Randomized Routing of Virtual Connections in Essentially Nonblocking Log N-Depth Networks

Ted Szymanski, *Member, IEEE*, and Chien Fang, *Member, IEEE*

***Abstract*—An optimal $N \times N$ circuit switching network with $\theta(N)$ bandwidth has a lower bound of $\theta(N \cdot \log N)$ hardware, which includes all crosspoints, bits of memory and logic gates, and a lower bound of $\theta(\log N)$ set-up time. To date no known self-routing circuit switching networks with explicit constructions achieve these lower bounds. In this paper we consider a randomized routing algorithm on a class of circuit switching networks called "Extended Dilated Banyans." It is proven that the blocking probability of an individual connection request is $O[\log_b N \cdot (k/d)^d]$, where d is the dilation factor and k is a constant. With a dilation of $\theta(\log \log N)$ and a loading <1 the blocking probability is shown to approach zero, yielding an "essentially nonblocking" network. The hardware complexity of these networks depends upon the internal node implementation. A space division node yields a network with $\theta(N \cdot \log N \cdot \log\log N)$ hardware and $\theta(\log N \cdot \log\log N)$ set-up time. A time division node, in which the bits from each connection are dynamically concentrated in time using a "Time-Bit-Concentrator" circuit, yields a network with an asymptotically optimal $O(N \cdot \log N)$ hardware and a slightly sub-optimal $\theta(\log N \cdot \log\log N)$ set-up time. Both implementations improve upon the best known explicit constructions of self-routing circuit switching networks with $\theta(N)$ bandwidth, and the TDM construction meets Shannon's lower bound on the cost of such networks. It is shown that Extended Dilated Banyans can carry significantly more traffic than the Batcher-banyan switch and its variants, given equivalent hardware complexity such as logic gates, bits of memory and crosspoints.**

I. Introduction

A CIRCUIT SWITCHING network is a network which can establish connections between specified input ports and specified output ports, over which data is transferred. An important criteria of circuit switching networks is the internal blocking probability, i.e., the probability that a connection cannot be established. Ideally a network will exhibit no internal blocking, but in practice a blocking probability of 10^{-8} or lower is acceptable for most purposes. An "essentially nonblocking network" can defined as one in which the internal blocking probability can be kept below an arbitrarily low threshold [22]. This paper is concerned with the design of self-routing circuit switching networks which can scale to asymptotically large sizes while simultaneously having very low blocking probabilities.

The *hardware cost* of a circuit switching network is defined as the number of logic gates required to build the network, where every logic gate has bounded fan-in and fan-out. This cost includes all crosspoints, all bits of memory and all logic gates. The *set-up* time is defined as the propagation delay (expressed in terms of individual logic gate delays) along the longest path from any input port to any output port. An optimal $N \times N$ circuit switching network with N input ports and N output ports has the following properties: 1) it has lower bound of $\theta(N \cdot \log N)$ hardware, 2) it would be "self-routing" and the "set-up" time would be $\theta(\log N)$ logic gate delays, and 3) it would be internally nonblocking regardless of the traffic pattern, provided that there are no destination conflicts.

"Self-routing" networks are useful in many applications such as ATM switching. Self-routing circuit switching networks are usually constructed with multiple stages of smaller crossbar switches. Connection requests are typically fed into the network bit-serially in synchronization with a "bit-clock," and are propagated forward one stage at a time. Each small crossbar switch contains sufficient hardware to buffer one or more incoming bits for each connection, perform and latch the routing decisions, and then propagate the connection to the next stage. The routing decisions at each stage must be made within each crossbar based on routing information extracted from the connection headers as they pass by, within a few gate delays.

Shannon established a lower bound of $\theta(N \cdot \log N)$ bits of memory for internally nonblocking circuit switching networks [20]. To date there are no known self-routing circuit switching networks with $\theta(N)$ bandwidth and with explicit space division constructions which meet this lower bound on the hardware cost. In practice, self-routing circuit switching networks proven to have low blocking probabilities and a worst case bandwidth of $\theta(N)$ are based on the Batcher sorting network [4] and related sorting and permutation networks, such as the Batcher-banyan switch [9], [13]. However, the Batcher sorting network requires $\theta(\log^2 N)$ hardware (which requires all connections to be bit-serial) and has a setup time of $\theta(\log^2 N)$ logic gate delays. To date space division self-routing circuit switching networks based on Batcher's sorting network have the best known explicit constructions, and these figures are summarized in Table I.

A dilated banyan has been defined as a banyan where every link is replaced by multiple parallel links [12], [17], [18]. The nodes in a dilated banyan can be called "dilated crossbars"; these are crossbars where each IO port can support multiple connections simultaneously. Dilated banyans can

Paper approved by G. P. O'Reilly, the Editor for Communications Switching of the IEEE Communications Society. Manuscript received August 4, 1992; revised August 5, 1993 and February 15, 1994. This work was supported by NSERC Canada under Grant OPGIN-007.

T. Szymanski is with the Department of Electrical Engineering, McGill University, Montreal, P.Q. H3A 2A7, Canada.

C. Fang is with the Sandia National Laboratories, Livermore, CA 94551 USA.

IEEE Log Number 9413170.

have much lower blocking probabilities than regular banyans [12], [17], [18], however they still suffer from severe worst case congestion.

To overcome the worse-case congestion of dilated banyans extra stages can be added to the network, following a technique used by Mitra *et al.* [15]. The resulting class of networks can be called "extended dilated banyans." In order to eliminate all worst case traffic patterns in an n-stage dilated banyan, it is sufficient to extend the dilated banyan by adding $n - 1$ extra stages. In this paper the class of self-routing circuit switching networks consisting of the concatenation of *any two dilated banyans* is considered. These "fully extended dilated banyans" may be operated with or without internal packet buffers. In this paper, we will consider only bit-serial circuit-switching networks which do not contain internal packet buffers.

A self-routing algorithm for these fully extended networks is proposed. Let the first dilated banyan acts as a "randomization" network, and the second acts as a traditional "routing" network. In the first dilated banyan, connections are routed to randomly selected output ports. In the second dilated banyan, connections are routed to their destinations. It is proven that the blocking probability of a connection in a dilated $N \times N$ banyan is $O[\log_b N \cdot (k/d)^d]$ where d is the link dilation factor and k is a constant. By increasing the dilation the blocking probability can be made arbitrarily small. With a dilation of $\theta(\log\log N)$and with each of the N input ports sourcing $O(\log\log N)$ connections, the dilated banyan becomes "essentially nonblocking." The importance of this result is the following: the fully extended dilated banyans are *immune* to worst case congestion: The randomization network transforms any given input-output mapping (including a worst case mapping) into a random input-output mapping. The random mapping is established in the routing network with a blocking probability which is guaranteed to be below an arbitrary specified threshold. Hence, the worst case bandwidth is therefore $\theta(N)$ as $N \rightarrow \infty$.

