

Packet-Mode Scheduling in Input-Queued Cell-Based Switches

Marco Ajmone Marsan, *Fellow, IEEE*, Andrea Bianco, *Member, IEEE*, Paolo Giaccone, *Student Member, IEEE*, Emilio Leonardi, *Member, IEEE*, and Fabio Neri, *Member, IEEE*

Abstract—We consider input-queued switch architectures dealing at their interfaces with variable-size packets, but internally operating on fixed-size cells. Packets are segmented into cells at input ports, transferred through the switching fabric, and reassembled at output ports. Cell transfers are controlled by a scheduling algorithm, which operates in packet-mode: all cells belonging to the same packet are transferred from inputs to outputs without interruption. We prove that input-queued switches using packet-mode scheduling can achieve 100% throughput, and we show by simulation that, depending on the packet size distribution, packet-mode scheduling may provide advantages over cell-mode scheduling.

Index Terms—Input queued switched, packet switching, scheduling algorithms, variable size packets.

I. INTRODUCTION

INPUT-QUEUED (IQ) switches with virtual output queuing (VOQ) buffering schemes [1]–[4] are today often adopted as the architecture for high-speed switches or routers, since all the components of an IQ switch (input interfaces, switching fabric, output interfaces) operate at a speed which is not larger than the data rate of input and output lines. Most of the implemented high-speed IQ switches internally operate on fixed-size data units (cells): the Lucent GRF [5], the Cisco GSR [6], the Tiny-Tera [7], the AN2/DEC [3], [8], the iPoint [9], and the MGR/BBN [10].

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 (SA) to avoid contention at output ports. Several SAs for IQ cell switches were proposed and compared in the literature (see, e.g., [2]–[4], [9], [11]–[16]). We call these cell-mode scheduling algorithms (CM-SAs). Good CM-SAs for IQ switches provide performance close to output-queued (OQ) architectures. We revisit in this paper some of these proposals, and develop novel variations to deal with variable-size packets; more precisely, we constrain the SA to deliver contiguously all the cells deriving from the segmentation of the same packet. In other words, variable-size packets are transformed into “trains of cells,” and cells belonging to the same train are scheduled so

that they remain contiguous in the delivery to the output card, i.e., they are not interleaved with cells of another train. This constraint permits savings in memory and complexity, since the reassembly of packets at the output becomes much easier. We call these packet-mode scheduling algorithms (PM-SAs).

In this paper, we use analysis and simulation to examine the performance of selected CM-SAs and PM-SAs, and to prove general results on PM-SAs. Although it is generally true that dealing with variable-size data units may bring about performance penalties [17], we prove in this paper that the maximum throughput achievable by IQ architectures using PM-SAs is 100%, as for IQ architectures adopting CM-SAs. Regarding packet delays, we show that the relative merits of switches using CM-SAs and PM-SAs depend on traffic characteristics, which accords with some results found in other contexts [18].

The paper is organized as follows. Section II describes the logical switch architecture and its major components, in the case of both fixed-size and variable-size packets. Section III briefly overviews the considered CM-SAs, and presents our modifications to turn them into PM-SAs. Section IV contains the proof that IQ switches operating in packet mode can achieve 100% throughput, provided that adequate SAs are adopted (see the statement of Theorem 2). Section IV also contains an approximate analytical model that helps to understand the behavior of packet delays. Section V presents simulation results for CM-SAs and PM-SAs. Finally, Section VI concludes the paper.

II. LOGICAL ARCHITECTURE

A. Input-Queued Cell Switches

We assume a switch with N inputs and outputs, all running at the same speed. The switch operates on fixed-size data units, which can be ATM cells, or have any other convenient format. Even though we refer to switches operating on fixed-size data units, our results are more generally applicable to any switch or router taking switching decisions at equally spaced time instants. The distance between two consecutive switching decisions is called the *time slot*, and the slot is the granularity in the allocation of switch resources.

Fig. 1 shows the logical structure for a *packet* switch based on an IQ *cell* switch. Consider, for now, only the IQ cell switch. Packets are stored at input interfaces and no buffers exist at output interfaces. Each input manages one first-in–first-out (FIFO) queue for each output, hence, a total of $N \times N = N^2$ queues are present. This queue separation permits to avoid performance degradations due to head-of-the-line (HOL) blocking

Manuscript received March 19, 2001; revised January 7, 2002; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor G. Rouskas. This work was supported in part by a research contract between CSELT and the Politecnico di Torino, and in part by the MURST MQOS Project. A preliminary version of this paper was presented at IEEE INFOCOM 2001, Anchorage, AK.

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

Digital Object Identifier 10.1109/TNET.2002.803939.

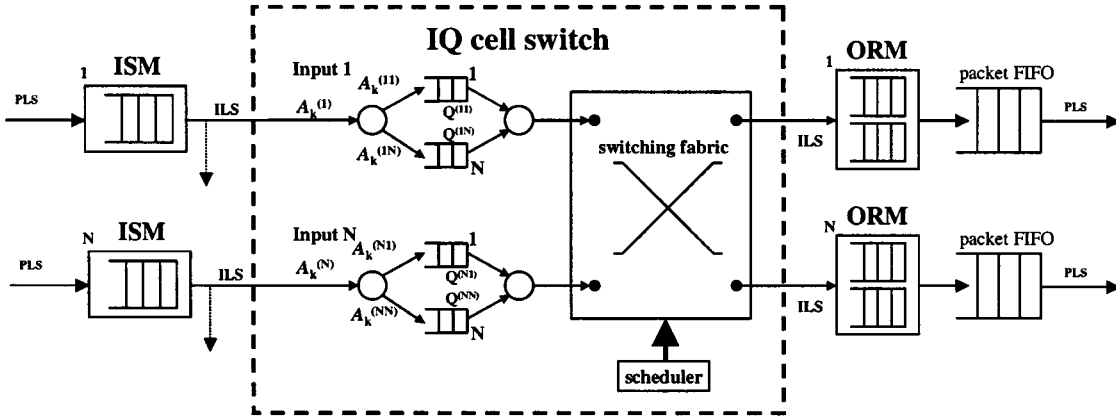


Fig. 1. Logical architecture for a packet switch based on an internal IQ cell switch.

[19], and is called virtual output queuing (VOQ), or advanced input queuing (AIQ), or destination queuing (DQ) [1]–[4].

Cells arrive at input i , $1 \leq i \leq N$, according to a discrete-time random process. At most one cell per slot arrives at each input, i.e., the data rate on input lines is no more than one cell per slot. We call $A_k^{(ij)}$, where k is a time slot index, the arrival process at input i for output j ; the average cell arrival rate is denoted by $\lambda^{(ij)}$. The aggregation of all arrival processes at input i is $A_k^{(i)} = \{A_k^{(ij)}, 1 \leq j \leq N\}$. The aggregation of all arrival processes is $A_k = \{A_k^{(i)}, 1 \leq i \leq N\}$. A_k is termed *admissible* if no input and no output is overloaded, i.e., if $\sum_{i=1}^N \lambda^{(ij)} < 1, \forall j$ and $\sum_{j=1}^N \lambda^{(ij)} < 1, \forall i$. Otherwise, A_k is called *inadmissible*. For later use, we define the cell arrival rate matrix $\mathbf{\Lambda} = [\lambda^{(ij)}]$, and the normalized cell arrival rate matrix $\mathbf{\Gamma} = [\gamma^{(ij)}]$, with $\gamma^{(ij)} = \lambda^{(ij)} / (\sum_{n=1}^N \sum_{m=1}^N \lambda^{(nm)})$. We define also the average *packet* arrival rate matrix, denoted by $\mathbf{\Lambda}_P = [\lambda_P^{(ij)}]$. Similarly to $\mathbf{\Gamma}$, we define the normalized packet arrival rate matrix $\mathbf{\Gamma}_P = [\gamma_P^{(ij)}]$.

When a cell with destination j arrives at input i according to process $A_k^{(ij)}$, it is stored in the FIFO queue $Q_k^{(ij)}$. The number of cells in $Q_k^{(ij)}$ at time k is denoted by $X_k^{(ij)}$. These FIFO queues have finite capacity: each queue can store at most Q_{\max} cells.

The switching fabric is nonblocking and memoryless; at most one cell can be removed from each input and at most one cell can be transferred to each output in every slot.

B. Input-Queued Packet Switches (or Routers)

Each router port has line interfaces where any data link and physical layer protocols can be used to receive and transmit IP datagrams. Within the IP layer at the input, routing functions are activated to associate an output port with the destination IP address. We neglect here issues related to the implementation of these functions, such as table lookup. Input IP datagrams are segmented into cells, that will be transferred to output ports by the switching fabric. Once cells are delivered to an output port, they are reassembled into the IP datagram, which is transmitted on the output line according to possibly different line formats.

The logical architecture for an IQ packet switch is shown in Fig. 1. At each input, an input segmentation module (ISM) segments the incoming packet into cells. PLS is the external

packet line speed. ISMs operate in store-and-forward mode, are equipped with enough memory to store a maximum-size packet, and start the segmentation process only after the complete reception of the packet. The cells resulting from the segmentation are transferred to the cell-switch input at a speed (called ILS) equal to the line speed PLS incremented to account for segmentation overheads. The capacity of each input queue at the cell switch is limited to Q_{\max} , hence, losses can occur. We assume that the entire packet is discarded if the input queue of the cell switch does not have enough free space to store all the cells deriving from the segmentation of the packet *when the first of these cells hits the queue*. This is a pessimistic assumption, but has the advantage of ease of implementation, and of avoiding the transmission of incomplete packet fractions through the switch.

