Design Principles for Practical Self-Routing Nonblocking Switching Networks with $O(N \cdot \log N)$ Bit-Complexity

Ted H. Szymanski, *Member*, *IEEE Computer Society*

Abstract—Principles for designing practical self-routing nonblocking $N \times N$ circuit-switched connection networks with optimal $\theta(N \cdot \log N)$ hardware at the bit-level of complexity are described. The overall principles behind the architecture can be described as "Expand-Route-Contract." A self-routing nonblocking network with w-bit wide datapaths can be achieved by expanding the datapaths to $w + z$ independent bit-serial connections, routing these connections through self-routing networks with blocking, and by contracting the data at the output and recovering the w-bit wide datapaths. For an appropriate redundancy z, the blocking probability can be made arbitrarily small and the fault tolerance arbitrarily high. By using efficient space domain concentrators, the architecture yields self-routing nonblocking switching networks with an optimal $O(N \cdot \log N)$ bits of memory or $O(N \cdot \log N \cdot \log \log \log N)$ logic gates. By using a linear-cost time domain concentrator, the architecture yields self-routing nonblocking switching networks with an optimal $\theta(N \cdot \log N)$ bits of memory or logic gates. These designs meet Shannon's lower bound on memory requirements, established in the 1950s. The number of stages of crossbars can match the theoretical minimum, which has not been achieved by previous self-routing networks. The architecture is feasible with existing electrical or optical technologies. The designs of electrical and optical switch cores with Terabits of bisection bandwidth for Networks-of-Workstations (NOWs) are described.

Index Terms—Multistage, networks, self-routing, nonblocking, circuit-switching, scalable, randomization, electrical, optical.

1 Introduction

An $N \times N$ nonblocking "Connection Network" is a circuit-switched network capable of achieving any of the $N!$ permutations of its N input ports onto its N output ports [35]. Such networks are often used for ATM switching or multiprocessor communication. The "hardware cost" is defined as the number of logic gates or bits of memory required in its construction. The "depth" is defined as the number of logic gates along the longest path between an input port and an output port. A "self-routing" network is one in which a circuit-switched connection can be established by the hardware as it propagates forward through the network, with reliance only on the local information available at each node; there is no need for any "off-line" path precomputation. The "setup time" is defined as the propagation delay of all logic gates traversed in the establishment of a connection between some input port and some output port.

This paper presents a design for practical and "scalable" connection networks, i.e., nonblocking switching networks which easily and efficiently scale to large sizes. This problem is historically significant and it is important in the design of Gigabit and Terabit optical networks [36]. Using recently developed free-space optical technologies [2], [12], [18], [21], complex electronic switching nodes can be implemented on an *Opto-Electronic Integrated Circuit* (OEIC) with optical I/O. Based upon industry projections [2], [18], [28], within a decade, OEICs containing millions of electronic logic gates and thousands of optical I/Os will be feasible. Each OEIC can implement one stage of an optical multistage network. The optical output of one stage can be fed into the next stage through free-space, where the permutation between stages can be implemented optically. The appeal of these networks is their very high bisection bandwidths (in the Terabit per second range) and the simplicity of their construction, since all interstage wires are implemented optically. It is important to minimize the number of stages in an optical network to minimize cost and maximize reliability. It is also important to avoid complicated backtracking control algorithms within the network, which are infeasible to achieve optically. Perhaps surprisingly, the new optical technologies are highlighting the need for good solutions to historic networking problems, such as the design of scalable switching networks with fast self-routing control algorithms. *(A complete set of graphs which allows a reader to design nonblocking networks is presented in Section 2.)*

OEICs with hundreds of binary circuit-switching nodes have been developed in many different smart pixel technologies, i.e., [8], [9], [12], [21], [34], [39]. Researchers at the former AT&T have demonstrated a circuit-switched optical multistage network with 60K optical channels which used off-line routing [8], [9], [12]. In spite of the considerable industrial interest in optical multistage networks, to date, there does not exist an efficient scalable nonblocking circuit-switching network architecture with fast self-routing algorithms and with a hardware complexity which is asymptotically optimal.

• *The author is with the Department of Electrical Engineering, McGill University, Montreal, PQ, Canada H3A 2A7.*
E-mail: teds@macs.ee.mcgill.ca.

Manuscript received 9 Aug. 1993; revised 10 June 1996 and 17 May 1997.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 105193.

A recent report sponsored by the U.S. National Science Foundation (NSF) entitled "*Research Priorities in Networking and Communications*" [36] defined 15 key research priorities for the next decade. These priorities include "*Dynamic Network Control*," i.e., the need for fast routing algorithms to control the Gigabit and Terabit networks of the future, and "*Switching Systems*," i.e., the need for optimally scalable connection networks. According to the NSF report, "*Realization of these goals require new advances in switching system theory and design. To date, there are no practical architectures for nonblocking multipoint virtual circuit switches that can meet the theoretical limits on optimal scaling with respect to all the characteristics of practical concern (switching network complexity, routing memory requirements) and most systems now being used have poor scaling properties*" [36].

The "Expand-Route-Contract" (ERC) network architecture proposed in this paper represents one step toward these goals. The ERC network can meet Shannon's asymptotic lower bound on the hardware complexity of self-routing nonblocking networks (see next paragraph), can scale optimally to Gigabit and Terabit bandwidths associated with optical technologies and allows for simple and very fast network control algorithms which are provably immune to congestion.

In the 1950s, Claude Shannon established a lower bound on the hardware complexity of self-routing nonblocking circuit-switching networks which are provably immune to congestion (equivalently, they never exhibit blocking given any permutation traffic pattern) [29]. According to Shannon's complexity argument, an optimal self-routing circuit-switched connection network would require $\theta(N \cdot \log N)$ hardware,[1] which includes all bits of memory [29], all logic gates, and all crosspoints, and would have a depth of $O(\log N)$ binary nodes. To date, no known self-routing nonblocking circuit-switching networks with explicit constructions meet Shannon's lower bounds. The famous AKS sorting network [1] meets Shannon's lower bounds in the limit of infinitely large sizes N, but it relies on linear cost concentrators which lack explicit constructions, i.e., it cannot be built in practice. The MultiBenes network proposed by Avora, Leighton, and Maggs [3] also meets Shannon's lower bounds, but it also relies on expander graphs which lack explicit constructions, relies on complex backtracking control algorithms, and it requires an AKS sorting network to acknowledge calls. These two networks are primarily of theoretical significance, and established that Shannon's lower bounds on the cost and depth of self-routing connection networks can be met in theory, but not in practice.

We point out that a "store-and-forward" packet-switched network, where packets are buffered in each stage, could not possibly meet Shannon's lower bound on cost. Each packet requires at least $O(\log N)$ bits to identify its destination, and, if packets are buffered in every stage, then the hardware complexity of the network is at least $\theta(N \cdot \log^2 N)$ bits of memory, which is suboptimal by a factor of $O(\log N)$. In addition, packet-switched networks are undesirable since they are slow [27]. A pipelined circuit-switched network with fast connection establishment can deliver a permutation of packets from its input side to its output side in roughly the same amount of time a packet-switched network requires to move packets forward one stage. For these reasons, pipelined circuit switching and the similar worm-hole routing technique have largely eliminated packet switching in recent multicomputer networks [27]. (Packet-switched networks using randomized routing are described in [23], [37], [38].)

In practice, many self-routing permutation networks are based on bit-serial circuit-switched versions of Batcher's sorting network [6], with $\theta(N \cdot \log^2 N)$ binary nodes, $\theta(\log^2 N)$ depth, and $\theta(\log^2 N)$ setup time. These complexities are expressed at the "bit-level," and include all crosspoints, all bits of internal memory, and all logic gates, where every logic gate is assumed to have bounded fan-in and bounded fan-out. There have been some innovative switch designs over the years. Douglass has proposed a rearrangeable network with $O(N \cdot \log^{2.5} N)$ hardware and $O(\log^{2.5} N)$ setup time [11]. Chien and Oruc propose permutation networks with $O(N \log N \cdot \log \log N)$ bit cost and with $O(\log^3 N)$ bit delay [7]. Using a numerical analysis, De Biase et al. proposed permutation networks with $O(N \cdot \log^{2+\varepsilon} N)$ bit cost (where $\varepsilon > 0$) and $O(\log N)$ bit delay [10]. The complexity of various networks is illustrated in Table 1. *However, the discrepancies between the best-known theoretical results and practical results are evident in Table 1.*

In this paper, we propose an architecture for self-routing nonblocking, circuit-switched connection networks, which we call the "*Expand-Route-Contract*" (ERC) architecture. An overview of the architecture is shown in Fig. 1. The nonblocking ERC architecture is based on the concept of expanding the incoming data, routing the bits through independent bit-serial networks which exhibit blocking, and compacting the data at the output. The architecture is also based on probabilistic schemes and randomization, rather than on deterministic schemes. The expansion can be accomplished in at least two ways:

1) A w-bit wide data word can be encoded with a Forward Error Correcting Code to yield a $w + z$ bit wide word or
2) The w-bit wide data word is submitted to an expander creating a $w + z$ bit wide word.

