as an uncoordinated scheduler.

4.1 Scheduling Algorithms for CICQ

Let x_{ij} be the length of VOQ from input i to output[4] j. Let b_{ij} be the length of the corresponding internal buffer; $0 \leq b_{ij} \leq L$, and when $b_{ij} = L$, the flow control mechanism inhibits the services from the corresponding VOQ: We assume that the flow control is immediate. Departure vector D comprises the services provided by the input arbiters and the output arbiters: d^I_{ij} represents the departure from the VOQ corresponding to x_{ij}, whereas d^O_{ij} represents the departure from the internal buffer corresponding to b_{ij}. The set $\mathcal{D}$ of all possible departing vectors is given by all D such that

$$\sum_{j=1}^{N} d^I_{ij} \leq 1 \quad \forall i \qquad \text{and} \qquad \sum_{i=1}^{N} d^O_{ij} \leq 1 \quad \forall j,$$

which describe the blocking constraints of (3) in the context of a CICQ switch.

We particularize the policy $\mathcal{P}_2$ by showing that it can be easily implemented in an uncoordinated fashion. Indeed, revisiting (15), $\mathcal{P}_2$ selects the departing vector according to:

$$\begin{aligned} D &= \arg\max_{D\in\mathcal{D}} \sum_{j=1}^{N^2} \frac{x_{j,1}}{L}\left(d_{j,1}(L - x_{j,2}) + d_{j,2}x_{j,2}\right) \\ &= \arg\max_{D\in\mathcal{D}} \sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}(d^I_{ij}(L - b_{ij}) + d^O_{ij}b_{ij}). \end{aligned} \tag{18}$$

As a consequence policy $\mathcal{P}_2$, operating in a CICQ switch (renamed Dual Maximum Weight First, DMWF) can be implemented according to the following simple algorithm. At each time slot:

1. each VOQ is associated with a weight $w^I_{ij} = x_{ij}$ $(L - b_{ij})$; a nonempty VOQ is marked as inhibited to packet transfer if the corresponding weight is null (i.e., the corresponding crosspoint buffer is full). Each internal buffer is associated with a weight $w^O_{ij} = x_{ij}b_{ij}$;
2. the arbiter of input i selects the non inhibited VOQ which maximizes w^I_{ij} over all $j = 1, \dots, N$;
3. the arbiter of output j selects the nonempty internal buffer which maximizes w^O_{ij} over all $i = 1, \dots, N$.

Thanks to Corollary 1, DMWF is ρ-efficient if $\rho < 0.5$ for any $L \geq 1$. Note that in the case $L = 1$, then DMWF degenerates into LQF-LQF scheduler.

If we now apply Theorem 3, in a CICQ switch $\|\mathbb{I}\| = N$ since N are the queues conflicting in the same input/output arbiter. Hence, in general, L should be set such that $L > N/(1-\rho)/2$. To summarize, we can claim the following:

Corollary 2. *Under admissible Bernoulli traffic, in a CICQ switch policy DMWF is ρ-efficient for $L \geq L_{\min}$ with*

$$L_{\min} = \begin{cases} 1 & \text{if} \quad \rho < 0.5 \\ \left\lceil \frac{N}{2(1-\rho)} \right\rceil & \text{if} \quad 0.5 \leq \rho < 1, \end{cases}$$

where ρ is the maximum offered load to an input and output port of the switch.

The result of Corollary 2 can be restated also as follows: The sustainable load is at least:

$$\rho = \begin{cases} 0.5 & \text{for } 1 \leq L \leq N \\ 1 - \frac{N}{2L} & \text{for } L > N, \end{cases}$$

or, equivalently:

Corollary 3. *Under admissible Bernoulli traffic, the minimum speedup to guarantee 100 percent throughput in a CICQ switch adopting DMWF policy, is*

$$S_P = \begin{cases} 2 & \text{for } 1 \leq L \leq N \\ \frac{2L}{2L-N} & \text{for } L > N. \end{cases}$$

This proves the existence of a trade-off between throughput (or speedup needed) and L under DMWF.

5 Finite Ingress Buffers

The results presented in this paper until now have assumed the infinite buffer availability at the ingress queues. However, there are practical situations when this is not feasible. In such a setup all buffers, including ingress queues, are finite. Next, we describe how we can use the above results to provide lower bounds on the achievable throughput.

The maximal throughput algorithms described above never overflow internal queues. However, the queues at ingress node can grow in an unbounded fashion. We compare the system with infinite ingress buffers, say $\mathcal{S}_1$, to the system with finite ingress buffers, say $\mathcal{S}_2$. Both systems use the same throughput maximal algorithm described above. In $\mathcal{S}_2$, when ingress queue overflows, packets are dropped in contrast to $\mathcal{S}_1$. By definition, the number of packets in $\mathcal{S}_1$ stochastically dominate[5] the number of packets in $\mathcal{S}_2$ for each flow. Hence, we can obtain an upper bound on the drop rate in $\mathcal{S}_2$ by calculating the probability that an arriving packet in $\mathcal{S}_1$ sees the ingress queue larger than the allowed buffer-size in $\mathcal{S}_2$.

