Multistage Network With Globally Controlled Switching Stages and Its Implementation Using Optical Multi-Interconnection Modules

Alvaro Cassinelli, Makoto Naruse, and Masatoshi Ishikawa

***Abstract*—Plane-to-plane guided-wave-based interconnection modules are proposed as building blocks for scalable opto-electronic multistage interconnection networks (MINs). This approach leads naturally to a MIN paradigm based not on cascading switching stages containing several size-reduced crossbars, as in the shuffle-exchange (SE) networks, but on cascading *permutation*-reduced crossbars instead, one per stage. The interest of such an architecture lies in the control simplicity and scalability potential. Transparent circuit switching for permutation routing is possible in such an unbuffered "globally switched" multistage interconnection network (GSMIN). Preliminary experiments using fiber-based interconnection modules are presented. Performance analysis and simulation of a buffered GSMIN is also studied for packet routing purposes.**

***Index Terms*—Circuit switching, column control, mechanically reconfigurable optical switch, optical multistage interconnection network, packet switching, parallel optical interconnection, permutation routing, stacked optical planar waveguides.**

I. Introduction

MULTISTAGE interconnection networks (MINs) have been studied extensively both from the point of view of their permutation capacity, which is required in specialized communication primitives for parallel computing, and from the point of view of their full-access capacity (point-to-point random access), which is useful for generic processor-memory requests in multiprocessor systems and of course in communication networks [1], [2]. In this paper, we will concentrate mainly on the latter issue by assuming a packet-switched path set-up protocol. We will consider as a starting point the class of self-routed, digitally controlled Delta networks, which covers a very large set of multistage interconnection networks, including the well-known shuffle-exchange (SE) networks [3], [4]. It has been noted previously that these Delta-networks somehow bridge the gap between low cost, time-shared bus architectures and the more efficient, though expensive, full crossbars when it comes to satisfy generic processor-memory requests [5]. Based on such multistage architecture, and extending our previous research on dense, plane-to-plane guided-wave-based permutation interconnection modules for pipelined optoelectronic systems [6], we propose here a new multistage paradigm and routing strategy that can be efficiently implemented by cascading *multi*permutation modules. A *multi*permutation module contains a reduced set of interstage global interconnections, that is to say, switching of individual channels is not allowed (see Fig. 1). This architecture will therefore be referred to as "globally switched (stage)" MINs (or GSMINs for short) in contrast with the well-known shuffle-exchange MINs (SEMINs), which have independent control of switches belonging to the same stage. An unbuffered GSMIN architecture may be of interest in the case of a time-division multiplexed (TDM), transparent circuit-switched permutation network [7], but it presents too much packet loss for packet switching purposes. However, we will show that a proper routing strategy combined with a moderately buffered GSMIN architecture may lead to performance competitive with most standard buffered MIN architectures under random traffic load.

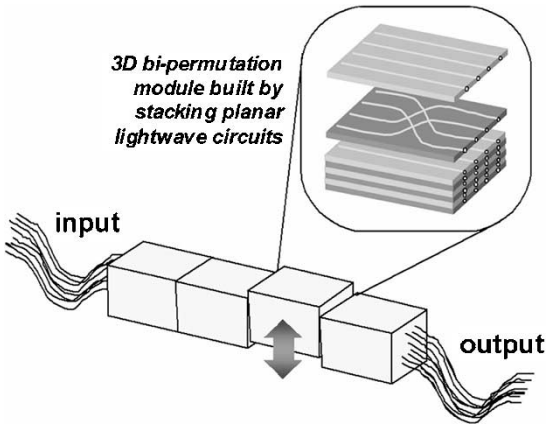

Fig. 1. Principle of a transparent circuit-switched network using cascaded guided-wave-based multipermutation modules. Minute displacement of each module may be used to address the required interstage permutation.

This paper is organized as follows. In Section II we review some of the basic concepts of SEMINs, and introduce (and intuitively justify) the idea of globally controlling some subsets of switches. In Section III we present an analytical model for a globally switched MIN without buffers, and compare its performance with the full crossbar and the equivalent unbuffered SEMINs. In Section IV, we model a buffered GSMIN and study its performance through computer simulation. In Section V we present experimental results on an all-optical unbuffered GSMIN setup using fiber-based interconnecting modules that

Manuscript received May 22, 2003; revised November 4, 2003.

A. Cassinelli and M. Ishikawa are with the Department of Information Physics and Computing, University of Tokyo, Tokyo 113-8656, Japan.

M. Naruse was with the Department of Information Physics and Computing, University of Tokyo, Tokyo 113-8656, Japan. He is now with the Ultrafast Photonic Network Group, Communication Research Laboratory (CRL), Tokyo 184-8795, Japan.

Digital Object Identifier 10.1109/JLT.2004.824385

can be mechanically reconfigured. In the conclusion section, we summarize our results and discuss further research directions.

II. The Stage-Switched MIN

A. Introduction

Although nonblocking multistage networks have been extensively studied as an alternative to the full-crossbar (most of them derived from the three-stage Clos/Benes architecture [8], [9]), attention has been focused on *blocking*, but still *full-access*,[1] multistage networks mainly because of their cost-effectiveness owing to their simplicity and moderate number of switching elements [10]–[14]

Both synchronous routing (or permutation routing) and asynchronous routing (or point-to-point routing) have been explored in blocking MINs [15]. In short, circuit switching using an unbuffered architecture is a good strategy for implementing permutation routing (the kind of routing typically required to implement communication primitives on single-instruction-multiple-dimension computers running highly parallelized algorithms since the relatively few global interconnection patterns required may still be provided by a blocking network). Unless the required permutations are known in advance, time-consuming setup algorithms may be required to set up the switches. If the reconfiguration speed is larger than the communication request rate, then a more or less adaptive TDM technique can be used to avoid such a computation intensive setup-phase, providing that the data to be transmitted can still be accommodated in a TDM time slot [16]. This paradigm is very interesting if the network is transparent, since despite short temporal communication slots, these may still accommodate large amounts of data owing to the huge bandwidth of optical channels.

On the other hand, when traffic is less regular (uncorrelated processor-to-processor or processor-memory requests on MIMD machines, either multiprocessor or multicomputer, and in telecom networks), the best performance is achieved by a buffered packet switching strategy. If the resulting network performance is acceptable, then a distributed control strategy can be implemented (and if possible deterministic—like self-routing), further reducing the setup overhead and enhancing the modularity of the network. The major drawback remains the resulting "opacity" of the network (state of the art technology still does not allow robust optical buffering and control flow, so that some form of optoelectronic conversion is needed at intermediate stages).

Both routing strategies will be studied in the following for a class of self-routing MINs known as shuffle-exchange networks (SEMIN), with the focus on a packet-switching strategy in a modified MIN with joint control of switches belonging to the same switching stage (GSMIN). To understand the basics of the GSMIN architecture, and how it relates to the standard SEMIN architecture, first we need to say a few words about the Delta-networks from which both are derived.

The well-known SE network, i.e., the Omega network and its reverse (Flip) [10], the Indirect Binary Cube and its reverse (the Generalized Cube) [14], and the Baseline network and its reverse [13], are all topologically equivalent[2] to the uniform bi-Delta network using regular (q-shuffle) permutations as interstage interconnections and 2×2 elemental crossbars as switching elements [17].

The differences between these SE networks lie uniquely in the choice of the interstage permutation. Since any network belonging to the Delta-class is digitally controlled and self-routing, SE networks are all self-routing as well.

A recursive definition of the multistage bi-Delta network in question is straightforward. We have

$$\begin{aligned}\Delta_n &= \mathcal{L}(\Delta_{n-1}); S_n; \mathrm{ES}_n(1) \\ &= \mathrm{ES}_n(1); S_n^{-1}; \mathcal{L}(\Delta_{n-1}).\end{aligned}$$

In the above formula, indexes represent the size (defined as the logarithm in base 2 of the number of input/outputs) of the whole network (Δ_n), its permutation stages (S_n), or its switching stages (ES_n) (formally, all of them are permutation-networks, denoted in general as P_n). The concatenation of permutation-networks (represented by the symbol ";") is to be read in lexicographic order to facilitate translation into a classical graphical representation having inputs at the left and outputs at the right. S_n represents the well-known (n-bit long) perfect shuffle permutation stage: if $[b_n, b_{n-1}, \ldots, b_1]$ represents the binary address of a given input channel, then the address of the corresponding perfect shuffled output channel will be given by: $S_n([b_n, b_{n-1}, \ldots, b_1]) = [b_{n-1}, \ldots, b_1, b_n]$. Positive superscripts denote repetitive concatenation of a given permutation stage (e.g., $S_n^2 = S_n; S_n$), and negative superscripts denote concatenation of its inverse (e.g., S_n^{-1} is the inverse perfect-shuffle). The replication operator $\mathcal{L}$ acts on permutation-networks; it takes a permutation-network P_n and gives the permutation-network $[\mathcal{L}(P_n)]_{n+1}$ which independently permutes the first 2^n inputs and second 2^n inputs, always following P_n (subscript indexes and parenthesis can be dropped when there is no risk of ambiguity, e.g., $[\mathcal{L}(P_n)]_{n+1} \equiv \mathcal{L}P_n$.) The operation can be made more explicit by showing the effect of the replicated permutation-network $\mathcal{L}P_n$ on the $n+1$ bit-long binary representation of a given input address: $(\mathcal{L}P_n)([b_{n+1}, b_n, b_{n-1}, \ldots, b_1]) = [b_{n+1}, P_n([b_n, b_{n-1}, \ldots, b_1])]$. In a classical planar representation of a network, this simply corresponds to placing two identical networks one above the other, as shown in Fig. 2. Successive network replication will be denoted using a superscripted replication operator: $\mathcal{L}^k(P_n) = \mathcal{L}^{k-1}(\mathcal{L}P_n)$. Last, $\mathrm{ES}_n(1)$ corresponds to the n-input/n-output first-order exchange-switching stage. It is composed of a set of 2^{n-1} first-order exchange switches ES_1, each attached to adjacent pairs of inputs (i.e., their addresses differ uniquely on the least significant bit). That is to say, $\mathrm{ES}_n(1) = \mathcal{L}^{n-1}\mathrm{ES}_1$. The first-order exchange switch ES_1 is a 2×2 reduced crossbar providing *either* the first-order exchange permutation E_1 *or* the identity (or by pass) permutation I_1. The first-order exchange permutation E_1(also known as the first-order cube permutation)

[1]Any input should be able to access any output in one pass through the network.