After the expansion, the $w + z$ bits of data are routed through $w + z$ independent bit-serial self-routing circuit-switched networks, which we call "bit-planes." Each self-routing bit-plane attempts to establish bit-serial circuit-switched connections, and each bit-plane is allowed to have an arbitrary blocking probability. (The bit-serial connections can also be bit-parallel). The expansion z is a design parameter which is chosen so that the probability that a w-bit wide connection is established is sufficiently large. Given any level of blocking in the bit-planes, it is always possible to pick the expansion z so that the "mean-time-between-blocking" of a w-bit wide connection is an arbitrarily large amount of time, for example 10^{50} years. It is important to recognize the strength of this probabilistic approach: There are only about 10^{15} years left in the life of our universe, and, hence, these networks can be designed so that the mean-time-between-blocking can exceed the remaining life of our universe.

1. All logarithms are to the base 2 unless otherwise indicated.

TABLE 1
ASYMPTOTIC COMPLEXITIES OF VARIOUS NONBLOCKING CIRCUIT-SWITCHED NETWORKS

Network	Self-Routing	Explicit ?	Memory Bit-Complexity	Setup-Time
Crossbar	yes	yes	$\Omega(N^2)$	$\Omega(N)$
Clos	No	yes	$\Omega(N^{3/2})$	-
Cantor	No	yes	$\theta(N \log N)$	-
Benes	No	yes	$\theta(N \log N)$	-
Douglass	No	yes	$\theta(N \cdot \log^{2.67} N)$	-
AKS sort	yes	No	$\theta(N \log N)$	$\theta(\log N)$
ALM	yes	No	$\theta(N \log N)$	$\theta(\log N)$
Batcher sort	yes	yes	$\theta(N \log^2 N)$	$\theta(\log^2 N)$
Koppelman-Oruc	yes	yes	$\theta(N \log^2 N)$	$\theta(\log^2 N)$
Chien-Oruc	yes	yes	$\theta(N \cdot \log N \cdot \log\log N)$	$\theta(\log^3 N)$
De Biase et. al.	yes	yes	$\approx \theta(N \log^{2+\varepsilon} N)$	$\theta(\log N)$
ERC	yes	yes	$\theta(N \log N \cdot \log\log N)$	$\theta(\log N)$
ERC-space	yes	yes	$\theta(N \cdot \log N)$	$\theta(\log N \cdot \log\log\log N)$
ERC-time	yes	yes	$\theta(N \cdot \log N)$	$\theta(\log N \cdot \log\log N)$

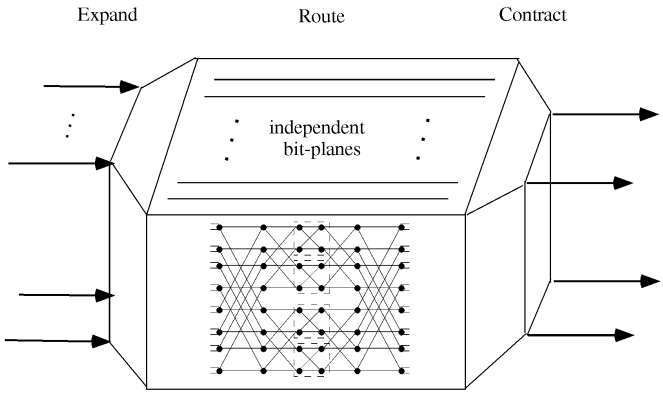


Fig. 1. Example of the proposed switch architecture, based on expansion, routing, and contraction.

In order to keep the overall cost at $\theta(N \cdot \log N)$, the self-routing bit-planes must have $\theta(N \cdot \log N)$ bit complexity, and the blocking probability of any bit-plane must not approach one, so that the required expansion (based on z) is bounded by a constant factor. It has recently been established that self-routing dilated banyans can be designed to have $\theta(N \cdot \log N)$ bit complexity and a blocking probability (denoted pb) which approaches zero as N approaches infinity [33]. In this paper, it is shown that the dilated banyans with low blocking probabilities can be used in the proposed "Expand-Route-Contract" architecture to yield hardware-efficient nonblocking switches with $\theta(N \cdot \log N)$ bit complexity (where the nonblocking property is based on probabilistic arguments). These nonblocking switches can meet Shannon's lower bound first established in the 1950s on the hardware complexity of self-routing nonblocking switches. (We note that the ERC architecture can yield a nonblocking network using any self-routing bit-planes, as long as the blocking probability in the bit-planes is less than one. We also point out that the bit-serial connections can be replaced by bit-parallel connections.)

In practice, each bit of high-speed memory has an equivalent cost of nearly 10 logic gates. Hence, minimization of memory requirements is often more important than minimization of logic gates [29]. The proposed ERC network can be designed with asymptotically optimal memory requirements. In summary, the proposed ERC connection network architecture has straightforward explicit constructions, uses very simple and fast routing algorithms which are easily implemented in hardware, and has very fast setup times when compared to other known networks. This paper is organized as follows. Section 2 presents a brief review of multipath delta networks, and derives some upper bounds on the blocking in multipath delta networks. Section 3 describes the principles behind the ERC architecture. Section 4 discusses SDM and TDM constructions of multipath delta networks and derives asymptotic complexities. Section 5 describes the application of the theory to the design of electrical and optical networks, and Section 6 contains concluding remarks.

2 BLOCKING IN MULTIPATH CIRCUIT-SWITCHED DELTA NETWORKS

Delta networks are banyan networks with the self-routing property [26]. A d-dilated delta network [4], [19], [31], [32], [33] can be obtained from a regular banyan by increasing the capacity of each edge to handle up to d connections simultaneously, and by replacing all the crossbar switches with dilated crossbar switches. Each logical input port to a dilated crossbar can receive up to d connections simultaneously, and each logical output port of a dilated crossbar can transmit up to d connections simultaneously. A two-dilated delta network is shown in Fig. 2a. The theorems in this paper will apply to dilated delta networks, and the more general *multipath* delta networks (see next paragraph) have comparable performance.

A "p-path delta" network can been defined as a multipath delta network with the following property: In every stage where a routing decision must be made, there exist p alternate paths which lead to a given destination [32]. Therefore, in every stage, a p-path delta network has at least p suitable alternate paths which a connection could take while moving toward its destination. A two-path delta network is shown in Fig. 2b.

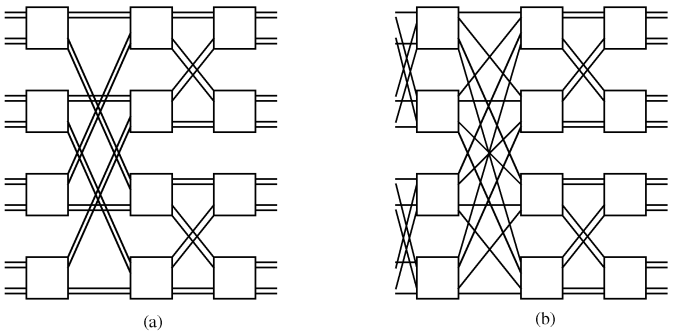


Fig. 2. (a) Dilated delta network built with two-dilated 2 × 2 crossbar switches. (b) Multipath delta network built with two-dilated 2 × 2 crossbar switches.

In this section, simple proofs are proposed which establish that

1) $\theta(N \cdot \log\log N)$connections will survive when routed through a p-path $N \times N$ delta network with a path multiplicity of $p = \theta(\log\log N)$, and
2) that the blocking probability pb of an individual connection approaches zero as N approaches infinity, given a loading of $\theta(\log\log N)$ connections per I/O port. (Some looser bounds were presented in [33].)

Readers who are not interested in the mathematical proofs may proceed directly to Section 2.5, where the numeric results are discussed, without a loss of continuity.

Consider a circuit-switched d-dilated $b^n \times b^n$ delta network, with $N \equiv b^n$ logical input ports and N logical output ports. (Note: A logical port has a capacity to support d connections.) Let there be h connection requests applied at each logical network input port ($h \le d$). Connection requests are randomly distributed over the logical output ports. Connection requests flow from the input side to the output side. Whenever $d + 1$ or more requests attempt to exit a logical output port in stage i, d requests are selected at random and propagated forward and the remainder are blocked.

Let the random variable N_{in} denote the number of requests entering the network, let N_{out} denote the number of requests leaving the network, and let $N_{blocked}$ denote the number of connection requests blocked within the network (each variable assumes a value given a specific state of the network). Define the acceptance probability as $pa \equiv E[N_{out}]/e[N_{in}]$ and the blocking probability as $pb \equiv E[N_{blocked}]/E[N_{in}]$, where the expectation is taken over all states of the network. It follows that $pb = 1 - pa$. (These probabilities are conditional on the fact that a request exists initially). Theorem 1 yields a concise upper bound on the blocking in a dilated crossbar switch and is stated without proof. The proof uses Valiant's version of Chernoff's bound [38] after application of Hoeffding's result [13], to yield a closed form expression on the tail of a binomial distribution.