Now, the proofs of stability of algorithm for $\mathcal{S}_1$ are based on polynomial Lyapunov function. They imply a bound on average queue-size at the ingress queues (see [18], [35] for details). This, in turn, implies a bound on the (time) stationary probability of queue-size being larger than ingress buffer-size. Since our arrival process is memoryless, the time stationary queue-size distribution and the queue-size distribution observed by arriving packets is identical. This, in turn, implies the upper bound on the loss-rate. Consequently, it gives a lower bound on achievable throughput region. We skip the details in the interest of space.

6 Conclusions

We studied the throughput property of network of queues with finite buffers. In particular, we obtain sufficient conditions on the required buffer-size of the internal queues so as to achieve maximal throughput. This was exhibited by producing algorithms that would provide maximal throughput under these conditions. The implementation considerations led us to consider simpler policies and, consequently, a natural trade-off between buffer-size and the achievable throughput.

[4] With abuse of notation, here, j stands either for a flow identifier or an output.

[5] To prove this, it is necessary that both systems use exactly the same schedule for identical arrival. In other words, system $\mathcal{S}_2$ obtains its schedules by simulating system $\mathcal{S}_1$ on side.

We applied our results in the context of $N \times N$ input queued switches based on buffered crossbars. In particular, we obtained a scheduling policy DMWF, in which each input and output arbiter makes decision independently. This policy naturally gives rise to a trade-off between speedup, throughput and buffer-size at the crosspoint.

Appendix A

Proof of Theorem 2

Proof. Consider the following Lyapunov[6] function: $\mathcal{L}(Y(n)) = Y(n)\mathbf{M}(n)Y(n)^T$, being $\Delta\mathcal{L}(n) = E[\mathcal{L}(Y(n+1)) - \mathcal{L}(Y(n))|Y(n)]$, to satisfy the stability criteria (7) it must be:

$$\lim_{\|Y(n)\|\to\infty} \frac{\Delta\mathcal{L}(n)}{\|Y(n)\|} < -\epsilon, \tag{19}$$

where $\Delta\mathcal{L}(n) = E[Y(n+1)\mathbf{M}(n+1)Y(n+1)^T - Y(n)\mathbf{M}(n)Y(n)^T|Y(n)]$. From now on, in the conditional expectation we omit $|Y(n)$ from the notation. Now, observe that $E[Y(n+1)\mathbf{M}(n+1)Y(n+1)^T]$ can be written as:

$$E\Big[Y(n+1)\mathbf{M}(n)Y(n+1)^T\Big] + \\ E\Big[Y(n+1)(\mathbf{M}(n+1) - \mathbf{M}(n))Y(n+1)^T\Big].$$

We show that the latter adder is an $o(\|Y\|)$ for $\|Y\| \to \infty$. Indeed,

$$\begin{aligned} &Y(n+1)(\mathbf{M}(n+1) - \mathbf{M}(n))Y(n+1)^T \\ &= \sum_{q\in\Phi_I\cup\Phi_M} y_q^2(n+1)\big(\mathbf{M}_{qq}(n+1) - \mathbf{M}_{qq}(n)\big) \\ &= \sum_{q\in\Phi_M} y_q^2(n+1)\big(\mathbf{M}_{qq}(n+1) - \mathbf{M}_{qq}(n)\big) \\ &\leq \sum_{q\in\Phi_M} l_{j(q)}^2 |\mathbf{M}_{qq}(n+1) - \mathbf{M}_{qq}(n)|, \end{aligned}$$

where the last equality holds since $\mathbf{M}_{qq}(n+1) - \mathbf{M}_{qq}(n) = 0$ for $q \in \Phi_I$. Now, since by hypothesis $E[A(n)A(n)^T]$ is upper bounded by some constant, then also

$$\begin{aligned} E\big[|\mathbf{M}_{qq}(n+1) - \mathbf{M}_{qq}(n)|\big] &= E\left[\frac{|y_{f(q)}(n+1) - y_{f(q)}(n)|}{l_{j(q)} - 1}\right] \\ &= E\left[\frac{|a_{f(q)}(n) - s_{f(q)}(n)|}{l_{j(q)} - 1}\right] \end{aligned}$$

results to be bounded and, consequently, the whole term $E[Y(n+1)(\mathbf{M}(n+1) - \mathbf{M}(n))Y(n+1)^T]$ is upper bounded by some constant, i.e., it is $o(\|Y\|)$ for $\|Y\| \to \infty$.

Hence, we can approximate $E[Y(n+1)\mathbf{M}(n+1)Y(n+1)^T]$ with $E[Y(n+1)\mathbf{M}(n)Y(n+1)^T]$ and obtain:

$$\begin{aligned} \Delta\mathcal{L}(n) &= E\Big[Y(n+1)\mathbf{M}(n)Y(n+1)^T\Big] - Y(n)\mathbf{M}(n)Y(n)^T \\ &= 2[\Lambda - S(n)(\mathbf{I} - \mathbf{R})]\mathbf{M}(n)Y(n)^T + E[A(n) - S(n)(\mathbf{I} - \mathbf{R})] \\ &\quad \mathbf{M}(n)[A(n) - S(n)(\mathbf{I} - \mathbf{R})]^T + o(\|Y\|). \end{aligned} \tag{20}$$

