A Time Domain Approach for Avoiding Crosstalk in Optical Blocking Multistage Interconnection Networks

Chunming Qiao, *Member, IEEE,* Rami Georges Melhem, Donald M. Chiarulli, and Steven P. Levitan

Abstract— **Crosstalk in Multistage Interconnection Networks can be avoided by ensuring that a switch is not used by two connections simultaneously. In order to support crosstalk-free communications among N inputs and N outputs, a *space domain* approach dilates an $N \times N$ network into one that is essentially equivalent to a $2N \times 2N$ network. Path conflicts, however may still exist in dilated networks.**

This paper proposes a *time domain* approach for avoiding crosstalk. Such an approach can be regarded as "dilating" a network in time, instead of space. More specifically, the connections that need to use the same switch are established during different time slots. This way, path conflicts are automatically avoided. The time domain dilation is useful for overcoming the limits on the network size while utilizing the high bandwidth of optical interconnects.

We study the set of permutations whose crosstalk-free connections can be established in just two time slots using the time domain approach. While the space domain approach trades hardware complexity for crosstalk-free communications, the time domain approach trades time complexity. We compare the proposed time domain to the space domain approach by analyzing the tradeoffs involved in these two approaches.

I. Introduction

BROADBAND switching networks can be built from 2×2 electro-optical switches such as lithium-niobate switches [2], [7], [8], [21]. Each switch has two active inputs and two active outputs. Optical signals carried on either input can be coupled to either output by applying an appropriate voltage to the switch. Fig. 1 shows the two logic states of such a switch, namely, "straight" and "cross," and a multistage interconnection network (MIN) with the *generalized cube* (GC) topology built from these switches.

One of the problems associated with these electro-optical switches is *crosstalk*, which is caused by undesired coupling between signals carried in two waveguides. For example, when a switch is in state "straight," a certain amount of signal power from input "A" may be coupled to "D". Similarly, a certain amount of signal power from input "B" may also be coupled to output "C". Crosstalk also occurs when the switch is in state "cross". When both inputs of the switch carry signals, output signals are affected by first order crosstalk. This is a significant factor which affects the signal-to-noise ratio (SNR).

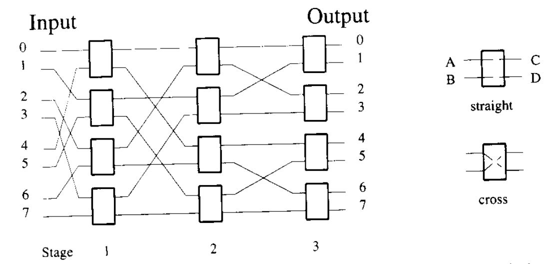


Fig. 1. An 8×8 generalized cube (GC) MIN built from 2×2 switches.

First order crosstalk at switches (hereafter abbreviated as crosstalk) can be avoided by ensuring that only one input of every switch is active at any given time. In other words, no two connections should use the same switch simultaneously. For MIN's, this can be accomplished with a space domain approach, called *network dilation* [9], [10], [24], [25]. Using this approach, an $N \times N$ network is dilated into a network that is essentially equivalent to a $2N \times 2N$ network, in which half of the input and output ports are used for N inputs and outputs, respectively. In the resulting dilated network, the connections of certain permutations can be established simultaneously without crosstalk. This concept has been generalized into *wavelength dilation* which is used to suppress crosstalks between closely separated wavelengths in wavelength-routing networks [19]. The space domain approach is reviewed in more details in Section II-A.

If the size of the network is limited to $N \times N$, crosstalk-free communications between $N/2$ inputs and outputs can be supported using the space domain approach. One of the objectives of this research is to support crosstalk-free communications among up to N inputs and outputs using the $N \times N$ network. For this, time-division multiplexing (TDM) can be employed to utilize the high bandwidth offered by the optical interconnects [1], [12]. An alternative would be to construct a $2N \times 2N$ network, which, depending on the value of N, may or may not be economically and/or technologically feasible.

The time domain approach proposed in this paper is a method that achieves the above objective. It extends the principle of the *reconfiguration with time division multiplexing*

Manuscript received June 30, 1993; revised June 9, 1994. This work was supported in part by the Air Force Office of Scientific Research under Contract AFOSR-89-0469 and in part by the Andrew Mellon Educational Trust in the form of a fellowship to the first author.

C. Qiao is with the Department of Electrical and Computer Engineering, State University of New York at Buffalo, Buffalo, NY 14260.

R. Melheim and D. Chiarulli are with the Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260.

S. Levitan is with the Department of Electrical Engineering, University of Pittsburgh, Pittsburgh, PA 15260.

IEEE Log Number 9404870.

0733-8724/94$04.00 © 1994 IEEE

(RTDM) paradigm [13], [15], [16], to be described in Section II-B. The RTDM resolves the path conflicts by partitioning a set of connections into conflict-free subsets and is thus applicable to dilated networks as well. The basic idea of the proposed time domain approach is to avoid crosstalk in a way similar to avoiding path conflict. More specifically, a set of connections is partitioned into several subsets such that the connections in each subset can be established simultaneously in a network not only conflict-free but also crosstalk-free. As such, the set of connections is established within several *time slots*, one for each subset. Clearly, the connections in some permutations realizable in a dilated network may need two time slots to be established using this time domain approach. The proposed approach can be regarded as "dilating" a network in the time domain, which is described in more details in Section II-C.

In Sections III and IV, we study the connectivity of networks dilated in the space domain and that of networks dilated in the time domain. More specifically, we determine the relationship between the set of permutations realizable in one time slot using the space domain approach and the set of permutations that require two time slots using the proposed time domain approach. In addition, we determine the number of time slots needed to establish arbitrary connections using either approach.

While the space domain approach can be regarded as a way to trade hardware complexity for crosstalk-free communications, the time domain approach trades time complexity, which is equivalent to communication bandwidth. Whether to trade hardware complexity or to trade bandwidth for crosstalk-free communications depends on the specific application being considered. For instance, for multiprocessor interconnections, the computation bandwidth of a processor is much lower than the optical bandwidth. Thus, it is more appropriate to utilize the excessive bandwidth to achieve crosstalk-free communications. In Section V, we discuss the tradeoffs involved in the time domain and space domain approaches. Finally we draw the conclusions in Section VI.

