

Multihop Control Schemes in Switches With Reconfiguration Latency

Valentina Alaria, Andrea Bianco, Paolo Giaccone, Emilio Leonardi, and Fabio Neri

Abstract—Optical switching fabrics (OSFs) are receiving increasing attention in the design of high-speed packet switches, due to their excellent properties in terms of available bandwidth and reduced power consumption. However, for most optical devices, the latency needed to reconfigure input/output switch port connections may not be negligible with respect to the packet transmission time and can adversely affect switch performance, creating high delays and reduced throughput. We consider OSFs, and we propose a multihop approach to schedule packet transfers; i.e., packets are sent to the final destination port by exploiting transmission through intermediate ports. We show that the multihop approach is a promising technique to control the trade-off between delay and throughput, and it permits us to partly decouple the switch reconfiguration rate from the packet duration. We propose a general framework to solve the issue of multihop transmission in input-queued packet switches. Furthermore, we examine the multihop approach when the direct exchange of packets among ports is based on multidimensional regular topologies. We discuss which sequence of intermediate ports should be traversed to reach the final output port (i.e., the internal routing) and the switch time-scheduling problem (i.e., when a pair of ports can exchange packets). Performance is analyzed both analytically and by simulation.

Index Terms—Multihop approach; Optical switching fabrics.

I. INTRODUCTION

Input queued (IQ) packet switches using hybrid optical/electronic switching architectures are considered today a very promising approach to design routers able to reach aggregate bandwidths up to 100 Tbits/s and above [1]. This is due to intrinsic limi-

tations of electronics at high speed and to the insufficient maturity of photonic technology that prevents all-optical packet switching from being a practical alternative in designing routers today. The IQ architecture is the only viable solution to sustain high-speed, large-size switching configurations, because the switch can operate internally at the same speed of input/output ports. Electronic buffering is largely prevalent, since optical memories based on fiber delay lines are bulky, costly, complex to manage and control, and typically provide worse performance. Optical switching fabrics provide a switching complexity largely independent of the data rate, no significant constraints on the physical size of the switch and on the length of internal switch interconnections (while electrical backplanes and interconnects have severe distance limitations), no need for parallelism in high-bit-rate interconnections, and very good scalability of power requirements.

As a consequence, hybrid optical/electronic switching architectures are very interesting for future high-capacity routers and switches. They exploit a fully optical switching fabric (OSF), typically located in a different rack with respect to the switch line cards and electronic buffering. Packets arrive at the router through optical links and, after optical/electronic conversion, are processed and buffered electronically in the line cards. After an electronic/optical conversion, packets are sent over optical fibers to the optical switching fabric.

Regardless of the specific technology used to implement OSFs, such as microelectromechanical systems (MEMS) [2,3], bubble switches [4], polarized lead zirconium titanate waveguides [5], or broadcast-and-select or arrayed-waveguide-based networks with tunable devices [6], a common feature is that, whenever the OSF configuration (input/output port connections) is changed, a reconfiguration latency is required before communication takes place, due to the physical behaviors of devices. We assume in this paper that all the ports of the switch are blocked during the reconfiguration phase of the switching fabric; thus, no packet transmissions occur in the reconfiguration phase. The reconfiguration latency, depending on the technology used, can be as large as microseconds to

Manuscript received December 9, 2008; revised June 18, 2009; accepted June 19, 2009; published July 30, 2009 (Doc. ID 113644).

V. Alaria (e-mail: valaria@cisco.com) is with CISCO Systems, Inc., 70 West Tasman Drive, San Jose, California 95134, USA.

A. Bianco (e-mail: bianco@tlc.polito.it), P. Giaccone (e-mail: giaccone@tlc.polito.it), E. Leonardi (e-mail: leonardi@tlc.polito.it), and F. Neri (e-mail: neri@tlc.polito.it) are with Dipartimento di Elettronica, Politecnico di Torino, Italy.

Digital Object Identifier 10.1364/JOCN.1.000B40

milliseconds and is not negligible with respect to packet transmission times. Thus, the reconfiguration latency can severely affect switch performance.

A possible solution to overcome these technological constraints is to adapt the switch scheduling algorithm, whose task is to select the switching configuration of the optical device, to minimize the number of reconfigurations required for efficiently transferring a given traffic pattern. On the one hand, to obtain high throughput, the scheduling should keep the same switching configuration for a long time, so as to reduce the negative effect of inactivity periods due to the reconfiguration time. On the other hand, low access delays when the switch is lightly loaded imply that the switching configuration should change quickly, so as to permit full connectivity between all ports to be obtained in a short time interval. To solve this throughput-delay trade-off, clever scheduling algorithms should be devised.

To the best of our knowledge, few works have been proposed that consider the reconfiguration latency constraints when defining the scheduling problem in IQ switches (see [7–10]). All of these works assume that, when input i is connected to output j , only packets stored at input port i and destined to output port j are transferred through the switching fabric; i.e., all the packets cross the switching fabric once. We name this approach single-hop transmission. Using a single-hop scheduler, when N packets destined to different N outputs are queued at a given input, at least N switching fabric reconfigurations are required to provide full connectivity between all input/output pairs and to transfer the N packets. The worst access delay experienced in an empty switch is $NT + \delta$, where T is the reconfiguration latency [11] and δ is the packet transmission time.

This delay can be unacceptable for large switches. Consider the following scenario: an optical switch with $N=1024$ ports and a reconfiguration latency $T=1$ ms. Assume that the link speed is 10 Gbits/s and that internally the switch operates on fixed-size data units of 64 bytes. Thus, the fixed-size data-unit transmission time is $\delta=51.2$ ns. Hence, the worst access delay is $N=1$ s, which is obviously unacceptable for any realistic application. Although this value decreases for smaller switch sizes and can be reduced by improvements in optical technologies, a rather different approach should be pursued to make possible, in the near future, the exploitation of optical switching fabrics with nonnegligible switching times in large-size high-speed packet switches. Note that the continuous increase in data rates on transmission lines keeps reducing the packet duration at a pace that has been observed to be faster than the switching latency reduction due to technological improvements.

To overcome this problem, we exploit a multihop approach, which was first proposed in [12] and later examined in [11,13]. The main idea of the multihop switch control is (i) to configure the switching matrix once in a while, on a time scale significantly larger than packet transmission time and (ii) to recirculate packets among ports; i.e. a packet at input port i may reach its destination port j via successive transmissions through one (or more) intermediate ports. In the same scenario previously considered, the worst-case access delay for a multihop approach is reduced to a much smaller value, of the order of $N\delta$, a likely acceptable value for practical implementations. An interesting consequence of our approach is to decouple the switch control rate (which should be related to performance and quality of service) from the packet rate (which instead is related to transmission technologies). If a set of properly chosen switching configurations is periodically repeated, we end up defining a virtual topology among switch ports in time division.

In this paper we discuss the framework under which it is possible to design efficient switch control schemes that exploit the multihop approach. We present some simple analytical modeling to evaluate the performance of such approach. Then, we show how to tailor the design framework for the specific choice of the Manhattan virtual topology, which is shown to be appealing because of its good trade-off between scheduler complexity and performance. The main novel contributions of this paper with respect to the preliminary results published in [11,13] are that (i) a general framework for the multihop approach is presented, defining the steps needed to compute efficient multihop scheduling; (ii) a simple, yet efficient, approach exploiting a multidimensional regular virtual topology is proposed, its advantages are highlighted, and its robustness is assessed; and (iii) possible enhancements to deal with general traffic matrices are finally discussed.

II. MULTIHOP APPROACH TO SWITCH CONTROL

We consider an IQ switch with N ports, each running at the same line rate. We assume that the switch has a synchronous behavior: fixed size packets are switched on a time-slot basis. The single all-optical switching fabric behaves as a buffer-less crossbar. To avoid any internal speedup, at each time-slot at most one packet can be sent from any input port and can be received at any output port. A centralized switch control scheme defines the switching fabric configuration in each time-slot. A *feasible* switching configuration is equivalent to a matching in a bipartite graph, where left-hand-side nodes represent input ports and right-hand-side nodes output ports. An edge connects left-hand node i to right-hand node j if input port i is con-

nected to output port j in a switching configuration. The task of the centralized switch control scheme is to select the sequence of switching configurations (matchings) to satisfy a given traffic request. Since the switching fabric exploits optical devices, when the switching configuration changes, the switching fabric cannot be used for T time slots: this time interval is denoted switching fabric *reconfiguration latency* in the paper.

While the switching configurations can be computed one after the other based upon the most recent traffic estimation, we refer in this paper to the case in which the switch is operated over a set of properly chosen switching configurations that are periodically repeated. The set of periodically repeated switching configurations can be changed over time to adapt to non-stationary traffic scenarios, as briefly discussed at the end of the paper.

Input queues are used to solve contentions among packets contending for the same output. Input queues are organized according to a virtual output queue (VOQ) buffering scheme, with one FIFO queue for each output port at each input port, to achieve high throughput [14].

We say that all packets arriving at the same input port and directed to the same output port belong to the same *traffic flow*. We refer mainly to stationary traffic conditions. Nonstationary traffic scenarios are outside the scope of this paper and are briefly addressed in Section VII. Hence, we can describe the traffic arrivals through a $N \times N$ matrix $\Lambda = [\lambda_{ij}]$, in which λ_{ij} is the average traffic load of the flow from input port i to output port j , $0 \leq i, j < N$. We also assume that $\lambda_{ii} = 0$, i.e., we do not send across the switching fabric a packet that has arrived at input port i and is directed to output port i : in this case, we assume, as in [1], that packets follow a dedicated data path in the line card from the input interface to the output interface of the same port. We assume that all λ_{ij} are known; these values might be either measured in real time or derived from flow quality of service requirements. The traffic is uniform if $\lambda_{ij} = r/(N-1)$, for $\forall i, j$ being r the offered traffic load for a generic input.