From now on, for the sake of readability, we will omit the variable n from our notations, when not necessary. Let us consider the second term in (20):

$$\begin{aligned} &E\Big[[A - S(\mathbf{I} - \mathbf{R})]\mathbf{M}[A - S(\mathbf{I} - \mathbf{R})]^T\Big] \\ &= E\Big[A\mathbf{M}A^T - 2A\mathbf{M}[S(\mathbf{I} - \mathbf{R})]^T + S(\mathbf{I} - \mathbf{R})\mathbf{M}[S(\mathbf{I} - \mathbf{R})]^T\Big]. \end{aligned} \tag{21}$$

Since $A\mathbf{M} = A$, the first and second terms of (21) are negligible with respect to $\|Y\| \to \infty$. Indeed:

$$E[A\mathbf{M}A^T] = E[AA^T]$$

and

$$E\Big[A\mathbf{M}[S(\mathbf{I} - \mathbf{R})]^T\Big] = \Lambda[S(\mathbf{I} - \mathbf{R})]^T = \sum_{q\in\Phi_I} s_q\lambda_q.$$

As a consequence, the rightmost member of (21) can be approximated with $f_2(S)$. Now, (20) becomes:

$$\Delta\mathcal{L} \approx 2\Lambda\mathbf{M}Y^T - 2f_1(S) + f_2(S). \tag{22}$$

If we now define $\Gamma = \Lambda(\mathbf{I} - \mathbf{R})^{-1}$, then $\Lambda\mathbf{M}Y^T$ can be written as $f_1(\Gamma)$. Since Λ is admissible, it results: $\|\Gamma\| = \rho < 1$; we can now define $\hat{\Gamma}$ such that $\Gamma = \rho\hat{\Gamma}$: $\|\hat{\Gamma}\| = 1$. Since f_1 is a linear function, then $f_1(\Gamma) = f_1(\rho\hat{\Gamma}) = \rho f_1(\hat{\Gamma})$. Recalling $f(S) = 2f_1(S) - f_2(S)$:

$$\Delta L \approx 2f_1(\Gamma) - f(S) = 2\rho f_1(\hat{\Gamma}) - f(S).$$

In addition, note that $\rho f_2(\hat{\Gamma}) = \rho^{-1}f_2(\Gamma) = \rho^{-1}\Lambda\mathbf{M}\Lambda^T$ is $o(\|Y\|)$, thus:

$$\Delta L \approx 2\rho f_1(\hat{\Gamma}) - \rho f_2(\hat{\Gamma}) - f(S) = \rho f(\hat{\Gamma}) - f(S).$$

According to the definition of policy $\mathcal{P}_1$, $f(S) \geq f(\hat{\Gamma})$, thus:

$$\Delta\mathcal{L} \leq \rho f(\hat{\Gamma}) - f(\hat{\Gamma}) = -(1-\rho)f(\hat{\Gamma}). \tag{23}$$

By neglecting terms which are $o(\|Y\|)$, we obtain:

$$f(\hat{\Gamma}) = 2f_1(\hat{\Gamma}) = \frac{2}{\rho}f_1(\Gamma) = \frac{2}{\rho}\sum_{q\in\Phi_I}\lambda_q y_q \geq \frac{2}{\rho}\lambda_{\min}\sum_{j=1}^{J} y_{j,1},$$

where $\lambda_{\min} = \min_{q\in\Phi_I}\{\lambda_q : \lambda_q > 0\}$. Reminding that according to (5) it results: $\|Y\| \leq \sum_{j=1}^{J} y_{j,1} + \sum_{j=1}^{J} h_j l_j$, which can be rewritten as: $\|Y\| \leq \sum_{j=1}^{J} y_{j,1} + o(\|Y\|)$. Hence, $f(\hat{\Gamma}) \geq 2\lambda_{\min}\|Y\|/\rho$ for $\|Y\| \to \infty$. (23) becomes:

$$\Delta\mathcal{L} \leq -2\frac{1-\rho}{\rho}\lambda_{\min}\|Y\|,$$

and this implies that, for any $l_j \geq 2$,

$$\lim_{\|Y\|\to\infty}\frac{\Delta\mathcal{L}}{\|Y\|} < -2\frac{1-\rho}{\rho}\lambda_{\min}.$$

□

[6] Note that $\mathcal{L}(Y(n)) \geq 0$ and $\mathcal{L}(Y_0) = 0$ if Y_0 is the null vector.

APPENDIX B
PROOF OF THEOREM 3

Proof. Consider again the Lyapunov function: $\mathcal{L}(X) = X\mathbf{M}X^T$. Equation (20) still holds:

$$\begin{aligned}\Delta\mathcal{L} &= 2[\Lambda - D(\mathbf{I}-\mathbf{R})]\mathbf{M}X^T \\ &\quad + E\Big[[A - D(\mathbf{I}-\mathbf{R})]\mathbf{M}[A - D(\mathbf{I}-\mathbf{R})]^T\Big].\end{aligned} \tag{24}$$