II. Approaches for Supporting Crosstalk-Free Communications

Throughout this study, we consider MIN's with the generalized cube (GC) topology [20] such as the one shown in Fig. 1. This topology is chosen partly because a large GC network can be easily constructed from smaller ones in a recursive way. For example, a 16×16 network is constructed from two 8×8 networks and an additional front stage, as shown in Fig. 2. The recursive characteristics of the topology will facilitate the discussions in later sections. Note that such a GC network is topologically equivalent to many blocking MIN's such as the Omega network [26]. For instance, the GC network in Fig. 1 becomes an 8×8 Omega network after positions of the two middle switches at the middle stage are interchanged. Similarly, by properly interchanging the positions of the switches at the (two) middle stages of the 16×16 GC network shown in Fig. 2, we can obtain a 16×16 Omega network. Based on this topological equivalence,

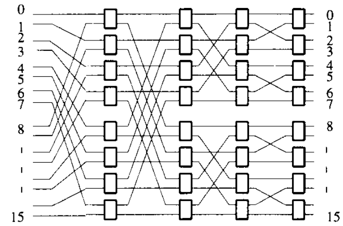


Fig. 2. A 16×16 GC network recursively constructed from 8×8 ones.

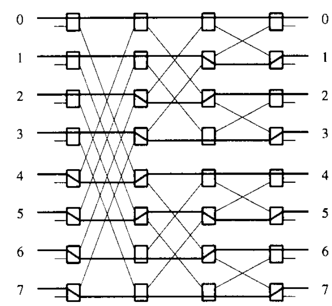


Fig. 3. The realization of the 8×8 identity permutation in a dilated 8×8 GC network.

we conclude that a GC network can realize the same set of permutation as an Omega network. Let this set of permutations be defined as follows.

Definition 1: Let Ω be the set of N by N permutations realizable by an $N \times N$ Omega or GC network.

The next subsection reviews the space domain approach for avoiding crosstalk, followed by a subsection describing RTDM for resolving path conflicts and a subsection describing the proposed time domain approach for avoiding crosstalk and resolving path conflicts as well.

A. Space Domain Dilation for Avoiding Crosstalk

A dilated $N \times N$ network is similar to a $2N \times 2N$ network. A major difference is that in a dilated network, only half of the input and output ports is used. Fig. 3 shows a dilated 8×8 GC network, in which only one of the two input (or output) ports of the switches at the first (or last) stage of the network is actually used. Other than that, this dilated 8×8 network is the same as the 16×16 network shown in Fig. 2. For this reason, we conclude that a dilated GC network is also topologically equivalent to a dilated Omega network. That is, they can realize the same set of permutations without crosstalk.

In a dilated network, a connection between an input and output is established by choosing an appropriate path in the network so that no switch in the network will have both input ports active at the same time. This avoids crosstalk as discussed earlier. By dilating a network, it becomes possible to establish a set of connections, or to realize certain permutations, without crosstalk. In [10], it is shown that a dilated Omega network has the same permutation capability as an Omega network. That is, the set of permutations realizable in a dilated Omega network without crosstalk is also Ω (see

Definition 1). Therefore, based on the topological equivalence mentioned previously, we establish the following lemma.

Lemma 1: Ω is also the set of permutation realizable in a dilated $N \times N$ GC network without crosstalk.

Since GC and Omega networks are blocking networks, Ω will contain some but not all possible permutations. Lemma 1 implies that certain permutations (such as the identity permutation as shown in Fig. 3) can be realized in a dilated GC network without crosstalk while others cannot. That is, dilated GC (or dilated Omega) networks are also blocking networks. This means that the network dilation approach can avoid crosstalks but cannot resolve path conflicts that exist in the original network. In the next section, we describe a time division multiplexing technique that may be used to resolve path conflicts in optically interconnected networks.

B. Reconfiguration with Time Division Multiplexing (RTDM)

The reconfiguration with time division multiplexing (RTDM) was proposed as a solution to the problem of relatively slow network control in resolving path conflicts [13]. The idea in RTDM is a generalization of techniques used in the time-space-time switching networks as in [8], [23], and [24].

With RTDM, a set of connections R is partitioned into several conflict-free subsets, called *mappings*. Thereafter, the connections in each mapping can be established in a network without path conflicts. Based on these mappings, a sequence of network configurations, or in other words, a set of states to which the switches in the network are set, can be determined. Each input and output is then informed of the sequence of configurations, which the network goes through, one configuration during each *time slot*. As a result, the connections in a mapping are established in one time slot and all the connections in R are established within several time slots in a time-division multiplexed fashion. Since no buffering or arbitration is needed, electro-optical switching devices such as lithium niobate directional couplers [2], [8], become suitable for implementing such networks.

If the set of connections R required by an application program is known and does not change during execution, a sequence of network configurations can be determined once at the beginning of execution of the application program. This is referred to as *static reconfiguration*, which is suitable whenever compile time analysis of the connection requirements can be done or a target communication structure is to be embedded in the interconnection network. Static reconfiguration with time-division multiplexing in multistage interconnection networks (MIN's) was studied in [15].

In the cases when the connection requirements change dynamically, the sequence of network configurations also needs to be changed. By time-multiplexing K configurations, K virtual networks are created in the time domain. The amount of control overhead involved in determining and setting the states of switches in these virtual networks can be amortized with concurrent processing of the requests. In addition, a sequence of configurations can capture communication locality in a similar way that a set of pages in a virtual memory system captures memory reference locality [5]. As a result, dynamic reconfiguration with time-multiplexing can effectively reduce control overhead and improve network performance [11], [16].

The principle of the above described RTDM can thus be used to resolve path conflicts in *dilated* networks as well. In doing so, a set of connections is partitioned into subsets that are not only conflict-free but crosstalk-free as well in a dilated network. Such subsets are called crosstalk-free (CF) mappings for the network. We will elaborate on the application of RTDM to path conflict resolution in dilated networks in Section IV. Next, we describe a way to integrate the solutions to crosstalk and path conflicts based on RTDM.