The cell-based switching fabric transfers cells from input to output queues, according to an SA. These cells are delivered to the output reassembly module (ORM) at speed ILS. Here, packets (i.e., IP datagrams) are reassembled. In general, cells belonging to different packets can be interleaved at the same output, hence, more than one reassembly machine can be active in the same ORM. However, at most one cell reaches each ORM in a slot time, hence, at most one packet is completed at each ORM in a slot time.

Once a packet is complete, it is logically added to an output packet queue, called *packet FIFO*, from which packets are sequentially transmitted onto the output line. The *packet FIFO* functionality is typically implemented by imposing a sequential transfer from the suitable ORM to the output line of all the cells belonging to the same reassembled packet.

In the case of IQ packet switches using PM-SAs, it is possible to further simplify the switch structure and to improve its performance by enforcing additional constraints on the SA. Indeed, cells belonging to the same packet are contiguous in the input queue of the internal IQ cell switch. By using PM-SAs (see Section III for details) cells belonging to the same packet are kept contiguous also in the output queue, and the ORM modules are no longer necessary (or at most one per output is used). In this case, the logical architecture could be simplified by removing both the ORM module and the output packet FIFO from Fig. 1. With this modification, since each ORM operates in store-and-forward mode, and hence, introduces a delay equal to

the packet size, the delays through the switch are reduced at least by a packet duration. Although this architectural simplification may lead to interesting performance and may be implementable with limited effort, we will not consider this possibility in this paper.

III. CELL AND PACKET SCHEDULING ALGORITHMS IN IQ SWITCHES

A. Problem Definition

This section describes SAs, i.e., the set of rules used to decide which input port is granted access to the switching fabric. Scheduling in IQ cell switch architectures can be modeled as a matching problem in bipartite graphs. The switch state can be described as a bipartite graph $G = [V, E]$ in which the graph vertices in set V are partitioned in two subsets: subset V_I , whose elements $v_I^{(k)}$ correspond to input interfaces, and subset V_O , whose elements $v_O^{(k)}$ correspond to output interfaces. Edges indicate the needs for cell transfers from an input to an output (an edge from $v_I^{(n)}$ to $v_O^{(m)}$ indicates the need for cell transfers from input n to output m), and can be labeled with a metric (or weight) that will be denoted by $w^{(nm)}$. The adopted metric is a key part of the SA. It can be binary to simply indicate that at least one cell exists to be transferred. Otherwise, the adopted metric can refer to the number of cells to be transferred or to the time the oldest cell has waited.

A matching M is a selection of an admissible subset of edges. A subset of edges is admissible if no vertex has two connected edges; this means that it never happens that two cells are extracted from the same input, or that two cells are transferred to the same output. A matching has *maximum size* if the number of edges is maximized; a matching has *maximum weight* if the sum of the edge metrics is maximized. A matching is *maximal* if adding any edge of G makes it inadmissible.

The need for good suboptimal matching algorithms derives from the fact that the optimal solutions of the problem have very high complexity. The complexity is $O(N^3)$ for the maximum weight matching (MWM) algorithm [20, ch. 8],¹ that can be proved to yield the maximum achievable throughput using as metrics either the number of cells to be transferred, or the time the oldest cell has waited [21]; it is $O(N^{5/2})$ for the simpler and less efficient maximum size matching algorithm [20].

The $N \times N$ matrix whose elements are the edge metrics in graph $G = [V, E]$ is called the *weight matrix*, denoted by $\mathbf{W} = [w^{(ij)}]$. This weight matrix \mathbf{W} varies with time, according to the changes in the system parameters from which its elements are computed. When necessary, denoting by k the current slot, we shall write $\mathbf{W}_k = [w_k^{(ij)}]$. We assume $w^{(ij)} = 0$ for missing edges in G , i.e., when no cells from input i to output j are waiting in input queues.

B. Considered Cell-Mode Scheduling Algorithms (CM-SAs)

A number of CM-SAs for IQ switch architectures have appeared in the technical literature; see, e.g., [2]–[4], [9],

¹Note that the literature regarding IQ switches generally reports a complexity $O(N^3 \log N)$, which is incorrect (see [20, ch. 8] for details).

TABLE I
CHARACTERIZATION OF THE CONSIDERED IQ SCHEDULING ALGORITHMS

Algorithm	Metric	Matching method
MWM-QL	QL: Queue Length	Maximum Weight
MWM-CA	CA: Cell Age	Maximum Weight
MSM	QO: Queue Occupancy	Maximum Size
iLQF	QL: Queue Length	Iterative Search
iOCF	CA: Cell Age	Iterative Search
iSLIP	QO: Queue Occupancy	Iterative Search

[11]–[16]. In this paper, we consider six proposals, namely, three optimal algorithms MWM-QL (MWM with queue lengths as weights), MWM-CA (MWM with cell ages as weights), MSM (maximum size matching), and three heuristics, iLQF [22], iSLIP [23], and iOCF [22]. The choice of algorithms is somewhat arbitrary; the motivations that guided our choice are discussed at the end of this section. For the sake of brevity, we do not provide here a description of these algorithms. The reader is referred to the original works for detailed descriptions.

In [24], we proposed a general taxonomy for SAs. Any SA can be decomposed into two main components:

1) *Metrics computation*. Computation of the weight matrix $\mathbf{W}_k = [w_k^{(ij)}]$. Each one of the possible N^2 edges in the bipartite graph is associated with a metric depending on the state of the corresponding queue, that is $w_k^{(ij)}$ depends on $X_k^{(ij)}$, the state of the queue $Q^{(ij)}$ at slot k . This metric will act as a priority for the cell transfer.

MSM and iSLIP adopt the queue occupancy (QO) metric: $w_k^{(ij)} = u(X_k^{(ij)})$, where $u(\cdot)$ is the unit step function. MWM-QL and iLQF adopt as metric the queue length (QL): $w_k^{(ij)} = X_k^{(ij)}$. MWM-CA and iOCF adopt as a metric the time already spent in the queue by the cell at the queue head; we called it cell age (CA).

2) *Matching method*. Computation of the matching. MSM and iSLIP aim at a maximum *size* matchings, whereas all other algorithms try to maximize the matching *weight*.

MWM-QL, MWM-CA, and MSM are optimal algorithms, able to compute the maximum weight or size matching [20]. iSLIP, iLQF, and iOCF generate maximal matchings using iterative search [24].

Table I summarizes the characteristics of the considered IQ SAs. Both metric and matching method have a deep impact on the performance and on the complexity of the algorithm.

Coming back to the choice of the algorithms that will be considered in the rest of this paper, we tried to select two algorithms for each type of metric, the first being the optimal algorithm and the second being a well-known heuristic approximating the optimal algorithm. With this choice it is possible to understand the effect of the type of metric and of the matching method on the overall system performance. MWM-QL, MWM-CA and MSM were chosen as optimum algorithms for the three types of metric. iSLIP aims to approximate MSM and was chosen because it is very well known for its simplicity, for its implementation in the Tiny-Tera switch, and for being the precursor of a commercial implementation. iOCF and iLQF were chosen as well-known approximations of MWM-QL and MWM-CA. All the three heuristics run for $\log N$ iterations.

Thus, our choice of algorithms aims at considering well-known representatives of classes of proposals, at the cost of paying little attention to more recent schemes, and to algorithms which slightly improve on well-known and established proposals.

C. Packet-Mode Scheduling Algorithms (PM-SAs)

PM-SAs introduce the additional constraint of keeping the cells belonging to the same packet contiguous in the switching fabric and in output lines. To achieve this, the SA must enforce that, once the transfer through the switching fabric of the first cell of a packet has started toward the corresponding output port, no cells belonging to other packets can be transferred to that output, i.e., when an input is enabled to transmit the first cell of a packet comprising k cells, the input/output matching must persist for the following $k - 1$ slots.

This is equivalent to having an infinite weight on the corresponding edge of graph G until all the cells belonging to the packet are transferred to the output port. Note that no conflicts can arise between infinitely weighted connections, since no more than one cell can reach a given output in one slot, hence, two connections directed to the same output cannot simultaneously have an infinite weight.

We propose to extend the six considered SAs to operate in packet mode. The only complexity increase in the implementation is to add a Boolean variable at each input to flag overprioritized connections.

IV. ANALYTICAL MODELS OF IQ PACKET-MODE SCHEDULING

In this section, we present analytical models of IQ packet switches. We first formally prove that the maximum throughput achievable by IQ packet switches operating in packet mode is identical to that achievable in IQ switches with CM-SAs and OQ cell switches. This means that packet-mode operation can achieve 100% throughput provided the input traffic is admissible. We then show, with a simplified model, that although relative delays of cell-mode and packet-mode switches depend on traffic characteristics, it is possible to define traffic conditions in which IQ switches using PM-SAs provide lower packet delays than those using CM-SAs.

A. Maximum Achievable Throughput

1) *Definitions and Preliminary Results:* Given a system of M (in our case, $M = N^2$) discrete-time queues of infinite capacities, let X_n be the row vector of queue lengths at time n ; i.e., $X_n = (x_n^1, x_n^2, \dots, x_n^M)$, where x_n^i is the number of customers in queue i at time n .

The evolution of the length of queue i is described by the expression $x_{n+1}^i = x_n^i + a_n^i - d_n^i$, where a_n^i represents the number of customers arrived at queue i in time interval $(n, n+1]$, and d_n^i represents the number of customers departed from queue i in time interval $(n, n+1]$. Let $A_n = (a_n^1, a_n^2, \dots, a_n^M)$ be the vector of the numbers of arrivals at all queues, and $D_n = (d_n^1, d_n^2, \dots, d_n^M)$ be the vector of the numbers of departures from all queues. With this notation, the system evolution equation can be written as $X_{n+1} = X_n + A_n - D_n$. We assume in this section that the entries a_n^i of vectors A_n are independent

and identically distributed (i.i.d.) for variable n with fixed i , and independent for variable i with fixed n , although this latter constraint could be partly relaxed.