The dilated crossbar nodes can be implemented with time division multiplexing (TDM) or space division multiplexing (SDM); each case affects the asymptotic cost and asymptotic set-up time of the fully extended dilated banyans. The proposed TDM construction yields a network with an asymptotically optimal $\theta(N \cdot \log N)$ hardware and a slightly sub-optimal $\theta(\log N \cdot \log\log N)$ setup time. The TDM construction has the lowest asymptotic cost and fastest asymptotic set-up time among known TDM networks with explicit constructions and with $\theta(N)$ bandwidth (see Table I). In fact, the proposed TDM switch meets Shannon's lower bound on the asymptotic cost of such switches. We point out that the TDM construction is circuit-switched, i.e., it does not require any internal packet buffers within the switching fabric (otherwise it could not possibly meet Shannon's lower bound on the cost). However, the TDM network requires $\theta(N \cdot \log N)$ bits of internally memory, since this is obviously a lower bound as established by Shannon [20].

The proposed space division construction yields a self-routing $N \times N$ circuit switching network with $\theta(N \cdot \log N \cdot \log\log N)$ hardware and $\theta(\log N \cdot \log\log N)$ set-up time. The SDM construction also has the lowest asymptotic cost and fastest asymptotic set-up time among known SDM networks with explicit constructions and with $\theta(N)$ bandwidth (see Table I). It is shown that these constructions provide attractive alternatives to the use of costly sorting networks in the design of practical robust self-routing circuit switching networks for ATM applications.

TABLE I
ASYMPTOTIC COMPLEXITIES OF VARIOUS NETWORKS

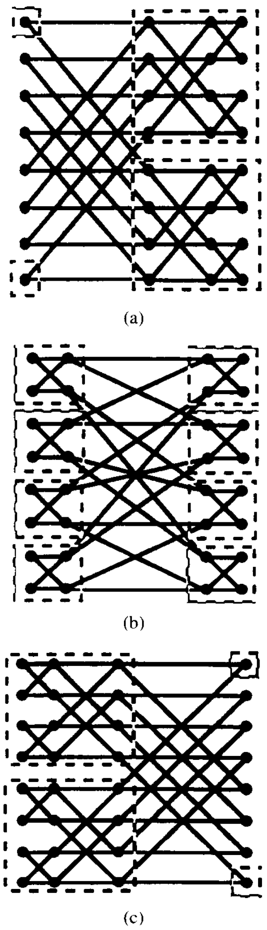
network	self-routing?	explicit?	hardware	setup-time
crossbar	yes	yes	$\Theta(N^2)$	$\Theta(N)$
Clos [7]	no	yes	$\Theta(N^{3/2})$	-
Cantor [8]	no	yes	$\Theta(N \cdot (logN)^2)$	-
Benes [7]	no	yes	$\Theta(N \cdot logN)$	-
Pippenger [16]	no	yes	$\Theta(N \cdot logN)$	-
Bassalygo [5]	no	no	$\Theta(N \cdot logN)$	-
Batcher [4]	yes	yes	$\Theta(N \cdot (logN)^2)$	$\Theta((logN)^2)$
Batcher-Banyan [9]	yes	yes	$\Theta(N (logN)^2)$	$\Theta((logN)^2)$
permutation networks	yes	yes	$\Theta(N (logN)^2)$	$\Theta((logN)^2)$
AKS [1]	yes	no	$\Theta(N \cdot logN)$	$\Theta(logN)$
ALM [3]	yes	no	$\Theta(N \cdot logN)$	$\Theta(logN)$
proposed TDM	yes	yes	$\Theta(N \cdot logN)$	$\Theta(logN \cdot loglogN)$
proposed SDM	yes	yes	$\Theta(N \cdot logN \cdot loglogN)$	$\Theta(logN \cdot loglogN)$

Finally, we point out that extended dilated banyans have been used commercially; the well known "BBN-Butterfly" parallel processor from Bolt, Beranek, and Newman used a 2-dilated delta network with extra stages for communications [24]. However, a proof that self-routing extended dilated banyans can be made essentially nonblocking given any worse-case traffic pattern has previously eluded researchers, i.e., see [14]. Furthermore, explicit constructions of practical essentially nonblocking self-routing connection networks proven to have $O(N \log N)$ bit complexity have also eluded researchers. The AKS sorting network [1] achieves $O(N \log N)$ bit complexity but lacks explicit constructions. This paper is organized as follows. Section II includes a review of banyan networks. Section III describes the randomized routing algorithm and the Extended Dilated Banyans. Section IV presents the proof of the essentially nonblocking property. Section V considers the hardware complexities of the TDM and SDM constructions. Section VI discusses some variations, and Section VII contains some concluding remarks.

II. TOPOLOGY AND ROUTING ALGORITHMS FOR MULTISTAGE BANYANS

A "square" $b^n \times b^n$ banyan network of size N consists of $n = \log_b N$ stages, where each stage consists of N/b nodes, where each node is a crossbar switch of size $b \times b$. The stages are labeled from 1 to n, and the N output ports from stage i are connected to the N input ports of stage $i + 1$ with N edges. Define a "*path*" through the banyan, from some input port i to some output port j, as a sequence of incident edges (or "links") which must be traversed in order to reach j from i. By definition every banyan has a unique path between each input port and each output port [23].

A binary "strict-buddy banyan" has been defined in [2] as one with the following property; in every stage except the last, each node has one "buddy" node that is connected to the same two successor nodes in the next stage. A radix-b strict-buddy banyan is defined here as a radix b banyan with the following


(a)

(b)

(c)

Fig. 1. Examples of various factorized binary banyans.

properties (1); all nodes are of degree b, and in (2) every stage except the last all b^{n-1} nodes can be partitioned into b^{n-2} sets where all b nodes in a set share the same b successors in the next stage. The following theorem stated without proof since it is used to obtain general theorems which are applicable to all members of the fully extended dilated banyan networks.

Theorem 1: All higher radix $(b^n \times b^n)$ strict buddy banyans are topologically equivalent.

Define a $b^n \times b^n$ "factor" as a $b^n \times b^n$ banyan or a $b^n \times b^n$ crossbar. The following claim will be essential to our proofs.

Claim: Any $b^n \times b^n$ banyan network can be topologically rearranged into two stages of "factors," with "factors" of size $b^s \times b^s$ in one stage, and "factors" of size $b^{n-s} \times b^{n-s}$ in the other stage (see Fig. 1 for examples of factorized banyans). The "factors" can be implemented with crossbars or by other instances of factorized banyans. By application of Theorem 1, it follows that properties 1 and 2 of the banyan are retained.

It is not difficult to verify that dilated banyan networks have severe worst case performance, whether they are circuit switched or packet switched. The worst case bandwidth of a d-dilated banyan is $O(\sqrt{N} \cdot d)$; for a 1-dilated banyan with 64 K IO ports the worst case bandwidth is a remarkably poor 256 connections or 0.039 of peak capacity. For an 8-dilated banyan with 64 K IO ports the worst case bandwidth is a remarkably poor 2048 connections or 0.031 of peak capacity. The worst case congestion grows more severe as the network size increases, severely limiting the usefulness of dilated banyans as robust self-routing circuit switching networks. In this paper, the worse case bandwidth problem will be eliminated by using a fully extended dilated banyan, as described in the next section.

III. Randomized Routing on Extended Dilated Banyans

Consider the concatenation of any two self-routing n-stage dilated banyans, yielding a fully extended dilated banyan with $2n$ stages. The first dilated banyan acts as a "randomization network" which attempts to route every connection request to a random output port. The second dilated banyan acts as a "routing network," which attempts to route every connection request to its intended destination.