[2]Two networks are said to be topologically equivalent if their corresponding graphs are *isomorphic* (an input and/or an output permutation suffices to make them look identical).

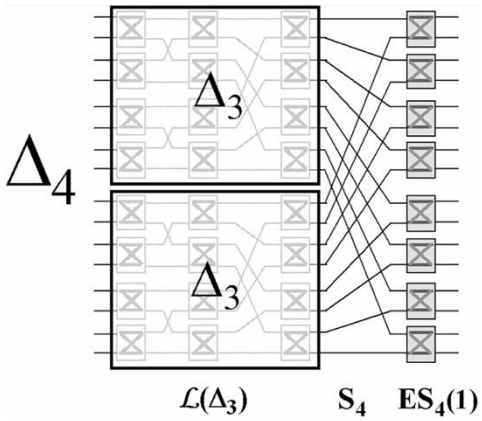

Fig. 2. A recursive definition of the uniform bi-Delta network using 2×2 crossbar switches.

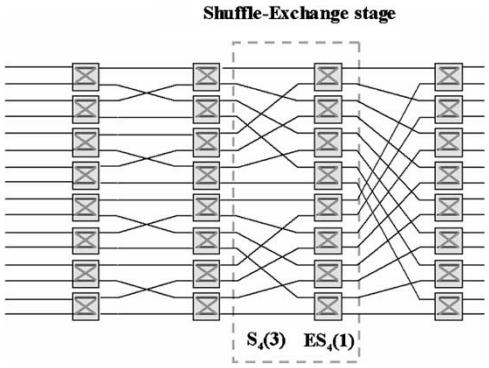

Fig. 3. Complete development of the recursive binary Delta-network definition gives a SE representation known as the Inverse Baseline.

is defined by $E_1([b_1]) = [\overline{b_1}]$, where b_1 represents a single bit coding the channel input or output number of the 2×2 crossbar.

So far, we defined a switching network as a particular concatenation of subnetworks elements (i.e., its definition string). However, when studying the permutation capacity of a switching network, it is possible to assimilate each subnetwork with the set of permutations it can generate (e.g., $\mathrm{ES}_1 = \{E_1, I_1\}$), and interpret the symbol ";" as the cartesian product of such permutation sets. Eventually, the whole network is assimilated to the set of its realizable permutations (practically, each permutation is realized by a particular configuration of the network switches). By identifying such permutations sets one can prove *functional equivalence* between networks, as in [18], [19]. However, it is also possible to prove *topological* equivalence between two networks by manipulating their definition strings, provided that the "integrity" of their switching stages is preserved (i.e., each switching stage may be replaced by a topological equivalent stage, but the switches belonging to a given switching stage remain always together, see Fig. 4).

Fig. 3 represents the complete development of the Delta-network (Δ_4) recursive formula. This network is also called the (4-stage) Inverse Baseline (IBS_4). (Sometimes a bit reversal permutation is appended to the network, so that when all switches are in bypass configuration, the whole network realizes the identity permutation.) The IBS_n has n SE stages.

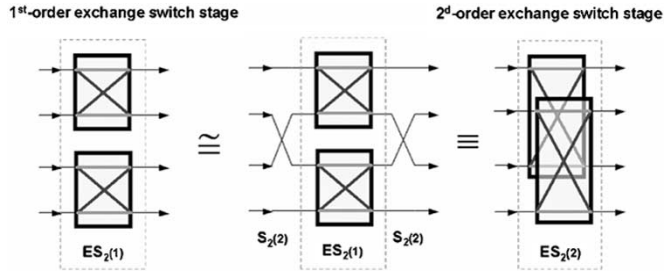

Fig. 4. First-order (left) and second-order (right) exchange switching (SE) stages are topological equivalent ($\cong$) permutation networks.

The interconnection at stage k is given by the $n-k$ replication of the k-bit perfect shuffle noted $S_n(k) = \mathcal{L}^{n-k}S_k$. We have

$$\mathrm{IBS}_n = S_n(1); \mathrm{ES}_n(1); \ldots; S_n(n); \mathrm{ES}_n(1),$$
$$\text{also noted } \mathrm{IBS}_n = \prod_{k=1}^{n} \{S_n(k); \mathrm{ES}_n(1)\}.$$

More generally, a k-order (n-input/n-output) exchange switching stage $\mathrm{ES}_n(k)$ is defined as the $n-k$ replication of the k-order exchange switch: $\mathrm{ES}_n(k) = \mathcal{L}^{n-k}\mathrm{ES}_k$. The k-order exchange switch $\mathrm{ES}_k = \{E_k, I_k\}$ is a 2-state switch providing either the k -order exchange permutation ES_k (also known as the k-cube permutation) or the identity. The k-order exchange permutation (or k-cube permutation) E_k is defined by $E_k([b_k \ldots b_1]) = [\overline{b_k} \ldots b_1]$. (Each k-order exchange switch is a 2×2 switch attached to nonadjacent pairs of inputs — their addresses differ on bit k.) As exemplified in Fig. 4, all these "generalized" exchange-switching stages are topologically equivalent networks. This is so because we have $\mathrm{ES}_n(m) = S_n^l(k); \mathrm{ES}_n(m+l); S_n^{-l}(k)$ (where $m < k \leq n$, and $l \leq k-m$), a transformation that preserves the integrity of the switching stages.

Electronic technology lends naturally to the use of first-order exchange switches, since switching typically takes place in a very narrow area composed of a few electronic gates. As a consequence, the electronic implementation of a binary Delta-MIN network (i.e., the SEMIN networks) rely exclusively on first-order SE-stages. On the other hand, since interstage permutations may be composed of long-range interconnections, they are strongly penalized by electronics, which has justified a lot of research directed toward their implementation using guided-wave or free-space optics.

The restriction to the first-order exchange switches when implementing a switching stage in a multistage interconnection network may not apply if the SE-stages are implemented using optical technology. This adds a new degree of freedom from the architectural point of view; in particular, the interstage permutation can be "absorbed" into the switching stage to form a "permutation switch" (a somehow similar consideration form the basis of the optical "3-D grid architectures" studied in [20]). For instance, in the following we will consider a network that is topologically equivalent to all SE networks, but uses *higher-order* exchange switching stages instead of first-order exchange switches. (In Section V we will discuss a possible hardware implementation of these higher-order switches.) It will be called

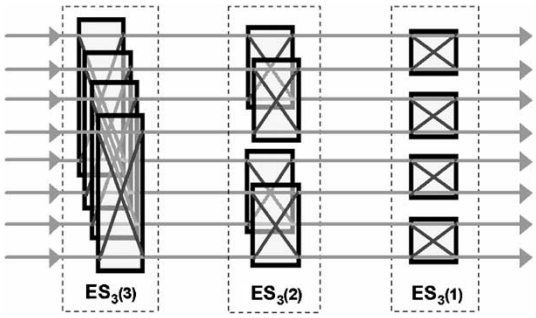

Fig. 5. The BHMIN network uses higher-order exchange switching stages. No particular permutating interconnection is needed between stages.

the Binary Hypercube MIN or BHMIN (see Fig. 5) since it can be seen as a multistage "spanned" version of a plain hypercube topology, where each stage provides a particular cube permutation. The plain hypercube, as defined in [21], is a weak hypercube — each processor uses at most one of its ports at any given cycle — that operates in SIMD mode with uniform addressing — all processors use the same interconnection dimension at any given cycle.)

More precisely, the BHMIN is defined as

$$\begin{aligned}\mathrm{BHMIN}_n &= \mathrm{ES}_n(n);\mathrm{ES}_n(n-1);\ldots;\mathrm{ES}_n(2);\mathrm{ES}_n(1)\\ &= \prod_{k=1}^{n}\mathrm{ES}_n(n-k+1).\end{aligned}$$

As can be seen, all the interstage permutations reduce to the identity. It is easy to prove topological equivalence with the Inverse Baseline by successively replacing, on the IBS_n formula, lower-order exchange switching stages with topological equivalent ones, but of a higher order. Indeed, using the fact that $\mathrm{ES}_n(m);S_n(k)=S_n(k);\mathrm{ES}_n(m+1)$, we have

$$\begin{aligned}\mathrm{IBS}_n &= \prod_{k=1}^{n}\{S_n(k);\mathrm{ES}_n(1)\}\\ &= S_n(1);\ \prod_{k=2}^{n}\{\mathrm{ES}_n(1);S_n(k)\};\ \mathrm{ES}_n(1)\\ &= S_n(1);\ \prod_{k=2}^{n}\{S_n(k);\ \mathrm{ES}_n(2)\};\ \mathrm{ES}_n(1)\\ &= S_n(1);\ S_n(2);\prod_{k=3}^{n}\{\mathrm{ES}_n(2);\ S_n(k)\};\ \mathrm{ES}_n(2);\ \mathrm{ES}_n(1)\\ &= \cdots\\ &= \prod_{k=1}^{n}S_n(k);\ \mathrm{BHMIN}_n.\end{aligned}$$

The Binary Hypercube MIN is topologically equivalent to the Inverse Baseline since a bit reversal permutation $R_n = R_n^{-1} = \prod_{k=1}^{n}S_n(k)$ at the output of the Inverse Baseline make both networks look identical.