We indicate with $\|Y\|$ the Euclidean norm of vector $Y = (y^1, y^2, \dots, y^K)$: $\|Y\| = \sqrt{\sum_{i=1}^K (y^i)^2}$. In addition, we use $E[\cdot]$ to indicate averages.

Definition 1: A system of queues is said to be **strongly stable** if $\lim_{n \rightarrow \infty} \sup E\|X_n\|$ is finite.

We assume that the stochastic process describing the evolution of the system of queues is an irreducible discrete-time Markov chain (DTMC), whose state vector at time n is X_n , $X_n \in \mathbb{N}^M$. Most systems of discrete-time queues of practical interest can be described with models that fall in the DTMC class. The following general criterion for the strong stability of systems falling into this class is, therefore, useful.

Theorem 1: Given a system of queues with state vector X_n , and a function $V(X_n) = X_n W X_n^T$ (called Lyapunov function), if there exists a symmetric copositive² matrix $W \in \mathbb{R}^{M \times M}$, and two positive real numbers $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$, such that $\forall X_n: \|X_n\| > B$

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

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

This is a rephrasing of the results presented in [25, sec. IV]. Readers are referred to [25] for a proof. Since the identity matrix I is a symmetric positive semidefinite matrix, and hence, a copositive matrix, it is possible to state the following.

Corollary 1: Given a system of queues with state vector X_n , if there exists $\epsilon \in \mathbb{R}^+$, $B \in \mathbb{R}^+$ such that $\forall X_n: \|X_n\| > B$

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

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

2) *Stability of Packet-Mode Scheduling IQ Switches:* Consider an IQ packet switch, and suppose that all input packet sizes are multiples of some unit length called UL (UL may correspond to a bit, a byte, or a cell). Consider the system of discrete-time queues comprising all input queues of the packet switch. The discrete time unit corresponds to a continuous-time increment equivalent to UL.

We assume that customers correspond to cells to be transferred from input to output ports. Since we consider an IQ switch, each element d_n^i of the departure vector D_n can only assume the values 0 and 1 $\forall i$ and $\forall n$. The arrival of a packet corresponds to the arrival of a group of customers, whose cardinality equals the packet size in UL units. Therefore, a_n^i can be larger than 1. However, if the traffic is admissible, $E[a_n^i] \leq 1, \forall i$.

Definition 2: A sequence of time instants $t_n \in \mathbb{N}^+$ is a *non-defective* sequence of regeneration instants (or stopping times) for the evolution of a system of queues if, for any t_n , the evolution of the system following t_n is conditionally independent of the evolution of the system before t_n , given the state $Y(t_n)$; moreover, letting $z_n = t_{n+1} - t_n$, $E[z_n] < \infty$ and $E[z_n^2] < \infty$.

Definition 3: An IQ packet switch follows a **renewal MWM-QL schedule** if at each stopping time t_n a new

²An $M \times M$ matrix Q is copositive if $XQX^T \geq 0, \forall X \in \mathbb{R}^+{}^M$.

switching configuration is selected according to the outcome of MWM-QL, and the switching configuration is kept constant until t_{n+1} .

Definition 4: An IQ packet switch follows an **incremental MWM-QL schedule** if at each stopping time t_n a new matching is selected according to the outcome of MWM-QL algorithm. Between two consecutive stopping times t_n and t_{n+1} , partial updates of the switching configuration are allowed. These reconfigurations are performed according to the outcome of MWM-QL, operating on a subset of input and output ports.

Lemma 1: An IQ packet switch following a renewal MWM-QL schedule is stable under any admissible i.i.d. input traffic pattern A_n such that $E[A_n A_n^T] < \infty, \forall n$.

Proof: The evolution of the system of discrete-time queues in the IQ packet switch is represented by a DTMC whose state is defined by the vector of queue lengths X_{t_n} ; between consecutive stopping times, the system evolution satisfies

$$X_{t_{n+1}} = X_{t_n} + \sum_{i=0}^{z_n-1} (A_{t_n+i} - D_{t_n+i}).$$

Note that all $D_{t_n+i}, i < z_n$, refer to the same matching; however, they need not be all equal, since some queue scheduled for transmission at time t_n may become empty before the next stopping time. If this happens, no packet can be transferred from empty queues.

By using the Lyapunov function $V(X_{t_n}) = X_{t_n} X_{t_n}^T$

$$\begin{aligned} & E[V(X_{t_{n+1}}) | X_{t_n}] - V(X_{t_n}) \\ &= E \left[2 \sum_{i=0}^{z_n-1} (A_{t_n+i} - D_{t_n+i}) X_{t_n}^T \right. \\ & \quad \left. + \sum_{i=0}^{z_n-1} (A_{t_n+i} - D_{t_n+i}) \sum_{i=0}^{z_n-1} (A_{t_n+i} - D_{t_n+i})^T \right]. \end{aligned}$$

Thus, under the assumption that $E[A_{t_n+i} A_{t_n+i}^T]$ is finite (which corresponds to assuming finite packet size variances), since $E[D_{t_n+i} D_{t_n+i}^T]$ is also finite

$$\begin{aligned} & \lim_{\|X_{t_n}\| \rightarrow \infty} \frac{E[V(X_{t_{n+1}}) | X_{t_n}] - V(X_{t_n})}{\|X_{t_n}\|} \\ &= \lim_{\|X_{t_n}\| \rightarrow \infty} \frac{2E \left[\sum_{i=0}^{z_n-1} (A_{t_n+i} - D_{t_n+i}) X_{t_n}^T \right]}{\|X_{t_n}\|}. \end{aligned}$$

Define now $D_\delta = z_n D_{t_n} - \sum_{i=0}^{z_n-1} D_{t_n+i}$; as noted before, this difference may be nonzero when some queues become empty before changes in the switch configuration. Thus

$$\begin{aligned} & \frac{E \left[\sum_{i=0}^{z_n-1} (A_{t_n+i} - D_{t_n+i}) X_{t_n}^T \right]}{\|X_{t_n}\|} \\ &= 2 \frac{E \left[\sum_{i=0}^{z_n-1} A_{t_n+i} X_{t_n}^T - z_n D_{t_n} X_{t_n}^T + D_\delta X_{t_n}^T \right]}{\|X_{t_n}\|}. \end{aligned}$$

Wald's theorem³ [26, sec. 2–13] can be applied, since t_n is a sequence of stopping times, thereby obtaining

$$\begin{aligned} & \frac{E \left[\sum_{i=0}^{z_n-1} A_{t_n+i} X_{t_n}^T - z_n D_{t_n} X_{t_n}^T + D_\delta X_{t_n}^T \right]}{\|X_{t_n}\|} \\ &= \frac{2E[z_n] (E[A_n] - D_{t_n}) X_{t_n}^T + 2E[D_\delta] X_{t_n}^T}{\|X_{t_n}\|}. \end{aligned}$$

Note that $E[D_\delta] X_{t_n}^T \leq ME[z_n^2] < \infty$, since at most M components of D_δ can be nonnull, no component of D_δ can exceed the value z_n , and, finally, a component of D_δ can be nonnull only if the corresponding queue length at time t_n is smaller than z_n . Moreover, for each admissible load and nonnull queue length vector, $(E[A_n] - D_{t_n}) X_{t_n}^T < 0$, as proved in [21]. Thus

$$\begin{aligned} & \lim_{\|X_{t_n}\| \rightarrow \infty} \frac{E[V(X_{t_{n+1}}) | X_{t_n}] - V(X_{t_n})}{\|X_{t_n}\|} \\ &= \lim_{\|X_{t_n}\| \rightarrow \infty} \frac{2E[z_n] (E[A] - D_{t_n}) X_{t_n}^T + 2E[D_\delta] X_{t_n}^T}{\|X_{t_n}\|} \\ &\leq 2E[z_n] \lim_{\|X_{t_n}\| \rightarrow \infty} \frac{(E[A] - D_{t_n}) X_{t_n}^T}{\|X_{t_n}\|} < -E[z_n] \epsilon \end{aligned}$$

where $E[z_n] > 0$, since $z_n > 0$ by definition. \square

Lemma 2: An IQ packet switch following an *incremental MWM-QL schedule* is stable under any admissible i.i.d. input traffic pattern A_n such that $E[A_n A_n^T] < \infty, \forall n$.

Proof: The proof can be easily obtained by applying the Lyapunov function used in Lemma 1.

Consider an IQ packet switch with a given packet arrival process running an incremental MWM-QL scheduler with stopping times $\{t_n\}$. A particular renewal MWM-QL scheduler can be defined under the same arrival process and with the same set of stopping times. The latter scheduler is stable due to Lemma 1.

Since for the two schedulers we have the same set of stopping times, we get

$$\sum_{i=0}^{z_n-1} D_{t_n+i}^I X_{t_n+i} \geq \sum_{i=0}^{z_n-1} D_{t_n+i} X_{t_n+i}$$

where $D_{t_n+i}^I$ is the departure vector at time t_n+i for the incremental MWM-QL schedule, and D_{t_n+i} is the departure vector for the renewal MWM-QL schedule. \square

Definition 5: An IQ packet switch follows a **packet MWM-QL schedule** if a new switching configuration is selected according to a MWM-QL algorithm, whenever either of the following conditions holds.

- All packet transmissions end at the same time.
- All the queues selected for transfer become empty.

Definition 6: An IQ packet switch follows a **packet incremental MWM-QL schedule** if both the following conditions hold.

- Whenever either all packet transmissions end at the same time, or all the queues selected for transfer become empty, a new switching configuration is selected according to a MWM-QL algorithm as in a packet MWM-QL schedule.

³Wald's theorem: Let $\{X_n\}$ be a sequence of i.i.d. random variables with finite expectation $E[X]$, and M a stopping time for the sequence X_n , with finite expectation $E[M] < \infty$, then $E[\sum_{i=1}^M X_i] = E[M]E[X]$.