If policy $\mathcal{P}_2$ is employed in the network, $D(n)$ is selected, according to (15), on the set of all possible service vectors Z such that $\|Z\| \le 1$. If we choose Z as: $Z = \Lambda(\mathbf{I}-\mathbf{R})^{-1} + (1-\rho)U$ with any U such that $\|U\| = 1$, then $\|Z\| \le 1$, since: $\|Z\| = \|\Lambda(\mathbf{I}-\mathbf{R})^{-1} + (1-\rho)U\| \le \|\Lambda(\mathbf{I}-\mathbf{R})^{-1}\| + \|(1-\rho)U\| = \rho + (1-\rho) = 1$. Now,

$$\begin{aligned}D(\mathbf{I}-\mathbf{R})\mathbf{M}X^T &\ge \Big[\Lambda(\mathbf{I}-\mathbf{R})^{-1} + (1-\rho)U\Big](\mathbf{I}-\mathbf{R})\mathbf{M}X^T \\ &= \Lambda\mathbf{M}X^T + (1-\rho)U(\mathbf{I}-\mathbf{R})\mathbf{M}X^T.\end{aligned} \tag{25}$$

Thanks to (25), we can bound the first term in (24):

$$[\Lambda - D(\mathbf{I}-\mathbf{R})]\mathbf{M}X^T \le \Lambda\mathbf{M}X^T - \Lambda\mathbf{M}X^T - (1-\rho)U(\mathbf{I}-\mathbf{R})\mathbf{M}X^T = -(1-\rho)UW^T. \tag{26}$$

The second term in (24) can be treated as the second term in (20), it results:

$$\Delta\mathcal{L} \le -2(1-\rho)\sum_{q=1}^{Q} u_q w_q + \sum_{q\in\Phi_M}\big(d_q - d_{u(q)}\big)^2 \frac{x_{f(q)}}{l_{j(q)}}. \tag{27}$$

Let $U = \mathbb{1}/\|\mathbb{1}\| \in \mathcal{D}$; it results:

$$\begin{aligned}\Delta\mathcal{L} &\le -2(1-\rho)\frac{1}{\|\mathbb{1}\|}\sum_{q=1}^{Q} w_q + \sum_{q\in\Phi_M}\big(d_q - d_{u(q)}\big)^2 \frac{x_{f(q)}}{l_{j(q)}} \\ &= -\frac{2(1-\rho)}{\|\mathbb{1}\|}\sum_{j=1}^{J} x_{j,1} + \sum_{j=1}^{J}\frac{x_{j,1}}{l_j}\sum_{h=2}^{h_j}(d_{j,h} - d_{j,h-1})^2 \\ &\le -\sum_{j=1}^{J} x_{j,1}\left[\frac{2(1-\rho)}{\|\mathbb{1}\|} - \frac{h_j - 1}{l_j}\right],\end{aligned}$$

where we exploited the fact that, for any j:

$$\sum_{h=1}^{h_j} w_{j,h} = \frac{x_{j,1}}{l_j}\left[(l_j - x_{j,2}) + \sum_{h=2}^{h_j-1}(x_h - x_{h+1}) + x_{h_j}\right] = x_{j,1}.$$

Furthermore, $\sum_{h=2}^{h_j}(d_{j,h} - d_{j,h-1})^2 \le h_j - 1$.

As a consequence, a sufficient condition to make the Lyapunov function drift negative is:

$$\frac{2(1-\rho)}{\|\mathbb{1}\|} - \frac{h_j - 1}{l_j} > 0,$$

which implies:

$$l_j > \frac{(h_j - 1)\|\mathbb{1}\|}{2(1-\rho)} \qquad \forall j.$$

□

APPENDIX C
PROOF OF COROLLARY 1

Proof. Equation (27), substituting D to U, can be written as follows, recalling (14):

$$\begin{aligned}\Delta\mathcal{L} &\le -2(1-\rho)\Bigg[\sum_{q\in\Phi_I} d_q x_q\left(1 - \frac{x_{p(q)}}{l_{j(q)}}\right) + \sum_{q\in\Phi_M} d_q x_{f(q)} \\ &\quad \left(\frac{x_q - x_{p(q)}}{l_{j(q)}}\right)\Bigg] + \sum_{q\in\Phi_M}\big(d_q - d_{u(q)}\big)^2\frac{x_{f(q)}}{l_{j(q)}} \\ &\le -2(1-\rho)\sum_{j=1}^{J}\frac{x_{j,1}}{l_j}\big[d_{j,1}(l_j - x_{j,2}) + d_{j,2}x_{j,2}\big] \\ &\quad + \sum_{j=1}^{J}\frac{x_{j,1}}{l_j}(d_{j,1} + d_{j,2} - 2d_{j,1}d_{j,2}) \\ &\le -2(1-\rho)\sum_{j=1}^{J}\frac{x_{j,1}}{l_j}\Bigg[d_{j,1}\left(l_j - x_{j,2} - \frac{1}{2(1-\rho)}\right) \\ &\quad + d_{j,2}\left(x_{j,2} - \frac{1}{2(1-\rho)}\right)\Bigg].\end{aligned}$$

Thanks to (15), being in this case:

$$D = \arg\max_{D\in\mathcal{D}}\sum_{j=1}^{J}\frac{x_{j,1}}{l_j}\big[d_{j,1}(l_j - x_{j,2}) + d_{j,2}x_{j,2}\big],$$

then a sufficient condition which ensures the Lyapunov function drift to be negative is:

$$\frac{1}{2(1-\rho)} < 1.$$

Indeed, $d_{j,1} = 1$ only when $l_j - x_{j,2} \ge 1$ and $d_{j,2} = 1$ only when $x_{j,2} \ge 1$. □

APPENDIX D
PROOF OF THEOREM 4

Proof. Through this proof, we assume $l_{j(q)} \ge 2\delta$ for every flow j, we emphasize that this assumption does not limit the validity of our result.