B. Global Control of Switches

The motivation for exploring joint control of all switches belonging to the same stage on a generalized SEMIN (using first or higher-order SE-stages) comes from hardware as well as routing considerations. First, from the hardware point of view it would be very advantageous in terms of simplicity and scalability if a whole switching stage could be implemented as a module having a unique control signal. (This configuration, known as "column-control," has been given early attention on first-order unbuffered SEMINs, see for instance [7]). As a consequence, the switching stage and its adjacent interstage permutation (i.e., the SE block) could be physically merged into a unique "switching module" providing two different interconnection patterns. The interconnections provided by such bipermutation switching module may be long range (i.e., the input and output line addresses may differ on higher order bits), a property that may be *directly* accommodated by optical switching technology [22], [23]. By directly we understands here that the switching module is not implemented as a combination of "local" switches and long range fixed interconnections, as it occurs if one performs column-control of switches in a first-order SEMIN.

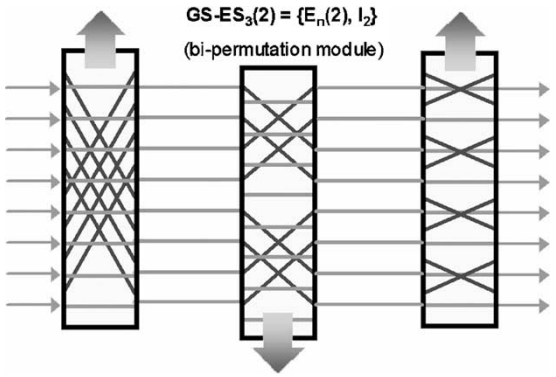

Fig. 6. The GSMIN architecture uses merged exchange/bypass switches: each stage can be configured as a *global* bypass or a *global* (first and higher-order) exchange permutation.

An example of a bipermutation switching module is given by a global-switched exchange stage of order k, or $\mathrm{GS}-\mathrm{ES}_n(k)$. This switching module is defined as the set of *two* permutations: $\mathrm{GS}-\mathrm{ES}_n(k) \equiv \{\mathcal{L}^{n-k}E_k, I_n\}(\neq \mathrm{ES}_n(k) = \mathcal{L}^{n-k}\mathrm{ES}_k$, which provides 2^n different permutations). As explained above, these bipermutation modules emerge naturally when merging shuffle stages and global-switched first-order SE-stages on the Inverse Baseline network. The resulting globally switched BHMIN (called GS—BHMIN, or GSMIN for short) is represented in Fig. 6. It uses cascaded bipermutation modules $\mathrm{GS}-\mathrm{ES}_n(k)$, providing each with two independent addressable permutations, namely the identity permutation and a particular cube permutation.

It is important to note that by column-controlling "binary" switches belonging to the same stage in a generalized SEMIN (i.e., derived from the *binary* Delta-network), we come up with *bi*-permutation modules, one per stage. However, we could as well have considered the joint operation of the certainly more complex $a\times b$ switches belonging to the same stage in the more general class of $a^n \times b^n$ -Delta-networks. This would produce an architecture composed of cascaded *multi*permutation modules, each module containing a larger number of independently addressable permutations (precisely $b!/(b-a)!$ global permutations, when $a > b$). Such GSMIN architecture is therefore

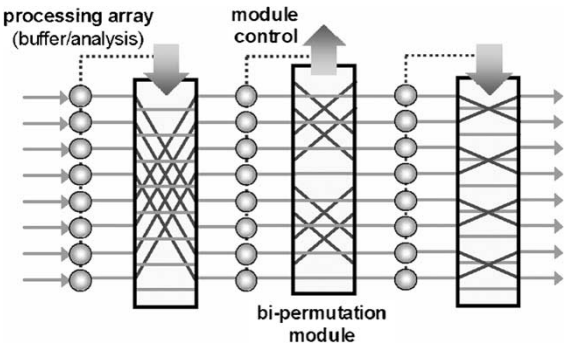

Fig. 7. The GSMIN buffered architecture using bipermutation modules.

formed by cascading *permutation-reduced* large crossbars, instead of cascading switching stages containing several *size-reduced* crossbars. However, this interesting issue goes beyond the scope of this paper.

Of course, the (unbuffered) GSMIN will have much less interconnection capacity than the (already blocking) SEMIN from which it is derived. However, we observe here that under very regular communication requests (such as the processor-to-processor requests generated on SIMD computers) a self-routing SEMIN can be overdimensioned in terms of interconnection capacity, since arbitrary point-to-point interconnections may not be required. In that case, permutation routing can be efficiently implemented by using the network as a circuit switch [16], [7]. Moreover, not even all the permutation states of these (blocking) networks may be required, only a very small subset may be sufficient (for instance, in a hypercube connected computer, any k-cube permutations may be requested at any given time by a parallel algorithm, but never a combination of these permutations at the same time). Even though a global controller may precalculate the states of the individual switches on a standard SEMIN based on the algorithm requirements [16], it is worth considering a more appropriate (multistage) decomposition of the set of required permutations, which obviates the need for independent control of switches belonging to the same switching stage when circuit switching for parallel computers is the target application.

Now, the most interesting observation comes from the behavior of an interstage-buffered GSMIN architecture under truly uniform random traffic (for a justification of this traffic model, see Section III). Suppose that conflicts generated by requests at the inputs of the first stage of a stage-buffered SEMIN are not resolved individually at each 2×2 switch (by dropping some packets), but rather globally at the stage level of the GSMIN by a competition between *all* the incoming requests. Since the traffic is random, it is likely that "votes" leading to the adoption of one of the two possible states of the global-switch will be evenly distributed. Such behavior takes place for all stages of the network, so that at each stage, half of the requests will be dropped and half will be able to pass to the next stage (on average). This represents an enormous amount of discarded requests, certainly much bigger than that occurring by internal blocking in the corresponding SEMIN; however, if one considers a buffered architecture (see Fig. 7), then presumably there will be no need to provide it with a large buffer memory, because the packets that have been retained in the buffers are very likely to go forward in the following cycle (since if they are made to participate in the competition, they will certainly bias the 50-50 voting distribution of the new arriving packets to "their advantage"). We can go even further and conjecture that in the particular case of truly random traffic, analysis of packet headers for the purpose of controlling globally-switched stages may be unnecessary: a continual "blind" alternation of switching states may perform just as well (i.e., TDM of stage-permutations). Analysis of packet headers is still necessary for allowing the packets to pass or to remain in the local buffer. Both conjectures (small buffers and efficiency of an alternating time-division permutation multiplexing technique) were verified by simulations, as we will show.

Last but not least, since the best way to enhance packet routing performance of a SEMIN involve primarily internal, stage-distributed buffering techniques, the proposed architecture has, at most, comparable buffer complexity, and presumably less hardware complexity. In the following we will evaluate the GSMIN performance on both unbuffered and buffered architectures. Analysis of cost-effectiveness will be left for further studies.

III. Analysis of an Unbuffered Architecture

We are now going to evaluate the performance of a synchronous, self-routed, unbuffered GSMIN using a memory-less virtual cut-through protocol suited for all-optical networks (the packet header is analyzed on the fly and is supposed to set the switch before the payload arrives. Since the architecture is unbuffered, when a resource conflict takes place, some packets will be dropped at intermediate stages). The following analysis would also apply to a circuit switched GSMIN, where each source attempts simultaneously to establish a communication path on the network (conflict resolution would be done by discarding requests at the *input* of the network instead of by dropping packets at intermediate stages).

A. Performance Evaluation

In order to model the functioning of the unbuffered GSMIN, we need to put forward several hypotheses. These include hypotheses on the way the network operates (a,b,c), and also about the characteristics of the requests to be routed (d,e)

a) Synchronous operation of the whole switching fabric, meaning that a transmission request (i.e., the presence of a request waiting for transmission at the input of each stage) occurs in a time slotted manner.
b) At each time slot—and since this is an unbuffered architecture—packets that can not be transmitted (because of blocking conflicts) are dropped and considered *lost* (this occurs at intermediate stages in an on-the-fly self-routed packet switched network, or at the input stage in the case of a circuit-switched network using a global controller).
c) Self-routing is assumed: when a packet is awaiting transmission at the input of stage k, the k-th bit of its n-bit path descriptor specifies the request or "vote" (Cross or Straight) made by *that* specific packet at stage k. As explained above, the new state of the global switch at stage

k is decided by a tournament among the votes made by *all* the incoming packets at stage k.

As one can expect, the performance of a specific switching network depends heavily on the kind of data traffic it is confronted with. Real traffic characteristics may be extremely difficult to grasp; however, the following are well-established models providing a good insight into network performance [24]:

d) The packet-arrival process (i.e., requests) at each input channel of the network is a Bernoulli process with parameter λ (i.e., λ is the probability that a request is made on a particular input at the beginning of each time slot). We assume the *temporal independence of requests* only for the input stage: traffic statistics are constant in time, and do not depend in particular on the previous requests. Temporal independence of requests (known as Strecker's approximation) is an extremely helpful simplification that may appear unrealistic considering the fact that packets that are dropped during the routing process will be, in a real system, resubmitted at the input soon or later. However, simulations show that, even in the case of a crossbar switch (where the transmission delay is minimal and thus resubmission will not be delayed for very long), deviations from the results using the simplifying assumption remain very small [25]. *Uniform load-distribution* is also assumed among inputs.

e) The traffic will be modeled using the *uniform reference model* (URM), implying that packets presented at the inputs are directed to any output with equal probability. This model, though unrealistic in most situations, has been chosen mainly for comparison purposes between the GSMIN and the SEMIN architectures (SEMIN performance has been extensively studied under the URM hypothesis). It is also interesting to note that all topologically equivalent SE networks will have equal performance under URM traffic (only the way the self-routing bits are handled by each switching stage may differ). Other interesting models that may be considered are the hot-spot model (a particular "hot" output is requested more often than the others [24]), or a hot-route model (suited for modeling communication requests of parallel programs where a particular "hot" global permutation, e.g., a dimension of an hypercube interconnected computer, is expected to be used intensively). Of course, modeling traffic nonuniformities represent a step toward a more realistic representation of the traffic generated by MIMD computers [24] as well as the data handled by ATM switches in broadband ISDN networks [26]. Techniques to cope with SEMIN performance degradation due to these traffic nonuniformities (message combining, packet randomization and rerouting for instance [27]) may be applied to the GSMIN as well.