C. Time Domain Dilation

The proposed time domain approach extends the principle of the RTDM paradigm for avoiding crosstalk. More specifically, the same ideas in RTDM can be applied except that a set of connections needs to be partitioned into several subsets to avoid not only path conflicts but crosstalk as well. These subsets are CF-mappings for an undilated network. By establishing the connections in each of these CF-mappings in a separate time slot, crosstalk is avoided without the need for a dilated network and path conflicts are automatically resolved. Note that, these CF-mappings are different from the CF-mappings for a dilated network. One of the differences, for example, is that a CF-mapping for an undilated $N \times N$ network can contain only up to $N/2$ connections, while a CF-mapping for a dilated $N \times N$ network can contain up to N connections.

Assuming that a permutation in Ω needs to be realized, such a time domain approach that avoids crosstalk by using two or more CF-mappings may be considered as "dilating" a network in the *time domain*. In particular, a network dilated in the time domain using two time slots can be regarded as a correspondence to a spatially dilated network. The next two sections study the issues related to the connectivity of the networks that are dilated either in the time or in the space domain.

III. Permutation Capability of the Time and Space Domain Approaches

In this section, we first describe the set of permutations that require just two CF-mappings using the time domain approach. This set is then compared to the set of permutation realizable in one time slot in a dilated network.

In the following discussions, the phase "in one time slot" is usually omitted following the word "realizable" whenever there is no confusion. We first introduce a definition which corresponds to Definition 1 in Section II.

Definition 2: Let Θ be the set of N by N permutations realizable with *two* CF-mappings by an $N \times N$ GC network.

The following theorems state the relationship between the Ω-permutations (see Definition 1) and the Θ-permutations. For the time being, we consider Ω as the set of permutations realizable by a GC network (although it could be considered as the set of permutations realizable by a dilated GC network, as in Lemma 1).

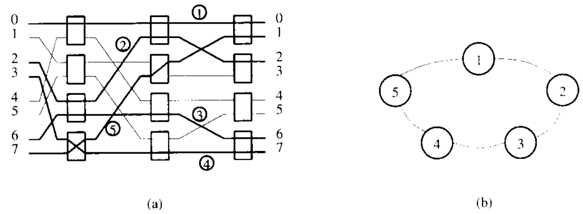


Fig. 4. Possible crosstalk among the five paths.

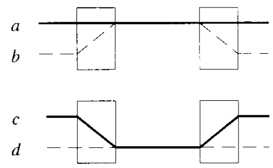


Fig. 5. An example 4×4 Θ-permutation.

Theorem 1: Not every Ω-permutation is a Θ-permutation in a network with $N \geq 8$.

Proof: We first consider the case $N = 8$ and show an example Ω-permutation which cannot be realized with just two CF-mappings. Five paths, numbered from 1 to 5, are drawn in bold lines in Fig. 4(a) as a part of the example Ω-permutation in an 8×8 GC network. Fig. 4(b) shows a graph of five vertices. Each vertex in the graph corresponds to a path in Fig. 4(a). Two vertices are connected by an edge if the two corresponding paths share a switch in the network. For example, vertex 1 and vertex 2 correspond to path 1 and path 2, respectively. Clearly, these two paths cannot be in the same CF-mapping. Since the graph in Fig. 4(b) is a ring of five vertices, it is impossible to establish the five paths in Fig. 4(a) with just two CF-mappings. This proves that the example Ω-permutation including these five paths is not a Θ-permutation. For $N > 8$, an Ω-permutation which includes the above five paths as *partial* paths can be constructed. The resulting Ω-permutation is not a Θ-permutation and therefore the theorem is proved. □

The above theorem indicates that Θ is not a superset of Ω. The following theorem shows the converse.

Theorem 2: Not every Θ-permutation is an Ω-permutation.

To prove this theorem, we first look at an example. Fig. 5 shows a 4×4 GC network and a permutation containing four paths labeled a, b, c, and d. This permutation, which we denote by $\pi(4)$, is not an Ω-permutation since these four paths cannot be established simultaneously without conflict in switch settings. However, since paths a and c can be established in one CF-mapping and paths b and d can be established in another CF-mapping, the permutation is a Θ-permutation.

The proof of the theorem is based on recursive constructions of a permutation $\pi(N)$ that is a Θ-permutation but not an Ω-permutation. $\pi(N)$ will be recursively constructed from the example permutation $\pi(4)$ shown in Fig. 5. Number the paths in $\pi(N)$ from 1 to N in a top-down order based on the position of their originating input ports. Denote the set of odd and even numbered paths by $\pi_{odd}(N)$ and $\pi_{even}(N)$, respectively. The induction hypotheses about $\pi(N)$ are as follows.

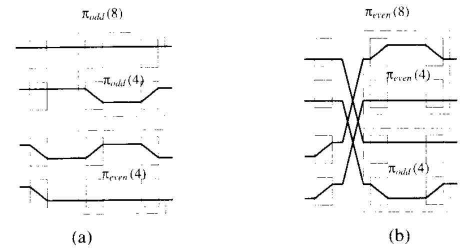


Fig. 6. Recursive construction of the two sets for $N = 8$.

1. Each of the sets, $\pi_{odd}(N)$ and $\pi_{even}(N)$ is a CF-mapping in an $N \times N$ network and their union, $\pi(N)$, is a Θ-permutation.
2. One path from set $\pi_{odd}(N)$ containing a as a partial path conflicts with one path from set $\pi_{even}(N)$ containing b as a partial path. Therefore, the union of the two sets, is not an Ω-permutation.

For $N = 4$, $\pi_{odd}(4)$ contains path 1 (which is a) and path 3 (which is c) while $\pi_{even}(N)$ contains path 2 (which is b) and path 4 (which is d). Clearly, the above hypotheses are true for $N = 4$.

Assume that the hypotheses are true for a network of size N. Based on this assumption, we will prove that the hypotheses are also true for a network of size $2N$. We do that by constructing the two sets, $\pi_{odd}(2N)$ and $\pi_{even}(2N)$, from $\pi_{odd}(N)$ and $\pi_{even}(N)$, in the following way.

To construct $\pi_{odd}(2N)$ in a network of size $2N \times 2N$, the top subnetwork (of size $N \times N$) is set to establish the paths in $\pi_{odd}(N)$ while the bottom subnetwork is set to establish the paths in $\pi_{even}(N)$. After the two subnetworks are set properly, the switches of the upper half of the first stage are then set to the "straight" state while those of the lower half are set to the "cross" state. This completes the construction of set $\pi_{odd}(2N)$. Fig. 6(a) shows the construction of $\pi_{odd}(8)$ from $\pi_{odd}(4)$ and $\pi_{even}(4)$.