THEOREM 1. *Given a random uniform traffic model, the conditional blocking probability pb in a d-dilated $N' \times N'$ crossbar, with h connection requests sourced at each logical input port is upper bounded by*

$$pb \le \left(\frac{eh}{d}\right)^d \cdot e^{-h}.$$

2.1 Blocking Probability in a Multipath Delta Network

THEOREM 2. *Given a random uniform traffic model, the conditional blocking probability in a self-routing d-dilated $b^n \times b^n$ delta network with h connection requests sourced at each logical input port is upper bounded by*

$$pb \le n \cdot \left(\frac{eh}{d}\right)^d \cdot e^{-h}.$$

PROOF. Suppose that all logical output ports in all stages are assigned unique labels $L_{i,j}$ for $0 \le i \le n$, $0 \le j \le N - 1$, where $N \equiv b^n$. A "path" through a network is defined as a sequence of dilated edges (i.e., edges with a capacity of d connections). A connection request (for a circuit-switched connection) follows a particular path to its destination as it is routed through the network; if it encounters a saturated edge, it blocks, otherwise, it survives. Define $B_{i,j}$ as the number of connection requests that block at $L_{i,j}$ in a given state. By definition, the end-to-end conditional blocking probability is given by

$$pb \equiv \frac{E\left[N_{blocked}\right]}{E\left[N_{in}\right]} = \frac{E\left[\sum_{s=1}^{n}\sum_{l=0}^{N-1} B_{l,s}\right]}{Nh}.$$

Consider a specific topology, the omega-inverse network, in this section (the result applies to all topologically equivalent delta networks). Due to the random uniform traffic model, the paths must be evenly distributed over the output ports in the last stage of the network. Assuming that there was no blocking in stages 1 to $n - 1$, then the entire network can be viewed as a d-dilated $N \times N$ crossbar (where $N = b^n$), where the blocking occurs only at the output ports. By applying the upper bound from Theorem 1, it follows that the expected number of requests which block at the output ports of the last stage is given by

$$E\left[\sum_{l=0}^{N-1} B_{l,n}\right] \le \left(\frac{eh}{d}\right)^d \cdot e^{-h} \cdot (Nh).$$

The first $n - 1$ stages of the network can be viewed as two smaller $N/2 \times N/2$ dilated banyans. By repeated application of the above argument on the rest of the network, the expected number of blocked requests in the d-dilated $b^n \times b^n$ delta network is upper bounded as follows;

$$E\left[\sum_{s=1}^{n}\sum_{l=0}^{N-1} B_{l,s}\right] \le n \cdot \left(\frac{eh}{d}\right)^d \cdot e^{-h} \cdot (Nh).$$

Therefore, the conditional blocking probability of the entire network is upper bounded as follows;

$$pb \le n \cdot \left(\frac{eh}{d}\right)^d \cdot e^{-h}.$$

□

Theorem 2 bounds the blocking probability in a *multipath* delta network given a random uniform traffic model.

2.2 Worst-Case Traffic Immunity and Randomized

Routing

In the worst case, d-dilated $N \times N$ delta network can establish only $O(\sqrt{N} \cdot d)$ connections simultaneously. For example, in an eight-dilated 64K × 64K banyan, the worst-case traffic pattern only allows about three percent of all connections to pass through. To ensure immunity to worst-case traffic, it is sufficient to "randomize" the traffic first, through the addition of another delta network [23], [38]. The concatenation of two dilated delta networks (one for randomization and one for routing, with the innermost stages merged) yields a class of general topologies which could be called "*Tandem Dilated Banyan*" networks, which include the "*Dilated Benes*" topology as one example. A more general network obtained from the concatenation of two multipath delta networks yields a class of topologies which could be called "*Tandem Multipath Delta*" networks, which includes the "*MultiBenes*" topology [3] as one example.

Every connection request picks a random destination at the output of the randomization network and then attempts to establish a connection to that destination. Any given traffic pattern, including a worst-case pattern, is transformed into a random traffic pattern in the randomization network [20], [23], [38]. Requests then attempt to establish connections to their original destinations through the routing network. This approach eliminates congestion due to worst-case traffic patterns, as will be shown. (Note: In practice, the randomizer can be operated in various manners; see Section 2.4.)

To date, no researchers have managed to rigorously derive a bound on the blocking probability of self-routing circuit-switching networks using randomized routing. Leighton addresses the problem of deriving a rigorous proof in his textbook. The connection requests which have survived through a randomizer are not randomly distributed over its output links: Their positions are correlated and it is not known how to bound the blocking given correlated traffic models. The difficulty of handling correlated traffic models and the unsolved nature of the problem is described in [20].

In this section, we present an alternative approach to bound the blocking in dilated delta networks using randomized routing. A key point of the proof is to note the distinction between "*paths*" and "*surviving connection requests*." At any stage, a "surviving connection request" is essentially the front-end of a pipelined circuit-switched connection, or, equivalently, the front-end of a worm-hole routed connection. The surviving connection requests exiting the randomizer are not randomly distributed over its output ports, as observed by Leighton. However, the paths of all connection requests must be randomly distributed over the output ports, since the path destinations are selected at random. Hence, if we assume all connection requests are surviving connection requests at any given stage, we can exploit the fact of random path destinations and thereby upper bound the blocking at the given stage. We may then exploit the symmetry between the randomizing network and the routing network to bound the blocking in the routing network. Since the loading at each end is deterministic and identical (h paths per logical port) and the loading distribution at the middle is identical, then the pattern of paths is symmetric about the middle. We may then establish that the upper bound from Theorem 2 is valid in each network; this is formalized in the next theorem.

THEOREM 3. *Given the concatenation of two d-dilated $N \times N$ banyan networks, with $h < d$ connection requests at each input port, which are randomly and uniformly distributed over the outputs of the first dilated banyan, such that each output port of the second dilated banyan is the destination of precisely h connection requests, the expected number of blocked requests in the second dilated banyan is upper bounded as follows:*

$$E\left[\sum_{s=n+1}^{2n}\sum_{l=0}^{N-1} B_{l,s}\right] = E\left[\sum_{s=1}^{n}\sum_{l=0}^{N-1} B_{l,s}\right] \le n \cdot \left(\frac{eh}{d}\right)^{d} \cdot e^{-h} \cdot (Nh)$$

$$\rightarrow pb \le n \cdot \left(\frac{eh}{d}\right)^{d} \cdot e^{-h}.$$

PROOF. Follows by symmetry from the proof of Theorem 2, noting that the expectation is upper bounded by considering paths which have random destinations, rather than surviving connection requests which have correlated destinations, and observing that the set of paths over which the expectation is taken in the routing network is symmetric with the set of paths in the randomization network, and observing that the direction of flow of connections is irrelevant. □

Theorem 3 establishes an upper bound on the blocking in the routing network given any worst-case traffic pattern, and is necessary to establish the existence of self routing nonblocking ERC networks which are immune to "worst-case" traffic patterns in Section 3.

2.3 Asymptotic Performance

We now consider the blocking in a dilated delta network as the network size scales toward infinity.

THEOREM 4. *Given the concatenation of two dilated banyans, one acting as a randomization network and one acting as a routing network, then $pb \rightarrow 0$ as $N \rightarrow \infty$ through the appropriate choice of h and d.*

PROOF. Let the dilation be $d = K \cdot \lceil \log \log N \rceil$ for constant $K \ge 1$ and the number of traffic sources at each input port h be such that $(eh/d) \le 1/2$, then

$$\lim_{n\to\infty} pb \le \lim_{n\to\infty} n \cdot \left(\frac{1}{2}\right)^{K\cdot\log\log N} \cdot e^{-K\cdot\log\log N/2e} \le \lim_{n\to\infty}\left(\frac{n}{n^{K+c}}\right) = 0$$

(since $c > 0$ and $n = \log N$). □

Theorem 4 establishes that $pb \rightarrow 0$ as $N \rightarrow \infty$ for the concatenation of two self-routing dilated Delta networks. These networks can be made to have an arbitrarily low pb and immunity to worst-case traffic patterns by the appropriate choice of h and d. (We note that even faster convergence to zero can be obtaining by selecting a wider dilation $d = K \cdot \lceil \log N \rceil$, in which case, $pb \le \lim_{N\to\infty} n/N^{K+c} = 0$, although, in practice, this larger dilation is not necessary. Wider dilations are useful in an asymptotic analysis where all N connections in a permutation do not block simultaneously.)