The performance indicator of interest will be the *probability of packet acceptance* (or normalized bandwidth) of the network, PA. Assuming steady state operation of the network, the probability of packet acceptance can be defined as the limit (when it exists), of the ratio between the m-cycles long output bandwidth ob_m (i.e., the total amount of delivered packets collected during m not necessarily consecutive network cycles), to the m-cycles long input bandwidth ib_m (i.e., the total amount of requests generated during m not necessarily consecutive network cycles). The limit is taken when the number of sampling cycles m tends to infinity, $\mathrm{PA} = \lim_{m\to\infty}(\mathrm{ob}_m/\mathrm{ib}_m)$. The acceptance probability is at best equal to one, and is general smaller than one since some packets may be dropped within the network because of internal conflicts. A consequence of the Law of Large Numbers is that if the quantity ob_m (resp. ib_m) is the sum of *uncorrelated* samples of the instantaneous (i.e., 1-cycle long) output bandwidth OB_1 (resp. input bandwidth IB_1), then $\lim_{m\to\infty}(\mathrm{ob}_m/m)$ and $\lim_{m\to\infty}(\mathrm{ib}_m/m)$ are respectively equal to the mean of the instantaneous output bandwidth $\langle \mathrm{OB}\rangle \equiv \langle \mathrm{OB}_1\rangle$ (i.e., the average number of packets actually delivered at the output of the whole switching fabric in a single network cycle) and to the mean of the instantaneous input bandwidth $\langle \mathrm{IB}\rangle \equiv \langle \mathrm{IB}_1\rangle$ (the average number of packet requests made at the input in a single network cycle). In that case, the probability of packet acceptance is just equal to the ratio of these averages. Moreover, since (d), we have $\langle \mathrm{IB}\rangle = \lambda N$, and therefore: $\mathrm{PA} = \langle \mathrm{OB}\rangle/(\lambda N)$. (The same result hold in the case of correlated samples though, provided that the underlying sample-generation mechanism is a stochastic Markov Chain -see Section IV.B.) In the case of a packet switched network *with* buffers, the average packet delay is another important performance parameter, but this quantity will not be considered in this preliminary study.

B. Comparison of Crossbar, SEMIN, and GSMIN

1) Analysis of a Crossbar: The full crossbar does not suffer from internal blocking. Its acceptance probability is only limited by output blocking conflicts—a phenomenon that also affects unbuffered multistage networks. Hence, the crossbar will be useful reference for *unbuffered* networks since these cannot achieve better figures of packet acceptance. Following Strecker [28], the expected output bandwidth $\langle \mathrm{OB}\rangle$ of an $a \times b$ (*single*-stage) crossbar under uniform traffic load is equal to $\langle \mathrm{OB}\rangle = b[1 - (1 - (\lambda/b))^a]$ (this is the average number of delivered packets per unit time). It follows that the probability of acceptance PA is equal to $\mathrm{PA} = \langle \mathrm{OB}\rangle/\lambda a = (b/\lambda a).[1 - (1 - (\lambda/b))^a]$. The limit for PA when both a and b grows very large is $\mathrm{PA} \simeq (b/\lambda a).[1 - e^{-(a\lambda/b)}]$ (a value within 1% of the actual value when a and b are greater than 30), and $(1 - e^{-\lambda})(1/\lambda)$ in the case of a square crossbar.

It is important to note that although input requests are assumed to be time and channel independent, the same is not true for the *output packet delivery* events, because of output request conflicts.

2) Analysis of an Unbuffered SEMIN: Using the above results for the crossbar, Patel [5] first studied the performance of a synchronous, unbuffered n-stage Delta network (having a^n inputs and b^n outputs and using digitally controlled, small intermediate $a \times b$ crossbars) under the uniform traffic hypothesis. If λ_k is the probability per unit time that a request is made at a certain input of an intermediate $a \times b$ crossbar belonging to stage k, then accordingly to Strecker's formula the following recurrence relation holds true: $\lambda_{k+1} = \langle \mathrm{OB}\rangle/b = 1 - (1 - (\lambda_k/b))^a$ (and of course

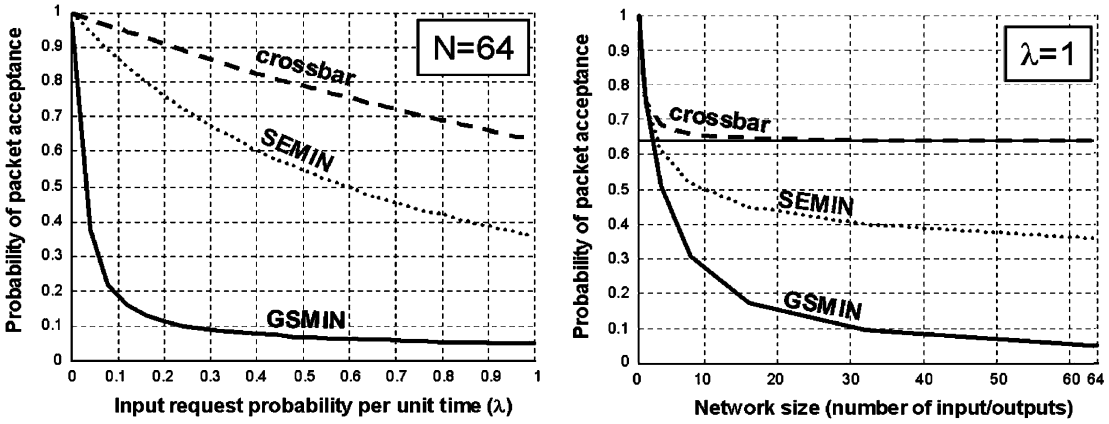

Fig. 8. Packet acceptance probability of three different unbuffered switching fabrics (crossbar, unbuffered SE network and unbuffered globally switched multistage network) as a function of (left) the traffic load for a fixed 64 × 64 size and (right) the network size (at full traffic load).

$\lambda_1 = \lambda$). This derivation assumes that the input requests to *any* small intermediate crossbar are, probabilistically speaking, *independent* events. Depending on the network topology, this may not be the case (as aforementioned, the output requests of any given crossbar are definitely *not* independent events). However, the assumption is true provided that the data streams arriving at different inputs ports of a given crossbar all passed through topologically *disjoint* subnetworks in previous stages: this is the case for all topologically equivalent SEMINs derived from the recursively built Delta-network. Using the recurrence relation, one can compute λ_{n+1}, i.e., the probability (per unit time) that there will be a packet delivered at any particular output of an n-stage Delta-network. The total bandwidth and probability of acceptance can then be computed by noting that $\langle \mathrm{OB} \rangle = b^n \lambda_{n+1}$ and $\mathrm{PA} = \langle \mathrm{OB} \rangle / \lambda a^n = (b^n \lambda_{n+1})/(a^n \lambda)$. Through some algebraic manipulations, Kruskal [29] showed an asymptotically valid *closed-form* expression for the acceptance probability of a square Delta-network using $k \times k$ switches: $\mathrm{PA} \simeq (2k)/((k-1)n + 2k/\lambda)$. For the interesting case where $k = 2$ (i.e., the binary Delta-networks known as SEMINs), we have: $\mathrm{PA} \simeq 4\lambda/(\lambda n + 4)$. This means that the probability of packet acceptance for the unbuffered SEMIN evolves as $1/\log_2 N$ and tends toward zero when N grows large ($N = 2^n$ is the number of network input/outputs).

3) Analysis of an Unbuffered GSMIN: In the case of an unbuffered GSMIN, it is possible to derive an exact recursive expression leading to the probability of packet acceptance. Let us define the random variable D_k corresponding to the number of transmission requests received at stage k ($k = 1, 2, \ldots, n$) at a given cycle. The event described as $\{D_k = d\}$ means that there are exactly d requests made at the input of stage k during the cycle under consideration. The probability distribution for D_k will be $p_k(d) \equiv \mathrm{prob}\{D_k = d\}$ for $d \in \{0, \ldots, N = 2^n\}$. The idea is to obtain a recurrence relation between the probability distributions p_k and p_{k+1}, instead of a recurrence relation between single-channel request probabilities λ_k (indeed, Strecker's formula cannot be used recursively here because request probabilities at intermediate stages will be strongly correlated as a result of the global-switching strategy employed). Since the probability distribution of D_1 is, given the uniform traffic hypothesis, equal to the binomial distribution $p_1(d) = \binom{N}{d}\lambda^d(1-\lambda)^{N-d}$, the recurrence relation will eventually lead to a computable expression for p_{n+1} (where $n = \log_2 N$) that can then be used to compute the average output bandwidth of the whole network by the formula: $\langle \mathrm{OB} \rangle = \langle D_{n+1} \rangle = \sum_{d=0}^{d=N} p_{n+1}(d).d$. In fact, such a recurrence relation is easy to obtain by noting that

$$p_{k+1}(d) = \sum_{r=d}^{\min(2d,N)} \mathrm{prob}\left\{ \begin{bmatrix} \text{exactly } r \text{ requests at} \\ \text{the input of stage } k \end{bmatrix} \text{ AND } \begin{bmatrix} d \text{ of these vote } S \text{ (traight)} \\ \text{while } (r-d) \text{ vote } C \text{ (ross)} \end{bmatrix} \text{ OR } \begin{bmatrix} d \text{ of these vote } C \\ \text{while } (r-d) \text{ vote } S \end{bmatrix} \right\}.$$