Since only one input (an odd numbered one) of each switch at the first stage is used, there will be no crosstalk at that stage when the connections in set $\pi_{odd}(2N)$ are established. Due to the induction hypotheses 1), no crosstalk will be present at later stages (of either subnetwork). Therefore, set $\pi_{odd}(2N)$ is a CF-mapping.

The set $\pi_{even}(2N)$ can be similarly constructed by reversing the roles of the two subnetworks in the above procedure. Fig. 6(b) shows the construction of $\pi_{even}(8)$. The resulting set $\pi_{even}(2N)$ is also a CF-mapping. Since different inputs and outputs are active in set $\pi_{odd}(2N)$ and set $\pi_{even}(2N)$, and each set contains N connections, the union of the two sets is thus a $2N$ by $2N$ permutation. Therefore, the hypotheses 1) is also true for networks of size $2N$.

Note that when the set $\pi_{odd}(2N)$ is constructed, the upper 4×4 subnetwork is set to establish a as a partial path. Similarly, when the set $\pi_{even}(2N)$ is constructed, the same 4×4 subnetwork is set to establish b as a partial path. Since paths a and b conflict in the subnetwork, the path from $\pi_{odd}(2N)$ containing a as a partial path conflicts with the path from $\pi_{even}(2N)$ containing b as a partial path. Thus, hypotheses (2) is also true for networks of size $2N$.

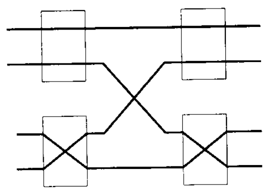


Fig. 7. An example 4×4 permutation in both Ω and Θ.

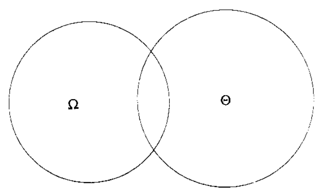


Fig. 8. Relationship between the set Ω and the set Θ.

Since both hypotheses are true, we have proved the theorem by induction. □

Having found from the above two theorems that the Θ-permutations and the Ω-permutations are not the same, we also observe the following.

Theorem 3: Some Ω-permutations are also Θ-permutations (and vice versa).

We will sketch the proof of the theorem by examining an example shown in Fig. 7. Note that the permutation in this figure, which is different from the one shown previously in Fig. 5, belongs to both Ω and Θ. Therefore, the theorem holds for a network with $N = 4$.

For networks with $N > 4$, a recursive construction procedure similar to the one used in Theorem 2 can be carried out and its description is thus omitted. It is worth noting, however, that the recursive construction is now based on the permutation in Fig. 7 instead of Fig. 5. Therefore, the theorem can be proved using a similar induction proof in which the hypotheses (2) becomes "no path in set $\pi_{odd}(2N)$ conflicts with paths in set $\pi_{even}(2N)$ and therefore the union of the two sets is an Ω-permutation." □

So far, we have found that Ω and Θ are two different sets with a nonempty common subset. Fig. 8 summarizes the relationship between the two sets. The figure also shows that Θ is larger (that is, contains more permutations) than Ω. This is because there is a one-to-one (but not onto) mapping from Ω to Θ, as proved below.

Denote by f a perfect shuffle on set $\{0, 1, \cdots, N-1\}$. That is, given any number D $(0 \leq D \leq N-1)$ and its binary representation $d_1 d_2 \cdots d_n$, a perfect shuffle f maps $D = d_1 d_2 \cdots d_n$ to its image $D' = d_2 \cdots d_n d_1$.

Let each connection be represented by an input–output pair (S, D) where $0 \leq S, D \leq N-1$. We may represent a permutation P by N pairs of input–output ports. That is, $P = \{(S_i, D_i) | i = 1, 2, \cdots, N\}$. Denote by F a mapping which maps P to P' by shuffling all output ports of the connections in P. More specifically, we have $P' = F(P) = \{(S_i, f(D_i)) | i = 1, 2, \cdots, N\}$. Fig. 9 illustrates the mapping from P to P' by F where $N = 8$ and $D_i = i - 1$.

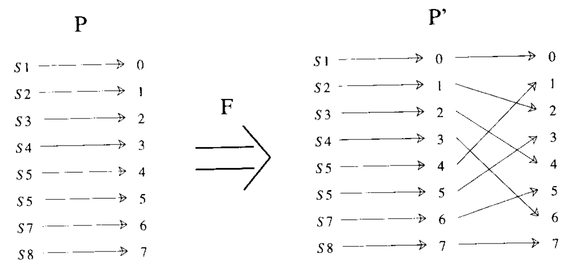


Fig. 9. Mapping permutation P to P' using F.

Lemma 2: F is a one-to-one mapping from Ω to Θ.

Proof: If P_1 and P_2 are two different permutations, then $F(P_1)$ and $F(P_2)$ are also different permutations. This is because function f is a one-to-one mapping between integers. To prove the lemma, we need to show that if P is an Ω-permutation, then its image $P' = F(P)$ is a Θ-permutation. So far, we have considered Ω as a set of permutations realizable by a GC-network as in Definition 1. In proving this lemma, we consider Ω as a set of permutations realizable by a *dilated* GC network as in Lemma 1. That is, we will show that if P can be realized by a dilated GC network with one CF-mapping, then P' can be realized by a GC network with two CF-mappings.

While presenting the general proof for any permutation P in an $N \times N$ network, we demonstrate the ideas used in the proof by applying them to the identity permutation realized by the 8×8 dilated GC network shown in Fig. 3. (Note that the identity permutation in an Ω-permutation according to Lemma 1.)

Divide the connections in P into two subsets, p and q, based on their destinations. More specifically, let p contain the connections whose destinations are $0, 1, \cdots, N/2 - 1$ and q contain the connections whose destinations are from $N/2$ to $N - 1$. For example, the 8×8 identity permutation is divided into

$$p = \{(0, 0), (1, 1), (2, 2), (3, 3)\} \tag{1a}$$

$$q = \{(4, 4), (5, 5), (6, 6), (7, 7)\}. \tag{1b}$$

When P is realized in a dilated network, the connections in p will use the upper subnetwork while the connections in q will use the lower subnetwork. The realization of p and q for the example identity permutation is shown in the left hand of Fig. 10.