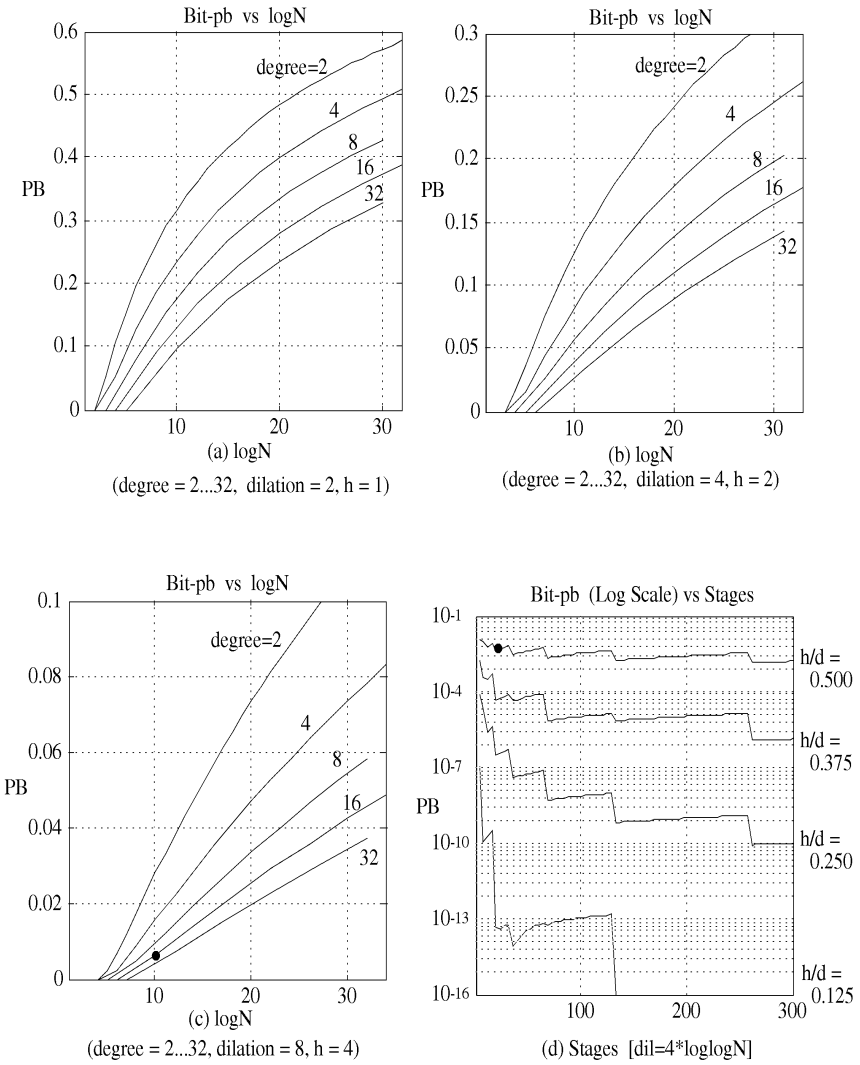


Fig. 3. Blocking probability (bit-pb) versus stages for various dilated banyans: (a) fixed dilation = 2, half-loaded, (b) fixed dilation = 4, half-loaded, (c) fixed dilation = 8, half-loaded, (d) dilation = load = O(log log N). Blocking probability approaches zero when dilation grows slowly. Bold dot represents design example discussed in Section 5.

2.4 Variations

In practice, a number of techniques can be used to either reduce the cost or lower the blocking probability. Blocking in the randomization network can be eliminated by having the crossbar switches always propagate connection requests forward in pseudorandom directions. Each switch in the randomizer can select a nonblocking state at random; the incoming connections are permuted and always propagated forward. The cost of the randomizer can also be reduced by using a one-dilated crossbar rather than *d*-dilated crossbars. This approach eliminates blocking in the randomizer and reduces the cost of the randomizer.

Alternatively, in practice, the blocking probability can be reduced by employing deflection routing in the randomization network, i.e., see [9]. The randomization network attempts to route requests to their intended destinations. If a request encounters a congested link in stage *i*, it is deflected and propagated out over the wrong link. After exiting the randomization network, some requests will arrive at their destinations, while others will arrive at incorrect destinations, and all requests are launched into the routing network. In the routing network, the requests which were deflected in the randomization network will have another chance to be routed to their destination. The deflection routing algorithm results in a lower end-to-end blocking probability than predicted by Theorems 2 and 4, but is more complex to implement electronically.

2.5 Numeric Results

In this section, the blocking probability of a dilated banyan based routing network is plotted against various parameters. Exact analytic models for the blocking probability of dilated banyans under random uniform traffic have been published in [19], [32]. The reader is referred to those papers for details.

The blocking probability of a dilated banyan can be reduced by operating at lower loads, i.e., by separating an "active" input port which supports connections by one or more "idle" input ports. This approach was also used by Avora, Leighton, and Maggs to lower the load in the

MultiBenes network [3]. To lower the load in our system, we assume that each input port which has the capacity to source up to d connection requests actually sources fewer requests, i.e., for a half-load, each dilated input port sources $h = d/2$ connection requests.

The blocking probabilities of various dilated banyans, with various dilations and, at half load ($h = d/2$), are plotted against the number of stages in Figs. 3a, 3b, and 3c. (Blocking in the last stage is eliminated in our traffic model, since at most $h \le d$ connections arrive at any one logical output port). Given a fixed dilation, the *pb* will approach one as $N \to \infty$, as indicated by Theorem 2, although it does so very slowly. (In all figures, the bold dots represent an eight-dilated banyan which will be used in the optical design in Section 5.)

According to Theorem 4, for *pb* to approach 0 as $N \to \infty$, the dilation and loading must grow slowly with N, i.e., $d = \lceil \log \log N \rceil$ and $h = O(d)$. The blocking probabilities of various dilated banyans which meet the conditions of Theorem 4, with a dilation of $\lceil 4 \cdot \log \log N \rceil$ and at four different loadings (h/d = 0.125, 0.25, 0.375, and 0.5), are plotted in Fig. 3d. The number of stages is set to a very large value (300 stages), so that the asymptotic limits of the curves can be examined. As established in Theorem 4, asymptotically *pb* $\to 0$ as $N \to \infty$. Hence, dilated banyans with fast self-routing algorithms can be designed to have arbitrarily small *pbs* as the network size scales to infinity. These results will be necessary in Section 3 to derive nonblocking ERC networks (based on a probabilistic approach) from networks with blocking .

3 THE "EXPAND-ROUTE-CONTRACT" NONBLOCKING SWITCH ARCHITECTURE

In conventional circuit-switching connection networks, the connection datapaths are usually many bits wide, typically eight, 16, or 32 bits. Typically, all the bits in a connection datapath are switched together as an indivisible entity. If the circuit-switched connection blocks, then all bits in the connection block simultaneously. Similarly, if one bit in the datapath fails, then the entire connection fails.

The proposed ERC architecture relies on a fundamentally different approach. In order to establish a w-bit wide circuit-switched connection in the ERC network, at the input side, $w + z$ independent bit-serial connection requests are inserted into the network (where $w \ge 1$). Each bit-serial connection is routed through a circuit-switched network called a "bit-plane," typically a one-bit wide dilated banyan with a finite blocking probability. At the output side, all surviving bit-serial circuit-switched connections are contracted together, and a w-bit wide datapath is established if w or more bit-serial connections have survived. In principle, the bit-planes can be any self-routing bit-serial circuit-switching networks with blocking, including the conventional bit-serial banyan network. (In principle, we could also use a Forward-Error Correcting code which can correct z bit-errors to expand a w-bit data word to $w + z$ bits at the input side, and the decoder to contract $w + z$ bits to a valid w-bit code word at the output side. A blocked bit-serial connection appears as a consistent bit-error which can be corrected by the error-correcting code.)

Suppose that every $N \times N$ bit-plane has a blocking probability denoted "*bit-pb*." The goal is to design an $N \times N$ switch with w-bit wide datapaths with a given blocking probability per connection (called the "*connection-pb*"), which can be arbitrarily small, given any level of blocking in the bit-planes. Typically, one may select the connection-pb to be 10^{-8}, although other probabilities, such as 10^{-20}, are easily designed. (Blocking in networks is occasionally defined as the event that a permutation cannot be routed in one pass. We assume the more practical measure in this paper, i.e., the event that a connection cannot be routed. Under the permutation model, our dilation in Theorem 4 must be wider, i.e., $O(\log N)$, and the ERC network still yields optimal memory bit-complexity.)

Given that we insert $w + z$ bit-serial connection requests into $w + z$ bit-planes, the probability that a w-bit wide data path is established at each output port is given by the following. (In practice, we could inject multiple bit-serial connections into the same bit-plane.) Let $pb = bit - pb$, and $pa = 1 - bit - pb$; and $w \ge 1$. Therefore, the Pr[w-bit wide data path established]

$$= \sum_{j=w}^{w+z} \binom{w+z}{j} pa^j (1-pa)^{w+z-j} \approx \sum_{j=w}^{\infty} \frac{e^{-pa} \cdot pa^j}{j!}.$$

In this section, define the "expansion" to be the number of extra bits required. For example, an expansion of four and a datapath width of w implies that $4 + w$ bit-serial connection requests must be operated in parallel to achieve the specified *connection-pb*. (If the bit-serial connections are replaced by bit-parallel connections, the expansion also applies to bit-parallel connections.)

The required expansion depends upon the blocking probability in the bit-plane and the datapath width. Wider datapaths require less expansion to achieve a given connection-pb. Fig. 4 applies for a datapath width of eight bits. Fig. 4a plots the expansion required to achieve a given *connection-pb* when the *bit-pb* is in the range of 0.001 to 0.01. Fig. 4b plots the expansion required to achieve a given *connection-pb* when the *bit-pb* is in the range of 0.01 to 0.1. Fig. 4c plots the expansion required to achieve a given *connection-pb* when the *bit-pb* is in the range of 0.1 to 0.5. The ERC architecture yields nonblocking networks regardless of the blocking probability in the bit-planes**.** Bit-planes with more blocking simply require a larger expansion to achieve a given connection-pb.