The first event described within square brackets is just $\{D_k = r\}$, whose probability is $p_k(r)$ by definition. The second event between square brackets accounts for the "fair" tournament which decides the new state of the switch as described earlier: given that there are exactly $r(\geq d)$ requests, we will get exactly d transmitted packets at the output of that stage only if there are at most d packets which vote straight *or* at most d packets voting cross (when $r = 2d$ the event should not be counted twice). Now, given that we have for any packet at stage k : $\mathrm{prob}\{\mathrm{vote} = S\} = 1 - \mathrm{prob}\{\mathrm{vote} = C\} \equiv p_{\mathrm{bit}}(k)$, we can write

$$\mathrm{prob}\left\{ \begin{matrix} d \text{ packets vote } S \\ \text{while } (r-d) \text{ vote } C \end{matrix} \right\} = \binom{r}{d} p_{\mathrm{bit}}^d(k)(1-p_{\mathrm{bit}}(k))^{r-d}.$$

But since voting values and number of voters are independent random events, we can write

$$p_{k+1}(d) = \sum_{r=d}^{\min(2d,N)} p_k(r) \cdot \binom{r}{d} \left(p_{\mathrm{bit}}^d(k)(1-p_{\mathrm{bit}}(k))^{r-d} + p_{\mathrm{bit}}^{r-d}(k)(1-p_{\mathrm{bit}}(k))^d \times [1 - \delta(r = 2d)] \right).$$

If we assume the URM model, then $p_{\mathrm{bit}}(k) = 1/2$ for all k and then the above expression becomes

$$p_{k+1}(d) = \sum_{r=d}^{\min(2d,N)} p_k(r) \cdot \binom{r}{d} \frac{1}{2^{r-1}} \cdot \frac{1}{2} \delta(r = 2d).$$

The graph on the left of Fig. 8 represents the acceptance probability as a function of the input load (parameter λ) for

a 64×64 crossbar, a standard 64×64 unbuffered SEMIN and a 64×64 unbuffered globally switched MIN, as described by the corresponding analytical formulas. As expected, the unbuffered GSMIN shows extremely poor performance under uniform traffic; the corresponding unbuffered SEMIN performs much better, but still 60% of the packets are dropped under heavy traffic load, which is almost twice the amount of packet loss occurring in a crossbar.

On the right of Fig. 8, it is shown how the packet acceptance evolves as a function of the network size, this time for a *fixed* maximum input load ($\lambda = 1$). The asymptotic value for the packet acceptance of the crossbar when N grows very large is represented as a horizontal straight line. While it is true that the GSMIN performance degrades much quicker than the standard MIN as the network grows , the figure clearly shows a significant difference between the unbuffered crossbar and the unbuffered multistage network (either the standard SEMIN or the derived GSMIN): the scaling of the number of inputs/outputs reduces packet acceptance toward zero, regardless of the "lightness" of the traffic load, while that of the crossbar remains always positive and greater than the asymptotic limit $(1-e^{-\lambda})\frac{1}{\lambda}$.

Despite the considerable amount of packet loss, an analysis of the cost-effectiveness of unbuffered networks may still advocate their implementation in the case of medium- and large-scale multiprocessors [5]. However, one has to remember that packet acceptance of the SEMINs decreases to zero with the network size, while the crossbar performance will be always larger than $(1-e^{-\lambda})\frac{1}{\lambda}$; therefore, if a minimum performance is required (which is always the case), then it may be necessary to implement a full crossbar or another switch architecture. "Augmenting" the network by *replication* or *dilation* may be an alternative solution worth considering [29], [30]. The same cost-effectiveness arguments could be drawn for the unbuffered GSMIN architecture, but will be left for further studies.

In terms of probability of acceptance alone, it is clear that the unbuffered architectures are better suited for handling data generated by parallel algorithms (permutation routing) if, of course, the interconnection capacity of these networks matches the algorithm requirements [16], [7]. Permutation modules composing the GSMIN architecture can be specifically tailored for this purpose (and/or selected at compile time from an optically-switched interconnection module "library"), the same way a configuration sequence can be optimized at compile time for use in a standard time-division multiplexed unbuffered SEMIN [16].

IV. Simulation of a Buffered Architecture

As aforementioned in this paper, we are mainly interested in networks having the ability to handle point-to-point requests, not permutation requests. Unbuffered GSMINs give very poor performance when it comes to the establishment of simultaneous, noncorrelated point-to-point requests. As one may expect, intermediate buffers can improve the overall packet acceptance; infinite buffer capacity would completely eliminate packet loss, but this is not realistic since it does not consider hardware limitations. Early results by Kruskal and Snir showed that as few as *four* packets per buffer at each switching node of a Delta network can approach infinite-buffer performances with uniform traffic [29]. In general, for a given maximum packet-loss probability target, a minimal buffer capacity can be derived. Buffered multistage networks perform well under uniform traffic but degrade severely with even slight nonuniformities (hot-spots) [24].

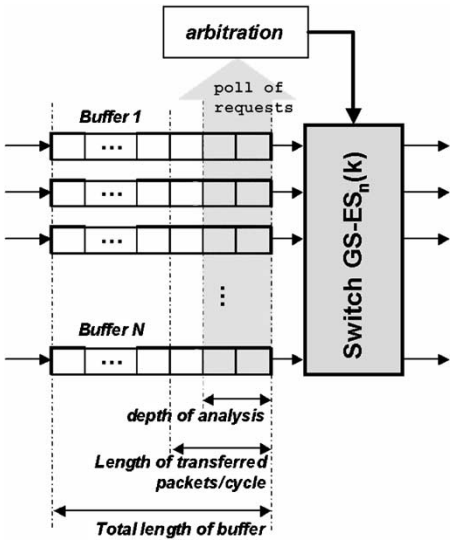

Fig. 9. Scheme of a buffered GSMIN stage, showing relevant flow control parameters.

There are several ways to introduce buffers on the switching network; in the following we consider only *FIFO input buffering* (i.e., FIFO buffers associated with a switching stage are placed right before that stage—Figs. 7 and 9). Even though input buffering is theoretically less efficient than output buffering, this choice is dictated by the fact that a true output buffering implementation may be exceedingly complex [31]. Additional hypotheses for the modeling of our buffered network are

1) the existence of an internal "back-pressure mechanism" activated when there is buffer contention, ensuring that no packet is lost in intermediate stages of the network (but only at the input);
2) no packet *contention* at the output channels (i.e., output channels are always ready to accept a new packet).

An exact analytical model of a buffered multistage network can be extremely difficult (if not impossible) to obtain in most cases, even assuming important simplifications (such as infinite buffer length and uniform traffic characteristics). This is so because of the extreme complexity of the modeling: the state of the network needs to be described in a multidimensional space with at least one dimension per buffer [32]. By introducing more or less arbitrary statistical independence assumptions, some authors identify the state of a whole switching stage to the state of a single switching buffer, which is then modeled using a more or less intricate Markov chain [33], [34]. Because of the very nature of the "global stage" switching decision, the state diagram of a single stage in a GSMIN architecture cannot be simplified using any statistical independency argument. Although it would be possible to describe the stages using a sufficiently intricate Markov chain, given the complexity of the model, it is

likely that the calculation of the steady-state probabilities of the chain would be extremely computationally intensive. Therefore, we preferred to proceed with a direct Monte Carlo simulation of the switching fabric. Moreover, this strategy enables a rapid and flexible control of the network model and traffic parameters.

A. Simulation Parameters

The routing cycle (which takes place synchronously in all stages) comprises 1) an analysis phase in order to select the new state of the global switch; 2) an actual setup phase of the switch; and 3) a data transfer phase. Fig. 9 represents a single stage in a buffered GSMIN architecture. As shown in the figure, a unique queue is associated with each of the N input lines of the global exchange switch $\mathrm{GS} - \mathrm{ES}_n(k)$. The buffer length is assumed to be equal for all ports. Following the remark by Kruskal and Snir [29] that a four-packet sized buffered network already approaches infinite-buffer performance, we are going to limit the buffer-size parameter exploration to a maximum of six packets per buffer. The "tournament" used to select the new state of the switch is done by collecting and analyzing the content of all the buffers up to a maximum "depth of analysis". The counting of votes up to that maximum depth is stopped separately in each buffer whenever there is a change in the value of the vote—otherwise, by taking into account packets that cannot be transferred in that particular cycle, packets that could move forward may be blocked. This "superficial" gathering of votes is another parameter of our model: in fact, we also tested a thoroughly "full depth" voting mode, which makes sense if the buffer queues may be accessed in a random manner during the actual transfer phase (or "hop" mode, see below). Once the number of straight votes (n_s) and cross votes (n_c) is determined, the actual selection of the new switch state (Straight or Cross) can be conducted either in a "fair" manner (i.e., if $n_s > n_c$, the S state is selected, if $n_s < n_c$ the C state is selected, and if $n_s = n_c$ either the S or C state is chosen with *equal* probability), in a "conservative" manner (similar to the fair manner except that when $n_s = n_c$, the state of the switch remains unaltered), or in an "alternate" manner (similar to fair manner, but when $n_s = n_c$, the state of the switch is forced to change). A "forced alternate" selection mode has also been considered: in that case, the state of the global switch is changed at each cycle, regardless of the result of the tournament. This "blind" path selection technique gives, in fact, a buffered TDM switch that would be relatively easy to implement.

During the transfer phase, packets that can move forward go to the next buffer stage (store and forward routing). These are the packets requesting the interconnection just made available during the last setup phase, and for which there is available buffer space in the following stage. Additional model parameters controlling the flow of data are the maximum amount of packets transferable from stage to stage in a single network cycle (if this quantity is larger than unity, then a "burst" of packets can be transferred, provided that there is no "blocking" packet in the queue, and of course provided that there is enough space in the following queue). It seems reasonable that such "internal burst size" should remain smaller than the depth of analysis, but in fact this condition is not mandatory. To further enhance the capacity of the network to avoid possible congestion, an additional "hop mode" of transfer has been studied, which, when enabled, transforms the single-queued FIFO buffer in a multi-queued (logically) segregated buffer: packets that could go if it were not for the head-of-line blocking ones are allowed to "jump" over these and move forward. Last, the traffic is modeled accordingly to the uniform reference model hypothesis.