If the resultant network is symmetric about its bisector then the innermost stages of each network can be merged into one, yielding a dilated Benes network with $2n-1$ stages. However, the class of fully extended dilated banyans includes many other networks which are not topologically equivalent to the Dilated Benes network. In other words, the particular choice of topology has little bearing on the main result. For example, if both banyans have the topology of an Omega network then the two innermost stages can also be merged into one, yielding a dilated network with $2n-1$ stages of shuffles, which could be called the fully extended dilated Omega network. This network is not topologically equivalent to the Dilated Benes network, but the main result of the paper still applies. Fig. 2(b) illustrates a typical end-to-end connection through a space division dilated Benes network. Once the connection is established, the data can be transferred over it, after which the connection is torn down.

IV. A Bound on the Blocking in Dilated Banyans

Consider a d-dilated $b^n \times b^n$ banyan with n stages. As a consequence of Theorem 1 on topological equivalence, all dilated strict-buddy banyans share the following properties; 1) all links leaving the same stage are topologically indistinguishable, 2) all nodes in the same stage are topologically indistinguishable, 3) the n-stage network can be factorized into a network with two stages of factors, with factors of size b^s in the first stage and factors of size b^{n-s} in the second stage. Furthermore, given a uniform random traffic model then 4) all nodes in the same stage are statistically identical and 5) all links leaving the same stage are statistically identical.

A. Binomial Approximation for Dilated Delta Networks

Patel presented an analysis of unique path $b^n \times b^n$ Delta networks in [23]. Delta networks are a subset of Banyan networks which have the self-routing digit-controlled property; at every stage a single digit in the destination tag is used to select a unique outgoing link. Patel's analysis is easily generalized to model d-dilated $b^n \times b^n$ Delta networks. Each d-dilated crossbar output port is incident to one "dilated link"

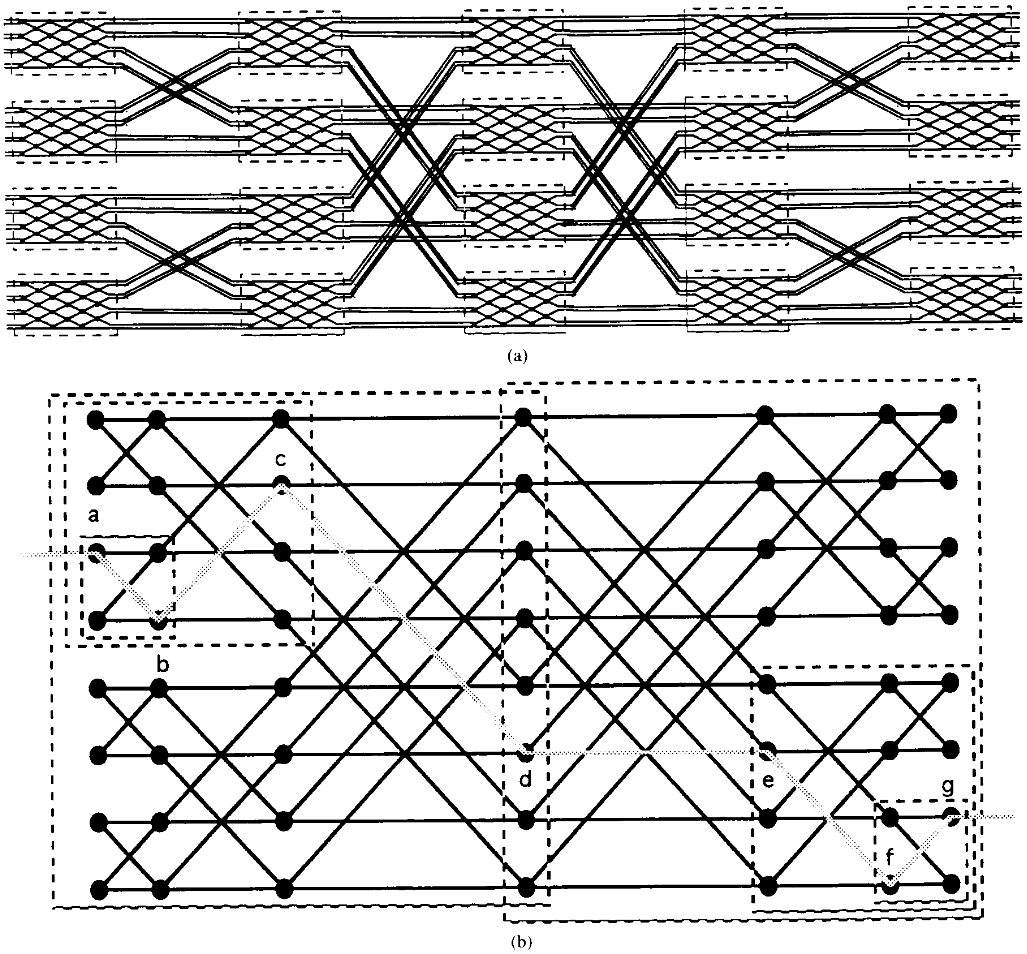


Fig. 2. (a) A space division construction of a fully extended dilated banyan. (b) A time division construction of a fully extended dilated banyan.

which can support up to d connections. Let $Y_{i,s}$ denote the probability that i connection requests traverse a dilated link leaving stage s of the network, where $0 \leq i \leq d$ and $0 \leq s \leq n$. The probabilities $Y_{i,s}$ determine the input loading to the network. Here after we will refer to "dilated links" as simply "links." An approximate analysis for a d-dilated $b^n \times b^n$ delta network under a random uniform traffic model is given by:

$$u_s \cong (1/d) \cdot \sum_{j=1}^{d} Y_{j,s} \cdot j$$

$$Y_{j,s+1} = \binom{bd}{j} \left(\frac{u_s}{b}\right) \left(1 - \frac{u_s}{b}\right)^{bd-j}, \quad 1 \leq j < d$$

$$Y_{d,s+1} = \sum_{j=d}^{bd} \binom{bd}{j} \left(\frac{u_s}{b}\right) \left(1 - \frac{u_s}{b}\right)^{bd-j}$$

where u_0 is a binomial approximation for the input loading. If $1 \leq j < d$ requests select an output port than all can propagate forward. Otherwise, d requests can propagate forward and the rest are blocked. The generalized analysis is approximate since the distribution of arrivals to a node in stage $s \in 1 \cdots n$ is actually a Multinomial distribution based on the probabilities $Y_{0,s}, Y_{1,s}, \cdots Y_{d,s}$ rather than a Binomial distribution based on u_s [17], [18]. The above analysis is nevertheless accurate to within 5 or 10%. In order to obtain a rigorous upper bound on the blocking probability, the Multinomial nature of the arrival distribution must be considered.

B. Rigorous Upper Bound

Theorem 2 will apply to all networks which meet properties 1)–5) stated earlier, including the dilated banyans, under a random uniform traffic model. The following two facts will be necessary.