Given that P is partitioned into p and q, the permutation $P' = F(P)$ is the union of the two subsets, p' and q', which correspond to p and q, respectively. While the connections in p' have the same sources as those in p, their destinations are $f(0), f(1), \cdots, f(N/2-1)$, that is, $0, 2, \cdots, N-2$. Similarly, the connections in q' have the same sources as those in q but their destinations are $1, 3, \cdots, N-1$. For example, for the 8×8 identity permutation, we have

$$p' = \{(0, 0), (1, 2), (2, 4), (3, 6)\} \tag{2a}$$

$$q' = \{(4, 1), (5, 3), (6, 5), (7, 7)\}. \tag{2b}$$

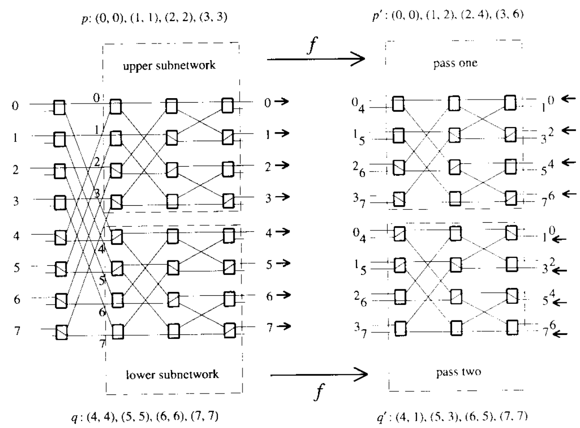


Fig. 10. Emulating the two subnetworks of a dilated network by an undilated network in two passes.

In order to show that P' is a Θ-permutation, it is sufficient to show that both p' and q' are CF-mappings in an undilated GC network.

We first observe that a dilated $N \times N$ GC network (e.g., the 8×8 dilated network in Fig. 3) consists of a first stage and two subnetworks. Each subnetwork is similar to an undilated $N \times N$ GC network (e.g., the 8×8 network in Fig. 1). The input ports of each subnetwork may be numbered in the same way as the undilated network while the output ports are numbered differently. For example, Fig. 10 shows that the output ports of the upper subnetwork of the 8×8 dilated network are numbered 0, 1, 2, and 3, respectively, while those of the undilated network (shown at top right) are numbered 0, 2, 4, and 6, respectively. The difference between these two numberings can be represented by the shuffle function f. Similarly, as shown in Fig. 10, by applying f to output ports 4, 5, 6, and 7 of the lower subnetwork, we obtain output ports 1, 3, 5, and 7 of the undilated network (shown at bottom right), respectively.

In general, by applying f to the numbering of the output ports of the upper subnetwork of an $N \times N$ *dilated* network, we get the numbering of the corresponding output ports of an $N \times N$ *undilated* network. Since p' is obtained by applying f to the destination of the connections in p, we may establish the connections p' in the undilated network in the same way that the connections in p are established in the dilated network. Similarly, we may establish the connections in q' in the undilated network in a way similar to that the connections in q are established in the lower subnetwork of the dilated network. Since both p' and q' are CF-mappings in an undilated $N \times N$ network, P' is a Θ-permutation.

For example, as shown in Fig. 10, by setting the switches of an undilated network (shown at the top right in Fig. 10) in the same way as the upper subnetwork, the four paths in p' (see (2a)) can be established without crosstalk in the same way that the paths in p (see (1a)) are established in the upper subnetwork. Similarly, we may set an undilated network (as shown at the bottom right) according to the lower subnetwork, except that the switches at the last stage should be set differently. This permits the four paths in q' (see (2b)) to be established in the network without crosstalk in a similar way that the four paths in q (see (1b)) are established in the lower subnetwork. This shows that both p' in (2a) and q' in (2b) are a CF-mapping and thus the union of the two subsets is a Θ-permutation. □

In order to show that Θ actually contains more permutations than Ω, we first note that a reverse mapping of F exists. Denote such a mapping by F^-. This mapping can be accomplished by a reverse shuffle of all output ports. More specifically, let a reverse shuffle be denoted by f^-, where $f^-(D = d_1 \cdots d_{n-1}d_n) = d_n d_1 \cdots d_{n-1}$. Given a permutation $P = \{(S_i, D_i) | i = 1, 2, \cdots, N\}$, its image under the reverse mapping F^- is $P' = F^-(P) = \{(S_i, f^-(D_i)) | i = 1, 2, \cdots, N\}$. Note that both $F(F^-)$ and $F^-(F)$ are identity mappings. That is, given any P, $F(F^-(P)) = F^-(F(P)) = P$.

Lemma 3: F is not an *onto* mapping from Ω to Θ.

To prove the lemma, it is sufficient to show that there exists a Θ-permutation π, such that $F^-(\pi)$ is not an Ω-permutation. Note that Ω will again be considered in the context of an undilated network according to Definition 1.

Let π be the Θ-permutation constructed recursively in Theorem 2. π contains connection $(0, 0)$ in $\pi_{odd}(N)$ and connection $(N/2, N/2)$ in $\pi_{even}(N)$. Thus, $F^-(\pi)$ contains $(0, 0)$ and $(N/2, N/4)$. These two connections share the same switch (for input 0 and input $N/2$) and their destinations are both in the upper subnetwork, thus resulting in a conflict in setting that switch. The more formal techniques used in [17] may be applied to show that these two connections in $F^-(Q)$ conflict with each other. Therefore, $F^-(\pi)$ is not an Ω-permutation. For example, when $N = 8$, π contains $(0, 0)$ and $(4, 4)$. Thus, $F^-(\pi)$ contains two conflicting connections, $(0, 0)$ and $(4, 2)$, and thus is not an Ω-permutation.

Since F is an one-to-one mapping from Ω to Θ and F^- is its reverse mapping, the proof that F is not an onto mapping is thus completed. □

Based on the above two lemmas, the following is established.

Theorem 4: There are more Θ-permutations than Ω-permutations.

Note that this theorem implies that the time domain "dilation" is more powerful than the space domain dilation in realizing permutations in blocking networks. As will be shown next, the time domain approach enjoys a similar property as it needs less than twice as many time slots as the space domain approach when establishing an arbitrary set of connections.