B. Simulation Results

Both a seven stage ($n = 7,128 \times 128$ input/ouput) IBS-SEMIN and GSMIN networks were simulated for different input request probabilities (Bernoulli parameter λ). The way the routing algorithm is designed implies that the network states (state of buffers and switches) form a Markov Chain (MC). Assuming ergodicity of the stochastic process [35] , the distribution of network states will reach stationarity, regardless of the initial state of the chain. (A mathematical proof of convergence for the proposed routing algorithms is not given here; convergence is a fact tested heuristically by plotting observables like the instantaneous output bandwidth, as is done in most papers dealing with complex buffered switching architectures). This enables us to evaluate steady-state network performance indicators using sample averages [36]. The m sample-long output bandwidth average, $\langle \mathrm{OB} \rangle_m = (1/m) \sum_{k=1}^{m} ob_{1,k}$, (where $ob_{1,k}$ are samples of the instantaneous output bandwidth OB_1), is a consistent and unbiased estimator of the mean of the instantaneous output bandwidth $\langle \mathrm{OB} \rangle$ (Law or Large Numbers for MCs [37]). Therefore, the quantity $\mathrm{PA}_m = (1/\lambda N) \langle \mathrm{OB} \rangle_m$ is a consistent and unbiased estimator of the network performance indicator of interest, namely the probability of packet acceptance, $\mathrm{PA} = \langle \mathrm{OB} \rangle / \lambda N$. The Central Limit Theorem (CLT) also holds for an ergodic MC [37]; however, since the consecutive samples may be strongly correlated, the variance of the $\langle \mathrm{OB} \rangle_m$ estimator includes the "uncorrelated" variance $\mathrm{var}(\mathrm{OB}_1)/m$, but also an m-long sum of sample autocorrelation lags. Therefore, the variance of the limiting normal distribution comprises an infinite sum of autocorrelation lags. These lags as well as the variance of the output bandwidth are unknown, and must be estimated from the samples themselves. A widely used method contouring these difficulties is the method of (nonoverlapping) Batch Means (BM) [38]. The BM method work around the correlation structure by rearranging the data into s subsets of length b (such that $m = s \times b$). Fairly uncorrelated b sample-long averages $\langle \mathrm{OB} \rangle_{b,l} (1 \leq l \leq s)$ can be formed between batches. The BM estimator of the variance of $\langle \mathrm{OB} \rangle_m$ is just the sample variance of the s batch means. We adopted the method by forming $s = 20$ batches each containing $b = 50$ samples of the instantaneous output bandwidth (i.e., $m = 1000$). To further decrease the bias of estimates, the sampling phase was started after discarding data from an initial transient found (heuristically) to be smaller than 50 network cycles (burn-in iterations). This gives a total of 1050 samples for each point in the graphs. Using the estimated variance and the assumption of normal distribution (MC-CLT), a 95% confidence interval for the probability of packet acceptance was calculated. It was found to be smaller than ± 0.03 for all points in the graphs.

A comparative study of the GSMIN performance under URM traffic led us to the following conclusions. First and as

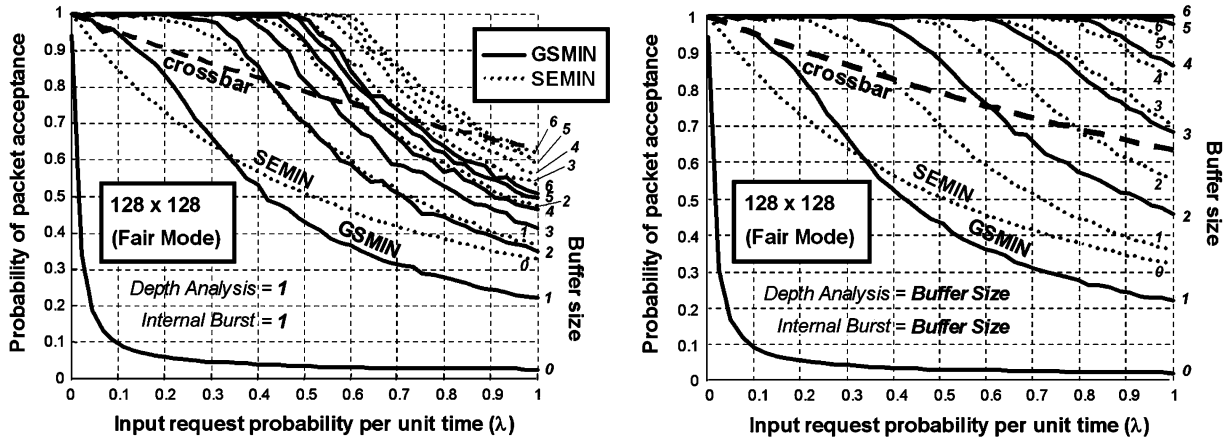

Fig. 10. Performance comparison between a buffered SEMIN (the Inverse Baseline) and the corresponding buffered GSMIN architecture, as a function of the traffic load. The graph on the left shows results using a "fair" switching strategy and a "superficial" depth of analysis and transfer, while the graph on the right represents the results obtained when depth of analysis and transfer are both equal to the total buffer size.

expected, by using intermediate buffers, the performance of both the SEMIN and the GSMIN improve significantly: Fig. 10 quantifies this improvement for different buffer sizes, and also shows the influence of the parameters "depth of analysis" and "internal burst size". The graph on the left has been computed by setting both parameters to one, which means that only the older packets waiting on the queues are given attention in the selection phase, and that only these packets are candidates to be transferred during the transfer phase. The graph on the right shows the performance results from what was observed to be the optimal situation: both parameters are set to their maximum, i.e., the actual buffer size.

Interestingly, as buffer size increase, the performance of a "fair" operated GSMIN improve quickly than does the performance of a standard ("fair" operated) SEMIN. This may be attributed to the fact that a global control of switches somehow breaks locally "frozen" switching states (occurring when the outcome of a request poll for a giving switch leads to a state that cannot be used since the corresponding buffers in following stage are full). For a buffer size equal to three, the GSMIN performance is already equivalent to that of a SEMIN (and is already better than the crossbar's — even at full load). Intermediate settings have been considered (e.g., more or less deep buffer analysis but limited size of transfer, or inversely superficial buffer analysis with, nevertheless, the possibility of a large transfer of buffer content per cycle, up to the maximum buffer size). As expected, these intermediate settings lead to intermediate performance that is not reported here in detail. On the other hand, "hop mode" seems to improve performance considerably: a three-packet-sized hop-enabling buffer leads to a performance equivalent to that of a five-packet-sized FIFO buffer. However, given the complexity required for maintaining a multi-queued buffer, the option will be left aside in the present study.

More interestingly, we found that the "alternate" switch selection gives better performance than "fair" selection for a SEMIN, while GSMIN performance is unaltered (see Fig. 11). Indeed, by using an "alternate" mechanism (which forces the switch to change its state each time the voting result is a draw), a slight improvement in performance is seen in the standard SEMIN for buffer sizes larger than one. However, SEMIN performance still

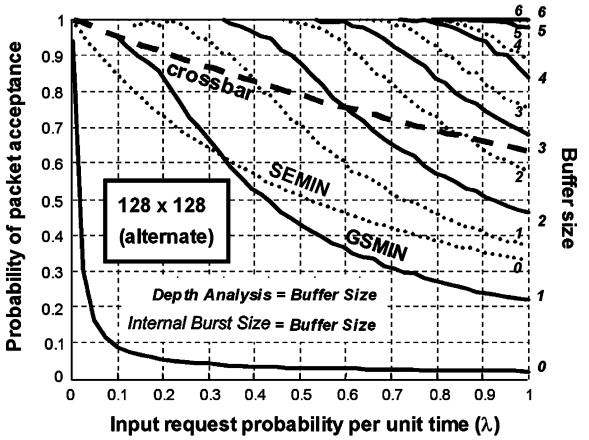

Fig. 11. SEMIN performance slightly improves with the adoption of an "alternate" switch-selection mechanism, while GSMIN performance remains unchanged (compare with Fig. 10).

seems to increase more slowly (with buffer size) than GSMINs, so that when the buffer size is equal to five (or four in the case of a 64×64 network), both networks show roughly the same performance. This improvement in performance may be explained by the fact that alternation of switch states when the poll is evenly distributed corresponds to giving priority to packets that had waited longer on the queues, something similar to the buffer allocation schemes widely used to avoid deadlock in store-and-forward networks [39], but further analysis should be conducted on this issue.

Last and perhaps most interestingly of all, we found that there is no significant change in performance between a buffered GSMIN with "fair" or "alternate" switch selection and a buffered GSMIN with "forced alternate" selection (i.e., no analysis phase at all), at least as far as URM traffic is concerned. At this point, one may think that even a *random* update of switch states will work just as well. However, while random update of switches may break locally frozen states, it does not gives "priority" to older packets, as in alternate (and forced alternate) mode. Moreover, a true random alternation of switch states (either global or individual) may be more complex to implement that a mere deterministic alternation of

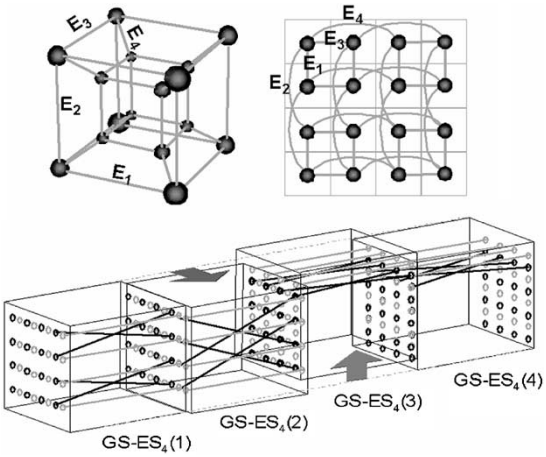

Fig. 12. 4-D hypercube topology mapped on a plane (top); the unbuffered GSMIN (here the GS-BHMIN) architecture using fiber-optic bipermutation modules implements the spanned version of the hypercube network (bottom).