Figs. 3 and 4 supply sufficient data so that a reader can design a self-routing nonblocking network of their choice. For example, to achieve a connection-pb of 10^{-8} given a bit-pb of 0.0066, the expansion is four bits (see the dot on Fig. 4a).

It is also possible that each connection is one bit wide. For this case, we set $w = 1$ and the expansion, or number of extra copies of the request, can be found from the previous equation. A connection request is successful if at least one bit-serial request reaches the destination.

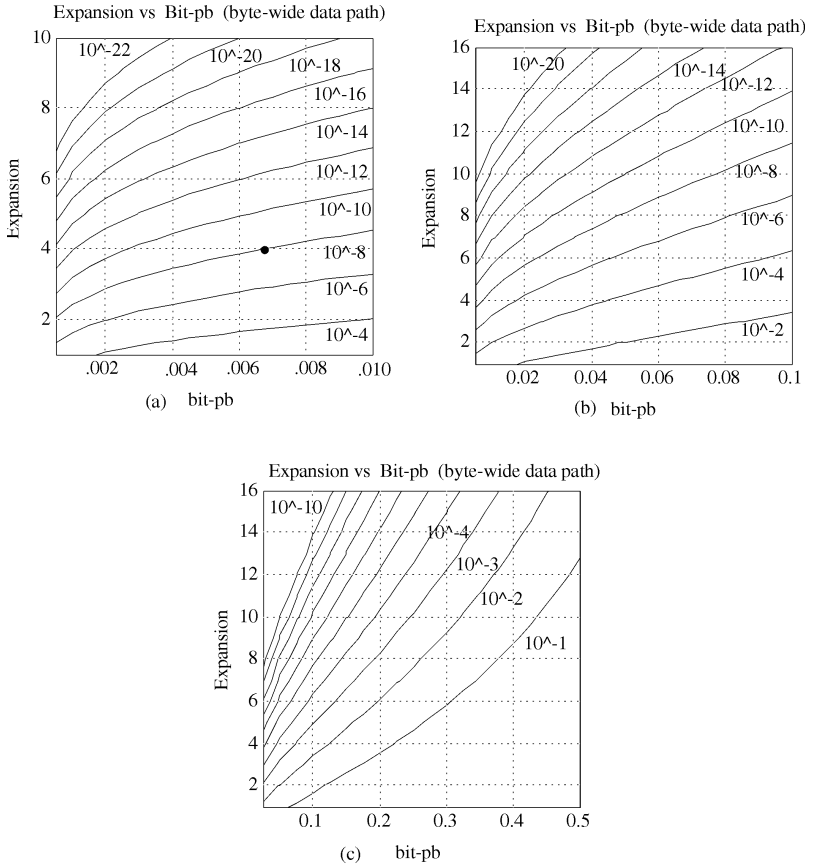


Fig. 4. Contour map of expansion versus bit-pb to achieve a given connection-pb for a datapath width of eight bits: (a) bit-pb in the range 0.001-0.01, (b) bit-pb in the range 0.01-0.1, (c) bit-pb in the range 0.1-0.5. Bold dot corresponds to example used in Section 5.

4 Hardware-Efficient TDM and SDM Constructions

In this section, the hardware complexity of the ERC network using multipath delta networks is examined.

4.1 Space Division Constructions

A *d-dilated k*k crossbar switch* consists of *k kd-to-d* concentrators; each concentrator collects the requests with a distinct routing bit between 0 ... $k-1$ and propagates *d* of them forward.

Concentrator Construction 1: A simple concentrator design called a "*Daisy-Chain Concentrator*" is shown in Fig. 5. The concentrator controller consists of an array of *kd-by-d* control cells, where an individual control cell is shown in Fig. 5a. Each control cell requires four logic gates and drives an associated crosspoint cell, shown in Fig. 5b. Each vertical column is essentially a "daisy-chain," which controls access to one output port. A busy signal travels down the daisy-chain and is asserted by the first active request. No other requests can claim the same output port once its busy signal is asserted. Other requests which encounter a busy signal in one daisy-chain column are forwarded to the next daisy-chain column to see if it is busy. An example state of a 6-to-4 concentrator is shown in Fig. 5c.

The *kd-to-d* daisy-chain concentrator requires five gates per cell and $kd \times d$ cells. For fixed *k*, the concentrator requires $O(d^2)$ logic gates and has a setup time of $O(d)$ logic gates. Each of the *kd* input ports requires $O(\log k)$ bits of memory to identify the requested logical output port. In a synchronous mode of operation, the degree *k* switch requires $O(kd \cdot \log k)$ bits of memory. For fixed *k*, the switch requires $O(d)$ bits of memory.

THEOREM 5. *The use of the daisy-chain concentrator in the ERC architecture, which satisfies the conditions of Theorem 4, yields a self-routing nonblocking $N \times N$ connection network with $O(N \cdot \log N)$ nodes, $O(N \cdot \log N)$ bits of memory, $O(N \cdot \log N \cdot \log \log N)$ logic gates, and a depth of $O(\log N \cdot \log \log N)$ logic gate delays.*

(The proof follows directly by substitution, where *N* is the number of distinct connections supported by the network.)

When the complexity is measured in terms of crossbar nodes, the cost is an optimal $O(N \cdot \log N)$ nodes. In terms of bits of memory, the complexity is an optimal $O(N \cdot \log N)$ bits, which meets Shannon's lower bound established in the 1950s. Hence, the switch scales optimally according to these important practical metrics. In terms of logic gates, the complexity is a slightly suboptimal $O(N \cdot \log N \cdot \log \log N)$ logic gates, and the depth is a slightly suboptimal $O(N \cdot \log N \cdot \log \log N)$ logic gate delays. As integrated circuits will soon support millions of gates, and as logic gate dimensions and delays will continue to shrink, these logic gate metrics have diminishing importance in practice. Furthermore, the grow rate in the term $O(\log \log N)$ is so slow as to be negligible in practice. The simplicity and regular VLSI

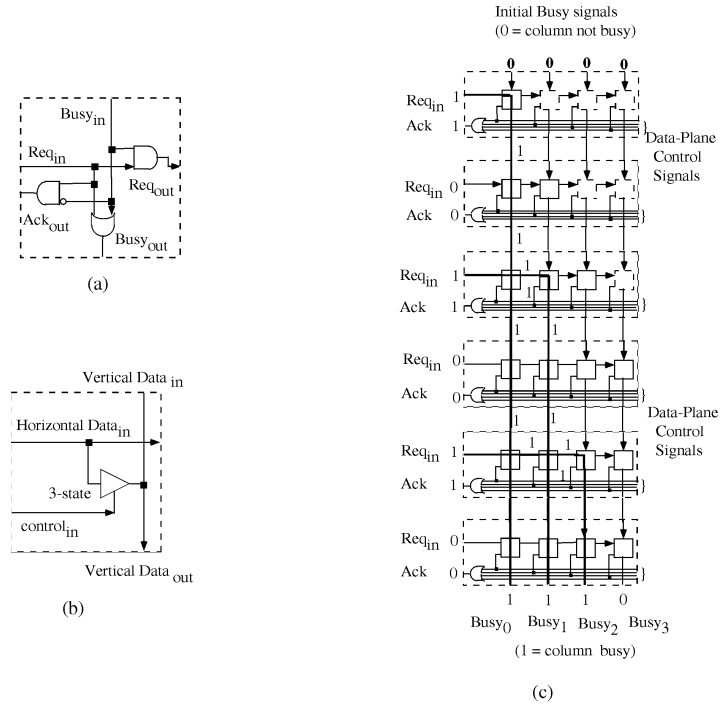


Fig. 5. (a) Daisy-chain concentrator control cell with four logic gates, (b) data plane crosspoint with tristate driver gate, (c) 6-to-4 daisy-chain concentrator with 6 × 4 array of concentrator cells. Three requests are routed to the first three output columns. (Dashed cells are not needed.)

layout of this concentrator circuit make it useful in practical designs (see Section 5). (The depth of this construction can be improved and will be reported elsewhere.)

Concentrator Construction 2: For sufficiently large dilations d, the logic gate complexity of the *d-dilated* $k \times k$ *crossbar* can be improved. The crossbar can be constructed with $O(kd \cdot \log kd)$ hardware and $O(\log kd)$ depth by using self-routing concentrators with logarithmic depth, as shown in Fig. 6. Each concentrator consists of a ranking circuit to assign ranks to the active inputs, followed by an omega-inverse network, which acts as a compact concentrator for $k = 2$. (It has recently been established that a single Omega network can act as a zero and one concentrator simultaneously [16], and, hence, only one Omega network is necessary in Fig. 6b.)

The ranking of kd inputs is achieved by using a bit-serial pipelined binary tree, as shown in Fig. 6a. Each box or circle is a bit-serial adder (the first stage of boxes is not necessary and drawn for symmetry). The ranks and partial sums are computed bit-serially, least significant bit first. This ranker is a pipelined multistage circuit based upon a binary tree ranker described in [20]. In the upward phase, each node propagates the sum up toward the root and propagates the count from its uppermost child horizontally. The root node (in the middle) propagates a zero count to its uppermost child, and propagates the incoming count from its upper child to the lower child. In the downward phase, each intermediate node propagates the count arriving from above directly to its uppermost child, and adds the count from above and the count arriving horizontally, and propagates the sum to its lower child. The leaves add the incoming count with a one if they have a connection, yielding the rank (from one to kd) of the request.