Fact 1 (Hoeffding [10]): If T is the number of successes in N independent Bernoulli trials with probabilities $p_1, \cdots, p_N$ then if $\sum p_i = Np$ and $m \geq Np + 1$ is an integer then

$$\Pr(T \geq m) \leq B(m, N, p)$$

where $B(m, M, p)$ is the probability of at least m successes in M Bernoulli trials, with the probability of success in any trial equal to p.

Fact 2: Valiant's bound [19] on Chernoff's bound states that for any $m > Mp$

$$B(m, M, p) \leq \left(\frac{eMp}{m}\right)^m \cdot e^{-Mp}.$$

The end-to-end acceptance probability PA is defined as the probability a connection request is established given that it was offered, and the end-to-end blocking probability PB is defined as probability a connection request is not established given that it was offered.

Theorem 2: Let each of the b^n input ports in a $b^n \times b^n$ d-dilated banyan source $h \leq d$ connection requests. The end-to-end blocking probability PB in an is upper bounded by

$$PB \leq 1 - \left[1 - \left(\frac{eh}{d}\right)^d \cdot e^{-h}\right]^n.$$

Proof: Let $N \equiv b^n$. Define the random variable $X_{i,j}$ as the number of connection requests traversing link i leaving stage j of the network for $0 \leq i \leq N-1$ and $1 \leq j \leq n$. The random variables $X_{i,0}$ determine the input loading. The random variables assume values for each state of the network, and the following expectations are defined over all destination assignments corresponding to a random uniform traffic model.

By symmetry, all links leaving stage s are statistically identical for $1 \leq s \leq n$, hence the end-to-end acceptance and blocking probabilities are given by

$$\begin{aligned} PA &= E[X_{0,n}]/E[X_{0,0}], \\ PB &= E[X_{0,0} - X_{0,n}]/E[X_{0,0}]. \end{aligned}$$

Computing limits on the distributions of the random variables X_{oj} for all j as $j \to \infty$ is one approach to the bounding the blocking probability. However, this approach is quite difficult when it is tractable, and it does not appear to be tractable in the general case (see [11], [14]). Therefore, it is worthwhile to re-formulate the problem. PA and PB can be expanded and re-expressed in terms of the conditional acceptance probabilities within each stage of the switching network.

$$\begin{aligned} PA &= \frac{E[X_{0,n}]}{E[X_{0,0}]} \\ &= \frac{E[X_{0,1}]}{E[X_{0,0}]} \cdot \frac{E[X_{0,2}]}{E[X_{0,1}]} \cdots \frac{E[X_{0,n}]}{E[X_{0,n-1}]}. \end{aligned}$$

The acceptance probability within each stage is denoted with small letters to distinguish it from the end-to-end acceptance probability. Hence, $PA = pa_1 \cdot pa_2 \cdots pa_n$, where pa_s is defined as $E[X_{0,s}]/E[X_{0,s-1}]$. Note that pa_s is equivalent to the exact probability a request is successfully routed out of stage s, given that it has survived up to stage s. The conditional blocking probability in stage s, denoted pb_s, is equivalent to the exact probability that a request is blocked while attempting to leave stage s, given that it survived up to stage s.

By symmetry all $b \times b$ nodes in the first stage are statistically identical and independent, and therefore we can consider any one node. The number of input ports that can reach any output link leaving stage 1 is b^1; these input ports submit exactly $h \cdot b^1$ connection requests. By symmetry these requests are evenly distributed over the b^1 links leaving each node in stage 1. The connection requests may appear at the inputs either synchronously or asynchronously, and can be viewed as being routed serially with all service orders equally likely. Thus, the exact probability that the jth request to be serviced encounters a saturated link and blocks is given by

$$B(d, j-1, 1/N')$$

where $N' = b$. It follows that an exact expression for the conditional blocking probability of all requests in the first stage, denoted pb_1, is given by

$$pb_1 = \frac{1}{N'h} \cdot \sum_{j=1}^{N'h} B(d, j-1, 1/N').$$

The probability the last request (out of $N'h$) to be serviced encounters a saturated link and blocks is therefore given by

$$\begin{aligned} B(d, N'h-1, 1/N') &\leq B(d, N'h, 1/N') \\ &\leq \left(\frac{eh}{d}\right)^d \cdot e^{-h} \end{aligned}$$

where the last inequality is obtained by inflating the number of trials to $N'h$ by the addition of one dummy trial with zero probability of success, and then applying Facts 1 and 2.

The following two claims are stated without proof.

Claim 1: The conditional blocking probability of the last request to be serviced is an upper bound on the conditional blocking probability of each and every request to be serviced.

Claim 2: The conditional blocking probability of the last request to be serviced is upper bounded when no blocking in the previous stages is assumed.

By applying Claims 1 and 2, a lower bound for pa_1 is given by the following:

$$pa_1 \geq \left[1 - \left(\frac{eh}{d}\right)^d \cdot e^{-h}\right]. \tag{1}$$

To establish a lower bound for pa_s for $2 \leq s \leq n$, factorize the network into 2 stages, with factors of size $b^s \times b^s$ in the first stage and factors of size $b^{n-s} \times b^{n-s}$ in the second stage. By symmetry we need only consider any one factor in the first stage and apply its bound to all factors in the first stage.

Consider an arbitrary connection request labeled X which has been successfully routed up to stage s, and it is now competing with other connections to exit stage s over some link L. By symmetry, it suffices to consider any one path leading up to some link L. Since there are exactly h requests at each input port, then exactly $b^s \cdot h - 1$ other connection requests may compete with X for access to link L. Due to the random traffic model, each path is equally likely to select any output port of the factor. It follows that the probability that X encounters a saturated link and blocks given that it did not block in all previous stages is again upper bounded by (1). Therefore, for $2 \leq s \leq n$

$$pa_s \geq \left[1 - \left(\frac{eh}{d}\right)^d \cdot e^{-h}\right].$$

Hence, the end-to-end blocking probability PB is upper bounded by

$$PB \leq 1 - \left[1 - \left(\frac{eh}{d}\right)^d \cdot e^{-h}\right]^n.$$

□

C. Worst Case Traffic Patterns

Theorem 2 bounds the blocking probability in a dilated banyan given a random uniform traffic model. In this section it is proven that the fully extended dilated banyans are immune to severe performance degradation caused by worst case traffic patterns. Historically, the worst case performance of a network is evaluated under a *"worst case permutation traffic model"* rather than a random uniform traffic model. This model excludes output port conflicts and only consider blocking due to internal link conflicts. In the permutation model, every input port of the fully extended dilated banyan sources precisely h connection requests and every output port is the destination of precisely h connection requests.

Given a worst case permutation, randomized routing will transform it into a random input-output mapping in the randomization network, since every connection request is routed to a random output port. Hence, the blocking probability in the randomization network is bounded by Theorem 2. However, in the permutation traffic model Theorem 2 will not apply to the routing network; see [14] for a description of the problem of bounding the blocking probability in circuit switched networks under this traffic model. To summarize the problem: The connection requests which have survived through the randomization network and which are entering the routing network are not randomly and uniformly distributed over the input ports or the output ports of the routing network. The connection requests are not randomly distributed over the input ports since the positions they occupy are *correlated.* The connection requests are not randomly distributed over the output ports, since it is known that every output port is the destination for at most h connection requests. Hence, Theorem 2 is not applicable under the worst case traffic model.