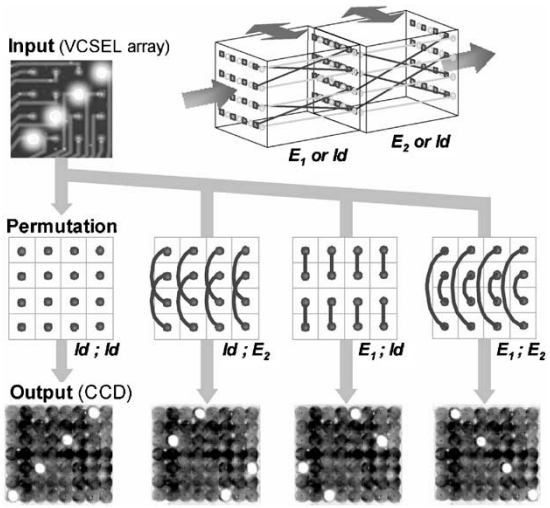

Fig. 13. Experimental demonstration of transparent permutation switching using a pair of fiber-based bipermutation modules.

switch states. Therefore, random update of switches has not been considered in this study.

When a standard SEMIN and a GSMIN are operated in forced alternate mode, they become strictly equivalent architectures (provided that the switch states of the SEMIN network were uniformly set from the start). Therefore, we can say that when a standard SEMIN is operated in forced alternate mode, its performance roughly degrades to that of a -fair or alternate operated- GSMIN (Fig. 11); on the other hand, when a GSMIN is operated in forced alternate mode, its performance remains substantially the same. This is an observation that would open the door to a simple, all-optical hardware implementation (time-division interconnection-multiplexed routing like in [16]), if not for the necessity of buffering (maybe additional interleaved optical functions — consisting in particular of recirculating fiber loops — could be integrated in the cascaded modules). Also, an interesting consequence of the forced alternate operation mode is that, if the addressing of the interleaved permutations is done by an electromechanical system (see SectionV), then such a device could be operated at its resonant frequency, allowing faster switching speeds without the need to provide feedback control mechanisms.

V. Implementation of an Optical GSMIN

As explained earlier, a multistage spanned version of most direct network topologies (hypercube, cube-connected-cycles, deBruijn, etc.) can be implemented as an unbuffered GSMIN architecture using specially designed multiinterconnection modules. Fig. 12 represents a spanned version of the 4-D weakly interconnected plain hypercube (16 nodes, 1 bit wide data bus). It uses four bipermutation modules, each providing a cube permutation and the identity permutation, which gives a total of $2^4 = 16$ global permutations for the whole network (alternatively, using only two of these modules, one can implement a hypercube of dimension 2, with a four bit-wide data-bus).

Two of these four modules were actually fabricated using interleaved optical fibers and the resulting four possible interconnection patterns observed (see Fig. 13). The coupling efficiency between modules (without additional optics, index matching oil, or antireflection coating) was measured to be 1.7 dB, demonstrating the viability of the simple cascaded architecture. More research is needed to establish the optical interconnection bandwidth given a maximum bit error rate in an optimized optomechanical system. Automatic alignment of modules is a critical issue now being studied, both dynamically [40] and statically (prealigned "plug-and-play" exchangeable blocks [41]). A more compact implementation of a bipermutation module may be achieved by interleaving printed lightwave circuits (PLC) implementing, alternately, a (1-D) permutation and the identity permutation, as represented in Fig. 1 (this is possible because the SE permutations are decomposable into row and column independent permutations [42]).

A small electromechanical switching device (much like a pick-up head, but with independent control in two directions — permutation interleaving is not limited to a unique dimension) has also been fabricated and is currently being tested. The switching speed seems to be limited to the millisecond range. Microelectromechanical (MEM) actuators may also be an interesting alternative when switching latency in the millisecond range is tolerable. Though the switching speed can be relatively slow, an appealing characteristic of the proposed mechanical reconfiguration mechanism is that the switch is inherently crosstalk free.

If switching times orders of magnitude faster are required, it is always possible to combine the control lines of individual (2×2) integrated electrooptical switches as proposed in [16]. The functionality of the resulting column-controlled SEMIN is equivalent to that of a (bipermutation based) GSMIN; however, the switching modules would not be "directly" integrated in the sense described in Section II-B. Moreover, the resulting network will suffer from crosstalk at the level of the individual switches. It is possible to contours that problem by "dilating" the network, as proposed in [7]. (The Dilated Slipped Banyan network described in [7] is dilated in the sense that at most one of the two inputs of any 2×2 switch is active at a time. In fact, the DSB can be seen as a particular implementation of a GSMIN, where each bi-permutation module is "indirectly" implemented

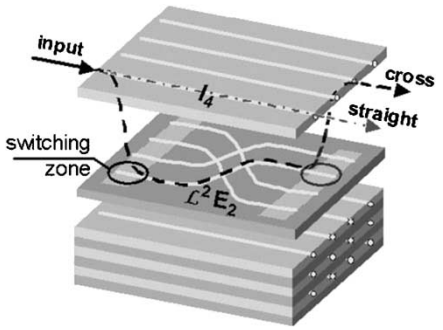

Fig. 14. Possible implementation of an all-optical global switch using sandwiched printed lightwave circuits and electrooptical coupling material.

by combining a column of first-order switches and a passive interconnection stage.) A more compact 3-D implementation of this architecture may be achieved by coupling the planar waveguide of a bipermutation module formed by stacking layers of planar lightwave circuits in the *normal* direction [43] via some interlayer electrooptical material as represented in the Fig. 14.

Still, an interesting research direction is to design higher-order electrooptical switches from the start. This may be done using nonmechanical liquid-crystal (LC)-based reconfigurable holograms [23], [44], or by combining acoustooptical (AO) beam-steering cells with fixed, passive multipermutation modules. Instead of actually translating the module, an acoustooptic (AO) cell placed between two fixed multipermutation modules would *globally* deflect the 2-D array of light beams from the output of one module in order to address the required array of channels at the input face of the following multipermutation module. Since the size of the array of beams may be very small (<1 mm^2 in our fiber-based prototype), an AO cell may be able to swap interconnections in the order of tens of microseconds or less.

VI. Conclusions and Further Research

The multistage interconnection network with globally switched stages (GSMIN) studied in this paper is derived from the standard binary Delta-MIN architecture (whose various representations are known as SEMINs) by operating same-stage sets of elemental switches using a unique control line. (The general problem of controlling subsets of switches in a more general class of Delta-networks is an open issue left for further studies.) The interest of this arrangement lies in its ease of control and implementation, particularly if the multistage architecture is built using dense, 2-D guided-wave-based optical interconnection modules containing several interleaved, independently addressable permutations. By using these plane-to-plane optical interconnection modules, a very compact and scalable system can be implemented. The resulting all-optical circuit-switched network can be tailored to provide the set of permutations satisfying the communication primitives of most static-network multiprocessors. We have presented in this paper preliminary experimental results demonstrating the merits of a simple optical architecture using cascaded fiber-based bipermutation modules. An optomechanical system is being developed that provides switching times on the order of milliseconds, making this architecture suitable for high-bandwidth, permutation routing interprocessor communications tolerating relatively slow reconfiguration times. However, to take full advantage of the huge optical bandwidth of the transparent architecture and the inherently free TDM arbitration mechanism, it would be necessary to further reduce the duration of the TDM time slot down to the nanoseconds range — then the system would be able to simulate asynchronous point-to-point interconnections, and could well represent a cost-effective alternative to the full crossbar.

The performance of an unbuffered GSMIN architecture for establishing arbitrary interconnections in a circuit-switched manner were, as expected, very poor compared to those of a standard unbuffered SEMIN architecture. This is due to the greatly reduced number of available network states, which translates as an extremely reduced overall permutation capacity. However, by simulating a 128×128 (and 64×64) switching fabric, it was qualitatively confirmed that a buffered GSMIN would not require excessive buffer size to achieve respectable performance under URM traffic. The most significant results found in this comparative study of buffered networks are that: a) the performance of a "fair" operated GSMIN evolve quickly with buffer size, and is already superior to that of a standard ("fair" operated) SEMIN for buffer sizes larger than four; and b) when operated in "alternate mode," SEMIN performance improves slightly, but still it seems to increase more slowly (with buffer size) than the GSMIN performance, so that when buffer size is set to five (or four for a smaller 64×64 fabric), both architectures have roughly the same performance ($>90\%$ throughput at full load). Last and most interestingly of all, it was also qualitatively confirmed by simulation that c) the switching mechanism could be reduced to a blind "forced alternation" of switch states without any degradation of performance (at least for URM traffic). Under such a forced alternation mechanism, the SEMIN and GSMIN fabrics become strictly equivalent architectures; hence, provided that the buffer size is chosen to be larger than four (or than three for a 64×64 fabric), this analysis-free strategy will provide a very simple arbitration mechanism for *standard* SEMIN networks. This is an interesting result on its own, and represents to our knowledge the first study on column-controlled *buffered* multistage interconnection networks. Moreover, using an optical module-based GSMIN architecture, this arbitration free paradigm may be very appealing for all-optical networks if optical buffering functions can be integrated on the cascaded global-switching modules themselves, which is an issue worth further investigation.

Also, we would like to determine, for larger switching fabrics, whether the GSMIN performance always evolves more quickly than that of the standard SEMIN as the buffer size increases, as suggested by our preliminary results. If so, given a certain network size, what would be the corresponding minimum buffer size that makes the GSMIN performance surpass the performance of the standard SEMIN? In the case of a forced-alternate mechanism, this size is equal to five for a 128×128 fabric, and to four for a smaller 64×64 fabric. Since complexity (and thus cost) of the GSMIN architecture can be presumably smaller than that of the corresponding SEMIN, this value represents the minimum buffer size that makes the GSMIN a more cost-efficient solution.