- Whenever some queues selected for transfer become idle (i.e., either they are empty, or packet transmissions end) a partial update of the switching configuration is allowed, according to an MWM-QL algorithm among idle ports.

Lemma 3: Consider an IQ packet switch, following either a packet MWM-QL schedule or a packet incremental MWM-QL schedule, whose input traffic is formed by variable size packets with i.i.d. random size. Packet sizes are expressed in integer multiples of UL. Assume that the average packet size is l and the packet size variance is σ^2 (both being finite). Assume that the transmission of packets from all queues selected by the MWM-QL algorithm starts at the same time with exactly the same rate. Consider the sequence of instants t_n at which either the transmission of all the packets at the head of the selected queues ends at the same time, or all selected queues become empty. The sequence of stopping times t_n is nondefective, i.e., $z_n = t_{n+1} - t_n$ are such that $E[z_n] < \infty$ and $E[z_n^2] < \infty$.

Proof: For simplicity, assume that the packet⁴ size distributions at all queues are aperiodic, i.e., the maximum common divisor of all possible packet sizes expressed in UL is equal to 1 (the proof can be easily extended to the case of periodicity) and further assume a switch operating according to a packet MWM-QL schedule (the proof can be easily extended to a switch operating according to a packet incremental MWM-QL schedule).

We suppose that switch queues have infinite length, so that we neglect the probability that switch queues become empty near traffic saturation; thus, we obtain an overestimate of $E[z_n]$ and $E[z_n^2]$, since times t_n are defined by only the sequence of instants in which the transmissions of all packets at the head of the selected queues end at the same time.

Each sequence of instants at which transmissions of packets end at queue k forms a discrete-time renewal point process⁵ (i.e., a lattice renewal point process with period equal to 1), thanks to the independence of packet sizes. Thus, for Blackwell's theorem⁶ [26, sec. 2–19], the average number $E[f_n^k]$ of packets whose transmissions end at queue k at time n satisfies

$$\lim_{n \rightarrow \infty} E[f_n^k] = \frac{1}{l}. \quad (1)$$

However, no more than one packet transmission can end at each queue at each time (assuming no packet is of size zero); thus, $E[f_n^k]$ equals the probability that a packet ends.

$$E[f_n^k] = P\{\text{transmission ends at time } n \text{ and at queue } k\}.$$

Limit (1) implies that, for any integer $m > 1$, there exists an instant n_k such that $\forall n > n_k$.

$$P\{\text{transmission ends at time } n \text{ and at queue } k\} > \frac{1}{ml} > 0.$$

Let N_s be the number of queues selected for transmission at instant n . Then we may easily compute the probability that the transmission of these N_s HOL packets ends at instant n , since

⁴Note that the state definition in our analysis changes from number of cells to number of packets from now on.

⁵A renewal point process is a sequence of times t_n , said renewals, such that $X_n = t_{n+1} - t_n$ are i.i.d., positive defined random variables.

⁶Blackwell's theorem: For discrete-time renewal processes with average inter-renewal time $1/\mu$, $\lim_{n \rightarrow \infty} P\{\text{observing a renewal at time } n\} = 1/\mu$.

no correlation exists among the queues' behavior. Thus, given m , for $n > n_k, \forall k$

$$\begin{aligned} & P\{\text{all transmissions end at time } n\} \\ &= \prod_{k=1}^{N_s} P\{\text{tx ends at time } n \text{ and at queue } k\} \\ &> \prod_{k=1}^{N_s} \frac{1}{ml} = \frac{1}{(ml)^{N_s}} > 0. \end{aligned}$$

Consider now the sequence of instants t_n at which either all packet transmissions end, or selected queues become empty. The sequence t_n forms a renewal process; thus Blackwell's theorem applies.

$$P\{\text{all transmissions end at time } n\} = E[f_n] = \frac{1}{E[z_n]}$$

where $E[f_n]$ is the average number of regenerations at time n ; since

$$P\{\text{all transmissions end at time } n\} > 0$$

we obtain $E[z_n] < \infty$.

To prove that also $E[z_n^2] < \infty$, consider all packets transmitted from queue k between two subsequent regenerations; let W be the number of such packets, and L_j be their sizes expressed in UL. We can write

$$\begin{aligned} E[z_n^2] - E^2[z_n] &= E \left[\left(\sum_{j=1}^W (L_j - E[L_j]) \right)^2 \right] \\ &= E \left[\sum_{j=1}^W (L_j^2 - E^2[L_j]) \right] \\ &\quad + E \left[\sum_{j=1}^W \sum_{\substack{i=1 \\ i \neq j}}^W (L_j L_i - E[L_j L_i]) \right]. \end{aligned}$$

The second term in the sum can be easily shown to be null by conditioning on the value of W ; it can, therefore, be eliminated. As a consequence

$$E[z_n^2] - E^2[z_n] = E \left[\sum_{j=1}^W L_j^2 \right] - \sum_{j=1}^W E^2[L_j]$$

and by Wald's theorem, since regeneration points are stopping times for the sequence L_j

$$E[z_n^2] - E^2[z_n] = E[W]E[L^2] - E[W]E^2[L_j] = E[W]\sigma^2.$$

Since $E[W]$ is finite (otherwise, $E[z_n]$ would be infinite) and σ is finite by assumption, we have that $E[z_n^2] < \infty$. \square

We can now state our main result.

Theorem 2: Any IQ packet switch following either a *packet MWM-QL schedule* or a *packet incremental MWM-QL schedule* is strongly stable, provided that:

- the input traffic is admissible;
- the input traffic is formed by variable size packets with i.i.d. random size having finite average and variance;
- the transmission of packets from all queues selected by the MWM-QL schedule starts at the same time with the same rate.

Proof: The proof is straightforward from Lemma 1 (for packet MWM-QL schedulers), or Lemma 2 (for packet incremental MWM-QL schedulers), and Lemma 3, since the assumptions of Theorem 2 satisfy the conditions under which Lemma 3 holds. \square

Note that the packet incremental MWM-QL schedule in Definition 6 corresponds to the packet mode MWM-QL described in Sections III-B and III-C, and used in the simulation results.

B. Approximate Packet Delay Estimation

An intuitive explanation of the relation between the average packet delay values in PM-SAs and CM-SAs can be provided with the help of an approximate queueing model.

We focus on one output port, and on packets directed from the different inputs to that output port, and consider only the packet delay component due to virtual output queueing (thus disregarding the effect of segmentation and reassembly modules). To estimate the packet delay with a queueing model, we also have to disregard the output conflicts in the switch; thus, our estimates will be reasonably accurate only for low to medium traffic loads. Still, the approximate model provides useful insight into the phenomena that will be observed in analyzing simulation results. The transfer toward the output port corresponds to the packet service, and packets are modeled as customers requiring a variable amount of service. As a further simplification, we describe the packet arrival process at the switch ingress with a Poisson process, without taking care of overlapping ingress times.

This simplified setting corresponds to an M/G/1 queue, where CM-SAs can be paralleled to processor-sharing (PS) or round-robin (RR) service, since all packets directed to the considered output are simultaneously served (again because of low traffic), and PM-SAs can be paralleled to FIFO service, since each packet is served separately, with no interleaving of cells of different packets. These assumptions hold especially for iSLIP under low load, since iSLIP implements a round-robin mechanisms. Since we do not model blocking of the server due to conflicts for the same output, the model is valid only for low load.

We know from queueing theory that the average delay $E[D_{\text{PS}}]$, in the case of PS service, is

$$E[D_{\text{PS}}] = \frac{\rho E[S]}{1 - \rho}$$

where $E[S]$ is the average service time and ρ is the queue traffic (or utilization factor). Instead, for an M/G/1 queue with FIFO service, we know that

$$E[D_{\text{FIFO}}] = \frac{\rho E[S]}{1 - \rho} \times \frac{1 + C_v^2}{2}$$

where C_v is the coefficient of variation of the service time, which, in our case, refers to packet sizes.

Note that $E[D_{\text{PS}}]$ is equal to the average delay in an M/M/1 queue with FIFO service, so that $E[D_{\text{PS}}] = E[D_{\text{FIFO}}]$ for negative exponential distribution of packet sizes (that is, for $C_v = 1$).

We define the *packet-mode gain*, denoted G , as the ratio between the average packet delay experienced with CM-SAs and the average packet delay experienced with PM-SAs. As noted before, in our simplified analysis these average packet

delays refer only to the waiting time in input queues, not comprising the delays due to segmentation and reassembly. From the simplified queueing model, G can be estimated as

$$G = \frac{2}{1 + C_v^2}.$$

Thus, the approximate analytical model predicts packet delay gains for PM-SAs in the case of packet size distributions with small variance ($C_v < 1$), whereas CM-SAs are expected to provide lower packet delays when the packet size variance is large ($C_v > 1$).

Note that similar results could have been obtained by modeling cell-mode operation with group arrivals, and by interpreting individual customers as cells, and groups as packets. Packet delays would in this case be equivalent to group delays. This approach was pursued in [18], to show some possible improvement of the delays experienced by packets in a variable-size packet network with respect to a fixed-size packet network (such as ATM).

V. SIMULATION RESULTS

We report simulation results for packet switches with N input/output interfaces, assuming that all input/output line rates are equal, and that only unicast traffic flows are present. We assume finite queue sizes in the simulation models. Each input queue $Q^{(ij)}$ has finite length Q_{max} ; when a cell directed to output j arrives at input i , and queue $Q^{(ij)}$ is full, the cell is lost. No buffer sharing among queues at the same input port is allowed.

A. Packet Size Distribution

Our simulation models do not explicitly describe the arrival of IP datagrams at packet-switch inputs. We instead model the arrival of cell bursts at the inputs of the internal cell switch. These cell bursts originate from the segmentation of a packet.