We consider the same Lyapunov function adopted in the proof of Theorem 2: $\mathcal{L}(X(n)) = X(n)\mathbf{M}(n)X(n)^T$. Likewise, (20) and (24):

$$\begin{aligned}\Delta\mathcal{L} &= 2[\Lambda - D^*(\mathbf{I}-\mathbf{R})]\mathbf{M}X^T + E\big[[A - D^*(\mathbf{I}-\mathbf{R})] \\ &\quad \mathbf{M}[A - D^*(\mathbf{I}-\mathbf{R})]^T\big].\end{aligned} \tag{28}$$

To evaluate the first term in (28), we estimate the difference between $D^*(\mathbf{I}-\mathbf{R})\mathbf{M}X^T$ with $D(\mathbf{I}-\mathbf{R})\mathbf{M}X^T$. Observe that we can write: $\mathbf{M}^* - \mathbf{M} = \mathbf{E}$, where $\mathbf{E}$ is a matrix, null outside the main diagonal, with:

$$\mathbf{E}_{qq} = \begin{cases} 0 & \text{if } q \in \Phi_I \\ (x_q^* - x_q)(l_{j(q)} - \delta) & \text{if } q \in \Phi_M \end{cases}$$

being $|\mathbf{E}_{qq}| < \delta/(l_{j(q)} - \delta)$ for every $q \in \Phi_M$. Let $X^* - X = \Delta X$ with $|\Delta X| \le \delta\mathbb{1}$; it results:

$$\begin{aligned} D^*(\mathbf{I}-\mathbf{R})\mathbf{M}X^T &= D^*(\mathbf{I}-\mathbf{R})(\mathbf{M}^*-\mathbf{E})(X^*-\Delta X)^T \\ &= D^*(\mathbf{I}-\mathbf{R})\mathbf{M}^*X^{*T} - D^*(\mathbf{I}-\mathbf{R})\mathbf{E}X^{*T} \\ &\quad - D^*(\mathbf{I}-\mathbf{R})\mathbf{M}^*\Delta X^T + D^*(\mathbf{I}-\mathbf{R})\mathbf{E}\Delta X^T, \end{aligned} \tag{29}$$

where the last term is $o(\|X\|)$. Considering the first term in (29), it results:

$$\begin{aligned} D^*(\mathbf{I}-\mathbf{R})\mathbf{M}^*X^{*T} &\geq D(\mathbf{I}-\mathbf{R})\mathbf{M}^*X^{*T} \\ &= D(\mathbf{I}-\mathbf{R})(\mathbf{M}+\mathbf{E})(X+\Delta X)^T = D(\mathbf{I}-\mathbf{R})\mathbf{M}X^T \\ &\quad + D(\mathbf{I}-\mathbf{R})\mathbf{E}X^T + D(\mathbf{I}-\mathbf{R})\mathbf{M}\Delta X^T + D(\mathbf{I}-\mathbf{R})\mathbf{E}\Delta X^T, \end{aligned} \tag{30}$$

where the first inequality is due to the definition (16) of $\mathcal{P}_\delta$; and the last term of (30) is again $o(\|X\|)$. Considering the second adder in (29), it results:

$$\begin{aligned} D^*(\mathbf{I}-\mathbf{R})\mathbf{E}X^{*T} &= D^*(\mathbf{I}-\mathbf{R})\mathbf{E}(X+\Delta X)^T \\ &= D^*(\mathbf{I}-\mathbf{R})\mathbf{E}X^T + D^*(\mathbf{I}-\mathbf{R})\mathbf{E}\Delta X^T, \end{aligned} \tag{31}$$

where the last term is negligible. Now, consider the third term in (29):

$$\begin{aligned} D^*(\mathbf{I}-\mathbf{R})\mathbf{M}^*\Delta X^T &= D^*(\mathbf{I}-\mathbf{R})(\mathbf{M}+\mathbf{E})\Delta X^T \\ &= D^*(\mathbf{I}-\mathbf{R})\mathbf{M}\Delta X^T + D^*(\mathbf{I}-\mathbf{R})\mathbf{E}\Delta X^T, \end{aligned} \tag{32}$$

where the last term is $o(\|X\|)$. By combining (30), (31), and (32), then (29) becomes:

$$\begin{aligned} D^*(\mathbf{I}-\mathbf{R})\mathbf{M}X^T &\geq D(\mathbf{I}-\mathbf{R})\mathbf{M}X^T + (D-D^*)(\mathbf{I}-\mathbf{R})\mathbf{E}X^T \\ &\quad + (D-D^*)(\mathbf{I}-\mathbf{R})\mathbf{M}\Delta X^T + o(\|X\|). \end{aligned} \tag{33}$$