IV. Establishing Arbitrary Connections with CF-Mappings

We now consider sets of arbitrary connections, which are not necessarily permutations. Such a set of connections may need to be established in the MIN if the MIN is used as a centralized switching hub. In [24], a dilated slipped banyan (DSB) architecture is proposed which emulates an $N \times N$ completely-connected network by repeatedly realizing N different permutations. Each of these permutations is realized in one CF-mapping for the duration of a time slot, and every input is connected to a different output each time a different permutation is realized. Therefore, a completely-connected network is emulated with N CF-mappings.

In the previous section, it was shown that not every permutation can be realized with two CF-mappings by an undilated network. Thus, it is not clear that an undilated GC network can emulate an $N \times N$ completely-connected network using just $2N$ CF-mappings. However, we establish the following theorem.

Theorem 5: An $N \times N$ completely-connected network can be emulated in $2N$ CF-mappings in an undilated GC network.

Proof: Let $P_1, P_2, \cdots, P_N$ be the N Ω-permutations used when a dilated network emulates a completely-connected network. Since all possible N^2 connections are contained in these permutations, we have $\bigcup_{i=1}^{N} P_i = I \times O$, where I denotes the set of N inputs and O denotes the set of N outputs.

To prove the theorem, we apply the F mapping in Lemma 2 to the above Ω-permutations and consider the permutations: $F(P_1), F(P_2), \cdots, F(P_N)$. Since each of these permutations is a Θ-permutation, it can be realized with two CF-mappings by an undilated network. Therefore, all these N Θ-permutations can be realized with $2N$ CF-mappings. It remains to be shown that any connection in $I \times O$ is in $\bigcup_{i=1}^{N} F(P_i)$.

Given any connection $(S, D) \in I \times O$. Let $D' = f^{-}(D)$. Since (S, D') is also a connection in $I \times O$, it belongs to P_k for some k $(1 \leq k \leq N)$. That is, $(S, D') \in P_k$. According to the definition of mapping F, we have $(S, f(D')) \in F(P_k)$. That is, $(S, f(f^{-}(D))) \in F(P_k)$ and thus $(S, D) \in F(P_k)$. □

For applications that may not require all possible connections at all times, using $2N$ CF-mappings results in low bandwidth utilization. The static and dynamic RTDM summarized in Section II-B are shown to be effective in achieving high communication efficiency [15], [16]. Although the results in [15], [16] do not consider crosstalk, the same ideas can be applied except that a set of arbitrary connections needs to be partitioned into several CF-mappings to avoid not only path conflicts but crosstalk as well. Note that, to establish a set of arbitrary connections R, more than one CF-mapping is needed even in a *dilated* network. This is because, as mentioned earlier, that space domain dilation does not resolve path conflicts. More specifically, a dilated blocking network is still a blocking network. Even in a dilated nonblocking (or rearrangeably nonblocking) network, path conflicts exist between connections with the same source or destination.

We are interested in determining the number of time slots (or CF-mappings) needed to establish a set of arbitrary connections using either the time domain or the space domain approach. To do so, let K_u and K_d be the number of CF-mappings resulted from the partitioning of R in an undilated and a dilated network, respectively. Clearly, $K_u \geq K_d$. Based on Theorem 5, $K_u = 2 \times K_d$ when $R = I \times O$.

When $R \subseteq I \times O$, simulations have been carried out to determine K_d and K_u. A set of random, distinct connection requests is generated from all possible N^2 connection requests. A greedy algorithm is used to partition this set into CF-mappings in either an undilated or a dilated network. Fig. 11 shows both K_d and K_u as a function of the number of requests when $N = 32$. From this figure, we see that $K_u < 2 \times K_d$. This result may be explained by the fact that the time domain approach integrates the solutions to both crosstalk and path conflict problems, and thus is more efficient than the space domain approach which deals with crosstalk avoidance only and still relies on the time multiplexing techniques to resolve path conflicts. For example, when establishing a set of connections that contains two connections with identical inputs or outputs, at least two CF-mappings are needed in either an undilated or a dilated network. As such, the space domain approach may lose its advantage of being able to save twice as much time as the time domain approach in such cases.

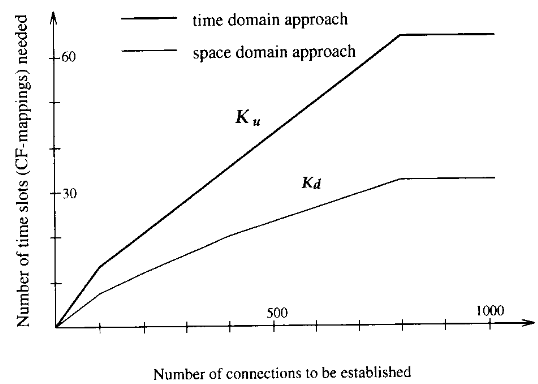


Fig. 11. Simulation results when $N = 32$.

V. Tradeoff Analysis

Ordinarily, a network having N input and output ports can support communications between N inputs and outputs. That is, up to N connections may be established in the network.

In order to support crosstalk-free communications among N inputs and outputs, the space domain approach increases the hardware complexity by dilating an $N \times N$ network into one that is essentially equivalent to a $2N \times 2N$ network. Although there may be other factors, we assume that the hardware complexity of a network is proportional to the number of switches, links and electronic driver circuits used in the network. As such, the hardware complexity of a dilated $N \times N$ (or a $2N \times 2N$) network can be considered as twice the hardware complexity of an (undilated) $N \times N$ network. Note that, however, the actual *cost* of a dilated $N \times N$ photonic switching network made of lithium niobate may be less than twice the cost of an $N \times N$ network, as long as constructing such a network is technologically and/or economically *feasible* [9], [25].

The time domain approach, on the other hand, increases time complexity by using more than one time slot to support crosstalk-free communications among N inputs and outputs. Note that this increase in the time complexity also represents a decrease in throughput, or usable bandwidth of the network. For example, in an 8×8 network with 2.5 Gb/s lasers, the time domain approach can only provide a (crosstalk-free) bandwidth of 10 Gb/s. If crosstalk had not been of any concern, an 8×8 configuration with 2.5 Gb/s lasers would have provided a bandwidth of 20 Gb/s. The usable bandwidth for crosstalk-free communications can only be increased if more hardware is used as in the space domain approach.