The cell arrival process at input i , $A_k^{(i)}$, is characterized with a two-state ON-OFF model. When the input port is in the ON state, a packet is being received. The number of slots spent in the ON state, i.e., the size in cells of the packet, is a discrete random variable $\Phi^{(ij)}$ for packets directed from input i to output j . No cells are received in the OFF state. The number of slots spent in the OFF state is geometrically distributed with average $E_{\text{OFF}} = (1 - p)/p$. The parameter p is set so as to achieve the desired input load.

We consider the following packet size distributions $\Phi^{(ij)}$.

UNIFORM(a, b). Packet sizes are uniformly distributed between a and b cells. In the following, $a = 1$ and $b = 192$. The value of b comes from the maximum transmission unit (MTU) of IP over ATM.

BIMODAL($a, b; p_a$). Packet sizes are chosen equal to either a cells with probability p_a , or b cells with probability $1 - p_a$.

EXPONENTIAL(a). Packet sizes have exponential distribution with mean a .

TRIMODAL($a, b, c; p_a, p_b$). Packets sizes are chosen equal to either a cells with probability p_a , or b cells with probability p_b , or c cells with probability $1 - p_a - p_b$. The trimodal distribution was chosen to accurately describe the IP packet-size distribution measured at the ports of the router gateway of the Politec-

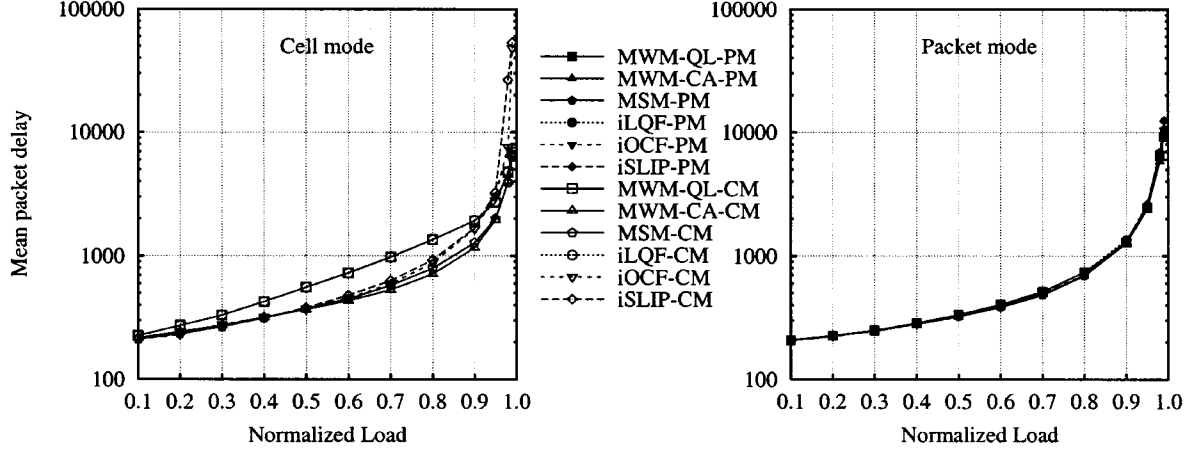


Fig. 2. Average packet delay for CM-SAs and PM-SAs in the uniform traffic scenario.

nico di Torino. Measurements were collected with TSTAT [27], [28], an IP-measurement tool developed by our research group. The collected traffic trace refers to 13 days of January 2001, and contains about 1 billion IP packets. Similar packet size distributions were found with other traffic traces. The measured distribution was approximated by TRIMODAL(1, 12, 32; 0.559, 0.200). TCP acknowledgments (ACKs) correspond to one cell, 576-byte packets to 12 cells, and 1500-byte packets to 32 cells.

B. Traffic Scenarios

We consider the following three traffic scenarios, which are described through their corresponding Γ and Γ_P matrices (see their definitions in Section II-A), and $\Phi^{(ij)}$ random variables.

Uniform Scenario. In this case $\gamma^{(ij)} = \gamma_P^{(ij)} = 1/N^2$, $\forall i, j$ (uniform traffic). Packet sizes are chosen according to UNIFORM(1, 192). Numerical results for this scenario will be shown in the case $N = 16$.

Spotted Scenario. In this case, an unbalanced load is generated toward different outputs, to emphasize the performance limitations of simpler SAs. In addition, packet sizes are chosen according to BIMODAL(3, 100; 0.5), to highlight possible starvation effects in the service of long or short packets. We assume $N = 8$, and set

$$\Gamma = \Gamma_P = \frac{1}{40} \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Diagonal Scenario. In this case $\gamma_P^{(ij)} = 1/N^2$ (the traffic is uniform at the packet level), and packets sizes are chosen according to BIMODAL(100, 1; $\mathbb{1}_{i=j}$), being $\mathbb{1}_{i=j}$ equal to 1 if $i = j$, and 0, otherwise (packets on the diagonal have fixed size equal to 100 cells, else equal to 1). Numerical results for this scenario will be shown in the case $N = 16$. The reason for considering this scenario is to highlight the starvation effect in the service of short packets due to the transfer of long packets.

Note that these traffic scenarios were chosen to illustrate critical differences in the behavior of PM-SAs and CM-SAs. Accu-

rate modeling of real traffic patterns in packet networks is outside the scope of this paper.

C. Performance Indices

Results are presented with graphs where the following performance indices are plotted versus the switch normalized traffic load. The latter is defined as the ratio between the input traffic load and the total capacity of input/output lines, both of them computed at the cell level (hence, the normalized load varies between 0 and 1).

- **Cell delay.** This is the time spent by cells in the cell-switch queues.

- **Packet delay.** This is the overall delay of a packet, considering the ISM module, the internal cell-switch queues, the ORM module, and the final packet FIFO. It is computed only for packets completely delivered at switch outputs, measuring the time from the ingress of the last cell of the packet into the ISM module until the egress of that same last cell from the final packet FIFO. Constant delay components are removed; hence, a single-cell packet traverses an empty packet switch in null time, and a packet comprising k cells has a best-case delay equal to $2(k - 1)$ slots, due to wait in the segmentation and reassembly phases.

- **Packet expansion.** This metric is computed as the ratio between the number of slots necessary to transfer all the cells of the packet from the input to the output of the switching fabric, and the size of the observed packet. For PM-SAs, the packet expansion is always one, but when a CM-SA is adopted, the transfer of the cells belonging to a packet can be interleaved with the cells of other packets, and the packet expansion can be larger than one.

Simulation runs were executed until the estimate of the average cell delay reached with probability 0.95 a relative width of the confidence interval equal to 2%. The estimation of the confidence interval width is obtained with a batch means approach.

D. Uniform Scenario

Fig. 2 shows the curves of the average packet delay for the considered CM-SAs and PM-SAs. No losses were experienced with queue lengths equal to 30 000 cells.

Consider first the curves of CM-SAs. Three different regions can be identified, corresponding to low, medium, and high load, respectively. When the load is low (say, $\rho < 0.1$), all algorithms yield almost identical performance, since contention arises with very low probability, and the matching is often unique. For medium load (in this case, with $0.1 < \rho < 0.8$), the effect of metrics is predominant on the SA performance. For high load (about $\rho > 0.8$), on the contrary, the matching method has the largest impact on performance. Indeed, looking at the medium load region, we see that iLQF-CM and MWM-QL-CM behave almost identically, and worse than all other algorithms, which yield practically the same average packet delay. This does not contradict the common belief that MWM-QL-CM yields the best average cell delay: If the average cell (rather than packet) delay is plotted, MWM-QL-CM actually yields the lowest curve. However, here we focus on packet delays, since they better reflect the performance experienced by higher layer protocols. To understand the reason for the different influence of the three considered metrics (QL, CA, and QO), we rephrase the result of Section IV-B: When the coefficient of variation of the packet size is small ($C_v < 1$), the interleaving of transfers of cells belonging to different packets should be avoided to minimize packet delays, because the packet-mode gain is greater than one. On the contrary, when $C_v > 1$, interleaving is beneficial, because the packet-mode gain is less than one. Observe that the QL metric under cell mode tends to interleave packets; to understand why, consider two packets of the same size contending for the same output: One cell from each packet will be transferred alternatively. Also, the QO metric under cell mode aims to distribute services among all nonempty queues, hence, tends to interleave the cells of all packets contending for the same output. However, the QL metric interleaves packets more than the QO metric, since it tends to equalize all queue lengths and, thus, interleaves for a longer period of time the (at most N) packets directed to the same output. On the contrary, the CA metric under cell mode tends to avoid interleaving. Indeed, whenever a packet is interleaved, the CA for all the remaining cells belonging to the interleaved packet grows, thus decreasing the probability of future interleaving. Under the uniform traffic scenario, C_v is very low ($C_v \approx 0.287$), and to minimize the average packet delay it is better to avoid interleaving: The QL metric is thus worse than QO and CA. To be better persuaded about these arguments, the reader can refer to the plots of the mean packet expansion for CM-SAs in Fig. 3. For medium load, MWM-QL-CM and iLQF-CM show the highest interleaving, whereas the algorithms based on the CA and QO metrics exhibit similar interleaving, but, as expected, CA minimizes the interleaving.

Return to CM-SAs in Fig. 2; for high load (about $\rho > 0.8$), the matching method has the most impact on the SA performance. All the optimal algorithms (MWM-QL, MWM-CA and MSM) show lower packet delays, whereas the suboptimal matching methods (except for iLQF, which is very efficient) show much higher delays, but only for very high load ($\rho > 0.99$).

Now consider the curves of the average packet delay for PM-SAs in Fig. 2. We can immediately observe that PM-SAs reduces the differences among the metrics and the matching methods. However, we will show later that this is true only when considering aggregate performance indexes, thus, possibly hiding some underlying unfairness among the delays experienced by packets with different sizes.