The considered performance metrics are the long term average throughput, the access delay, and the queueing delay. The *access delay* is the time experienced by a packet from the arrival to the input port until delivery to the output port in an empty switch, i.e., not facing contention with other packets. It accounts for reconfiguration latencies and for waiting for the proper switching configuration, possibly repeated due to the multihop approach. The *queueing delay* is the delay experienced by a packet from the arrival to the input port until delivery to the output port due to contentions, solved via buffering, in the access

to the switching fabric. It does not include the delays captured by the access delay. The sum of access delay and queueing delay is called total packet delay or simply *packet delay*. Propagation delays are considered negligible. Note that the access delay is also a lower bound on the total delay experienced by a packet; in Section V we show that it is a good approximation of the delays experienced at low–medium loads.

A. Single Hop Versus Multihop

To transfer all the packets according to a classical single-hop approach, under a generic traffic matrix with nonzero entries outside its diagonal, full connectivity among switching ports is necessary (i.e., each input port must be connected to every output port). As a consequence, the switch control must use at least N matchings. This may create unacceptable access delays for large reconfiguration latencies, even in a lightly loaded switch, as previously explained. In terms of throughput, a reduction of the maximum acceptable load is caused by inactivity during reconfigurations.

In contrast, according to the multihop approach, even a partial connectivity among ports may guarantee the transfer of packets for all port pairs through the switching fabric. Using a reduced set of switching configurations, input port i is directly connected (in single hop) by the scheduler only to a subset of other ports to which it can directly transmit packets. Packets directed to any port j that is not directly connected to port i reach the destination in a multihop fashion, i.e., through some intermediate ports. Line cards must be able to send packets back to the switching fabric if needed; i.e., they are assumed to provide a data path from output ports back to input ports.

As the number of switching configurations can be much smaller than N , access delays can be drastically reduced. However, as a side effect of the multihop approach, the same packet may require multiple transmissions across the switching fabric, thus increasing the effective traffic load to input ports and hence the queueing delay, thus reducing the switch throughput. Clearly, throughput–delay trade-offs arise in the switch design.

The multihop scheduler is more complex than the single-hop one, since it has to define the sequence of intermediate ports (i.e., hops) followed by each flow and when the flow has to be sent across the switching fabric for each hop.

B. Multihop Switch Control Scheme

The multihop switch control scheme identifies a sequence of switching configurations that are cyclically

repeated to best serve a given traffic matrix. We describe the multihop switch control problem as a sequence of steps.

In the first step, a virtual topology is defined as an interconnection pattern among switch ports. Given the virtual topology, a routing scheme defines the paths followed in the virtual topology by each possible flow. In the second step, a set of covering matchings is chosen to permit packet transmission among ports on all the logical links of the virtual topology. Finally, a sequence among the identified matchings is selected to transfer the packets across the switching fabric through the multihop approach. We explain in detail the proposed approach in the next subsections, using Fig. 1 as a specific example.

C. Virtual Topology and Routing

A virtual interconnection topology \hat{G}_σ is overlaid on the set of switch ports. We use the term “node” to refer to the topology and the term “port” to refer to the switch. Each node in \hat{G}_σ corresponds to a switch port. Let σ be the correspondence between the topology nodes and the switch ports; let $\sigma(i)$ be the node associated with port i . In the simplest case, $\sigma(i)=i$. The chosen topology is a directed graph (di-graph) that must be strongly connected, i.e., there must exist a directed path between any pair of nodes, to permit packet transfers between any input/output port pair. Hence, each edge of \hat{G}_σ is a directed edge (di-edge). Figure 1 shows an example of a virtual topology that can be overlaid on a 6×6 switch. Note that no direct (i.e., single-hop) packet transfer is enabled, e.g., between node 0 and node 2.

Given a virtual topology, a deterministic routing algorithm computes the shortest paths, the path length being measured in terms of number of hops, to connect any pair of nodes. When more than one path may connect two nodes, we force the algorithm always to

choose one path, to prevent missequenced delivery of packets belonging to the same flow.

Note that the choice of the topology is somehow arbitrary, although criteria to optimize the virtual topology according to traffic needs can be defined, as discussed below.

D. Covering Matchings Selection

Given a particular virtual topology, the scheduler should select a set of matchings to switch packets in a multihop fashion, following the routing paths chosen by the routing algorithm. To simplify such computation, the algorithm starts finding a minimal set of matchings, called *covering matchings*, that cover all edges of the virtual topology. Note that this procedure considers just the topology and not the routing defined on it.

In more detail, we define a *di-matching* \hat{M} as a subset of di-edges of \hat{G}_σ such that each node is connected to at most one incoming and one outgoing di-edge. The algorithm computes a minimal set \mathcal{C} of di-matchings on \hat{G}_σ , such that the union of all the di-edges in \mathcal{C} covers all di-edges in \hat{G}_σ . This condition guarantees that each di-edge in the virtual topology is covered by at least one di-matching. Hence, any routing path can be mapped into a sequence of di-edges belonging to di-matchings. Equivalently, throughout any sequence of all covering di-matchings, it is possible to ensure full connectivity among the nodes in \hat{G}_σ .

Finding a minimal set of di-matchings in regular topologies can be simple (as we will show in Section IV). For general topologies, the set can be computed by exploiting the classical algorithm for Birkhoff–von Neumann (BvN) decomposition [15]. Indeed, we can consider the adjacency matrix $A(\hat{G}_\sigma)=[A_{ij}]$ of \hat{G}_σ , i.e., the $N \times N$ binary matrix with $A_{ij}=1$ iff a di-edge connects

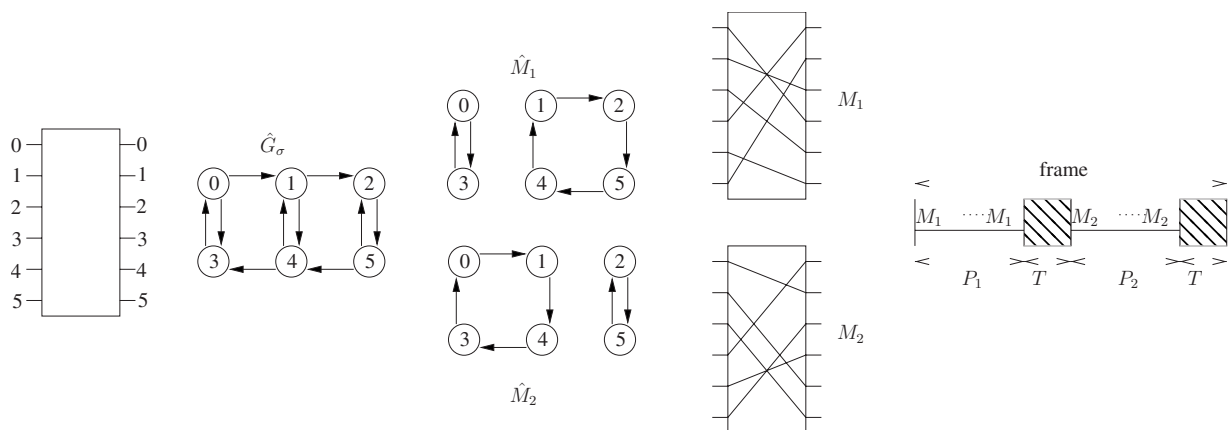


Fig. 1. Example of multihop scheduling for a 6×6 switch. \hat{G}_σ is covered by two di-matchings \hat{M}_1 and \hat{M}_2 (i.e., $\mathcal{C}=\{\hat{M}_1, \hat{M}_2\}$), corresponding to the switching configurations M_1 and M_2 . The final frame will be composed by M_1 for P_1 time slots and by M_2 for P_2 time slots.

node i to j in \hat{G}_σ . We wish to decompose matrix A into a sum of permutation matrices (i.e., binary matrices with only one element set to 1 in each row and in each column, all other elements being set to 0). Note that a permutation matrix corresponds to a di-matching by construction. Here, with a slight abuse of notation, we let \hat{M} denote both the di-matching and its corresponding permutation matrix. The BvN decomposition of $A(\hat{G}_\sigma)$ into the set $\{\hat{M}_k\}$ guarantees that

$$A(\hat{G}_\sigma) \leq \sum_{k=1}^{m_A} \hat{M}_k,$$

with a minimum number of di-matchings equal to m_A , which is the maximum among the column and row sums in $A(\hat{G}_\sigma)$ or, equivalently, the maximum among the in degrees and out degrees of all the nodes in \hat{G}_σ . In the example of Fig. 1,

$$A(\hat{G}_\sigma) = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix},$$

where $m_A=2$ can be decomposed as $A(\hat{G}_\sigma) \leq \hat{M}_1 + \hat{M}_2$ and $\mathcal{C} = \{\hat{M}_1, \hat{M}_2\}$ with

$$\hat{M}_1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\hat{M}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

E. Frame Scheduling Computation

Observe that a di-edge from node $\sigma(i)$ to $\sigma(j)$ corresponds to connecting port i to port j in the switching fabric. The set of all the di-edges belonging to a di-matching corresponds to a feasible matching for the

switching fabric. Thus, the set \mathcal{C} of all possible matchings that cover all paths selected by the routing algorithm on the virtual topology has been defined.