To overcome these problems, we assume that all connection requests survive through the randomization network. By Claims 1 and 2 the upper bounds on the blocking probability derived with this assumption will still apply. Hence, it can be said with certainty that all *paths* are randomly and uniformly distributed over the input ports of the routing network since we are assuming no blocking in prior stages. (Note the distinction between *paths* and *surviving connection requests*). The following theorem derives an upper bound on the blocking probability in the routing network.

Theorem 3: The conditional blocking probability PB of a $b^n \times b^n$ d-dilated banyan acting as a routing network in a fully extended dilated banyan is upper bounded by

$$PB \leq 1 - \left[1 - \left(\frac{eh}{d}\right)^d \cdot e^{-h}\right]^n.$$

Proof: Within the routing network $PA = pa_{n+1} \cdot pa_{n+2} \cdots pa_{2n}$. Consider any arbitrary connection request labeled X which has been successfully routed up to stage $n+s$ and which is now competing for access to an outgoing link labeled L leaving stage $n+s$. It is convenient to factorize the routing network into two stages of factors, with factors of size $b^s \times b^s$ in the first stage and factors of size $b^{n-s} \times b^{n-s}$ in the second.

Due to the randomization every path is equally likely to be mapped to each and every input port of the routing network. Therefore within the routing network each of the Nh paths is equally likely to arrive at every factor in the first stage of the factorized routing network. (Again, note the distinction between paths and surviving requests).

Each factor in the second stage of the factorized network has exactly $N' = b^{n-s}$ output ports. Given the permutation traffic model, at most $h \cdot N'$ connection requests are destined for any second stage factor. Therefore, connection request X will compete with at most $N'h - 1$ other paths for access to the same factor in the second stage. By symmetry and due to the randomization, each path destined to the same factor in the second stage is equally likely to attempt to enter that factor over all incident links leading into that factor. Hence, the probability that any other path will select the same link L as connection request X is $1/N'$. Thus, the probability that X encounters a saturated link and blocks is upper bounded as follows:

$$pb_{n+s} \leq B(d, N'h - 1, 1/N') \leq \left(\frac{eh}{d}\right)^d \cdot e^{-h}.$$

Therefore in an n stage routing network the end-to-end blocking probability PB is again upper bounded by

$$PB \leq 1 - \left[1 - \left(\frac{eh}{d}\right)^d \cdot e^{-h}\right]^n. \quad (2)$$

□

Theorems 2 and 3 establish rigorous upper bounds on the average case and worst case blocking probabilities for the class of fully extended dilated banyans. A connection request will successfully make it through both halves with probability $\geq PA^2$ and it will be unsuccessful with probability $\leq 1 - PA^2$.

In the permutation traffic model, we can tighten the upper bound slightly. In this model, the last stage in the routing network cannot block any requests, since there is never any blocking at any output port. Furthermore, if the last stage of the randomization network was combined with the first stage of the routing network, then blocking can only occur in $2n - 2$ stages.

D. Numerical Results

Fig. 3(a) illustrates the upper bound on the blocking probabilities of asymptotically large dilated banyans operating at a light offered loads ($h = 1$). Curves are shown for 8 dilated banyans, 12 dilated banyans and 16 dilated banyans. For a fixed dilation, the upper bounds approach unity as $N \rightarrow \infty$, although they do so very slowly. This observation is consistent with Koch's result in [11], which proved that for fixed dilations, the blocking probability approaches 1 as $N \rightarrow \infty$.

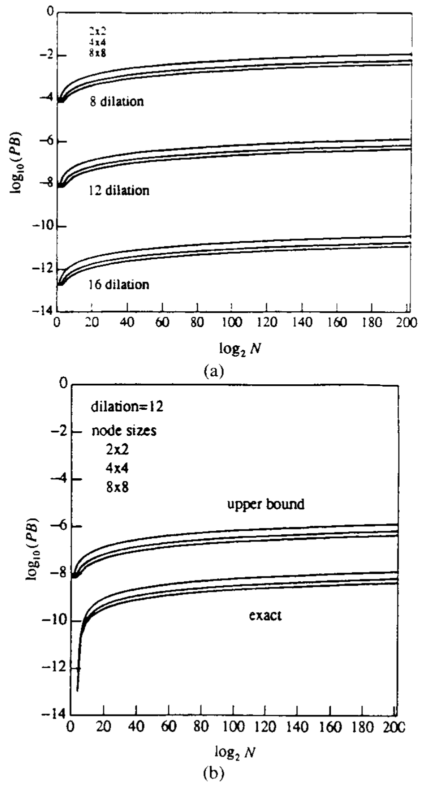


Fig. 3. (a) Upper bounds on PB versus $\log_2 N$ for fixed dilations ($d = 8, 12,$ and 16). Each dilation represents three curves for banyans of degree 2, 4, and 8 (the smaller degrees have the higher PB). (b) Upper bound versus exact PB for 12 dilated banyans of degree 2, 4, and 8. (Upper bounds from Theorem 2; exact analysis from [17].)

Fig. 3(b) illustrates the exact blocking probability for 12-dilated banyans (of degree 2, 4, and 8), and also the upper bound computed from Theorem 2. The analysis for the exact blocking probability of dilated banyans is detailed and can be found in [17] or [21]. The upper bound is about two orders of magnitude larger than the exact blocking probability, but the "shapes" of the curves are remarkably similar.

E. Asymptotic Performance, Approximation

Using the Binomial expansion $(1 - x)^n \approx 1 - n \cdot x$ for $x \ll 1$, (2) can be approximated. Letting $x = (eh/d)^d \cdot e^{-h}$ then for $x \ll 1$ the upper bound on the blocking probability can be approximated:

$$PB \le \log_2 N \cdot (eh/d)^d \cdot e^{-h}$$

For small eh/d this approximation is good to many digits, and it is very useful since (2) often yields numeric underflow (since the PB can be extremely small). For sufficiently small eh/d, the second and higher order terms in the binomial expansion can be ignored, and therefore $PB = O(\log_b N \cdot (eh/d)^d)$.

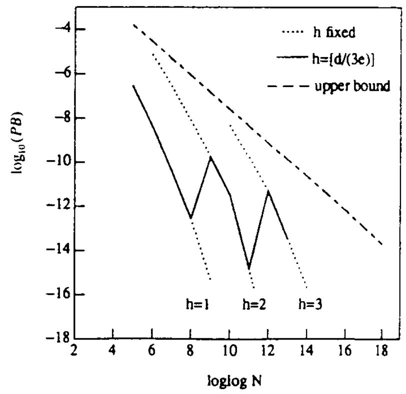


Fig. 4. PB versus $\log_2 \log_2 N$ for a dilation $d = 2 \cdot \log_2 \log_2 N$. Each dotted line is an exact PB curve for a fixed loading h. The solid line is an exact piecewise linear PB curve for $h = \lfloor d/3e \rfloor$,m which exhibits discontinuities when h increases by 1, but approaches zero as $N \to \infty$. Dashed curve is an upper bound on PB from Theorem 2.