Observe that the second term of (33) is $o(\|X\|)$, being the $q\mathrm{th}$ element of vector $\mathbf{E}X^T$: 1) equal to zero if $q \in \Phi_I$ and 2) strictly less than $\delta x_q/(l_{j(q)}-\delta) \leq 2\delta$ if $q \in \Phi_M$, under the assumption $l_{j(q)} \geq 2\delta$. Now, consider the third term in (33):

$$\begin{aligned} &\left|(D-D^*)(\mathbf{I}-\mathbf{R})\mathbf{M}\Delta X^T\right| \\ &= \left|\sum_{q\in\Phi_I}\left(d_q-d_q^*\right)\left(x_q^*-x_q\right) + \sum_{q\in\Phi_M}\left(d_q-d_{u(q)}-d_q^*+d_{u(q)}^*\right)\right. \\ &\quad \left.\frac{x_{f(q)}}{l_{j(q)}-\delta}\left(x_q^*-x_q\right)\right| < \sum_{q\in\Phi_I} d_q\delta + \sum_{q\in\Phi_M}\left(d_q+d_{u(q)}^*\right)\frac{x_{f(q)}}{l_{j(q)}-\delta}\delta \\ &\approx \sum_{q\in\Phi_M}\left(d_q+d_{u(q)}^*\right)\frac{x_{f(q)}}{l_{j(q)}-\delta}\delta \leq 2\delta\sum_{q\in\Phi_M}\frac{x_{f(q)}}{l_{j(q)}-\delta}. \end{aligned}$$

Hence, (33) becomes for $\|X\| \to \infty$:

$$D^*(\mathbf{I}-\mathbf{R})\mathbf{M}X^T > D(\mathbf{I}-\mathbf{R})\mathbf{M}X^T - 2\delta\sum_{q\in\Phi_M}\frac{x_{f(q)}}{l_{j(q)}-\delta}, \tag{34}$$

which can approximated as in (25). By also treating the second term of (28) as in the proof of Theorem 3 we get:

$$\begin{aligned} \Delta\mathcal{L} < &-2(1-\rho)\sum_{q=1}^{Q} u_q w_q + 4\delta\sum_{q\in\Phi_M}\frac{x_{f(q)}}{l_{j(q)}-\delta}\sum_{q\in\Phi_M}\left(d_q-d_{u(q)}\right)^2 \\ &\frac{x_{f(q)}}{l_{j(q)}-\delta}. \end{aligned} \tag{35}$$

From which, proceeding as in the proof of Theorem 3, we set $U = \mathbb{I}/\|\mathbb{I}\| \in \mathcal{D}$ and obtain:

$$\Delta\mathcal{L} < -\sum_{j=1}^{J} x_{j,1}\left[\frac{2(1-\rho)}{\|\mathbb{I}\|} - \frac{(h_j-1)(1+4\delta)}{l_j-\delta}\right].$$

As a consequence, a sufficient condition to make the Lyapunov function drift negative is:

$$\frac{2(1-\rho)}{\|\mathbb{I}\|} - \frac{(h_j-1)(1+4\delta)}{l_j-\delta} > 0,$$

which implies:

$$l_j > \frac{(h_j-1)(1+4\delta)\|\mathbb{I}\|}{2(1-\rho)} + \delta \qquad \forall j.$$

□

Appendix E
Proof of Theorem 5

Proof. Following the same scheme of the proof of Theorem 4, similarly to (33):

$$\begin{aligned} D^*(\mathbf{I}-\mathbf{R})\mathbf{M}X^T &\geq D(\mathbf{I}-\mathbf{R})\mathbf{M}X^T + (D-D^*)(\mathbf{I}-\mathbf{R})\mathbf{E}X^T \\ &\quad + (D-D^*)(\mathbf{I}-\mathbf{R})\mathbf{M}\Delta X^T + o(\|X\|), \end{aligned} \tag{36}$$

where the second term is again $o(\|X\|)$ and the third term:

$$\begin{aligned} &\left|(D-D^*)(\mathbf{I}-\mathbf{R})\mathbf{M}\Delta X^T\right| \\ &= \left|\sum_{q\in\Phi_I}\left(d_q-d_q^*\right)\left(x_q^*-x_q\right) + \sum_{q\in\Phi_M}\left(d_q-d_{u(q)}-d_q^*+d_{u(q)}^*\right)\right. \\ &\quad \left.\frac{x_{f(q)}}{l_{j(q)}-\delta}\left(x_q^*-x_q\right)\right| \leq \sum_{q\in\Phi_I} d_q C + \sum_{q\in\Phi_M}\left(d_q+d_{u(q)}^*\right)\frac{x_{f(q)}}{l_{j(q)}-\delta}\delta \\ &\approx \sum_{q\in\Phi_M}\left(d_q+d_{u(q)}^*\right)\frac{x_{f(q)}}{l_{j(q)}-\delta}\delta \leq 2\delta\sum_{q\in\Phi_M}\frac{x_{f(q)}}{l_{j(q)}-\delta}, \end{aligned}$$

since all the components relative to $q \in \Phi_I$ are $o(\|X\|)$. Then, proceeding as in the proof of Theorem 4, we get the result. □

Acknowledgments

The work of P. Giaccone and E. Leonardi was supported in part by the Italian Ministry for Education, University of Research within the project Famous.