Now the switch control scheme has to determine the temporal sequence and duration of the matchings selected to control the switching fabric. Since we refer mainly to stationary traffic, we adopt a frame scheduling approach [16,17], in which this sequence is fixed and periodic. The frame is organized in a sequence of $|\mathcal{C}|$ epochs; during the k th epoch, a matching in \mathcal{C} is adopted for P_k time slots. A reconfiguration latency has to be paid between epochs. Note that scaling all P_k by a factor $\alpha > 1$ increases the throughput (as the scaling does not apply to T), but also increases delays. P_k must therefore be chosen as a trade-off between delays (due to epoch durations) and throughput (due to the frequency of blackouts during switch reconfigurations). Hence, the overall frame duration F , measured in time slots, is simply

$$F = \sum_{k=1}^{|\mathcal{C}|} (P_k + T) = |\mathcal{C}|T + \sum_{k=1}^{|\mathcal{C}|} P_k.$$

In the example of Fig. 1, the frame duration is $F = P_1 + P_2 + 2T$ time slots.

F. Complexity of Multihop Switch Control

We now discuss the complexity of the multihop switch control framework. Under stationary traffic, assuming that the virtual topology is given, the routing computation is simple and can also be done off line. Finding the set of covering matchings in general requires the solution of the BvN decomposition [15], whose complexity is $O(N^{4.5})$. However, this computation can be done off line. Furthermore, for some regular topologies, as shown in Section IV, finding the set of covering matchings can be solved by following a straightforward procedure.

When the frame scheduling is considered, if the traffic is uniform and the routing policy is balancing the load across all the di-edges of the topology, we can assume a *constant* epoch duration P for all matchings: $P = P_k$ for $k = 1, \dots, |\mathcal{C}|$. This means that the frame scheduling phase complexity is $O(1)$, since all matching sequences can be computed off line.

In general, when the traffic is not uniform and/or the routing policy is not balancing the load across the topology, finding the optimal frame may require a significant computational effort because of the large number of degrees of freedom in the design: choice of the topology, choice of topology mapping, choice of the routing policy, choice of the covering matchings, choice of epoch duration. Indeed, as shown in Section III, finding the optimal mapping of the switch ports onto the virtual topology, i.e. a mapping that maximizes

the throughput given a traffic matrix Λ , is equivalent to minimize the mean internodal distance of the topology, weighting the distance with the traffic load. This is equivalent to the node placement problem in a WDM network, which has been proved to be NP-hard [18] for even the simplest linear bus regular topology.

To simplify the problem, in Section IV, we investigate the multihop switch control framework, using only regular virtual topologies. Indeed, under the simplifying assumption of uniform traffic, finding a proper routing algorithm that balances traffic is easy. For such topologies, in Subsection VI.A we will discuss some polynomial time algorithms to improve switch performance by properly mapping ports to nodes also under nonuniform traffic. Their effectiveness is studied by simulation. We will also show that suboptimal choices of topology, node mapping, and routing provide good performance under general traffic patterns.

As a final comment, if the traffic is not stationary, the multihop switch control algorithms should be re-run any time the traffic changes significantly to optimize performance, leading to changes in the scheduling and/or in the routing over the virtual topology, and/or in the virtual topology itself. As a side effect, any change in the routing path may induce out-of-order packet delivery. This problem can be neglected (since it occurs for a short transient period) or addressed by adding more complexity either in the queueing structure or in the switch control scheme.

G. Queueing Architecture

In our design, for stationary traffic, packets of the same flow follow the same precomputed path. Let us assume that

$$\mathcal{P} = (\sigma(i), \sigma(k_1), \dots, \sigma(k_{H-1}), \sigma(j))$$

is an H -hop path followed by the flow from input port i to output port j through $H-1$ intermediate nodes. During the first hop, a packet of the flow from i to j is sent from port i to port k_1 , then from port k_1 to port k_2 , and so on, till reaching the destination port j . The packet will be stored in this sequence of VOQs:

$$\text{VOQ}_{ik_1} \rightarrow \text{VOQ}_{k_1k_2} \rightarrow \dots \rightarrow \text{VOQ}_{k_{H-2}j},$$

where VOQ_{ij} is the VOQ at input port i storing packets destined to output port j . Hence, the classical bijective correspondence between VOQs and input/output port pairs is lost. Packets belonging to different flows can be stored in the same VOQ, in FIFO order. At the same time, only the VOQs corresponding to the di-edges in \hat{G}_σ are used (since any path should follow these di-edges). This permits a large reduction in the complexity of the queueing structure. Indeed, it is not necessary to have one separate queue for each destination port in each input port, i.e., a total of N^2

queues as in the classical IQ switch. At each port, a number of VOQs equal to the node out degree in the virtual topology is sufficient. Referring to Fig. 1, only two queues are needed for the input ports in the set $\{0, 1, 4, 5\}$ and one queue for ports in $\{2, 3\}$. Overall, only 10 FIFO queues are needed, instead of 36 queues needed in the classical VOQ system for single hop.

III. PERFORMANCE OF MULTIHOP APPROACH

The access delay experienced by packets belonging to the flow from input i to output j depends on the corresponding path in \hat{G}_σ and on the matching sequence defined in the frame. Given an edge $e = (\sigma(i), \sigma(j))$ in \hat{G}_σ , we define $f_{\text{next}}(t, e)$ as the first time slot after t during which a matching in the frame sequence connects port i to port j . If $\mathcal{P} = (e_1, \dots, e_H)$ is the sequence of H di-edges (hops) in a path, then the access delay experienced by a packet arrived at time slot t_0 can be evaluated as $t_H - t_0$ where

$$t_h = f_{\text{next}}(t_{h-1}, e_h) \quad \forall h = 1, \dots, H.$$

An upper bound on the access delay, denoted the *worst-case access delay*, can be simply evaluated as $|C|H(P+T)$ time slots, since a packet may always wait for a new frame before being served. Note that this bound can be tight for specific frame sequences. Since we could not obtain a simple model for the queueing delay, we omit its analytical investigation, and we defer its study through simulations to Section V.

The throughput depends on the traffic matrix and is affected mainly by two factors: multihop transmissions and reconfiguration latency. Indeed, throughput is affected by the waste of bandwidth due to multiple transmissions of the same packet across the switching fabric. Under uniform traffic, this effect can be evaluated by considering the average internodal distance $E[d^\sigma]$ induced by the routing policy in \hat{G}_σ , which is the average number of hops experienced by a packet:

$$E[d^\sigma] = \frac{1}{N^2} \sum_{i,j} d_{ij}^\sigma, \quad (1)$$

where d_{ij}^σ is the distance between port i and port j , in number of hops.

In addition, the throughput performance is affected by the reconfiguration latency. We can measure throughput efficiency by η , the *holding factor* of the switching fabric, defined as the fraction of time in which the switching fabric is available for packet transfer. Under uniform traffic, by construction, $\eta = P/(P+T)$. To achieve high utilization, the epoch duration should be much larger than the reconfiguration latency. As a consequence, P is chosen as a function of T to obtain a high throughput efficiency.

TABLE I
PARAMETERS CONSIDERED FOR THE CONSIDERED SWITCHING ARCHITECTURE

Parameter	Symbol	Value
Reconfiguration latency	T	0.12 ms
Scheduling period	P	1.2 ms
I/O link rate		10 Gbits/s
Packet size		1500 bytes
Packet transmission time	δ	1.2 μ s
VOQ size		400 packets

Appendix A shows the following upper bound on the maximum throughput $\hat{\lambda}$ achievable when the traffic is uniform:

$$\hat{\lambda} = \frac{1}{E[d^\sigma]} \frac{P}{P + T}.$$

A. Throughput Improvement Due to Speedup

To compensate for throughput reductions due to multihop transmissions and reconfiguration latencies, it is possible to exploit two types of internal bandwidth speedup S . In the case of *temporal speedup*, the switching fabric runs S times faster than the line rate. During each epoch up to SP packets are served at each port. Note that the total frame duration is not modified, and hence the access delay does not vary. In the case of *spatial speedup*, S switching fabrics run in parallel, configured according to different covering matchings. Each port must be able to transmit packets simultaneously on the S fabrics. The frame duration is reduced by a factor S , also affecting the access delay. In addition, when $S=|C|$, there are enough switching planes to cover all the topology without reconfiguration, and the reconfiguration latency is null: $T=0$. In contrast, for single hop, the spatial speedup required to neutralize the reconfiguration costs is N , because one separate switching fabric must be available for the N switching configurations used in the single-hop approach.

If $\hat{\lambda}$ is the maximum normalized throughput achievable without any speedup (i.e., with unitary speedup), with a speedup S the maximum throughput becomes simply $\min\{1, S\hat{\lambda}\}$. From another perspective, the minimum speedup needed to obtain 100% throughput is $1/\hat{\lambda}$.