F. Asymptotic Performance, Upper Bound

A rigorous bound on PB in either half, given a dilation $d = \theta(\log\log N)$ and a loading $h = \theta(\log\log N)$, can be found as follows. The number of stages n is set to $n = \log_2 N$, the dilation d is set to $d = \log_2 \log_2 N$, and the loading on the input/output ports h is set to $h = d/(2e)$, so that $eh/d = 1/2$. Hence, PB is upper bounded by

$$\begin{aligned} PB &\le 1 - \left[1 - \left(\frac{1}{2}\right)^d \cdot e^{-h}\right]^n \\ &\le 1 - \left[1 - \left(\frac{1}{n^{1+\log_2 e/2e}}\right)\right]^n. \end{aligned}$$

Letting $y \equiv (1 - 1/n^{1+\log_2 e/2e})^n$, then $\log_e y = \log_e [1 - n^{-(1+\log_2 e/2e)}]/[1/n]$. The limit of $\log_e y$ as $n \to \infty$ is ∞/∞. By applying L'Hospitals rule twice and taking the limit as $n \to \infty$, one rigorously establishes that

$$\lim_{n\to\infty} PB \le 1 - 1 = 0. \tag{3}$$

Equation (3) establishes rigorously that the blocking probability of any connection request asymptotically approaches zero as $N \to \infty$, given a dilation and loading that grow with $\theta(\log\log N)$.

Fig. 4 illustrates the exact blocking probability of a dilated banyan when $d = \theta(\log\log N)$ and $h = \theta(d)$. The exact PB is computed from the analysis in [17]. For a fixed loading (i.e., $h = 1$ or $h = 2$), the exact blocking probability drops extremely rapidly as $N \to \infty$ as proven in Theorem 2. Even when the loading is allowed to increase with $\theta(\log\log N)$ (i.e., the solid curve represents $h = \lfloor d/3e \rfloor$), the exact blocking probability drops rapidly as $N \to \infty$. The dashed curve is the upper bound on the blocking probability computed from Theorem 2, for a loading of $h = \lfloor d/3e \rfloor$. The blocking probability approaches zero as $N \to \infty$, as the asymptotic analysis indicates.

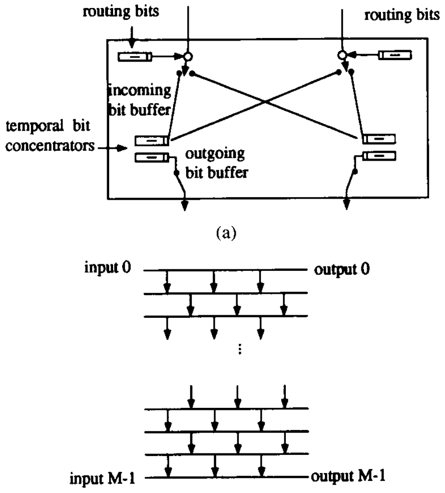


Fig. 5. TDM and SDM constructions of a dilated 2 × 2 cross bar node. (a) A TDM construction using linear cost "temporal bit concentrators". (b) An SDM construction using a lattice-like sorting circuit; the 0's rise and the 1's sink.

V. Hardware Complexity Analysis

Dilations can be implemented using two techniques, Time Division Multiplexing (TDM) and Space Division Multiplexing (SDM).

A. *"Time Bit Concentrators" with Linear Cost*

An $N \times N$ binary banyan requires $O(N \cdot \log N)$ nodes (of degree two) arranged in $O(\log N)$ stages. Suppose each 2×2 node is time multiplexed to "simulate" a dilation of $\theta(\log\log N)$. Basically, TDM will be used to implement a novel bit concentrator in the time domain, which we can call a "Time Bit Concentrator." This concentrator operates on individual bits and not entire packets, and its cost and latency grows linearly with the number of bits it is concentrating.

Claim: The use of TDM in a $b \times b$ node (for bounded b) to simulate a dilation of $O(\log\log N)$ incurs a cost of $\theta(\log\log N)$ hardware per node and a latency of $O(\log\log N)$ logic gate delays per node.

Proof: An implementation of a 2×2 node, using "Time Bit Concentrators," is shown in Fig. 5(a). The circuit is synchronized to a global "bit-clock." In every bit clock, each input port receives a pair of bits from the last stage, and each output port transmits a pair of bits to the next stage. These bits have been concentrated in time, in order to implement a dilation of $O(d)$.

Each input port requires $2d$ bits of memory to store the direction bits for up to d time multiplexed connections (either "0" or "1" or "null" if there is no connection). These bits must be loaded into a circular buffer initially as the connection headers pass by, and are retained for the duration of the connections (which may be hundreds of bit-times). For simplicity, each output port uses 2 sets of bit-buffers; one set stores incoming bits and one set supplies outgoing bits. In Fig. 5(a), the arriving bits are directed to the appropriate output port where they are temporarily stored (concentrated in time). After servicing the d time multiplexed connections, the incoming bit-buffers will become the outgoing bit-buffers and visa-versa. The bit-buffers themselves are "push/pop" stacks which can hold $2d$ bits, which can be pushed or popped in constant time. Therefore, the entire d-dilated 2×2 node requires $O(d)$ logic gates and $O(d)$ bits of memory. □

The entire TDM $N \times N$ switching network requires $O(N \cdot \log N)$ binary nodes, where each node uses a Time-Bit-Concentrator to implement a dilation of $O(\log\log N)$. Since each binary node requires $O(\log\log N)$ hardware and incurs a delay of $O(\log\log N)$, then the entire network requires $O(\log N \cdot \log\log N)$ hardware (which includes all logic gates, bits of memory and crosspoints) and has an end-to-end set-up time of $O(\log N \cdot \log\log N)$.

In order to express the cost and delay of this TDM network fairly, its cost and delay must be re-expressed in terms of its size, or equivalently the number of connection requests which it can handle. The switching network can be viewed as having a capacity of $M = N \cdot \log\log N$ sources and $M = N \cdot \log\log N$ sinks (since each of the N input ports and output ports shares $h = O(\log\log N)$ time-multiplexed sources/sinks). Each of the N output ports requires $O(h^2)$ hardware to deliver the traffic to the proper sink (out of h sinks multiplexed onto that output port), but this overhead is asymptotically negligible.

Theorem 4: An $M \times M$ switching network of the above TDM construction has a worst case bandwidth of $\theta(M)$, requires $\theta(M \cdot \log M)$ hardware, and has a propagation delay of $\theta(\log M \cdot \log\log M)$. The hardware cost is asymptotically optimal and the delays are slightly suboptimal. (The proof follows by substitution.) This construction has the fastest asymptotic setup times among known self-routing circuit switches with $\theta(N)$ bandwidth with explicit time division constructions.

B. *Space Division Implementation*

An $N \times N$ binary banyan requires $O(N \cdot \log N)$ nodes (of degree two) arranged in $O(\log N)$ stages. Suppose each 2×2 node is space multiplexed to implement a dilation of d (which will increase its degree to $2d$).

Claim: The use of SDM in a $b \times b$ node (for bounded b) to implement a dilation of $O(\log\log N)$ incurs a cost of $O[(\log\log N)^2]$ hardware per node and a delay of $O(\log\log N)$ per node.