We note that to achieve crosstalk-free communications with about the same amount of hardware (e.g., a time-dilated $N\times N$ network or a space-dilated $N/2 \times N/2$ network), the space domain approach would have been able to connect only $N/2$ inputs to $N/2$ outputs. But if each of these $N/2$ inputs is time multiplexed with two input sources, and each of these $N/2$ outputs is time multiplexed with two output destinations, then the (crosstalk-free) bandwidth connecting the N inputs to N outputs in the space domain approach is the same as with the time domain approach.

If the future technology allows the transmission rate to scale up faster than the network size, the proposed time domain approach will be a useful way to support crosstalk-free communications. That is, if the cost of increasing the bandwidth of each connection should become as "cheap" as (or even cheaper than) the cost of constructing a network whose size is twice its original size, then one may consider using the time domain approach instead of the space domain approach. For example, let us assume that a goal is to provide 32 channels between 32 input and output pairs with each channel having 2 Gb/s bandwidth. Using the proposed time domain approach, one can achieve the goal by using a 32×32 *undilated* network and lasers each operating at 4 Gb/s. This is appropriate *when* (and if) building a 32×32 *dilated* network which needs lasers that operate only at 2 Gb/s, is *less feasible* than building the smaller (undilated) network that needs lasers at 4 Gb/s. Of course, it would be ideal to have a 32×32 dilated network and use these high speed lasers (each operating at 4 Gb/s) to provide higher bandwidth. In summary, whenever the limit on the network size is reached before the limit on the available bandwidth, the time domain approach may be used as a way to trade the bandwidth for the desired connectivity, as many other multiplexing techniques would do.

We also note that so far, we have assumed that each switch in a network, whether dilated or undilated, is controlled individually. A network control algorithm will be responsible for processing a given set of connection requests and determining the state of each switch in the network. The time, as well as memory, needed for such an algorithm to determine a sequence of network configurations that can establish a given set of requested connections is on the same order whether the network is dilated or not (although, as can be seen from Fig. 11 for example, the number of configurations, i.e., the number of time slots, needed will be different). If column control (or stage control) [3] is used in either the dilated or the undilated network, the connectivity of the network in terms of its capability in realizing permutations will be reduced and the network control will be simplified. It would be interesting to investigate if results that are similar to those theorems and simulation results previously obtained under individual switch control can also be obtained under column control.

In [14], a space–time complexity measure is defined as the product of the number of switches (and links) used in the network and the number of time slots needed to establish a set of crosstalk-free connections. This is similar to the measure used in VLSI design [4], [6], [18], [22]. Simulation results presented in [14] show that the time domain approach improves the space-time tradeoffs over the network dilation approach when used to establish a set of arbitrary connections. An intuitive explanation is that the time domain approach integrates the solutions to the crosstalk and path conflict problems while space domain dilation only deals with crosstalk avoidance and still relies on time multiplexing techniques to resolve path conflicts. Note that although only blocking networks are simulated, these results are expected to be applicable to nonblocking networks as well since path conflicts exist among the connections having the same source or destination. This is in spite of the fact that the set of permutations realizable with two time slots by a nonblocking network would be the same as the set of permutations realizable by a *dilated* nonblocking network (both consist of all possible permutations).

VI. Summary

In this paper, a time domain approach which is an extension of RTDM is proposed. Using this approach, a set of connections to be established is partitioned into several subsets, so that the connections in each subset can be established simultaneously in the network without crosstalk. Such an approach may be regarded as a way to "dilate" a network in the time domain as opposed to the space domain network dilation approach.

The relationship between Θ, which is the set of permutations realizable with two CF-mappings by a GC network (or its equivalence) and Ω, which is the set of permutations realizable by a dilated GC network (or its equivalence), is studied. In particular, it is shown that there is a one-to-one but not onto mapping from Ω to Θ. In other words, the set Θ contains more permutations than the set Ω. This implies that the time domain "dilation" is more powerful in realizing permutations than space domain "dilation."

Note that since dilating a network usually increases the number of stages in the network as well, the time domain approach for avoiding crosstalk will result in less propagation delay and optical path loss than the network dilation approach. It is also more flexible since the same architecture may be used for applications with or without crosstalk budget problems. Finally, it is worth noting that the proposed time domain approach is a useful approach for overcoming the limits imposed on the network size, especially if future technology allows optical bandwidth to scale up faster than network sizes. On the other hand, if the technology allows the feasible construction of large networks and the bandwidth does not scale up accordingly, then the network dilation approach should be used.

Acknowledgment

The authors would like to thank Dr. M. Karol and the anonymous reviewers for their comments which helped us to improve this paper.

References

[1] N. K. Ailawadi, R. C. Alferness, G. D. Bergland, and R. A. Thompson, "Broadband photonic switching using guided-wave fabrics," *IEEE LTS Mag.*, pp. 38–42, May 1991.

[2] R. C. Alferness, "Guided-wave devices for optical communication," *IEEE J. Quantum Electron.*, vol. QE-17, pp. 946–957, 1981.

[3] K. E. Batcher, "The multi-dimensional access in STARAN," *IEEE Trans. Comput. (Special Issue on Parallel Processing)*, pp. 174–177, Feb. 1977.

[4] R. P. Brent and H. T. Kung, "The area-time complexity of binary multiplication," *J. ACM*, vol. 28, no. 3, pp. 521–534, 1981.

[5] D. Chiarulli, S. Levitan, R. Melhem, and C. Qiao, "Locality based control algorithms for reconfigurable optical interconnection networks," *Appl. Opt.*, vol. 33, pp. 1528–1537, Mar. 1994.

[6] H. Habelson and P. Andreae, "Information transfer and area-time tradeoffs for VLSI multiplication," *Commun. ACM*, vol. 23, no. 1, pp. 20–23, 1980.

[7] H. Hinton, "A non-blocking optical interconnection network using directional couplers," in *Proc. IEEE Global Telecommun. Conf.*, Nov. 1984, pp. 885–889.

[8] S. Korotky *et al.*, "An experimental synchronized optical network using a high-speed time-multiplexed Ti:$LiNbO_3$ switch," presented at the Optical Fiber Communication Conf., 1989.

[9] T. O. Murphy, C. T. Kemmerer, and D. T. Moser, "A 16×16 Ti:$LiNbO_3$ dilated benes photonic switch module," presented at the Topical Meeting on Photonic Switching, Postdeadline Paper PD3, Salt Lake City, Ut, Mar. 1991.