B. Topology Comparison Under Uniform Traffic

Many virtual interconnection topologies can be mapped onto the switching ports. Previous work [12] has considered ring topologies, whereas [11] has shown theoretically the advantages of multidimen-

sional topologies. The ranking of these topologies is somehow arbitrary, because each topology shows different throughput and delay trade-offs, depending on system parameters. Hence, we fix some parameters according to Table I, which reflects reasonable design choices and constraints. T is derived by technological constraints related to MEMS reconfiguration latencies (see, e.g., [2,10]), and P is set to guarantee a switching efficiency $\eta \approx 90\%$, corresponding to 10% throughput reduction in the single-hop case. With the packet size equal to the Ethernet maximum transmission unit, the slot duration is 1.2 μ s, corresponding to $T=100$ time slots and $P=1000$ time slots.

Assuming uniform traffic and using the values of Table I, Fig. 2 shows the minimum speedup S necessary to achieve 100% throughput under uniform traffic for the following well-known topologies [19,20]: single hop (SH), unidirectional ring (UR), bidirectional ring (RI), x -dimensional Manhattan (MH x), x -dimensional shuffle (SF x), and x -dimensional Kautz (KN x), for $x=2,3,4$. We observe that the speedup scales with the number of nodes according to the average distance (i.e., average hop count). Hence the topologies with average distances proportional to the logarithm of the number of nodes (shuffle and Kautz) exhibit the best scaling behavior. Note, however, that for ports $N \leq 1024$, the values of S are not dramatically different.

Figure 3 shows instead the worst-case access delay at low loads for the same topologies. Delays depend on the connectivity degree for nodes in the virtual topologies, and the topologies with the least average number of node neighbors (2D Manhattan) exhibit the lowest delays.

Appendices B and C show how to evaluate the theoretical performance for the specific cases of single-hop and Manhattan topologies.

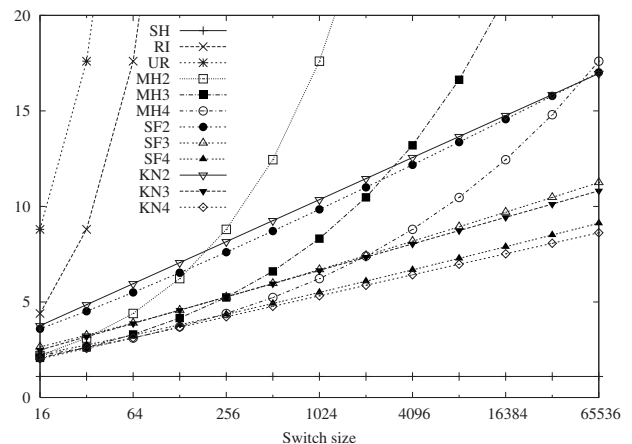


Fig. 2. Minimum speedup S necessary to achieve 100% throughput in uniform traffic as a function of the port number.

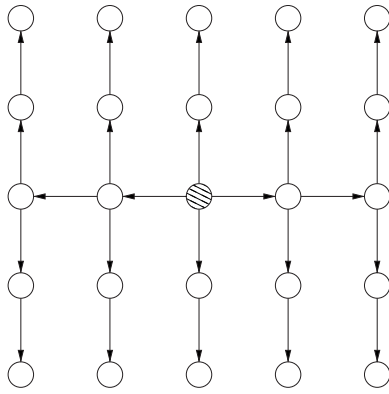


Fig. 4. Routing paths, according to PDR, for the central node of a 5×5 Manhattan network corresponding to a 25×25 switch; at most two directions are needed to reach any destination node.

IV. MULTIHOP FOR MANHATTAN TOPOLOGIES

Among all the topologies, in this work we analyze in some detail multidimensional bidirectional Manhattan topologies. Indeed, they permit good routing strategies to be easily defined, and they provide a good delay-throughput trade-off, as shown in Figs. 2 and 3.

We start by considering the mapping of a 2D Manhattan topology of N nodes, organized in M rows and M columns where $N=M^2$, into an $N \times N$ switch. In Section VI we will discuss the mapping when \sqrt{N} is not an integer. Each input/output port of the switch corresponds to a node of the virtual topology, according to the following bijective mapping: node (i, j) , located in row i and column j , with $0 \leq i, j \leq (M-1)$, corresponds to port $k = M \times i + j$, $0 \leq k \leq N-1$.

Given that we rely on a regular topology with node out degree 4, port (i, j) can directly (i.e., in single-hop) reach four ports, one for each direction: $(i, |j+1|_M)$, $(i, |j-1|_M)$, $(|i+1|_M, j)$, and $(|i-1|_M, j)$ ($|\cdot|_n$ denotes the modulo- n operator, i.e., the remainder of the division by n). All other destinations must be reached in a mul-

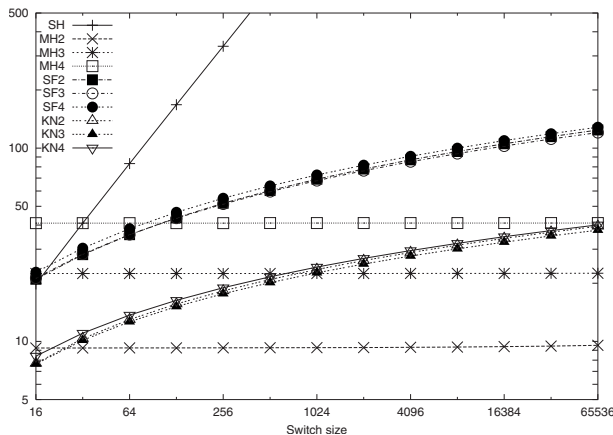


Fig. 3. Worst-case access delay W in milliseconds for uniform traffic versus the number of ports.

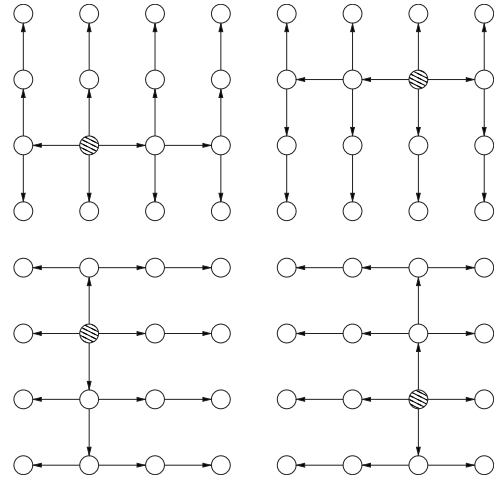


Fig. 5. Routing paths, according to the balanced version PDR, for four nodes of a topology with M even.

ti-hop fashion. Only four di-matchings are needed to cover the topology, independently of N , one for each direction. Each corresponding matching is used for the first P time slots of a scheduling epoch; the total frame duration is $4P + 4T$.

This example can be extended to multidimensional Manhattan topologies of generic dimension c , with degree $2c$ at each node; in this case, each side of the corresponding hypercube comprises $\sqrt[c]{N}$ nodes, and the frame duration is $2c(P + T)$. Note that a bidirectional ring topology is obtained by setting $c = 1$.

Many routing algorithms on a Manhattan network can be devised. In our work we consider the following routing scheme, called “privileged directions routing” (PDR), which described for a bidimensional Manhattan network for simplicity but can easily be extended to multidimensional networks. Among all the possible shortest paths from a node (i, j) to a node (k, l) , consider the path through node (i, l) , following (possibly) first the row direction and then (possibly) the column direction. Figure 4 shows the minimum distance routing paths followed by the central node of a 5×5 Manhattan topology to reach all the other nodes. Note that the PDR scheme has the following properties: (i) unique routing path between any pairs of nodes, (ii) easy computation, and (iii) symmetric distances: $d_{ij}^r = d_{ji}^r$. When M (i.e., the number of rows and columns) is odd, PDR also guarantees that the load across all the edges is balanced under uniform traffic. When M is even, PDR does not balance this load. To address this problem, it is possible to consider a balanced version of PDR, in which half of the nodes follow the path along first row and then column directions, and half of them follow the path along first column and then row directions. Figure 5 shows an example of directions followed by four nodes in a 4×4 topology.

Packets could be stored according to a classical VOQ

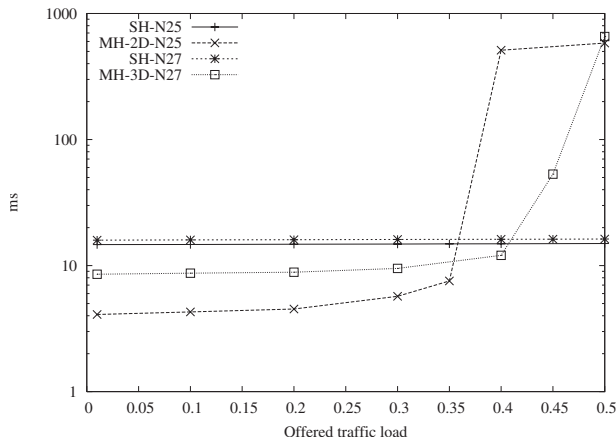


Fig. 6. Average delay under uniform traffic for 2D or 3D multihop topologies (2D for $N=25$ and 3D for $N=27$).

scheme: in the example of Fig. 4, the rightward direction allows the central node to reach ten destination nodes. Hence, during the matching corresponding to the rightward direction the packets stored in ten VOQs are served. However, thanks to the considerations in Subsection II.G, it is possible to use only $2c$ queues per input, one for each possible direction, instead of N VOQs at each input port, with evident benefits for the scalability of the queueing system in large switches.

V. SIMULATIVE PERFORMANCE STUDY

Table I shows the parameters considered in the switch under study. We start from the performance under uniform traffic and then, keeping the same frame scheduling plan, under nonuniform traffic. Section VI will discuss how to optimize the design for nonuniform traffic. Simulation runs were executed until the estimate of the average packet delay reached with probability 0.95 a relative width of the confidence interval less than 3%. The estimation of the confidence interval width is obtained with a batch means approach.