Proof: A SDM implementation is shown in Fig. 5(b). Each d-dilated 2×2 node can be implemented using a type of lattice sorting circuit with $2d$ inputs and $2d$ outputs. Connection requests arrive bit-serially at the left side, and use a single bit to denote the desired output port; a 0 implies any output link in the upper half of the outputs and a one implies any output link in the lower half of the outputs. The arrows represent bit-serial compare-exchange modules controlled by a single bit; the 0's rise and the 1's sink; once the state of a comparator is set it remains there for the duration of the connection, which may be hundreds of bit-times. Each d-dilated 2×2 node requires $O(d^2)$ crosspoints and $O(d^2)$ logic gates, and has a propagation delay of $O(d)$ gate delays. □

The entire SDM switching network then requires $O[N \cdot \log N \cdot (\log\log N)^2]$ hardware and has a propagation delay of $O(\log N \cdot \log\log N)$ gate delays. Once again, the cost and delay of the SDM network should be expressed in terms of M, the number of connections requests which can be handled where $M = N \log\log N$.

Theorem 5: An $M \times M$ switching network of the above SDM construction has a worst case bandwidth of $\theta(M)$, requires $\theta(M \cdot \log M \cdot \log\log M)$ hardware, and has a propagation delay of $\theta(\log M \cdot \log\log M)$ gate delays. (The proof follows by substitution.)

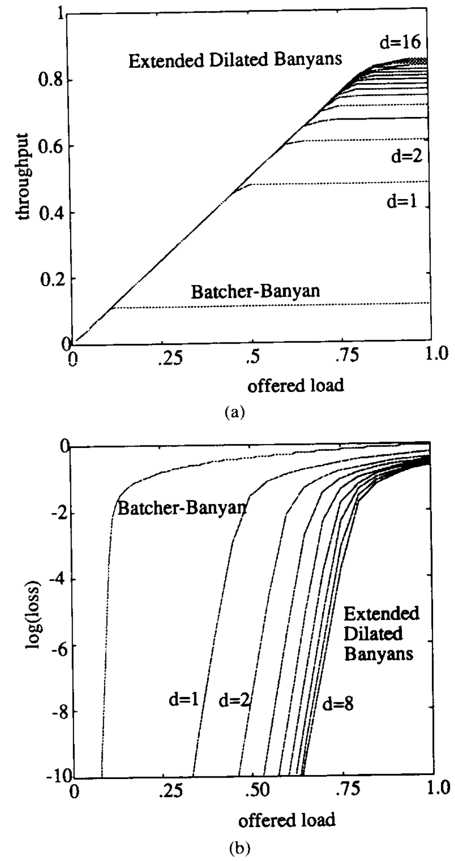
These SDM cost figures are sub-optimal by factors of $\theta(\log\log M)$ only, and represent improvements over the prior best known space division networks. These figures are among the lowest asymptotic costs and the fastest asymptotic setup times of known self-routing circuit switches with $\theta(N)$ bandwidth with explicit space division constructions. (These SDM figures can be improved upon, but the improvement is beyond the scope of this paper and will appear elsewhere [25].)

VI. Practical Considerations

The exact analysis of an ATM packet switch using a fully extended dilated banyan is rather intricate. The exact analysis must model the Input and/or Output queues external to the switch fabric, which have Batch arrival and Batch departure processes. An exact analysis must also compute the exact blocking probability within the circuit switching fabric. The reader is referred to [21] for detailed analytic models.

Fig. 6 compares the performance of a three-stage dilated Clos network to a Batcher-banyan. Each network is circuit-switched with $N = 1024$ IO ports. Each switching fabric uses external input queues to store packets which have temporarily blocked in any one time slot, which will be re-submitted in the next time slot. When the dilation is >1 in the Dilated Clos network, each input port attempts to establish d circuit switched connections in every time slot. It will then transfer one ATM cell over every established connection, after which the connections are torn down and the time slot ends. A random uniform traffic model is assumed.

To equate their costs both circuit switching fabrics use the same number of gate-array integrated circuits (IC). Fig. 6 compares a three-stage dilated Clos network built with d-dilated 32×32 crossbars versus a Batcher-banyan switch built with 32×32 bitonic sorting circuits. Each d-dilated 32×32 crossbar and each 32×32 bitonic sorting circuit are implemented on one gate array IC, respectively. Each IC is assumed to have the same number of IO pins, reflecting the constraints of existing integrated circuit packaging technology. For example, existing pin-grid-array (PGA) integrated circuits typically have 256 IO pins. It follows that as the dilation increases, the bandwidth of a connection decreases [21], since more connections are using the same number of IO pins. The Batcher-banyan requires at least 11 stages of IC's. The dilated Clos network requires only three stages of IC's. Hence, to equate the hardware cost approximately 11/3 copies of the Dilated Clos network are operated in parallel (see [21] for the detailed calculations).



(a)

(b)

Fig. 6. Practical comparison of extended dilated banyans with the Batcher-banyan. Both networks have equivalent hardware cost (in gate array integrated circuits) and use input queueing. (a) Normalized throughput versus offered load. (b) Probability of packet loss versus offered load.

According to Fig. 6, the dilated Clos networks carry significantly more traffic than the Batcher-banyan switch. Furthermore, because they carry significantly more traffic, the probability that a packet is lost due to input queue overflow is much lower. (*Note:* both networks use the same number of packet buffers in the external input queues.) From the graphs, the throughput of the Dilated Clos network increases with increasing dilation, and the improvement is attributed to the improved link utilization. In effect, the TDM and SDM implementations use a type of statistical multiplexing of connections over the hardware, thereby improving the utilization. While this comparison uses a three-stage dilated Clos network, similar results hold even with five stage, seven stage, and in general $2n - 1$ stage fully extended dilated banyans.

Note that there is a tradeoff between maximized bandwidth and minimized blocking probability. For maximized bandwidth the Dilated Clos network should be operated at the highest possible loads, by setting $h = d$. While this increases the internal blocking probability, the carried traffic is nevertheless maximized since more connections are established in each pass on average. The above graphs are based on maximized bandwidth. The important point is that even at full loading, these networks are still immune to severe worst case congestion.

A. Hardware Experiences

An ATM-like switch fabric based upon a TDM dilated banyan network has been developed by various students at McGill University in a series of student projects. The TDM nodes have been specified in the VHDL hardware description language on the Mentor Graphics CAD environment. Multiple Time-Bit-Concentrator nodes can be implemented on a single Xilinx Field Programmable Gate Array (FPGA) IC. The degree of time multiplexing is programmable and can be changed in real-time by down-loading the appropriate bit-stream to the Xilinx FPGA's. The blocking probability can be made arbitrarily low by adjusting the time multiplexing. We have demonstrated in VHDL a self-routing digital switch with four space division links and with a time multiplexing factor of 16, yielding 64 logically addressable IO ports. The key features of the architecture are: it can scale to arbitrarily large sizes while maintaining an arbitrarily low blocking probability, while remaining provably immune to congestion, and while maintaining $O(N \log N)$ hardware bit-complexity.