REFERENCES

[1] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "Packet-Mode Scheduling in Input-Queued Cell-Based Switch," *IEEE/ACM Trans. Networking,* vol. 10, no. 5, pp. 666-678, Oct. 2002.

[2] M. Ajmone Marsan, P. Giaccone, E. Leonardi, and F. Neri, "On the Stability of Local Scheduling Policies in Networks of Packet Switches with Input Queues," *IEEE J. Selected Areas in Comm.,* vol. 21, no. 4, pp. 642-655, May 2003.

[3] M. Andrews and L. Zhang, "Achieving Stability in Networks of Input-Queued Switches," *Proc. IEEE INFOCOM '01,* pp. 1673-1679, Apr. 2001.

[4] N.J. Boden et al., "Myrinet: A Gigabit-per-Second Local Area Network," *IEEE Micro,* vol. 15, no. 1, pp. 29-36, Feb. 1995.

[5] N. Chrysos and M. Katevenis, "Weighted Fairness in Buffered Crossbar Scheduling," *Proc. IEEE High Performance Switching and Routing Conf. (HPSR '03),* pp. 17-22, June 2003.

[6] S.T. Chuang, S. Iyer, and N. McKeown, "Practical Algorithms for Performance Guarantees in Buffered Crossbars," *Proc. IEEE INFOCOM '05,* Mar. 2005.

[7] J.G. Dai and W. Lin, "Maximum Pressure Policies in Stochastic Processing Networks," *Operations Research,* vol. 53, pp. 197-218, 2005.

[8] J.G. Dai and B. Prabhakar, "The Throughput of Data Switches with and without Speedup," *Proc. IEEE INFOCOM '00,* pp. 556-564, Mar. 2000.

[9] J. Duato, "A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 6, no. 10, pp. 1055-1067, Oct. 1995.

[10] T. Javadi, R. Magill, and T. Hrabik, "A High Throughput Scheduling Algorithm for a Buffered Crossbar Switch Fabric," *Proc. IEEE Int'l Conf. Comm. (ICC '01),* pp. 1581-1591, June 2001.

[11] M. Katevenis, G. Passas, D. Simos, I. Papaefstathiou, and N. Chrysos, "Variable Packet Size Buffered Crossbar (CICQ) Switches," *Proc. IEEE Int'l Conf. Comm. (ICC '04),* June 2004.

[12] P. Kermani and L. Kleinrock, "Virtual Cut-Through: A New Computer Communication Switching Technique," *Computer Networks,* vol. 3, no. 3, pp. 267-286, Sept. 1979.

[13] P.R. Kumar and S.P. Meyn, "Stability of Queueing Networks and Scheduling Policies," *IEEE Trans. Automatic Control,* vol. 40, no. 2, pp. 251-260, Feb. 1995.

[14] H.T. Kung and K. Chang, "Receiver-Oriented Adaptive Buffer Allocation in Credit-Based Flow Control for ATM Networks," *Proc. IEEE INFOCOM '95,* pp. 239-252, Apr. 1995.

[15] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes.* Morgan Kaufmann, 1991.

[16] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan, "On the Stability of Input-Queued Switches with Speedup," *IEEE/ACM Trans. Networking,* vol. 9, no. 1, pp. 104-118, Feb. 2001.

[17] E. Leonardi, M. Mellia, M. Ajmone Marsan, and F. Neri, "On the Throughput Achievable by Isolated and Interconnected Input-Queueing Switches under Multiclass Traffic," *IEEE Trans. Information Theory,* vol. 45, no. 3, pp. 1167-1174, Mar. 2005.

[18] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan, "Bounds on Average Delays and Queue Length Averages and Variances in Input Queued and Combined Input/Output Queued Cell-Based Switches," *J. ACM,* vol. 50, no. 4, July 2003.

[19] X. Lin, P.K. McKinley, and L.M. Ni, "The Message Flow Model for Routing in Wormhole-Routed Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 6, no. 7, pp. 755-760, July 1995.

[20] R.B. Magill, C.E. Rohrs, and R.L. Stevenson, "Output Queued Switch Emulation by Fabrics with Limited Memory," *IEEE J. Selected Area in Comm.,* vol. 21, no. 4, May 2003.

[21] S. Mascolo, D. Cavendish, and M. Gerla, "ATM Rate Based Congestion Control Using a Smith Predictor: An EPRCA Implementation," *Proc. IEEE INFOCOM '96,* pp. 569-576, Mar. 1996.

[22] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% Throughput in an Input-Queued Switch," *IEEE Trans. Comm.,* vol. 47, no. 8, pp. 1260-1272, Aug. 1999.

[23] S.P. Meyn and R. Tweedie, *Markov Chain and Stochastic Stability.* Springer-Verlag, 1993.

[24] L. Mhamdi and M. Hamdi, "MCBF: A High-Performance Scheduling Algorithm for Buffered Crossbar Switches," *IEEE Comm. Letters,* vol. 7, no. 9, pp. 451-453, Sept. 2003.

[25] M. Nabeshima, "Performance Evaluation of a Combined Input and Crosspoint Queued Switch," *IEICE Trans. Comm.,* vol. E83-B, no. 3, Mar. 2000.