A. Performance Under Uniform Traffic

Figure 6 shows the average packet delay (comprising access and queuing delays) with respect to the offered load r for single-hop (SH) and two scenarios:

- multihop Manhattan (MH) 2D topology, with $N=25$;
- multihop Manhattan 3D topology, with $N=27$.

Both scenarios refer to small-size switches. Table II shows the maximum throughput and the worst-case access delay estimated by the theoretical models of Appendices B and C. In Fig. 6, the single-hop topology shows almost a flat delay with respect to r , since the load is much lower than $\hat{\lambda}_{SH}$ and the performance is

TABLE II
THEORETICAL PERFORMANCE UNDER UNIFORM TRAFFIC FOR SMALL SWITCHES

Algorithm	$\hat{\lambda}$	Worst-Case Access Delay (ms)
SH-N25	0.91	33.0
MH-2D-N25	0.36	7.9
SH-N27	0.91	35.6
MH-3D-N27	0.40	19.8

dominated by the frame scheduling; the average delay is well bounded by the worst-case access delay. The delays for multihop schemes, both 2D and 3D, are well estimated by the theoretical models when $r < \hat{\lambda}$. When the offered load becomes nonadmissible, the delays are dominated by the queueing process inside the finite-size buffers. This graph confirms our expectations: when increasing the dimension of the topology (from 2D to 3D), the maximum throughput increases at the expenses of a larger access delay.

The throughput reduction due to multihop schemes can be compensated by some speedup. Here, we consider explicitly the effect of temporal speedup. Figure 7 shows the average delay for a large switch ($N=121$) for single-hop and 2D multihop schemes, in the case of variable speedup $S=1, 3, 5$. Also in these scenarios, the average delay for low load and the maximum throughput are well bounded by the approximated models of Appendices B and C, as reported in Table III.

It is important to observe that, because of the scheduling approach based on frames, the average delay for the single-hop case is not affected by the speedup; of course, single hop can benefit from a very small speedup ($1/\eta=1.1$) to achieve the maximum throughput. In contrast, throughput for the multihop cases is affected by speedup: a speedup S is able to increase the maximum achievable throughput by a factor S .

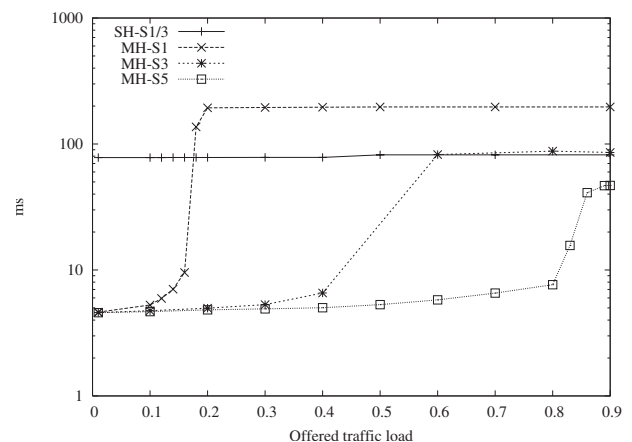


Fig. 7. Average delay under uniform traffic, for switch size $N=121$, 2D multihop topologies, and variable speedup S .

TABLE III
THEORETICAL PERFORMANCE UNDER UNIFORM TRAFFIC FOR
LARGE SWITCHES ($N=121$)

Algorithm	$\hat{\lambda}$	Worst-Case Access Delay (ms)
SH-S1	0.91	160
SH-S3	1.00	160
SH-S5	1.00	160
MH-2D-S1	0.17	7.9
MH-2D-S3	0.50	7.9
MH-2D-S5	0.83	7.9

The delay for low load is independent of the speedup, as confirmed by our arguments to estimate the worst-case access delay.

B. Performance Under Nonuniform Traffic

To test the robustness of the multihop switch control, we considered the performance achievable under well known nonuniform traffic matrices, using the same frame sequence computed by assuming uniform traffic.

The conclusions drawn in the previous section for uniform traffic hold for many nonuniform traffic matrices. Indeed, in our simulations, we observed the same qualitative behavior for different switch sizes, under the following traffic matrices:

- linear (lin)-diagonal traffic: each diagonal is loaded linearly with its position from left to right; in formulas, $\lambda_{ij} = \lceil 2r/N(N-1) \rceil |j-i|_N$, for any $j \neq i$,
- log-diagonal traffic: each diagonal is loaded twice the load of its left diagonal; in formulas, $\lambda_{ij} = \lceil r/(2^{N-1}-1) \rceil 2^{|j-i|_N}$, for any $j \neq i$.

For the sake of space, we discuss only the results for lin-diagonal traffic. Figure 8 shows the effects of 2D and 3D multihop schemes under lin-diagonal traffic; a higher dimension of the topology can lead to higher

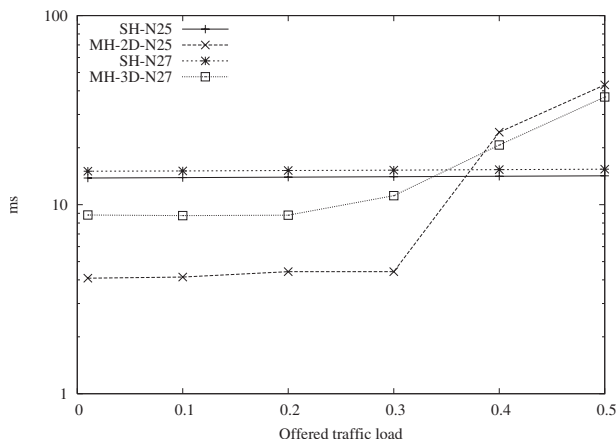


Fig. 8. Average delay under lin-diagonal traffic, for 2D and 3D multihop topologies and small switches.

throughput but at the expenses of a larger access delay (which is the same for uniform traffic).

Figure 9 shows the speedup effect under lin-diagonal traffic. The same qualitative behavior found for uniform traffic holds.

C. Fairness Issues

When a packet experiences a multihop transfer across the switching fabric, it contends many times for the access. Hence, under multihop operation we can expect better performance for packets experiencing fewer hops. This implies unfair performance between packets entering the same input port and directed to different output ports.

Figures 10 and 11 investigate the unfairness issue, showing, for $N=27$, the throughput and the average delay achievable under uniform traffic for different classes of packets, each one associated with a particular distance (in terms of hops) from its input port to its output port. Table II estimates that the 3D multihop scheme achieves 0.40 throughput and 9.9 ms access delay, averaging over all possible distances. This is in accordance with the curves labeled “Average” in both figures.

Packets with lower distance experience lower average delay. At the same time, they experience higher throughput when the switch is overloaded; indeed, when considering many packet flows entering a work conserving queueing system, throughput may be unfair between flows only when the system capacity is exceeded. Note that sustained overload conditions are typically avoided in real networks by flow and congestion control schemes (e.g., TCP in the Internet).

VI. TOPOLOGY OPTIMIZATION

We investigate in this section the effect of some adversarial nonuniform traffic matrix on a switch de-

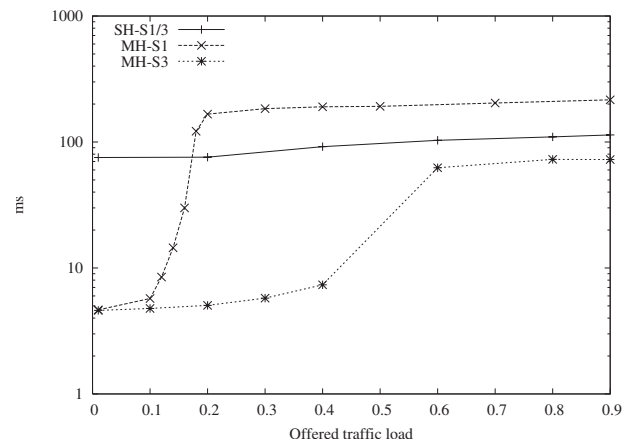


Fig. 9. Average delay under lin-diagonal traffic, for 2D topology, variable speedup S , and switch size $N=121$.

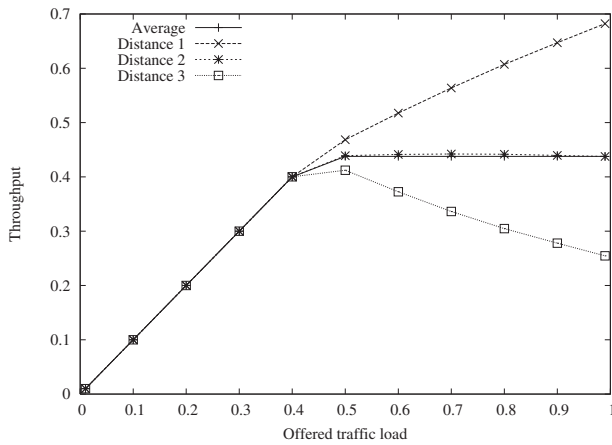


Fig. 10. Throughput under uniform traffic on a 3D multihop topology with $N=27$.

signed for uniform traffic, highlighting the need for topology adaptation to the traffic matrix.

When the traffic is nonuniform, the performance of the system can be affected by the actual correspondence between the switch ports and the topology nodes. Intuitively, two ports exchanging a large amount of traffic should be placed in topologically close nodes, whereas two ports exchanging a small amount of traffic can be placed in nodes that are far.