THEOREM 6. *The use of the log-depth concentrator in the ERC architecture, which satisfies the conditions of Theorem 4, yields a self-routing nonblocking* $N \times N$ *connection network with* $O(N \cdot \log N)$ nodes, $O(N \cdot \log N \cdot \log \log \log N)$ *bits of memory,* $O(N \cdot \log N \cdot \log \log \log N)$ *logic gates, and a depth of* $O(\log N \cdot \log \log \log N)$ *logic gate delays.*

(Proof follows by substitution.)

When the complexity is measured in terms of crossbar nodes, the cost is an optimal $O(N \cdot \log N)$ nodes. In terms of bits of memory, the hardware complexity is $O(N \cdot \log N \cdot \log \log \log N)$ bits, which is slightly suboptimal by a small factor of $O(\log \log \log N)$ when compared to Shannon's lower bound. In terms of logic gates, the complexity is $O(N \cdot \log N \cdot \log \log \log N)$ and the depth is $O(\log N \cdot \log \log \log N)$, which is an improvement over the daisy-chain concentrator design. In situations where gate delays are more important than bits of memory, the log-depth concentrator may be preferable over the daisy-chain concentrator.

4.2 TDM Constructions

A time-division *d-dilated* $k \times k$ *crossbar* can be implemented with a hardware cost of $O(k^2 \cdot d)$ logic gates and bits of memory and a latency of $O(kd)$ by using a circuit called a "time-bit concentrator" [33]. A TDM-dilated crossbar switch with four incoming and four outgoing space-division links is shown in Fig. 7. Each link carries up to d time-multiplexed bit-serial connections. On each incoming link, the bits from the d-multiplexed connections keep arriving in the same order (i.e., for a dilation of eight, bits belonging to connections arrive in order 1, 2, 3, ..., 8, and the cycle repeats). At each input port, a circular buffer

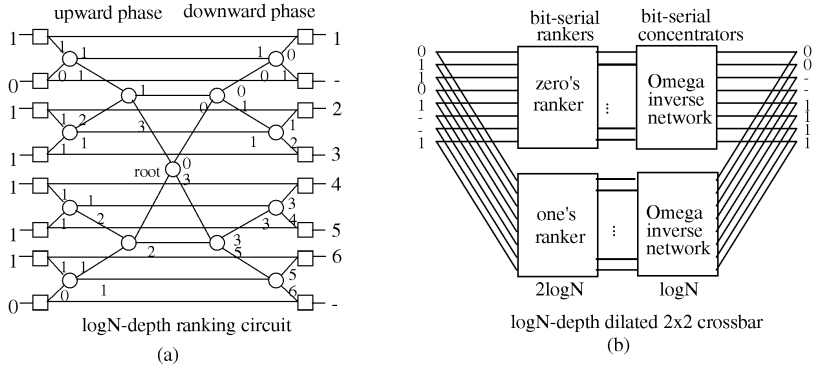


Fig. 6. (a) An O(log N)-depth ranking circuit, (b) a d-dilated 2 × 2 crossbar with O(log d) depth and O(d log d) bit-complexity.

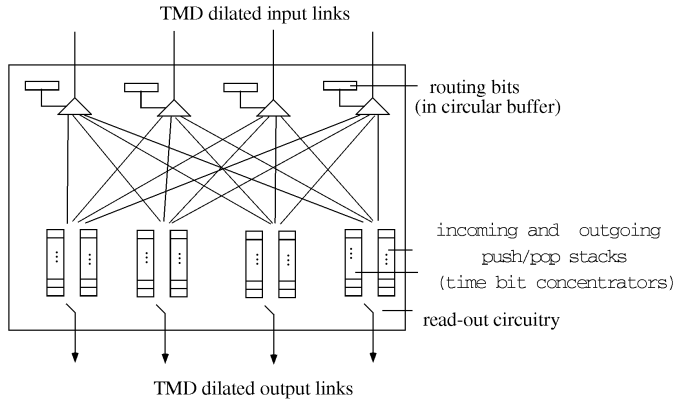


Fig. 7. A 4 × 4 dilated crossbar using time-bit concentrators.

stores the routing bits for the connections. The circular buffer is loaded initially as the connection-request headers pass by.

Once the circular buffers are loaded with routing bits, the multiplexed data bits arrive, and each bit is routed to a push-pop stack at the desired logical output port. In Fig. 7, we have two dedicated push-pop stacks at each output port, so that there is no contention for pushing the stack. One stack is used to store incoming bits, while the other is emptying outgoing bits. (Each stack must allow four simultaneous pushes in constant time, and can be designed as four separate push-pop stacks for simplicity). Once all d incoming bits have been routed to the output ports, d of them are then transmitted forward over each space-division link by having a read-out circuit pop the stack(s) (in constant time) in a repeatable order for d time units. Any bits left in the stack(s) after this represent blocked connections, and they are dropped. During the time one stack is emptying, another stack is storing a new set of incoming bits, which supplies the outgoing bits when the cycle repeats itself. It can be verified that this circuit acts as a *d-dilated crossbar*, has $O(k^2 \cdot d)$ hardware and $O(kd)$ latency in gates. We point out that the time-bit concentrator is fully pipelineable, i.e., new bits can enter and exit the concentrator on every link during every clock cycle.

THEOREM 7. *The use the time-bit concentrator in the ERC architecture, which satisfies the conditions of Theorem 4, yields a self-routing nonblocking $N \times N$ connection network with $O(N \cdot \log N)$ nodes, $O(N \cdot \log N)$ bits of memory, $O(N \cdot \log N)$ logic gates, and a depth of $O(\log N \cdot \log \log N)$ logic gate delays.*

(Proof follows by substitution.)

When the complexity is measured in terms of crossbar nodes, the cost is an optimal $O(N \cdot \log N)$ nodes. In terms of bits of memory and logic gates, the complexity is an optimal $O(N \cdot \log N)$, which meets Shannon's lower bound. Hence, the switch scales optimally according to these important practical metrics. The depth is $O(\log N \cdot \log \log N)$ logic gate delays, which is slightly suboptimal by the small factor of $O(\log \log N)$. This design is particularly useful in optical networks, since TDM is a natural mechanism to exploit the bandwidth advantage that optics offers over electronics. (It is interesting to observe that the complexity of the ERC network is an optimal $O(N \cdot \log N)$ hardware for wider dilations of $d = O(\log N)$, provided that $h = O(d)$ and k is constant.)

Variations: In practice, the depth of all the ERC self-routing networks can be reduced to an optimal $O(\log N)$ logic gate delays by keeping the dilation fixed and letting the expansion increase slowly with N. One may select a multistage network with a fixed dilation d and loading h, which has a complexity of $O(N \cdot \log N)$ hardware and a depth of $O(\log N)$ logic gates. From Fig. 3, for fixed dilations, it is observed that pb will rise very slowly as N increases. To keep the *connection_pb* below a prescribed value, the expansion can be read from Fig. 4. The analysis and numerical results indi-

TABLE 2
SEMICONDUCTOR INDUSTRY ASSOCIATION PROJECTIONS FOR CMOS TECHNOLOGY [28]

Year	Feature Size (μ)	Gates	Area (Sq. mm)	Electrical I/O pins	On-Chip Clock	Off-Chip Clock
1995	0.35	0.8 M	400	900	200 Mhz	100 Mhz
1998	0.25	2 M	600	1,350	350 Mhz	175 Mhz
2001	0.18	5 M	800	2,000	500 Mhz	250 Mhz
2004	0.12	10 M	1,000	2,600	700 Mhz	350 Mhz
2007	0.10	20 M	1,250	3,600	1 Ghz	500 Mhz

cate that the expansion is $\approx \theta(\log \log N)$. Hence, in practice, ERC networks can also be designed to have $O(N \cdot \log N \cdot \log \log N)$ hardware complexity and $O(\log N)$ logic gate delays.

5 A TERABIT SELF-ROUTING NONBLOCKING ATM SWITCH CORE FOR NETWORKS-OF-WORKSTATIONS

One application of fast self-routing switching networks is the "*Networks of Workstations*" (NOWs) distributed computer architecture. NOWs interconnected with a centralized ATM switch core based on a multistage delta network are described in [27]. The performance of microprocessors and communication networks have been growing exponentially over the last decade, and these trends are expected to continue well into the future [22], [27], [28]. By the year 2017, the single chip micros are expected to have performances of a few Teraflops per second, and networks are expected to have capacities of several Terabits per second [22]. In this section, we consider the design of a scalable ATM switch core, which can interconnect a large number of networked workstations.

Each workstation typically has a CMOS *Message-Processor* (MP) to handle the communication protocols. Messages are supplied to the Message-Processors, where they may be fragmented into fixed sized packets, assigned sequence numbers for error control, replicated for broadcasting, queued, and then transmitted over electrical or fiber links to the centralized switch core. The MPs also perform the receiving protocols. The switch core could support *distributed shared memory* over a NOW.