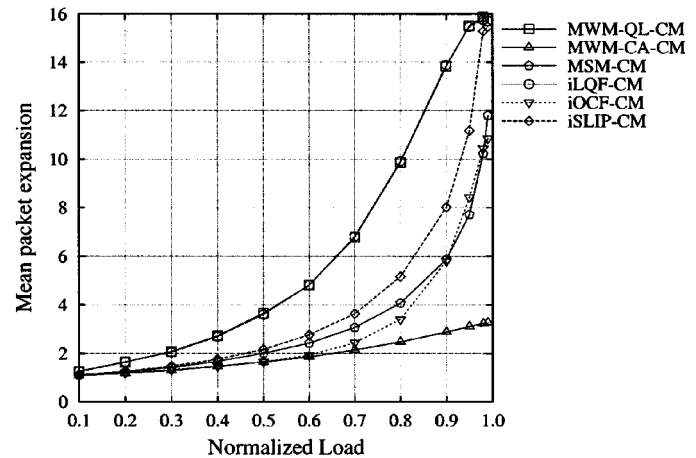


Fig. 3. Average packet expansion for CM-SAs in the uniform traffic scenario.

Fig. 4 plots the coefficient of variation of the packet delay for CM-SAs and PM-SAs. Under cell mode, metrics have the highest influence on the packet delay variance, but also the matching method becomes significant for high load. As expected, the algorithms based on the CA metric yield the lowest variance, exactly because they select cells to be transferred based on the time they spent in input queues. On the contrary, under packet mode, only the metric determines the packet delay variance, whereas the matching method has hardly any influence. Hence, for the variance of packet delays, PM-SAs emphasize the effect of the metric and reduce the effect of the matching method.

E. Spotted Scenario

Fig. 5 plots the average packet delays when CM-SAs and PM-SAs are adopted, under spotted traffic. In this case, as for uniform traffic, it is possible to identify three regions, the first one (low load, not shown) where all algorithms behave almost identically, the second one (medium load) where the metric drives performance, and the third one (high load) where the matching method dominates. Also in this case C_v is less than one ($C_v \approx 0.94$), so that interleaving should be avoided to minimize delays. Qualitatively, the performance at medium load is very similar to the case of uniform traffic. As soon as the load increases, the matching method becomes less efficient for all suboptimal algorithms, and losses are experienced (here, the maximum queue size is set equal to 10 000 cells). The flattening of delay curves to horizontal asymptotes is an indication of packet losses: Each packet enters an almost full queue and faces an almost constant delay. On the contrary, optimal algorithms yield much lower delays and do not experience losses. This traffic scenario thus highlights the performance differences between optimal and suboptimal algorithms working in cell mode.

The two graphs of Fig. 5 again show that PM-SAs decrease the effect of both metric and matching method. Thanks to PM-SAs, suboptimal algorithms with very simple metrics (such as iSLIP) can achieve throughputs very similar to those of much more complex algorithms. In this particular case, however, iSLIP-PM can experience some delay penalty with respect to the other PM-SAs, since the traffic scenario was expressly designed so as to “break” the round robin mechanisms of iSLIP.

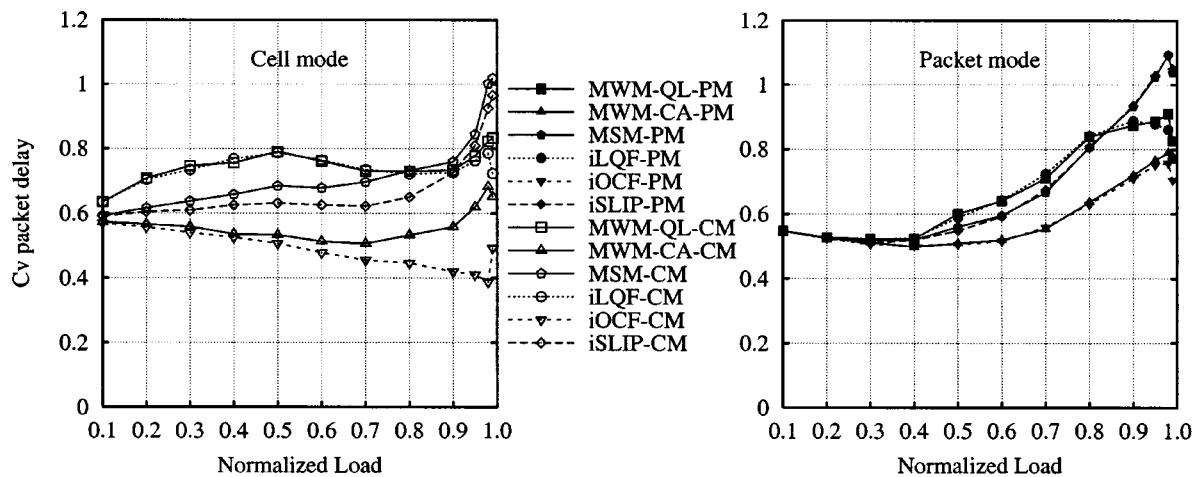


Fig. 4. Coefficient of variation of packet delay for CM-SAs and PM-SAs in the uniform traffic scenario.

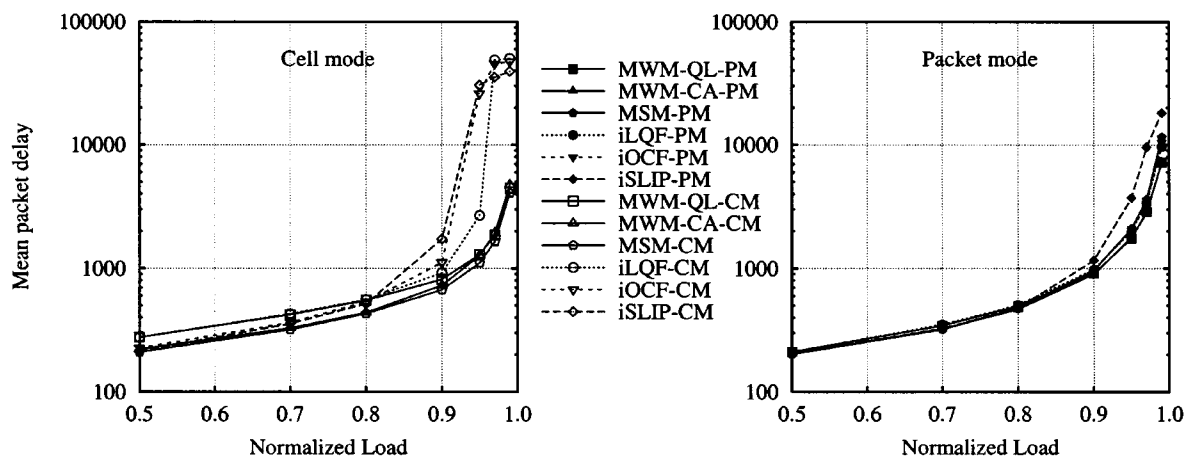


Fig. 5. Average packet delay for CM-SAs and PM-SAs in the spotted traffic scenario.

The throughput improvement for suboptimal PM-SAs can be explained in the following way. When a packet transmission is completed, PM-SAs compute the new matching only for all those inputs and outputs that are not currently involved in packet transfers; whereas the outcome of a complete matching is generally not maximal, by reducing the number of inputs to be matched, it becomes easier to compute a maximal matching. For example, iSLIP and iOCF always find a maximal matching when the number of iterations of the algorithm is not smaller than the number of inputs available for the matching. When a PM-SA is adopted, the matching changes very little in subsequent time slots. In fact, results not reported here indicate that the average number of edges which are common to two successive matchings is always greater than 97% of the total number of edges, under uniform and spotted traffic. This means that usually at most one or two edges are changed.

F. Diagonal Scenario

The diagonal traffic scenario is used to highlight the different delays experienced by packets belonging to two classes: short and long packets. We want to quantify the temporary starvation in the service of a class of packets due to the transfer of packets of the other class. Simulation results will tell us that the main responsible of this temporary starvation phenomena is the SA metric, and that starvation, which is related to unfairness in services, exists for both CM-SAs and PM-SAs.

Fig. 6 plots the average packet delay experienced by long packets (on the main diagonal of the traffic matrix, with size 100 cells), for CM-SAs and PM-SAs. Fig. 7 plots the average packet delay experienced by short packets (with size equal to one cell), for CM-SAs and PM-SAs. Consider first the delays for CM-SAs. For high load, iSLIP-CM and MSM-CM yield very short delays for short packets, but very long delays for long packets. This is mainly due to the QO metric, which is insensitive to the arrivals of large batches of cells generated by long packets. Long packets experience losses under iSLIP-CM, and the maximum throughput of the overall packet flow is about 0.89. Short packets are served a short time after arrival, since their weight is equivalent to that of long packets at the head of their queues. Hence, short packets experience very short delays, as can be seen in Fig. 7, at the price of the very long delays incurred by long packets observed in Fig. 6. Thus, the QO metric induces temporary starvation for long packets. On the contrary, the QL metric of MWM-QL-CM and iLQF-CM has the opposite effect: When a large batch of cells is generated due to a long packet arrival, the corresponding input queue increases its weight, and a long packet is served after a short delay, at the expense of short packets. This behavior induces temporary starvation for short packets. Finally, the CA metric minimizes the delay variance, as showed before, hence, minimizes the differences in the delays experienced