To compute an optimal port placement, it is necessary to define a cost function to be minimized that is able to capture the dependency between the performance and the topology allocation. Since performance is affected both by the congestion across each link of the topology and by the number of hops to traverse for each traffic flow, we can use the traffic-weighted inter-nodal distance as the cost function:

$$\min_{\sigma \in \Omega} \sum_{i,j} \lambda_{ij} d_{ij}^{\sigma} \tag{2}$$

with Ω being the set of all possible mappings between ports and nodes of the Manhattan topology. Note that $|\Omega|=N!$.

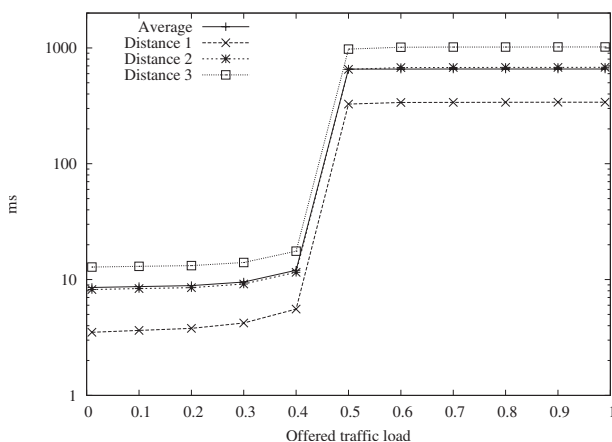


Fig. 11. Average delay under uniform traffic on a 3D multihop topology with $N=27$.

Finally, the routing and the scheduling are computed following the standard steps developed for uniform traffic.

The placement algorithm also permits us to solve the multihop frame-scheduling design when the Manhattan topology is overlaid on an $N \times N$ switch, where N is not the square of an integer number. In this case, it is possible to consider a slightly larger switch of size $N' \times N'$ mapped onto a $M' \times M'$ Manhattan topology, with $M' = \lceil \sqrt{N} \rceil$, where $\lceil x \rceil$ is the smallest integer $\geq x$. Then, it is sufficient to set $\lambda_{ij} = 0$ for each additional $N' - N$ input and $N' - N$ output ports that have been added artificially and to apply the topology placement algorithm for such nonuniform traffic pattern.

A. Algorithms for Topology Placement

To efficiently compute a suboptimal node placement for a c -dimensional Manhattan topology, we propose the following three heuristics.

1) *Minimum Link Traffic*: The greedy minimum link traffic (MLT) algorithm tries to minimize the maximum traffic flowing on any topology edge, named the *network congestion index*. At procedure startup, all ports are set as unmapped, and all nodes are set as available. For each pair of ports i and j , the bidirectional traffic flowing between them is computed: $m_{ij} = \lambda_{ij} + \lambda_{ji}$. Furthermore, an (arbitrary) ordering relationship among topology nodes is established by computing a Hamiltonian path on the topology (i.e., a path visiting once all nodes). This ordering operation is simple on the Manhattan topology.

At each step, among all the port pairs (i, j) , such that port j is unmapped, the algorithm selects the pair (i_0, j_0) , which corresponds to the largest value $m_{i_0 j_0}$.

If port i_0 is also unmapped, i_0 is mapped to the first available node, according to the defined node ordering relationship. The node is marked as unavailable. If there are available neighboring nodes of the node to which port i_0 was mapped, the algorithm selects among them node n_0 that makes it possible to minimize the network congestion index created on the current topology by unavailable nodes (i.e., nodes previously associated with switch ports). If no neighbors of the node associated with port i_0 are available, the first available node n_0 is selected following the ordering relationship previously defined. Port j_0 is mapped to node n_0 and node n_0 is marked as unavailable.

When all the ports have been placed, the algorithm ends.

2) *Swap Neighbors*: The iterative swap neighbors (SN) algorithm tries to minimize the cost function in Eq. (2) by swapping the placement of node pairs. The procedure starts operating on a generic, randomly cre-

ated node placement σ . At each iteration, the algorithm finds the pair of ports i and j [respectively mapped to nodes $\sigma(i)$ and $\sigma(j)$] such that $d_{ij}^\sigma m_{ij}$ is maximized. The algorithm first considers port i . Among all the $2c$ neighbors of node $\sigma(i)$, the algorithm selects the node $\sigma(k)$ that corresponds to the minimum value of the exchanged traffic m_{ik} . The algorithm swaps the node $\sigma(j)$ with node $\sigma(k)$ if the following condition is satisfied:

$$m_{ik} < m_{ij}. \quad (3)$$

The same procedure is then applied to port j , considering the possible swap between one of its neighbors and i . The algorithm ends either when no swapping satisfying the previous condition exists or when a maximum number of iterations is reached.

To understand the reason why it is necessary to check condition (3), let us consider nodes $\sigma(i)$, $\sigma(j)$, and $\sigma(k)$ and their contribution to cost function (2), equal to

$$\delta^\sigma = m_{ij}d_{ij}^\sigma + m_{ik} + m_{kj}d_{kj}^\sigma, \quad (4)$$

since $d_{ik}^\sigma = 1$ by construction. We now evaluate the contribution if nodes $\sigma(j)$ and $\sigma(k)$ were swapped, leading to a new topology definition through placement σ' ; i.e., $\sigma'(j) = \sigma(k)$ and $\sigma'(k) = \sigma(j)$:

$$\delta^{\sigma'} = m_{ij} + m_{ik}d_{ik}^{\sigma'} + m_{kj}d_{kj}^{\sigma'}. \quad (5)$$

Now observe that by swapping j and k , $d_{ij}^\sigma = d_{ik}^{\sigma'}$ and, thanks to the fact that the distances are symmetric, $d_{kj}^\sigma = d_{kj}^{\sigma'}$. By combining Eqs. (4) and (5), the swap results are favorable when $\delta^{\sigma'} < \delta^\sigma$. As a consequence,

$$m_{ij} + m_{ik}d_{ik}^{\sigma'} < m_{ij}d_{ij}^\sigma + m_{ik},$$

$$m_{ik}(d_{ik}^{\sigma'} - 1) < m_{ij}(d_{ij}^\sigma - 1),$$

which implies Eq. (3).

3) Random placement: In random placement (RND) each switch port is mapped randomly to one node of the topology. This algorithm is useful mainly for performance comparison and can be used as a reference case for a node placement algorithm oblivious of the traffic matrix.

B. Performance Evaluation

To compare the performance of the above described placement algorithms, we consider a worst-case scenario, in which it is easy to define the optimal solution. This traffic scenario is called “localized traffic” and is built according to the following rules. Map each switch port randomly into a topology node. Let σ_{OPT} be the corresponding node placement. Set the traffic load flowing from each node to each of the $2c$ neighbors

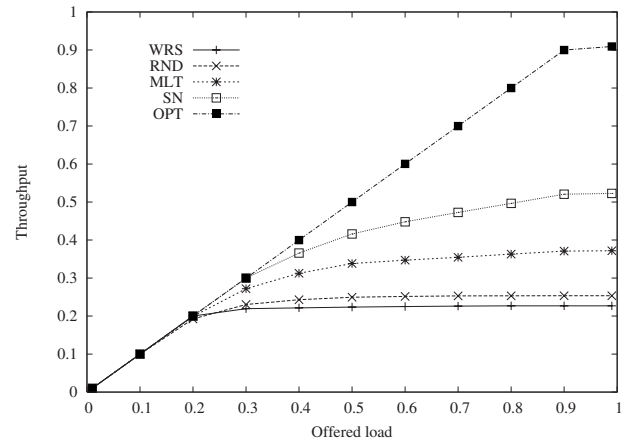


Fig. 12. Throughput for different node placement algorithms.

equal to $r/2c$: the resulting load entering and leaving each node (switch port) is r . An example of a neighbors traffic matrix for $N=9$ is given here:

$$\Lambda = \frac{r}{4} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Under localized traffic, the optimal placement is given by σ_{OPT} , which minimizes the cost function, because traffic is exchanged only between neighbor nodes. Note that, according to this optimal port placement, each packet is transferred at most once across the switching fabric, and the multihop scheme works as a single-hop scheme. We can also define the worst-case placement for localized traffic, called σ_{WRS} , such that all nodes exchanging traffic are located at the maximum possible distance.

Figures 12 and 13 show the throughput and the average delay obtained by the different node placement algorithms under localized traffic for a switch size $N=25$. “OPT” and “WRS” refer to multihop with σ_{OPT} and σ_{WRS} node placement. In the case of σ_{OPT} , $E[d^{\sigma_{\text{OPT}}}] = 1$, and the maximum throughput $\hat{\lambda}$ is 0.909. In the case of σ_{WRS} , $E[d^{\sigma_{\text{WRS}}}] = 4$, and $\hat{\lambda} = 0.227$. These values are met by the simulation results of Fig. 12. The best heuristic is SN, guaranteeing slightly more than 50% of the bandwidth available when the optimal placement is used. RND provides performance very close to WRS. Figure 13 shows the delay–

TABLE IV
AVERAGE COST FUNCTION $E(D^2)$ FOR DIFFERENT PLACEMENT ALGORITHMS UNDER RANDOMLY GENERATED DOUBLY STOCHASTIC TRAFFIC MATRICES

Algorithm	$N=9$	$N=25$
MLT	1.581	2.598
SN	1.487	2.339
RND	1.687	2.603

throughput trade-off. This adversarial scenario shows that clever node placement algorithms may provide nonmarginal performance benefits.