Consider a pipelined circuit-switched switch core, where the connections are established and torn down on a per-packet basis. (The network could be synchronous or asynchronous.) As a connection moves forward, a packet is transferred in a pipelined manner byte-by-byte. Pipelined circuit switching is similar to worm-hole routing [27], in that every intermediate node buffers a few bits or bytes of a packet, and the packet can be spread out over many intermediate nodes as it moves through the network.

Consider the design of a 1K × 1K switch core with byte-wide data paths, with a blocking probability per connection of 10^{-8}. In practice, protocols will ensure that no destination is overloaded, i.e., that a destination receives at most h packets at a time (for some number h). Attempts to transmit to an overloaded destination can be flagged with a "busy" signal and deferred until a short time later. Assume the switch is to be designed using CMOS technology available in the year 1998. Table 2 illustrates some of the Semiconductor Industry Association (SIA) estimates for CMOS technology over the next decade [28]. The figures in Table 2 will influence the design example, and are traditionally conservative.

Multistage networks can be designed with many stages of simple binary switches, or fewer stages of larger switches. To minimize the IC count, we will consider multistage networks with fewer stages of moderate size switches. A one-stage switch has minimal cost in terms of ICs. However, it requires very large crossbar switches. Due to electronic pin limitations, it is not possible to implement a 1K × 1K switch with eight bit-wide data paths on a single IC (yielding a one-stage network). However, using the design principles proposed in this paper, one can design a three, five, seven stage, or arbitrary $2n - 1$ stage networks with moderate size crossbars and with arbitrarily low blocking probabilities, which overcome the electrical pin-limitation problem.

Consider a three stage CLOS network with moderate size 16 × 16 crossbar switches in each stage, where each crossbar switch is bit-serial and eight-dilated. Each bit-serial Clos network represents an independent bit-plane in our ERC switch, and requires 16 crossbars per stage for three stages. Assume that the effective input loading is one half, i.e., each eight-dilated input port supports four processors, rather than eight. The blocking probability of dilated networks can be read directly from Fig. 3c. According to Fig. 3c, the "bit-pb" of this bit-plane is 0.0066. In other words, less than one percent of the bit-serial connections will block on average, given a permutation traffic model. (For a uniform random traffic model, the blocking is higher and the required expansion can be recomputed following the method in Section 3.)

To achieve a connection-pb of 10^{-8}, the expansion can be read from Fig. 4a. The required expansion is four bits. Hence, to establish a byte-wide connection, we launch 12 independent bit-serial connection requests into the ERC network. At each output port, a byte-wide connection is established if eight or more bit-serial connections survive. An 8-to-12 expander is needed at each switch input port, to create 12 independent bit-serial connections from the original eight. Also, a 12-to-8 concentrator is needed at each switch output port, to compact up to 12 bit-serial requests down to an eight-bit datapath. The additional gates needed to implement these components are negligible. (A 12-to-8 concentrator requires about 500 gates, which is negligible compared to the number of gates in the MP. Acknowledgment signals can be used to identify valid bit-serial connections.)

The self-routing dilated crossbar switches can be designed by using the daisy-chain concentrators of Section 4.1. It can be verified that each eight-dilated bit-serial 16 × 16 crossbar requires about 1 KBit of memory, which is very small when compared to the memory requirements of a

buffered crossbar. (A buffered crossbar supporting ATM cells would require at least one cell buffer per link, or at least 54 KBits of memory.) Due to electrical I/O pin limitations, each IC has enough I/O pins to support only five of these crossbar switches. It follows that the electrical ERC network requires 116 ICs, and has an aggregate bandwidth of 1.4 Terabit/sec. The architecture scales to five, seven, or more stages. The five-stage switch has a bandwidth of ≈ 22.9 Terabit/sec, and the seven-stage switch has a bandwidth of ≈ 367 Terabit/sec.

The proposed architecture addresses two key networking problems, as identified in the NSF report [36]. The problem of fast control of Gigabit/Terabit networks is partially addressed by using the ERC architecture, since it uses very fast self routing algorithms and is provably robust and immune to congestion (as demonstrated by Theorems 1-4). The problem that existing switch architectures have suboptimal hardware and memory scaling properties is addressed, as the hardware and memory complexity of the proposed architecture scales optimally or nearly optimally. Perhaps the only major remaining problem with "all electrical" architectures is the large number of wires between stages. This problem can be solved by using optics as the next design example illustrates.

5.1 Design of an Opto-Electronic Switch Core

The design of an opto-electronic switch core is summarized. To minimize cost, a one-stage switch would be preferable. However, a one-stage switch will require a 1K × 1K crossbar, which is very large. Table 3 illustrates projections for the electrical and optical I/O properties of OEICs (hereafter called ICs). Column 2 is from [18]. Using the daisy-chain concentrator, a 1K × 1K crossbar with byte-wide datapaths will require in excess of 10M gates, exceeding the gate capacity of the ICs. Hence, a three-stage ERC construction, using moderate size crossbars, can be used.

TABLE 3
PROJECTED CAPACITIES FOR SINGLE CHIP OEICS
(BASED ON DATA IN [2], [18])

Year	Max # Optical I/O	Optical Clock	Max. Optical I/O BW
1995	6,000	200 Mhz	0.6 Tb/s
1998	12,000	350 Mhz	2.1 Tb/s
2001	24,000	500 Mhz	6 Tb/s
2004	40,000	700 Mhz	14 Tb/s
2007	50,000	1 Ghz	25 Tb/s

Note: BW is product of optical I/O time optical clock divided by two.

It can be verified that each IC has sufficient optical I/O to support a large number of eight-dilated bit-serial 16 × 16 crossbar switches. However, each IC is limited by logic gates to implement only about 25 of these crossbar switches, which we assume. It follows that the ERC switch core requires 24 OEICs, and has an aggregate bandwidth of 2.8 Terabit/sec (double the bandwidth of the electrical version, since the optical datapaths operate at the faster optical clock rate). A five-stage network has a bandwidth of ≈ 46 Terabit/sec, and a seven-stage network has a bandwidth of ≈ 734 Terabit/sec.

These designs are technologically feasible with existing OEIC technology. The datapaths to and from the switch core can be realized with commercially available parallel fiber ribbons, such as the Motorola OPTOBUS [24]. A field programmable logic device with optical I/O, which can be dynamically programmed to implement dilated crossbars, has been developed [30]. Using these technologies and the proposed design principles, one may design arbitrarily large optical switching networks using multiple stages of moderate size crossbars.

5.2 Comparison with the ALM Network

Avora, Leighton, and Maggs described a self-routing "MultiBenes" network which is nonblocking and which has an asymptotically optimal cost of $O(N \cdot \log N)$ gates and bits of memory [3]. Like the ERC network, the MultiBenes network can be viewed as the concatenation of two multipath delta networks. However, the ALM routing algorithms and nodes are considerably more complex than the proposed ERC schemes. The ALM network requires complicated backtracking routing algorithms and partial packet buffering in the nodes, which render it unattractive for optical implementations. In addition, to achieve the optimal hardware complexity, the ALM network requires linear cost expanders for which no explicit construction is known. Nevertheless, in order to draw a comparison, we assume that their expanders can be built. The design in [3] describes a network with a path multiplicity of 10, a spacing between active logical input ports of 300, and the use of binary switching nodes. It follows that each stage in an ALM network with 1K active input ports requires at least 300K connection datapaths. Hence, an optical version of the ALM network has a cost which is several orders of magnitude more than the ERC network. In practice, one may be able to improve the performance ALM network. However, it is more efficient to apply the ERC design principles on the MultiBenes or Multipath Delta topology, which will yield an optimal or near-optimal network.

6 CONCLUSIONS

Principles for designing practical self-routing nonblocking switching networks, such as those used in ATM switch cores, were proposed. These principles lead to a large class of self-routing nonblocking switching networks, which are based on the concepts of expansion, routing, and contraction. These networks address two research priorities identified in a recent NSF sponsored report, the need for fast algorithms to control the Gigabit and Terabit networks of the future, and the need for optimally scalable switching networks [36]. The proposed space domain constructions yield self-routing nonblocking switching networks with an optimal $O(N \cdot \log N)$ bits of memory or $O(N \cdot \log N \cdot \log\log\log N)$ logic gates. The proposed time domain construction yields self-routing nonblocking switching networks with an optimal $\theta(N \cdot \log N)$ bits of memory or $\theta(N \cdot \log N)$ logic gates. These designs meet Shannon's lower bound on memory requirements established in the 1950s, and they readily scale to large sizes. The proposed architecture bridges the discrepancies between the best-known theoretical and practical results, and are attractive for both electrical and optical implementations. Fast self-routing switching networks with Terabits of bisection bandwidths can be designed using these principles.

ACKNOWLEDGMENTS

We gratefully acknowledge the comments of the referees which have improved the presentation of the paper. This research was funded by NSERC Canada Grant OPG0-0121601.

REFERENCES

[1] M. Ajtai, J. Komlos, and E. Szemeredi, "An O(NlogN) Sorting Network," *Proc. 15th ACM Symp. Theory of Computation*, pp. 1-9, 1983.