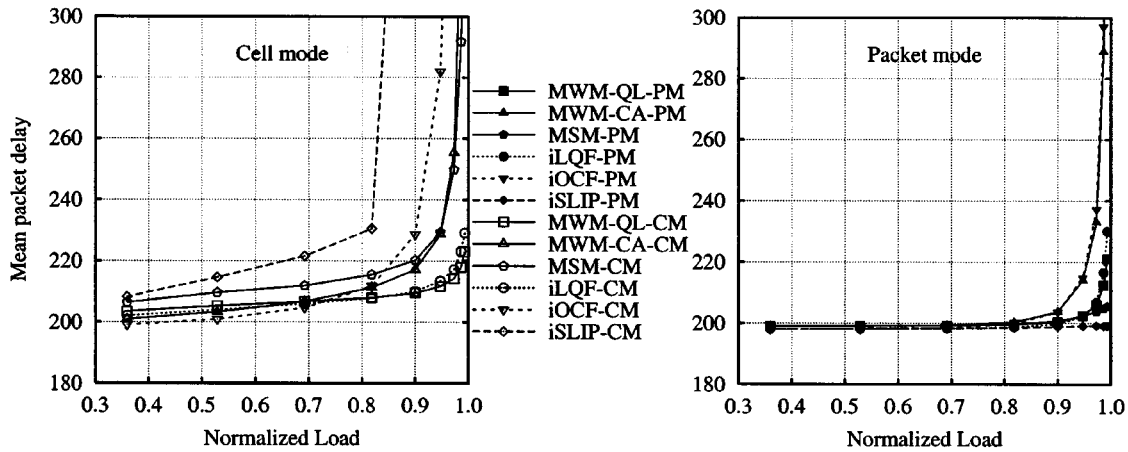


Fig. 6. Average packet delay for long packets in the diagonal traffic scenario with CM-SAs and PM-SAs.

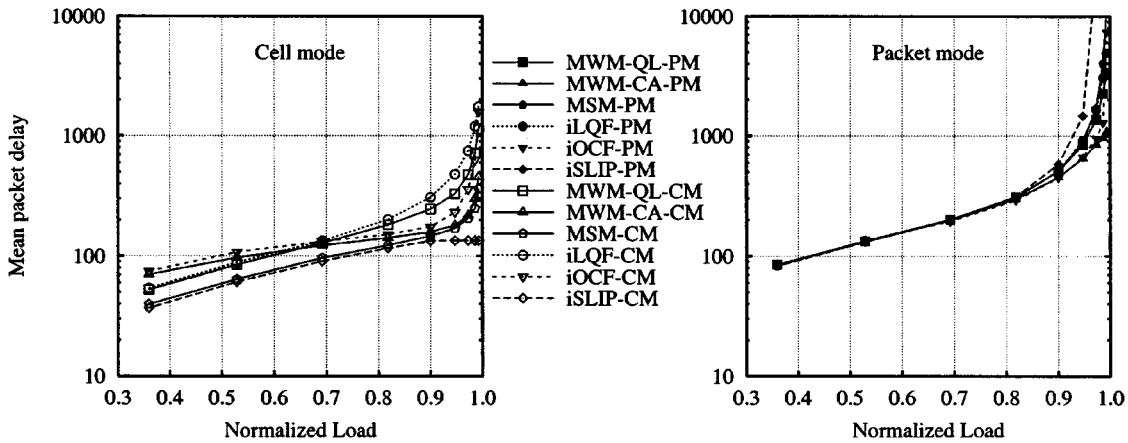


Fig. 7. Average packet delay for short packets in the diagonal traffic scenario with CM-SAs and PM-SAs.

by long and short packets. This effect is visible by comparing Figs. 6 and 7, where MWM-CA-CM shows packet delays equal to about 280 slots for both long and short packets at load $\rho = 0.95$. It can also be observed that, with cell mode, the delay difference produced by different matching methods is relevant; indeed, optimal algorithms are always able to achieve 100% throughput, even if the metric induces starvation for a class of packets (like in the case of QO for long packets). When PM-SAs are considered, although the differences between the curves are again reduced, the effect of metrics is predominant, and the effect of the matching method is almost negligible. The QO metric, used by iSLIP-PM and MSM-PM, gives very short delays for long packets, at the expense of short packets, as shown by plots in Figs. 6 and 7. iSLIP-PM does not delay long packets, since its round-robin mechanism tends to synchronize the services of long packets of the traffic matrix diagonal. Note that now losses are experienced for short packets, but the overall maximum throughput for iSLIP-PM is about 0.97, which is much better than the value 0.89 reached by iSLIP-CM. For MWM-CA-PM and iOCF-PM, the CA metric, as usual, tends to equalize packet delays experienced by long and short packets. The QL metric of MWM-QL-PM and iLQF-PM behaves very similarly to cell mode. Only the delays of short packets increase slightly, because their temporary starvation increases due to the complete transmission of long packets,

whereas in cell mode short packets could be interleaved with long packets.

In conclusion, under diagonal traffic, the QO metric can induce starvation in the service of a class of packets at expense of the other; this can also reduce the maximum achievable throughput, if the optimal matching method (MSM) is not adopted. This conclusion holds both for CM-SAs and PM-SAs. However, when a PM-SA is adopted, the overall maximum throughput is higher. The CA metric avoids starvation phenomena for both CM-SAs and PM-SAs, whereas the QL metric behaves very similarly in both modes, also if it always suffers temporary starvation for short packets.

G. Packet-Mode Gains

In this section, we show simulation results concerning the actual packet-mode gain values under different packet size distributions, for iSLIP, which is the SA that is better described by the approximate model of Section IV-B. We assume packet sources and destinations to be uniformly distributed, as in the uniform traffic scenario. We consider four different packet-size distributions: uniform, bimodal, exponential, and trimodal. Table II shows, for each distribution, the coefficient of variation C_v , which is the parameter that drives the delay performance, according to our simplified model. We chose uniform, exponential, and bimodal distributions to study the packet mode

TABLE II
THEORETICAL PACKET-MODE GAINS FOR DIFFERENT
PACKET SIZE DISTRIBUTION

Distribution	C_v	Theoretical packet mode gain G
UNIFORM(1, 192)	0.287	1.84
EXPONENTIAL(100)	1.00	1.00
TRIMODAL(1, 12, 32; 0.559, 0.200)	1.20	0.82
BIMODAL(1, 101; 0.99)	4.975	0.077

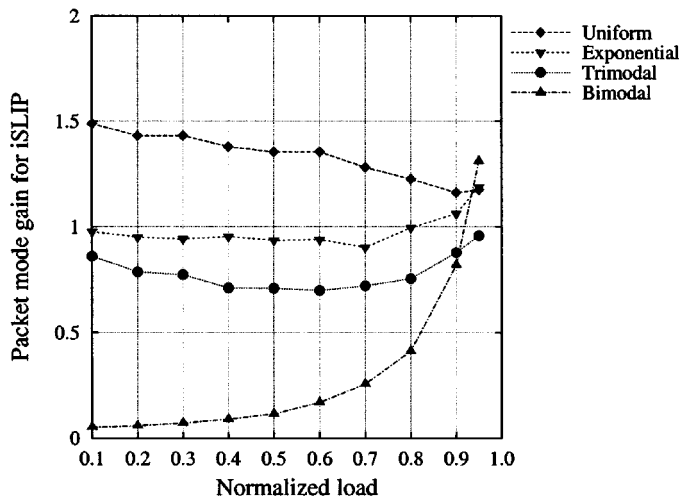


Fig. 8. Packet-mode gain G for iSLIP under different packet size distributions.

gain for $C_v < 1$, $C_v = 1$, and $C_v > 1$. We also chose trimodal distribution to study a more realistic traffic pattern.

Fig. 8 and Table II show the simulated and theoretical packet mode gains for iSLIP. Simulation results clearly show that the predictions of the approximate queueing model of Section IV-B are reliable for low loads. For high load, our approximation fails, because of the contentions among HOL packets destined to the same output, which is not described in the approximate model.

Note also that, according to the results just shown for the trimodal distribution, CM-SAs for iSLIP could be considered better suited to Internet traffic than PM-SAs. This conclusion is however incorrect, since the gain is not very low (being always greater than 0.7) and iSLIP-PM is much more robust to unbalanced traffic patterns than iSLIP-CM, as shown by the simulation results presented in the previous sections. In addition, in the cases where we observed an improvement in the maximum achievable throughput (as under spotted and diagonal traffic), the packet-mode gain becomes infinite, since the packet delay is finite only for PM-SAs.

VI. CONCLUSION

This paper focused on architectures and scheduling algorithms for IQ packet switches dealing with variable-size packets.

We considered six previously proposed scheduling algorithms for the transfer of fixed-size data units, and proposed novel modifications of these scheduling algorithms to deal with variable-size packets, having in mind IP routers internally using cell-based switching engines. These packet-mode scheduling algorithms require a negligible complexity increase with respect to cell scheduling, and can yield performance advantages in terms of packet delays over cell-mode schedulers.

We analytically proved that no throughput limitations exist by operating a switch in packet mode, showed that delays in IQ switches operating in cell or packet mode depend on the traffic characteristics, and validated our analytical findings with simulation experiments.

In particular, this paper contains the first proof that packet-mode scheduling in an IQ switch allows 100% throughput to be obtained (under the conditions of Theorem 2), as well as an approximate analytical model to show the dependence on the packet size distribution of the delay improvements possible with packet-mode schedulers. Another important contribution of this paper is the insight provided by the simulation results, which clearly indicates that packet-mode schedulers can offer very good performance, almost independent from the matching method and the metric. This is quite an interesting result, which indicates that by adopting packet-mode scheduling it is possible to use simple metrics as well as matching methods with reduced complexity with little sacrifice in performance.