However, from a practical point of view, it is important to understand the behavior of the placement algorithms under more general traffic scenarios. We randomly generated 1000 doubly stochastic traffic matrices (with the constraint $\lambda_{ii}=0$) and run the placement algorithm on them. Table IV shows the average of the cost function $E(d^o)$ obtained for two different switch sizes. Although the SN heuristic still provides the best performance, the difference between the algorithms is not so significant. Thus from a practical point of view, simple node placement algorithms may only slightly reduce switch performance and are a viable solution to exploit multihop schemes.

In summary, while for particular worst-case traffic scenarios (e.g., localized traffic) the topology placement is crucial for performance, when the traffic is general and relatively uniform (as in the case of doubly stochastic traffic matrices) the node placement slightly affects performance. This confirms the observation of Subsection V.B: For most traffic matrices, it is quite efficient to *a priori* design the multihop frame scheduling by assuming uniform traffic, because it is fairly robust to variations in the traffic matrix.

VII. CONCLUSIONS

In this paper we studied the multihop approach to schedule packets across an optical switching fabric

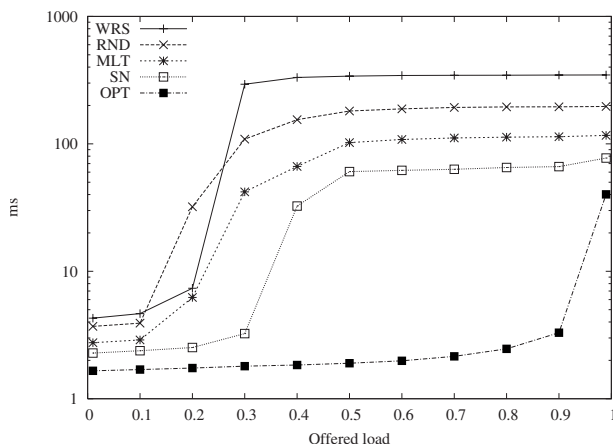


Fig. 13. Average delay for node placement algorithms.

with large reconfiguration latencies. The main idea is to send a packet from an input port to an output port across the switching fabric through (possibly) several intermediate ports, to reduce the number of switching reconfigurations and to avoid unacceptable access delays.

We proposed a generic framework to compute multihop frame scheduling, which cycles over a set of switching configurations, thereby defining a virtual topology in time division. We investigated with some detail the special case of a multihop frame based on Manhattan virtual topologies, under uniform and non-uniform traffic. We showed the trade-off between throughput, speedup, and delays. This trade-off is well estimated by simple formulas in selected scenarios, whereas simulation results confirm these findings in more general scenarios.

The multihop approach is the only viable switch control technique to overcome the large reconfiguration penalty of optical devices in large-size IQ switches. Although many degrees of freedom are available in the definition of the switch control scheme, which would be asked to optimize performance for particular traffic matrices, a simple approach defined for uniform traffic was shown to be robust enough to provide good performance for general traffic scenarios.

We considered only stationary traffic patterns in our analysis. Obviously in real life traffic is not stationary, and multihop switch control should be adapted to the current traffic conditions. This adaptation of the scheduling to traffic cannot happen on a very short scale in packet-switched networks, where packet flows can be intrinsically bursty and affected by end-to-end flow control and congestion avoidance algorithms. The adaptation of the switch control scheme can be done at different complexity and performance levels (i) by modifying the time scheduling (i.e., the duration P_k of scheduling epochs), (ii) by changing the routing on the current virtual topology, or (iii) by changing the virtual topology. In the paper we showed that some virtual topologies (we focused on the multidimensional bidirectional Manhattan in Section V.B) are rather robust to changes in the traffic pattern. Hence they can serve nonextreme nonstationary traffic scenarios with limited performance degradation. In any case, changes in the scheduling require both internal signaling to reconfigure network ports and some form of synchronization of the scheduling changes among the different ports to avoid inconsistencies in the scheduling decisions and out-of-sequence packet delivery. An analysis of the trade-offs between better performance due to a scheduling well matched to the current traffic, and the extra cost to sustain frequent scheduling changes, is outside the scope of this paper, and left for future investigations.

We finally note that recently commercial optical cross connects became available and gained significant market shares. They target circuit switching, and hence offer reconfiguration times that are not suited to packet switching. The approaches proposed in this paper may help router designers to take advantage of the optical switching technologies that are finding technical maturity and large-scale production in current optical cross connects.

APPENDIX A: MAXIMUM THROUGHPUT UNDER MULTIHOP AND RECONFIGURATIONS

An upper bound to the maximum throughput under uniform traffic can be computed by combining the effects of multihop transmissions and of the reconfigurations.

We start to evaluate the former, affecting the load offered to the switch. We assume that routing in the topology is able to distribute the traffic uniformly among all links; when the topology is symmetric, this assumption can usually be met. Then, the average traffic offered to a port is due to the traffic entering the port from outside and the traffic traversing that port to reach its final destination. Given a topology mapping σ , let d_{ij}^σ be the length of the path (in terms of number of hops or edges) along which traffic from port i to port j is routed, i.e. the distance between ports i and j (or, equivalently, between nodes $\sigma(i)$ and $\sigma(j)$) under the particular chosen σ . The total traffic flowing on the topology is

$$\rho_{\text{tot}} = \sum_{i,j} \lambda_{ij} d_{ij}^\sigma = \sum_{i,j} \frac{\lambda}{N} d_{ij}^\sigma,$$

where λ is the total arrival rate at the switch. The overall load offered to a port is

$$\rho = \frac{\rho_{\text{tot}}}{N} = \frac{\sum_{i,j} \lambda d_{ij}^\sigma}{N^2} = \lambda E[d^\sigma],$$

with $E[d^\sigma]$ being the average internodal distance according to the selected routing strategy, defined in Eq. (1). As a consequence, an upper bound to the maximum throughput is given by the traffic load at which the port load equals the port capacity.

Now we evaluate the effect of the reconfigurations, affecting the available capacity at each port. Since during an epoch of duration $(P+T)$, the port is able to serve the traffic for a duration P , then the port capacity is given by

$$\mu = \frac{P}{P+T}.$$

By imposing $\rho < \mu$, the maximum offered load to a port $\hat{\lambda}$ is

$$\hat{\lambda} = \frac{P}{P+T} \frac{1}{E[d^\sigma]}. \quad (6)$$

APPENDIX B: PERFORMANCE OF SINGLE-HOP APPROACH

In the case of the single-hop configuration, we consider a frame scheduling approach, adopting a sequence of N disjoint matchings given by the BvN decomposition [15] of the traffic matrix. Under uniform traffic, a frame is composed by N scheduling epochs; during the k th scheduling epoch ($0 \leq k < N$), input port i will be connected to output port $|i+k|_N$ for a duration P . Hence, the maximum throughput for single hop under uniform traffic is

$$\hat{\lambda}_{\text{SH}} = \frac{P}{P+T}.$$

For single-hop, the worst-case access delay \bar{D}_{SH} can be bound simply by

$$\bar{D}_{\text{SH}} = N(P+T).$$

APPENDIX C: PERFORMANCE OF MANHATTAN TOPOLOGIES

In the case of Manhattan topologies and uniform traffic, PDR (described in Section IV) allows us to balance the offered traffic across all the links. To apply Eq. (6), we estimate the average distance for bidirectional Manhattan topologies. For each dimension, two possible directions can be chosen; hence $\sqrt[2]{N}/4$ is the approximate average distance traversed along the same direction (precise evaluation of the average distance is possible, but this approximation gives an upper bound that is tight enough for our purposes). Since c dimensions are allowed,

$$E[d^\sigma] = c \sqrt[2]{N}/4. \quad (7)$$

The maximum throughput for multihop Manhattan (MH) topologies under uniform traffic, obtained by combining Eqs. (6) and (7), is

$$\hat{\lambda}_{\text{MH}} = \frac{P}{P+T} \frac{4}{c \sqrt[2]{N}}.$$

Note that, according to the reasoning in Subsection III.A, a spatial speedup equal to $2c$ (independent of N) is sufficient for this topology to obtain the maximum throughput without paying any reconfiguration penalty.

For multidimensional Manhattan topologies, we can easily estimate an upper bound on the average ac-

cess delay. Observe that only $2c$ (twice the degree of the topology) matchings constitute the frame, lasting $2c(P+T)$ time slots. Furthermore, the shortest path between two generic nodes can be associated with an ordered sequence of h directions (one for each dimension; hence $1 \leq h \leq c$), corresponding to h different matchings. The average number of directions $E[h]$ taken by a packet is given by

$$E[h] = \begin{cases} \frac{2(N - N^{1/2})}{N - 1} & \text{for } c = 2 \\ \frac{3(N - N^{2/3})}{N - 1} & \text{for } c = 3 \end{cases},$$

which can be shown by simple geometrical reasonings and counting arguments; for N large enough, $E[h] \asymp c$. In the worst case, a packet should follow a path corresponding to the farthest (in time) matching in the frame. For each direction, this implies to wait the whole frame minus one epoch: $(2c-1)(P+T)$. Hence, the worst-case access delay is

$$\bar{D}_{MH} = E[h](2c-1)(P+T) \approx c(2c-1)(P+T).$$

ACKNOWLEDGMENTS

The work described in this paper was performed with the support of the BONE project (Building the Future Optical Network in Europe), funded by the European Commission through the 7th Framework Programme.