[26] M.J. Neely, E. Modiano, and C.E. Rohrs, "Dynamic Power Allocation and Routing for Time Varying Wireless Networks," *Proc. IEEE INFOCOM '03,* vol. 1, pp. 745-755, Mar. 2003.

[27] M.L. Neely, E. Modiano, and C.E. Rohrs, "Power Allocation and Routing in Multibeam Satellites with Time-Varying Channels," *IEEE/ACM Trans. Networking,* vol. 11, no. 1, pp. 138-152, Feb. 2003.

[28] L.M. Ni and P.K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," *Computer,* vol. 26, no. 2, pp. 62-76, Feb. 1993.

[29] R. Rojas Cessa, E. Oki, Z. Jing, and H.J. Chao, "CIXB-1: Combined Input-One-Cell-Crosspoint Buffered Switch," *Proc. IEEE High Performance Switching and Routing Conf. (HPSR '01),* pp. 324-329, 2001.

[30] R. Rojas Cessa, E. Oki, and H.J. Chao, "CIXOB-k: Combined Input-Crosspoint-Output Buffered Packet Switch," *Proc. IEEE GLOBECOM '01,* pp. 2654-2660, Nov. 2001.

[31] R. Rojas-Cessa and E. Oki, "Round Robin Selection with Adaptable Size Frame in a Combined Input-Crosspoint Buffered Switch," *IEEE Comm. Letters,* vol. 7, no. 11, Nov. 2003.

[32] K. Ross and N. Bambos, "Local Search Scheduling Algorithms for Maximal Throughput in Packet Switches," *Proc. IEEE INFOCOM '04,* Mar. 2004.

[33] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Trans. Automatic Control,* vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

[34] L. Tassiulas, "Scheduling and Performance Limits of Networks with Constantly Changing Topology," *IEEE Trans. Information Theory,* vol. 43, no. 3, pp. 1067-1073, May 1997.

[35] D. Shah, "Randomization and Heavy Traffic Theory: New Approaches for Design and Analysis of Switch Algorithms," PhD thesis, Dept. of Computer Science, Stanford Univ., Oct. 2004.

[36] L. Tassiulas, "Linear Complexity Algorithms for Maximum Throughput in Radio Networks and Input Queued Switches," *Proc. IEEE INFOCOM '98,* Apr. 1998.

[37] R. Telikepalli, T. Drwiega, and J. Yan, "Storage Area Network Extension Solutions and Their Performance Assessment," *IEEE Comm. Magazine,* vol. 42, no. 4, pp. 56-63, Apr. 2004.

[38] K. Yoshigoe and K.J. Christensen, "An Evolution to Crossbar Switches with Virtual Output Queueing and Buffered Crosspoint," *IEEE Network,* pp. 48-56, Sept. 2003.

Paolo Giaccone received the DrIng and PhD degrees in telecommunications engineering from Politecnico di Torino in 1998 and 2001, respectively. He is currently an assistant professor in the Dipartimento di Elettronica at Politecnico di Torino, Italy. During the summer 1998, he visited the High Speed Networks Research Group at Lucent Technology, Holmdel. From 2000-2001 and during the summer of 2002, he visited Professor Balaji Prabhakar in the Electrical Engineering Department at Stanford University. Between 2001 and 2002, he held a postdoctorate position at Politecnico di Torino, and during the summer of 2002 at Stanford University. His main area of interest is the design of scheduling policies for high performance routers. He is a member of the IEEE.

Emilio Leonardi received the DrIng degree in electronics engineering in 1991 and the PhD degree in telecommunications engineering in 1995, both from Politecnico di Torino, where he is currently an associate professor. In 1995, he visited the Computer Science Department at the University of California Los Angeles (UCLA). In the summer of 1999, he joined the High Speed Networks Research Group, at Bell Laboratories. In the summer of 2001, he visited the Electrical Engineering Department at Stanford University and, finally, in the summer of 2003, the IP Group at Sprint Labs, Burlingame, California. He has coauthored more than 150 papers published in international journals and presented in leading conferences. He participated to the program committee of several conferences including: IEEE Infocom, IEEE Globecom, and the IEEE International Communications Conference. He was guest editor of two special issues of the *IEEE Journal of Selected Areas of Communications*, focused on high speed switches and routers. He received the IEEE TCGN best paper award for a paper presented at IEEE Globecom '02, "High Speed Networks Symposium." His research interests are in the field of: performance evaluation of communication networks, queueing theory, packet switching, all-optical networks, and wireless networks. He is a member of the IEEE.

Devavrat Shah received the BTech degree in computer science from IIT-Bombay in 1999, with the honor of the President of India Gold Medal. He received the PhD degree from the Computer Science Department at Stanford University in 2004. He is currently an assistant professor with the Departments of Electrical Engineering and Computer Science and ESD at the Massachusetts Institute of Technology (MIT). His primary research interests are in the theory and practice of networks. Specifically, he is interested in algorithms for networks, stochastic optimization, and network information theory. He was coawarded the IEEE INFOCOM best paper award in 2004. He received 2005 George B. Dantzig best desseration award from INFORMS. He is a member of the IEEE.