REFERENCES

- [1] T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Trans. Commun.*, vol. COM-27, pp. 1449–1455, Oct. 1979.
- [2] Y. Tamir and G. Frazier, "High performance multi-queue buffers for VLSI communication switches," in *15th Annu. Symp. Computer Architecture*, Washington, DC, June 1988, pp. 343–354.
- [3] T. Anderson, S. Owicki, J. Saxe, and C. Thacker, "High speed switch scheduling for local area networks," *ACM Trans. Comput. Syst.*, vol. 11, no. 4, pp. 319–352, Nov. 1993.
- [4] R. Lamaire and D. Serpanos, "Two dimensional round-robin schedulers for packet switches with multiple input queues," *IEEE/ACM Trans. Networking*, vol. 2, pp. 471–482, Oct. 1994.
- [5] (2000, Apr.) GRF-multiGigabit routers. Product Overview. [Online]. Available: <http://www.lucent.com>.
- [6] (2000, Apr.) Cisco 12000 Gigabit switch router [Online]. Available: <http://www.cisco.com>.
- [7] N. McKeown, M. Izzard, A. Mekkittikul, B. Ellesick, and M. Horowitz, "The Tiny Tera: A packet switch core," *IEEE Micro*, vol. 17, pp. 27–40, Feb. 1997.
- [8] A. Hung, G. Kesidis, and N. McKeown, "ATM input-buffered switches with guaranteed-rate property," in *Proc. 3rd IEEE Symp. Computers and Communications (ISCC'98)*, Athens, Greece, July 1998, pp. 331–335.
- [9] H. Duan, J. W. Lockwood, S. M. Kang, and J. D. Will, "A high performance OC12/OC48 queue design prototype for input buffered ATM switches," in *Proc. IEEE INFOCOM 1997*, vol. 1, Los Alamitos, CA, pp. 20–28.
- [10] C. Partridge *et al.*, "A 50-Gb/s IP router," *IEEE/ACM Trans. Networking*, vol. 6, pp. 237–248, June 1998.
- [11] M. Ajmone Marsan, A. Bianco, E. Leonard, and L. Milia, "RPA: A flexible scheduling algorithm for input buffered switches," *IEEE Trans. Commun.*, vol. 47, pp. 1921–1933, Dec. 1999.
- [12] A. Mekkittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," in *Proc. IEEE INFOCOM 1998*, vol. 2, New York, pp. 792–799.
- [13] N. McKeown and T. E. Anderson, "A quantitative comparison of scheduling algorithms for input-queued switches," *Comput. Netw. ISDN Syst.*, vol. 30, no. 24, pp. 2309–2326, Dec. 1998.
- [14] K. J. Christensen, "Design and evaluation of a parallel-pollled virtual output queued switch," in *IEEE Int. Conf. Communications (ICC) Conf. Rec.*, vol. 1, Helsinki, Finland, June 2001, pp. 112–116.
- [15] D. N. Serpanos and P. I. Antoniadis, "FIRM: A class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues," in *Proc. IEEE INFOCOM 2000*, vol. 2, Tel Aviv, Israel, pp. 548–555.
- [16] H. Chen, J. Lambert, and A. Pitsilleddes, "RC-BB switch. A high performance switching network for B-ISDN," in *Proc. IEEE GLOBECOM '95*, vol. 3, Singapore, pp. 2097–2101.
- [17] A. Shaw, "Fixed-length packets versus variable-length packets in fast packet switching networks," Massachusetts Inst. Technol., Cambridge, Tech. Rep., Mar. 1994.
- [18] I. Cidon, J. Derby, I. Gopal, and B. Kadaba, "A critique of ATM from a data communications perspective," *J. High Speed Networks*, vol. 1, no. 4, pp. 315–336, 1992.

- [19] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queueing on a space division switch," *IEEE Trans. Commun.*, vol. COM-35, pp. 1347–1356, Dec. 1987.
- [20] R. E. Tarjan, *Data Structures and Network Algorithms*. Philadelphia, PA: SIAM, Nov. 1983.
- [21] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, pp. 1260–1267, Aug. 1999.
- [22] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. dissertation, Univ. California, Berkeley, 1995.
- [23] —, "iSLIP: A scheduling algorithm for input-queued switches," *IEEE Trans. Networking*, vol. 7, pp. 188–201, Apr. 1999.
- [24] M. Ajmone Marsan, A. Bianco, E. Filippi, P. Giaccone, E. Leonardi, and F. Neri, "On the behavior of input queueing switch architectures," *Eur. Trans. Telecommun.*, vol. 10, no. 2, pp. 111–124, Mar. 1999.
- [25] P. R. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 251–260, Feb. 1995.
- [26] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [27] R. Lo Cigno, M. Mellia, and A. Carpani. TSTAT web page. [Online]. Available: <http://www.tlc-networks.polito.it/Tstat>.
- [28] M. Mellia, A. Carpani, and R. Lo Cigno, "Measuring IP and TCP behavior on an edge node," in *Proc. IEEE GLOBECOM'02*, Taipei, Taiwan, Nov. 2002, [Online]. Available: <http://www.tlc-networks.polito.it/planet-ip/cour02/g1s1p4.ps>.



Marco Ajmone Marsan (F'99) holds degrees in electronic engineering from the Politecnico di Torino, Torino, Italy, and the University of California at Los Angeles (UCLA). In 2002, he was awarded an *Honoris Causa* degree in telecommunication networks from the Budapest University of Technology and Economics, Budapest, Hungary.

He is currently a Full Professor in the Department of Electronics, Politecnico di Torino. From 1975 to 1987, he was with the Department of Electronics, Politecnico di Torino, first as a Researcher, then as an Associate Professor. From 1987 to 1990, he was a Full Professor in the Department of Computer Science, University of Milan, Milan, Italy. During the summers of 1980 and 1981, he was with the Research in Distributed Processing Group, Department of Computer Science, UCLA. During the summer of 1998, he was an Erskine Fellow in the Department of Computer Science, University of Canterbury, Christchurch, New Zealand. He has coauthored over 300 journal and conference papers in the areas of communications and computer science, as well as two books, *Performance Models of Multiprocessor Systems* (Cambridge, MA: MIT Press) and *Modeling With Generalized Stochastic Petri Nets* (New York: Wiley). His current research interests are in the fields of performance evaluation of communication networks and their protocols.

Dr. Ajmone Marsan received the best paper award at the Third International Conference on Distributed Computing Systems, Miami, FL, in 1982. He participates in a number of editorial boards of international journals, including the IEEE/ACM TRANSACTIONS ON NETWORKING.



Andrea Bianco (M'98) was born in Torino, Italy, in 1962. He received the Dr.Ing. degree in electronics engineering and the Ph.D. degree in telecommunications engineering from Politecnico di Torino, Torino, Italy, in 1986 and 1993, respectively.

He is currently an Associate Professor in the Department of Electronics, Politecnico di Torino. From 1994 to 2001, he was an Assistant Professor with the Politecnico di Torino, first in the Production Systems Department, later in the Department of Electronics. In 1993, he was with Hewlett-Packard Laboratories,

Palo Alto, CA. In the summer of 1998, he was with the Department of Electronics, Stanford University, Stanford, CA. He has co-authored over 80 papers published in international journals and presented in leading international conferences in the area of telecommunication networks. His current research interests are in the fields of protocols for all-optical networks and switch architectures for high-speed networks.

Dr. Bianco has participated in the technical program committees of several conferences, including the IEEE Infocom 2000, IFIP ONDM (Optical Network Design and Modeling) 2002, and Networking 2002. He is the Technical Program Co-Chair of the High Performance Switching and Routing 2003 Workshop.



Paolo Giaccone (S'99) received the Dr.Ing. and Ph.D. degrees in telecommunications engineering from the Politecnico di Torino, Torino, Italy, in 1998 and 2001, respectively.

He currently holds a Postdoctoral position in the Department of Electronics, Politecnico di Torino. During the summer of 1998, he was with the High Speed Networks Research Group, Lucent Technology–Bell Labs, Holmdel, NJ. During 2000–2001, he was with the Department of Electrical Engineering, Stanford University, Stanford, CA. His

main area of interest is the design of scheduling policies for high-performance routers.



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

He is currently an Assistant Professor in the Department of Electronics, Politecnico di Torino. In 1995, he was with the Department of Computer Science, University of California at Los Angeles. In the summer of 1999, he was with the High Speed Networks Research Group, Lucent Technology–Bell

Labs, Holmdel, NJ, and in the summer of 2001, he was with the Department of Electrical Engineering, Stanford University, Stanford, CA. He has coauthored over 100 papers published in international journals and presented in leading international conferences. His areas of interest are all-optical networks, queueing theory, and scheduling policies for high-speed switches.



Fabio Neri (M'99) was born in Novara, Italy, in 1958. He received the Dr.Ing. and Ph.D. degrees in electrical engineering from the Politecnico di Torino, Torino, Italy, in 1981 and 1987, respectively.

He is currently a Full Professor in the Department of Electronics, Politecnico di Torino. His teaching duties include graduate-level courses on computer communication networks and on the performance evaluation of telecommunication systems. He leads a research group on optical networks at the Politecnico di Torino. He has recently been involved in several

European projects on WDM networks, including the ACTS project SONATA, which envisaged a single-layer optical transport network encompassing all concentration, distribution, transmission, switching and routing functions within a national network, and the IST project DAVID, which investigates the potential of optical packet switching in metropolitan and backbone networks. He coordinated the participation of his research group to several national Italian research projects. He has coauthored over 100 papers published in international journals and presented in leading international conferences. His research interests are in the fields of performance evaluation of communication networks, high-speed and all-optical networks, packet-switching architectures, discrete event simulation, and queueing theory.

Dr. Neri is a Member of the IEEE Communications Society. He has served on the boards of several IEEE conferences and journals, and participated in the Technical Program Committees of several conferences, including the IEEE Infocom and the IEEE Globecom. He was General Co-Chair of the 2001 IEEE Local and Metropolitan Area Networks Workshop and of the 2002 IFIP Working Conference on Optical Network Design and Modeling.