Further analysis is required to evaluate other performance indicators (such as average packet delay), and most importantly, the performance of a GSMIN routing paradigm (in a buffered or unbuffered architecture) for other models of traffic. If routing hot-spot traffic seems intuitively a difficult task for the GSMIN, the performances when routing "hot links" may be, on the contrary, very good if the "hot" permutations correspond to the ones made available in the cascaded architecture (the uncorrelated requests being tractable as they were in the URM model).

ACKNOWLEDGMENT

The authors would like to acknowledge two anonymous referees whose valuable comments have been helpful in improving the overall quality of the paper.

REFERENCES

[1] A. Varma and C. Raghavendra, *Interconnection Networks for Multiprocessors and Multicomputers: Theory and Practice*. Los Alamitos, CA: IEEE Computer Society Press, 1994.
[2] *Data Networks*, 1992.
[3] J. Patel, "Processor-memory interconnections for multiprocessors," in *Proc. 6th Ann. Symp. Comput. Architectures*, 1979, pp. 168–177.
[4] D. Dias and M. Kumar, "Packet switching in NlogN multistage networks," in *Proc. IEEE GLOBECOM'84*, Atlanta, GA, 1984, pp. 114–120.
[5] J. Patel, "Performance of processor-memory interconnections for multiprocessors," *IEEE Trans. Comput.*, vol. C-30, pp. 771–780, 1981.
[6] M. Naruse, A. Cassinelli, and M. Ishikawa, "Two-dimensional fiber array with integrated topology for short-distance optical interconnections," in *IEEE LEOS Annual Meeting (Glasgow)—Conf. Proc.*, 2002, pp. 722–723.
[7] R. A. Thompson, "The dilated slipped banyan switching network architecture for use in an all-optical local-area network," *J. Lightwave Technol.*, vol. 9, pp. 1780–1787, Dec. 1991.
[8] C. Clos, "A study of nonblocking switching networks," *Bell Syst. Tech. J.*, vol. 32, pp. 406–424, 1953.
[9] V. Benes, "On rearrangeable three-stage connecting networks," *Bell Syst. Tech. J.*, vol. 41, pp. 1481–1492, 1962.
[10] D. Lawrie, "Access and alignement of data in array processor," *IEEE Trans. Comput.*, vol. C-24, pp. 1145–1155, 1975.
[11] H. Siegel and R. McMillen, "Using the augmented data manipulator network in PASM," *IEEE Comput.*, vol. 14, pp. 25–33, 1981.
[12] L. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. First Ann. Symp. Comput. Arch.*, 1973, pp. 21–28.
[13] C. Wu and T. Feng, "On a class of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-29, pp. 694–702, 1980.
[14] M. Pease, "The indirect binary n-cube microprocessor array," *IEEE Trans. Comput.*, vol. C-26, pp. 458–473, 1977.
[15] I. Scherson and A. Youssef, *Interconnection Networks for High-Performance Parallel Computers*. Los Alamitos, CA: IEEE Computer Society Press, 1994.
[16] C. Qiao and R. Melhem, "Reconfiguration with time division multiplexed MIN's for multiprocessor communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 5, pp. 337–352, 1994.
[17] C. Kruskal and M. Snir, "A unified theory of interconnection networks structure," *Theoretical Computer Science*, vol. 48, pp. 75–94, 1986.
[18] D. Parker, "Notes on Shuffle/Exchange type switching networks," *IEEE Trans. Comput.*, vol. C-29, pp. 213–222, 1980.
[19] M. Sheeran, "Puzzling permutations," *Glasgow Functional Programming Workshop*, 1996.
[20] J. Giglmayr, "Geometrical analysis and design of integrated 3-d lightwave circuits," *Optics in Computing 2000*, vol. SPIE 4089, pp. 969–980, 2000.
[21] R. Cypher and J. L. C. Sanz, *The SIMD Model of Parallel Computation*. New York: Springer-Verlag.
[22] G. Marsden, P. Marchand, P. Harvey, and S. Esener, "Optical transpose interconnection system architectures," *Opt. Lett.*, vol. 18, pp. 1083–1085, 1993.
[23] N. McArdle, M. Naruse, H. Toyoda, Y. Kobayashi, and M. Ishikawa, "Reconfigurable optical interconnections for parallel computing," in *Proc. IEEE*, vol. 88, June 2000, pp. 829–837.
[24] G. F. Pfister and V. A. Norton, "Hot spot contention and combining in multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-34, pp. 943–948, 1985.
[25] D. Bhandarkar, "Analysis of memory interference in multiprocessors," *IEEE Trans. Comput.*, vol. 24, pp. 897–908, 1975.
[26] S. Fong, M. Atiquzzaman, and S. Singh, "An analytical model and performance analysis of shared buffer ATM switches under nonuniform traffic," *Int. J. Comput. Syst. Sci. Eng., Special Issue on ATM Networks*, p. 8194, 1997.
[27] T. Lang and L. Kurisaki, "Nonuniform traffic spots (NUTS) in multistage interconnection networks," *J. Parallel Distrib. Comput.*, vol. 10, pp. 55–67, 1990.
[28] W. D. Strecker, "Analysis of the instruction execution rate in certain computer structures," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 1970.
[29] C. P. Kruskal and M. Snir, "The performance of multistage interconnection networks for multiprocessors," *IEEE Trans. Comput.*, vol. C-32, pp. 1091–1098, Dec. 1983.
[30] M. Kumar and J. Jump, "Performance of unbuffered shuffle exchange networks," *IEEE Trans. Comput.*, vol. C-35, pp. 573–578, 1986.
[31] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. COM-35, pp. 1347–1356, 1987.
[32] T. Theimer, E. Rathgeb, and M. Huber, "Performance analysis of buffered banyan networks," *IEEE Trans. Commun.*, vol. 39, pp. 269–277, 1991.
[33] H. Yoon, K. Y. Lee, and M. T. Liu, "Performance analysis of multibuffered packet-switching networks in multiprocessor systems," *IEEE Trans. Comput.*, vol. 39, pp. 319–327, 1990.
[34] Y. C. Jenq, "Performance analysis of a packet switch based on single-buffered banyan network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 1014–1021, 1983.
[35] C. Geyer, "Practical markov chain monte carlo," *Stat. Sci*, vol. 7, no. 4, pp. 473–483, 1992.
[36] L. Tierney, "Markov chains for exploring posterior distributions," *Ann. Stat.*, vol. 22, pp. 1701–1762, 1994.
[37] K. Chan and C. Geyer, "Comment on 'Markov chains for exploting posterior distributinos' by L. Tierney," *Ann. Stat.*, vol. 22, pp. 1747–1758, 1994.
[38] A. Law and W. Kelton, *Simulation Modeling and Analysis*, 3rd ed. New York: McGraw-Hill, 2000.
[39] P. M. Merlin and P. J. Schweitzer, "Deadlock avoidance in store-andforward networks," *IEEE Trans. Commun.*, vol. 28, pp. 345–354, 1980.
[40] M. Naruse, S. Yamamoto, and M. Ishikawa, "Real-time active alignment demonstration for free-space optical interconnections," *IEEE Photon. Technol. Lett.*, vol. 13, pp. 1257–1259, 2001.
[41] A. Goulet, M. Naruse, and M. Ishikawa, "Integration technique to realize alignment-free opto-electronic systems," *OC2001 Int. Topical Meeting on Opt. Computing (Lake Tahoe), Tech. Dig.*, pp. 122–124, Jan. 2001.
[42] A. Cassinelli, M. Naruse, M. Ishikawa, and F. Kubota, "A modular, guided wave approach to plane-to-plane optical interconnects for multistage interconnection networks," *Extended Abstr. Opt. Japan 2002 Conf., JSAP, Koganei, Tokyo*, pp. 124–125, 2002.
[43] B. Kim, A. Shakouri, B. Liu, and J. Bowers, "Improved extinction ratio in ultra short directional couplers using asymmetric structures," *Japan J. Appl. Phys.*, vol. 37, pp. 930–932, 1998.
[44] X. Wang, D. Wilson, R. Muller, P. Maker, and D. Psaltis, "Liquid-crystal blazed-grating beam deflector," *Appl. Opt.*, vol. 39, pp. 6545–6555, Dec. 2000.

Alvaro Cassinelli was born in Montevideo, Uruguay. He received a Graduate Engineering diploma from the Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1996. He received the Doctoral Qualifying degree (DEA) in laser and matter interaction from the University of Paris-XI/ENST/Ecole Polytechnique the same year. In 2000, he received the Ph.D. degree from the University of Paris-XI Orsay for his work on optoelectronic stochastic parallel processors for image processing.

Since 2001, he has been a Research Fellow with the Department of Information Physics and Computing, University of Tokyo, Tokyo, Japan. His present research interests lie in the area of the fundamental aspects of computing and telecommunications, and their practical implementations.

Makoto Naruse received the B.E., M.E., and Dr. Eng. degrees from the University of Tokyo, Tokyo, Japan, in 1994, 1996, and 1999, respectively.

From 1999 to 2002, he was a Postdoctoral Researcher and a Research Associate with the University of Tokyo. In 2002, he joined the Communications Research Laboratory, Incorporated Administrative Agency, Tokyo, where he researches in the field of optical interconnections and photonic networks.

Masatoshi Ishikawa received the B.S., M.S., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 1977, 1979, and 1988, respectively, all in engineering.

He is a Professor with the Department of Information Physics and Computing, Graduate School of Engineering, University of Tokyo. He has been employed by the Ministry of International Trade and Industry in the Industrial Products Research Laboratory. His research interests include optoelectronic computing, parallel processing, machine vision, smart sensors, tactile sensors, sensor data fusion, and robotics.