[10] K. Padmanabhan and A. Netravali, "Dilated network for photonic switching," *IEEE Trans. Commun.*, vol. COM-35, no. 12, pp. 1357–1365, Dec. 1987.

[11] C. Qiao, R. Melhem, D. Chiarulli, and S. Levitan, "Efficient channel allocation for routing in optically interconnected multiprocessor systems," in *SPIE Proc. Conf. Advances in Optical Information Processing V*, vol. 1704, Apr. 1992, pp. 428–439.

[12] C. Qiao and R. Melhem, "Time-division optical communications in multiprocessor arrays," *IEEE Trans. Comput.*, vol. 42, no. 5, pp. 577–590, 1993.

[13] C. Qiao, "A high speed interconnection paradigm for multiprocessors and its applications to optical interconnection networks," Ph.D. dissertation, Dept. Comput. Sci., Univ. Pittsburgh, 1993.

[14] C. Qiao, R. Melhem, D. Chiarulli, and S. Levitan, "A time domain approach for avoiding crosstalk," in *OSA Proc. Photonics in Switching*, vol. 16, Mar. 1993, pp. 133–137.

[15] C. Qiao and R. Melhem, "Reconfiguration with time-division multiplexed MINs for multiprocessor communications," *IEEE Trans. Parallel Distributed Syst.*, vol. 5, no. 4, pp. 337–352, 1994.

[16] C. Qiao, R. Melhem, D. Chiarulli, and S. Levitan, "Dynamic reconfiguration of optically interconnected networks with time division multiplexing," *J. Parallel Distributed Comput.*, vol. 22, pp. 268–278, 1994.

[17] C. Raghavendra and A. Varma, "Fault-tolerant multiprocessors with redundant-path interconnection networks," *IEEE Trans. Comput.*, vol. C-35, no. 4, pp. 307–316, Apr. 1986.

[18] J. E. Savage and S. Swamy, "Space-time tradeoffs for oblivious sorting and integer multiplication," Brown Univ., Tech. Rep. CS-37, 1979.

[19] J. Sharony, K. W. Cheung, and T. E. Stern, "The wavelength dilation concept in lightwave networks—Implementation and system considerations," *IEEE J. Lightwave Technol.*, vol. 11, no. 5/6, pp. 900–907, 1993.

[20] H. J. Siegel and S. D. Smith, "Study of multistage SIMD interconnection networks," in *Proc. Fifth Annu. Symp. Computer Architecture*, April 1978, pp. 223–229.

[21] R. Spanke, "Architectures for large non-blocking optical switches," *IEEE J. Quantum Electron.*, vol. QE-22, pp. 885–889, Aug. 1986.

[22] C. D. Thompson, "Area-time complexity for VLSI," in *Proc. 11th Annu. ACM Symp. Theory of Computing*, May 1979, pp. 81–88.

[23] R. A. Thompson, R. V. Anderson, J. V. Camlet, and P. P. Giordano, "Experimental modular switching system with a time-multiplexed photonic center stage," in *Proc. OSA Topical Meeting Photonic Switching*, Mar. 1989, pp. 212–218.

[24] R. A. Thompson, "The dilated slipped banyan switching network architecture for use in an all-optical local area network," *IEEE J. Lightwave Technol.*, vol. 9, no. 12, pp. 1780–1787, Dec. 1991.

[25] J. E. Watson *et al.*, "A low-voltage 8×8 Ti:$LiNbO_3$ switch with a dilated Benes architecture," *J. Lightwave Technol.*, vol. 8, no. 5, pp. 794–800, May 1990.

[26] C.-L. Wu and T. Y. Feng, "On a class of multistage interconnection networks," *IEEE Trans. Comput.*, vol. C-29, pp. 694–702, Aug. 1980.

Chunming Qiao (S'89–M'93) received the B.S. degree in computer science and engineering from the University of Science and Technology of China in 1985, and the M.S. and a Ph.D. degrees, both in computer science, from the University of Pittsburgh in 1990 and 1993, respectively.

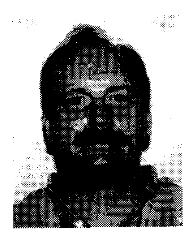
He joined the Department of Electrical and Computer Engineering, State University of New York at Buffalo, as an Assistant Professor in 1993. He has published in the areas of multiprocessor interconnection networks, time-division multiplexed optical communications, and photonic switching.

Dr. Qiao is a member of the International Society for Optical Engineering and the IEEE Computer Society and Communication Society.

Rami Georges Melhem was born in Cairo, Egypt, in 1954. He received the B.E. degree in electrical engineering from Cairo University in 1976, the M.A. degree in mathematics and the M.S. degree in computer science from the University of Pittsburgh in 1981, and the Ph.D. degree in computer science from the University of Pittsburgh in December 1983.

Since 1989, he has been an Associate Professor of Computer Science at the University of Pittsburgh.

Dr. Melhem as served on program committees of several conferences and workshops and is on the editorial board of the IEEE TRANSACTIONS ON COMPUTERS. He was a Guest Editor of a Special Issue of the *Journal of Parallel and Distributed Computing* on Optical Computing and Interconnection Systems. He is a member of the IEEE Computer Society, the Association for Computing Machinery, and the International Society for Optical Engineering.



Donald M. Chiarulli received the B.S. degree in Physics from Louisiana State University in 1976, the M.S. degree in computer science from Virginia Polytechnic Institute in 1979, and the Ph.D. degree in computer science from Louisiana State University in 1986.

He is an Associate Professor of Computer Science at the University of Pittsburgh. He has been active in high performance computing research since 1980. More recently, he and his colleagues at the University of Pittsburgh have successfully demonstrated the first all-optical addressing circuit for use in high performance backplanes.

Steven P. Levitan received the B.S. degree from Case Western Reserve University in 1972 and the M.S. and Ph.D. degrees, in computer science, from the University of Massachusetts, Amherst in 1979 and 1984, respectively.

He is the Wellington C. Carl Associate Professor of Electrical Engineering at the University of Pittsburgh. In 1987 he joined the Electrical Engineering faculty at the University of Pittsburgh. His research interests include computer aided design for VLSI, parallel computer architecture, parallel algorithm design, and VLSI design.

Dr. Levitan is a member of the IEEE Computer Society, ACM, SPIE, and OSA.