Variations: a) Randomization may be unnecessary in applications where the traffic is relatively random, and a "partial randomization" scheme may be sufficient. In this case, the number of extra stages to be added can be lowered or even eliminated. b) The randomization network can be operated in a nonblocking mode by having each node always forward all requests. Each node can select a state from a small lookup table of pseudo-random nonblocking states. c) It may be useful to employ a deflection routing algorithm and have every connection attempt to reach its real destination in the first dilated banyan; only the deflected connections need be routed through the second dilated banyan. While such a scheme may be useful in practice, it seems difficult to formally prove that it is immune to worst case congestion.

VII. Conclusions

It was proven that blocking probability of a dilated banyan decreases rapidly as the dilation factor grows. With a dilation of $O(\log \log N)$ and a loading of $O(\log \log N)$ connections on each IO port, the blocking probability of an individual connection approaches zero. At a sufficiently light loading, dilated banyans can be used as self-routing "essentially nonblocking" circuit switching networks. By increasing the loading, they can be used as circuit-switching networks which carry high amounts of traffic.

In order to provide immunity to worst case traffic patterns, it is sufficient to randomize the traffic through a randomization network before routing through a dilated banyan. The resultant networks are provably immune to severe internal congestion problems by nature of the randomization network. The TDM construction of a fully extended dilated banyan requires $O(N \cdot \log N)$ bits of internal memory and hardware and meets Shannon's asymptotic lower bound on the cost of essentially nonblocking networks. The SDM construction requires $O(N \cdot \log N \cdot \log \log N)$ bits of internal memory and hardware, which is slightly sub-optimal. However, asymptotically it is significantly less expensive than the well known Batcher-banyan switch and other circuit switches which are based on sorting networks. For realistic sizes, the performance improvements of the dilated banyans over the Batcher-banyan are about an order of magnitude.

Acknowledgment

The authors wish to thank an anonymous referee for his/her thoughtful and detailed comments, which have greatly improved the presentation of the paper.

References

[1] M. Ajtai, J. Komlos, and E. Szemeredi, "An $O(N \log N)$ sorting network," in *Proc. 15th ACM Symp. Theory of Computation,* 1983, pp. 1–9.

[2] D. P. Agrawal, "Graph theoretical analysis and design of multistage interconnection networks," *IEEE Trans. Comput.,* vol. C-32, pp. 637–648, July 1983.

[3] S. Arora, F. T. Leighton, and B. Maggs, "On line algorithms for path selection in a nonblocking network," in *Proc. 22nd Annu. ACM Symp. on Theory of Comput.,* 1990, pp. 149–158.

[4] K. E. Batcher, "Sorting networks and their applications," in *Proc. 1968 Spring Joint Comput. Conf.*

[5] L. A. Bassalygo and M. S. Pinsker, "On the complexity of optimal nonblocking switching networks without rearrangement," *Prob. Peredach. Inform.,* vol. 9, Jan. 1973.

[6] A. Borodin and J. E. Hopcroft, "Routing, merging, and sorting on parallel models of computation," in *Proc. 14th Annu. ACM Symp. on Theory of Comput.,* ACM, 1982, pp. 338–344.

[7] V. E. Benes, *The Mathematical Theory of Connecting Networks and Telephone Traffic.* New York: Academic, 1965.

[8] D. Cantor, "On nonblocking switching networks," *Networks,* vol. 1, Dec. 1971.

[9] A. Huang and S. Nauer, "Starlite: A wideband digital switch," in *Proc. GLOBECOM,* Dec. 1988.

[10] W. Hoeffding, "On the distribution of the number of successes in independent trials," *Ann. Math. Statist.,* vol. 27, pp. 713–721, 1956.

[11] R. R. Koch, "Increasing the size of a network by a constant factor can increase performance by more than a constant factor," in *Proc. 29th Annu. Symp. Foundations of Comput. Sci.,* Oct. 1988, pp. 221–230.

[12] 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.

[13] T. T. Lee, "A modular architecture for very large packet switches," *IEEE Trans. Commun.,* vol. 38, pp. 1097–1106, July 1990.

[14] F. T. Leighton, *Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* New York: Morgan–Kaufman, 1992.

[15] D. Mitra and A. Cieslak, "Randomized parallel communications on an extension of the Omega network," *J. ACM,* vol. 34, no. 4, pp. 802–824, Oct. 1987.

[16] N. Pippenger, "On crossbar switching networks," *IEEE Trans. Commun.,* vol. COM-23, pp. 646–659, June 1975.

[17] T. H. Szymanski and V. C. Hamacher, "On the permutation capability of multistage interconnection networks," *IEEE Trans. Comput.,* vol. C-36, pp. 810–822, July 1987.

[18] ——, "On the universality of multipath multistage interconnection networks," *J. Parallel and Distrib. Comput.,* no. 7, pp. 541–569, 1989.

[19] L. G. Valiant and G. J. Brebner, "Universal schemes for parallel communications," in *Proc. 13th Annu. ACM Symp. on Theory of Comput.,* 1981, pp. 263–277.

[20] C. E. Shannon, "Memory requirements in a telephone exchange," *Bell Syst. Tech. J.,* vol. 29, pp. 343–349, 1950.

[21] T. H. Szymanski, "A performance analysis of channel sharing in multistage ATM switches," in *Int. Conf. Commun., ICC'93,* Geneva, Switzerland, May 1993, pp. 1679–1685.

[22] M. Schwartz, *Telecommunications Networks: Protocols, Modeling, and Analysis.* Reading, MA: Addison-Wesley, 1987.

[23] J. H. Patel, "Performance of processor-memory interconnections for multiprocessors," *IEEE Trans. Comput.,* vol. C-30, pp. 771–780, Oct. 1981.

[24] *Butterfly Parallel Processor Overview.,* BBN Labs. Inc., 1985.

[25] T. H. Szymanski, "Design principles for self-routing nonblocking connection networks with $O(N \cdot \log N)$ bit-complexity," to be published.

Ted Szymanski (S'82–M'87) received the B.Sc. degree in engineering science and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto in 1980, 1982, and 1988, respectively.

From 1982 to 1984 he was consulting for various companies in the Toronto area, working primarily in distributed operating systems. From 1987 to 1991, he was an Assistant Professor at Columbia University and a Principle Investigator at the NSF Center for Telecommunications Research, working on switching network archtectures and WDM optical architectures. He is currently an Associate Professor at McGill University and a Project Leader in the Canadian Institute for Telecommunications Research, leading a project on Large ATM Architectures Based on Terabit Photonic Backplanes. His research interests include switching and computing architectures, congestion control in ATM networks, performance analysis, and optical architectures based on WDM and smart pixel arrays.

Dr. Szymanski is a member of the IEEE Computer and Communications Societies.

Chien Fang (S'87–M'92) received the B.S., M.S., and Ph.D. degrees from Columbia University, New York, NY, in 1983, 1984, and 1993, respectively, all in electrical engineering.

From 1984 to 1987 he worked as a systems and design engineer for Semi-Films division of National Micronetics, Kingston, NY, where he was responsible for real-time software development. He was a summer research student at IBM Research, Yorktown Heights, NY, in 1989, where he worked on image compression. He joined Sandia National Laboratories, Livermore, CA, in 1993, as a member of technical staff in research, where he has worked on performance studies of ATM networks. His research includes switching architectures, congestion control for ATM networks, and lightweight protocols for gigabit networks.

Dr. Fang is a member of Eta Kappa Nu and Tau Beta Pi.