[2] ARPA/COOP/AT&T Hybrid-SEED Workshop Notes, George Mason Univ., July 1995.

[3] S. Avora, F. Leighton, and B. Maggs, "On Line Algorithms for Path Selection in a Nonblocking Network," *Proc. 1990 ACM Symp. Theory of Computing*, pp. 149-158, 1990.

[4] B.D. Alleyne and I. Scherson, "Expanded Delta Networks for Very Large Parallel Computers," *Proc. Int'l Conf. Parallel Processing*, pp. 127-131, 1992.

[5] A. Bassalygo and M.S. Pinsker, "Complexity of Optimum Nonblocking Switching Network without Reconnections," *Problems of Information Transmission*, vol. 9, pp. 64-66, 1974.

[6] K.E. Batcher, "Sorting Networks and Their Applications," *Proc. 1968 Spring Joint Computer Conf.*, 1968.

[7] M.V. Chien and A.Y. Oruc, "Adaptive Binary Sorting Schemes and Associated Interconnection Networks," *Proc. Int'l Conf. Parallel Processing*, pp. 289-293, 1992.

[8] T.J. Cloonan, G.W. Richards, A.L. Lentine, F.B. McCormick, and J.R. Erickson, "Free-Space Photonic Switching Architectures Based on Extended Generalized Shuffles," *Applied Optics*, vol. 31, no. 35, pp. 7,471-7,492, Dec. 1992.

[9] T.J. Cloonan, "Comparative Study of Optical and Electronic Interconnection Technologies for Large Asynchronous Transfer Mode Packet Switching Applications," *Optical Eng.*, vol. 33, no. 5, pp. 1,512-1,523, May 1994.

[10] G.A. De Biase, C. Ferrone, and A. Massini, "An O(logN) Depth Asymptotically Nonblocking Self Routing Permutation Network," *IEEE Trans. Computers*, vol. 44, no. 8, pp. 1,047-1,051, Aug. 1995.

[11] B. Douglass, "Rearrangeable Three-Stage Interconnection Networks and Their Routing Properties," *IEEE Trans. Computers*, vol. 42, no. 5, pp. 559-567, May 1993.

[12] H.S. Hinton, T.J. Cloonan, F.B. McCormick, A.L. Lentine, and F.A.P.Tooley, "Free-Space Digital Optical Systems," *Proc. IEEE*, vol. 82, no. 11, pp. 1,632-1,649, Nov. 1994.

[13] W. Hoeffding, "On the Distribution of the Number of Successes in Independent Trials," *Annals of Math. Statistics*, vol. 27, pp. 713-721, 1956.

[14] A. Huang and S. Knauer, "Starlite: A Wideband Digital Switch," *Proc. Globecom*, Dec. 1988.

[15] C.Y. Jan and A.Y. Oruc, "Fast Self-Routing Permutation Switching on an Asymptotically Minimal Cost Network," *IEEE Trans. Comm.*, vol. 42, no. 12, Dec. 1993.

[16] R. Kannan, H.F. Jordan, K.Y. Lee, and C. Reed, "A Bit-Controlled MultiChannel Time Slot Permutation Network," *Proc. Second Int'l Conf. Massively Parallel Processing Using Optical Interconnects*, pp. 271-278, 1995.

[17] D.M. Koppelman and A.Y. Oruc, "A Self-Routing Permutation Network," *J. Parallel and Distributed Computing*, vol. 10, pp. 140-151, 1990.

[18] A.V. Krishnamoorthy and D.A.B. Miller, "Scaling Optoelectronic-VLSI Circuits into the 21st Century: A Technology Roadmap," *IEEE J. Selected Topics in Quantum Electronics*, vol. 2, no. 1, pp. 55-76, Apr. 1996.

[19] C.P. Kruskal and M. Snir, "The Performance of Multistage Interconnection Networks for Multiprocessors," *IEEE Trans. Computers*, vol. 32, no. 12, pp. 1,091-1,098, Dec. 1983.

[20] F.T. Leighton, *Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes*. Morgan-Kaufmann, 1992.

[21] A.L. Lentine et al., "700 Mb/s Operation of Optoelectronic Switching Nodes Comprised of Flip-Chip-Bonded GaAs/AlGaAS MQW Modulators and Detectors on Silicon CMOS Circuitry," *Proc. Conf. Lasers and Electrooptics*, 1995.

[22] T. Lewis, "The Next $10{,}000_2$ Years: Part 1," *Computer*, Apr. 1996, pp. 64-70, "Part 2," pp. 78-86, May 1996.

[23] D. Mitra and R.A. Cieslak, "Randomized Parallel Communications on an Extension of the Omega Network," *J. ACM*, vol. 34, pp. 802-824, 1987.

[24] Motorola, "OPTOBUS Data Sheet," Logic Integrated Circuits Division, 1995.

[25] D. Nassimi and S. Sahni, "Parallel Permutation and Sorting Algorithms and a New Generalized Connection Network," *J. ACM*, pp. 642-667, 1982

[26] J.H. Patel, "Performance of Processor-Memory Interconnections for Multiprocessors," *IEEE Trans. Computers*, vol. 30, no. 10, pp. 771-780, Oct. 1981.

[27] J.L. Hennessey and D.A. Patterson, *Computer Architecture, A Quantatative Approach*, second edition. San Francisco: Morgan-Kauffman, 1995.

[28] Semiconductor Industry Association, "The National Technology Roadmap for Semiconductors," San Jose, Calif.: SIA, 1994.

[29] C.E. Shannon, "Memory Requirements in a Telephone Exchange," *Bell. Systems Technical J.*, 1953.

[30] S. Sherif, T.H. Szymanski, and H.S. Hinton, "Design and Implementation of a Field Programmable Smart Pixel Array," *Proc. LEOS 96 Conf. Smart Pixels*, Keystone, Colo., Aug. 1996.

[31] B. Supmonchai and T.H. Szymanski, "Fast Self-Routing Concentrators for Optoelectronic Systems," submitted.

[32] T.H. Szymanski and V.C. Hamacher, "On the Universality of Multipath Multistage Interconnection Networks," *Interconnection Networks*, I. Scherson and Youseff, eds., IEEE CS Press, 1994.

[33] T.H. Szymanski and C. Fang, "Randomized Routing of Virtual Connections in Essentially Nonblocking log N-Depth Networks," *IEEE Trans. Comm.*, pp. 2,521-2,531, Sept. 1995.

[34] T.H. Szymanski and H.S. Hinton, "Reconfigurable Intelligent Optical Backplane for Parallel Computing and Communications," *Applied Optics*, pp. 1,253-1,268, Mar. 1996.

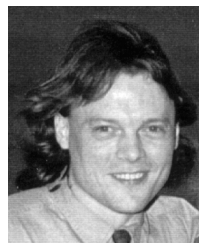
[35] C.D. Thompson, "Generalized Connection Networks for Parallel Processor Intercommunication," *IEEE Trans. Computers*, vol. 27, no. 12, pp. 1,119-1,125, Dec. 1978.

[36] U.S. National Science Foundation, "Research Priorities in Networking and Communications," Report to the NSF Division of Networking and Communications Research and Infrastructure, May 12-14, 1994, Arlington, Va.

[37] E. Upfal, S. Felperin, and M Snir, "Randomized Routing with Shorter Paths," *IEEE Trans. Parallel and Distributed Systems*, vol. 7, no. 4, pp. 356-362, Apr. 1996.

[38] L.G. Valiant and G.J. Brebner, "Universal Schemes for Parallel Communications," *Proc. 13th Ann. ACM Symp. Theory of Computing*, pp. 263-277, 1981.

[39] M. Yamaguchi, and K-I Yukimatsu, "Recent Free-Space Photonic Switches," *IEICE Trans. Comm.*, vol. E77B, no. 2, Feb. 1994.



Ted H. Szymanski received his BSc degree in engineering science, and the MASc and PhD degrees in electrical engineering from the University of Toronto. From 1987 to 1991, he was an assistant professor at Columbia University and a principle investigator at the U.S. National Science Foundation Center for Telecommunications Research working on ATM switching networks and WDM optical architectures. He is currently an associate professor and the director of the Microelectronics and Computer Systems Laboratory at McGill University, and a project leader in the Canadian Institute for Telecommunications Research, leading a project on "Optical Architectures and Applications." An intelligent optical backplane architecture developed by this project is being constructed in Canada. He is active professionally, and has served on the program committees for the 1998 and 1997 Workshops on Optics in Computer Science, the 1998 and 1997 International Conferences on Massively Parallel Processing using Optical Interconnects, the 1998 International Conference on Optical Computing, the 1997 Innovative Systems on Silicon Conference, the 1995 Workshop on High-Speed Network Computing, and the 1998, 1995, and 1994 Canadian Conferences on Programmable Logic Devices. He has also served as Session Organizer for IEEE Infocom, the International Computing and Communication Conference, and the International Conference on Parallel Processing. His personal interests including snowboarding and cycling, and his research interests include telecommunication and computing architectures, performance modeling, and optical networks. He is a member of the IEEE Computer and Communications societies.