REFERENCES

- [1] I. Keslassy, S. T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, "Scaling internet routers using optics," in *Proc. 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Karlsruhe, Germany, Aug. 2003, pp. 189–200.
- [2] M. C. Wu, O. Solgaard, and J. E. Ford, "Optical MEMS for lightwave communication," *J. Lightwave Technol.*, vol. 24, no. 12, Dec. 2006, pp. 4433–4454.
- [3] P. D. Dobbelaere, K. Falta, and S. Gloeckner, "Advances in integrated 2D MEMS-based solutions for optical network applications," *IEEE Commun. Mag.*, vol. 41, no. 5, May 2003, pp. S16–S23.
- [4] S. Hengstler, J. J. Uebbing, and P. McGuire, "Laser-activated optical bubble switch element," in *2003 IEEE/LEOS Int. Conf. on Optical MEMS*, Waikoloa, HI, Aug. 18–21, 2003, pp. 117–118.
- [5] K. Nashimoto, N. Tanaka, M. LaBuda, D. Ritums, J. Dawley, M. Raj, D. Kudzuma, and T. Vo, "High-speed PLZT optical switches for burst and packet switching," in *BroadNets 2005. 2nd Int. Conf. on Broadband Networks*, Boston, MA, Oct. 7, 2005, vol. 2, pp. 1118–1123.
- [6] S. L. Danielsen, C. Joergensen, B. Mikkelsen, and K. E. Stubkjaer, "Optical packet switched network layer without optical buffers," *IEEE Photon. Technol. Lett.*, vol. 10, no. 6, June 1998, pp. 896–898.
- [7] K. Kar, D. Stiliadis, T. V. Lakshman, and L. Tassioulas, "Scheduling algorithms for optical packet fabrics" *IEEE J. Sel. Areas Commun.*, vol. 21, no. 7, Sept. 2003, pp. 1143–1155.

- [8] I. Keslassy, M. Kodialam, L. T. Lakshman, and D. Stiliadis, "On guaranteed smooth scheduling for input-queued switches," in *IEEE INFOCOM 2003. 22nd Annu. Joint Conf. of the IEEE Computer and Communications Societies.*, vol. 2, San Francisco, CA, March 30–Apr. 3, 2003, pp. 1384–1394.
- [9] B. Towles and W. J. Dally, "Guaranteed scheduling for switches with configuration overhead," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 835–847, Oct. 2003.
- [10] Li Xin and M. Hamdi, "On scheduling optical packet switches with reconfiguration delay," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 7, Sept. 2003, pp. 1156–1164.
- [11] V. Alaria, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "Design of switches with reconfiguration latency," in *IEEE Int. Conf. on Communications, 2006. ICC '06*, Istanbul, Turkey, Jun. 2006, vol. 6, pp. 2599–2605.
- [12] A. Bianco, P. Giaccone, E. Leonardi, F. Neri, and P. R. Brusin, "Multi-hop scheduling for optical switches with large reconfiguration overhead," presented at HPSR 2004, High Performance Switching and Routing, Phoenix, AZ, April 2004.
- [13] V. Alaria, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "Multi-hop Scheduling Algorithms in Switches with Reconfiguration Latency," in *2006 Workshop on High Performance Switching and Routing*, Poznan, Poland, June 7–9, 2006, pp. 1–6.
- [14] 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, Nov. 1993, pp. 319–352.
- [15] C. S. Chang, W. J. Chen, and H. Y. Huang, "Birkhoff-von Neumann input buffered crossbar switches," in *IEEE INFOCOM 2000. 19th Annu. Joint Conf. of the IEEE Computer and Communications Societies. Proceedings*, Tel Aviv, Israel, March 26–30, 2000, vol. 3, p. 1614–1623.
- [16] T. Weller and B. Hajek, "Scheduling nonuniform traffic in a packet-switching system with small propagation delay," *IEEE/ACM Trans. Netw.*, vol. 5, no. 7, pp. 813–823, Dec. 1997.
- [17] A. Bianco, M. Franceschinis, S. Ghisolfi, A. Hill, E. Leonardi, F. Neri, and R. Webb, "Frame-based matching algorithms for input-queued switches," in *Workshop on High Performance Switching and Routing, 2002. Merging Optical and IP Technologies*, Kobe, Japan, May 2002, pp. 69–76.
- [18] K. L. Yeung and T.-S. P. Yum, "Node placement optimization in ShuffleNets," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, June 1998, pp. 319–324.
- [19] W. J. Dally and B. Towles, *Principles and Practise of Interconnection Networks*, San Fransisco, CA: Morgan Kaufmann, 2004.
- [20] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, San Fransisco, CA: Morgan Kaufmann, 1992.



Valentina Alaria was born in Torino, Italy, in 1980. She received her M.S. in Telecommunication Engineering from Politecnico di Torino, Italy in July 2004. In August 2004, Valentina joined Cisco Systems in San Jose, California, first in the Architectural Team for the Datacenter Business Unit and now in the Advanced Architecture and Research Group. Her main areas of interest span from Data Center and storage architectures to applications classification and monitoring, enterprise services and databases virtualization.



Andrea Bianco is Associate Professor at the Electronics Department of Politecnico di Torino, Italy. He has coauthored more than 130 papers published in international journals and presented in leading international conferences in the area of telecommunication networks. He was technical program co-chair of the HPSR (High Performance Switching and Routing) 2003 and 2008 workshops and of DRCN (Design of Reliable Communication Networks) 2005. He has

been guest or guest coeditor of special issues in international journals, including *IEEE Communications Magazine* and *Computer Networks*. He was TPC member of several conferences, including IEEE INFOCOM, IEEE GLOBECOM, and IEEE ICC. His current main research interests are in the fields of protocols and architectures for all-optical networks and switch architectures for high-speed networks.



Paolo Giaccone received the Dr. Ing. and Ph.D. degrees in telecommunications engineering from the Politecnico di Torino, Torino, Italy, in 1998 and 2001, respectively. He is currently an Assistant Professor 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, New Jersey. During 2000–2001, he was with the Department of Electrical

Engineering, Stanford University, Stanford, California. His main area of interest is the design of scheduling policies for high-performance routers and for wireless networks.



Emilio Leonardi was born in Cosenza, Italy, in 1967. He received a Dr. Ing. degree in Electronics Engineering in 1991 and a Ph.D. in Telecommunications Engineering in 1995, both from Politecnico di Torino. He is currently an Associate Professor at the Dipartimento di Elettronica of Politecnico di Torino. In 1995, he visited the Computer Science Department of the University of California, Los Angeles (UCLA), in summer 1999 he joined the High Speed Networks

Research Group, at Bell Laboratories/Lucent Technologies, Holmdel (New Jersey); in summer 2001, the Electrical Engineering Department of the Stanford University; and finally in summer 2003, the IP Group at Sprint, Advanced Technologies Laboratories, Burlingame, California. He participated in several national and European projects such as IST-SONATA and IST-DAVID and the NoE e-Photon-One Euro-FGI. He has also been involved in several con-

sulting and research projects with private industries, including Lucent Technologies-Bell Labs, IBM, British Telecom, Alcatel, and TILAB. He is the scientific coordinator of the European 7th FP STREP project “NAPA-WIN E” on P2P streaming applications, involving 11 European research institutions, operators and manufacturers. He has coauthored more than 150 papers published in international journals and presented in leading international conferences, all of them in the area of telecommunication networks. He participated in the program committees of several conferences, including IEEE Infocom, IEEE Globecom, and IEEE ICC. He was guest editor of two special issues of *IEEE Journal of Selected Areas of Communications* focused on high-speed switches and routers. He has been recipient of the best paper award at the following conferences: IEEE Globecom—High Speed Networks Symposium, 2002; IEEE High Performance Switching a Routing Symposium (HPSR), 2006; 14th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MAS-COTS), 2006. His research interests are in the fields of performance evaluation of wireless networks, point-to-point systems, queueing theory, and packet switching.



Fabio Neri was born in Novara, Italy, in 1958. He holds a M.S. and Ph.D. in Electrical Engineering, both from Politecnico di Torino, Turin, Italy. His research interests are in the fields of performance evaluation of communication networks, high-speed and all-optical networks, packet switching architectures, discrete event simulation, and queueing theory. He is Full Professor at the Electronics Department of Politecnico di Torino (see www.tlc-networks.polito.it). His

teaching duties include graduate-level courses on computer communication networks and on the performance evaluation of telecommunication systems. He leads research activities on optical networks and on switching architectures at Politecnico di Torino. He coordinated the participation of his research group in several national Italian research projects. He was involved in a number of European projects on WDM networks, and was the coordinator of the FP6 Network of Excellence e-Photon/ONe on optical networks, which involved 40 European institutions. He has coauthored more than 150 papers published in international journals and presented in leading international conferences. Dr. Neri participated in the technical program committees of several conferences, including IEEE Infocom and IEEE Globecom. He was general cochair of the 2001 IEEE Local and Metropolitan Area Networks (IEEE LANMAN) Workshop, of the 2002 and 2007 IFIP Working Conference on Optical Network Design and Modelling (ONDM), and of the Optical Networks and Systems Symposium at IEEE Globecom 2008. He served on the editorial board of *IEEE/ACM Transactions on Networking*, and is co-editor-in-chief of the Elsevier *Optical Switching and Networking* journal.