Permutation Capability of Optical Multistage Interconnection Networks

Yuanyuan Yang*
Department of Computer Science
University of Vermont, Burlington, VT 05405
yang@cs.uvm.edu

Jianchao Wang
GTE Laboratories
40 Sylvan Road, Waltham, MA 02254
jwang@gte.com

Yi Pan†
Department of Computer Science
University of Dayton, Dayton, OH 45469
pan@hype.cps.udayton.edu

Abstract

In this paper, we study optical multistage interconnection networks (MINs). Advances in electro-optic technologies have made optical communication a promising networking choice to meet the increasing demands for high channel bandwidth and low communication latency of high-performance computing/communication applications. Although optical MINs hold great promise and have demonstrated advantages over their electronic counterpart, they also hold their own challenges. Due to the unique properties of optics, crosstalk in optical switches should be avoided to make them work properly. Most of the research work described in the literature are for electronic MINs, and hence, crosstalk is not considered. In this paper, we introduce a new concept, semi-permutation, to analyze the permutation capability of optical MINs under the constraint of avoiding crosstalk, and apply it to two examples of optical MINs, banyan network and Benes network. For the blocking banyan network, we show that not all semi-permutations are realizable in one pass, and give the number of realizable semi-permutations. For the rearrangeable Benes network, we show that any semi-permutation is realizable in one pass and any permutation is realizable in two passes under the constraint of avoiding crosstalk. A routing algorithm for realizing a semi-permutation in a Benes network is also presented. With the speed and bandwidth provided by current optical technology, an optical MIN clearly demonstrates a superior overall performance over its electronic MIN counterpart.

*Research supported by the U.S. Army Research Office under Grant No. DAAH04-96-1-0234 and by the National Science Foundation under Grant No. OSR-9350540 and MIP-9522532.

†Research supported by the National Science Foundation under Grant No. CCR-9211621 and CCR-9503882.

1 Introduction

Communications among processors in a parallel computing system are always the main design issue when a parallel system is built or a parallel algorithm is designed. With advances in silicon and Ga-As technologies, processor speed will soon reach the gigahertz (GHz) range. Traditional metal-based communication technology used in parallel computing systems is becoming a potential bottleneck. This requires either that significant progress need to be made in the traditional interconnects, or that new interconnect technologies, such as optics, be introduced in parallel computing systems.

Advances in electro-optic technologies have made optical communication a promising networking choice to meet the increasing demands for high channel bandwidth and low communication latency of high-performance computing/communication applications. Fiber optic communications offer a combination of high bandwidth, low error probability, and gigabit transmission capacity. They have been extensively used in wide-area networks and have received much attention in parallel processing community as well. In fact, many commercial massively parallel computers such as Cray C90 use optical technology in their communication subsystems.

Using new optical technologies in parallel computers may require us to rethink how we design interconnection networks, and write parallel algorithms. Fully exploring the capabilities of the optical technology requires careful analysis of the properties of optics, proposal of new performance measures, design of new interconnection networks and routing algorithms, and new parallel application algorithms. Some research results in this area are described in [10].

Multistage interconnection networks (MINs) have been an important interconnecting scheme for parallel computing systems. A MIN can be blocking such as a banyan network [6], rearrangeably nonblocking such as a Benes network [1], nonblocking such as crossbar [15], or with variable connecting capabilities, from rearrangeable for permutation to nonblocking for multicast, such as a Clos network [1], [4], [19], depending on the number of stages, the number of switches, the switch capability, and the interconnection patterns used between stages. MINs have been extensively studied in the literature, e.g., see [1], [6], [4], [18], [19]. However, most of the research reported are for electronic MINs.

As optical technology advances, there are a lot of interests in using optical technology for implementing interconnection networks and switches. Although electronic MINs and optical MINs have many similarities, there are some fundamental differences between them. Because of some unique properties of optics, traditional routing algorithms and results are not applicable here. New research is needed to address the issues associated with optical MINs. In this paper, we will mainly consider the permutation capability of blocking and rearrangeably nonblocking optical MINs.

2 Preliminaries

In this section, we give some background and assumptions which are necessary for the analysis of optical MINs.

2.1 The switching model

An optical MIN can be implemented with either free-space optics or guided wave technology. Several optical crossbars and MINs have been proposed in the literature [8], [16]. Some implementations have also been reported [7], [15]. To exploit the huge optical bandwidth of fiber, Wavelength Division Multiplexing (WDM) technique can also be used, where the optical spectrum is divided into many different logical channels, and each channel corresponding to a unique wavelength [12]. In this paper, we consider optical implementation with guided wave technology. The switching model can be described briefly as follows.

Optical switching, as the name implies, involves the switching of optical signals, rather than electronic signals as in conventional electronic systems. Two types of guided wave optical switching systems can be identified. The first is a hybrid approach in which optical signals are switched, but the switches are controlled electronically. With this approach, the use of electronic control signals implies that the routing will be carried out electronically. As such, the speed of the electronic switch control signals can be much less than the bit rate of the optical signals being switched. The second approach is all-optical switching. This potentially overcomes the problem associated with the hybrid approach. However, such systems will not become practical until far into the future [9], and hence only the hybrid optical MINs are considered in this paper.

In hybrid optical MINs, the electronically controlled optical switches, such as Lithium Niobate directional couplers [5], can have switching speeds in the range from hundreds of picoseconds to tens of nanoseconds. If packet switching is used, the address information in each packet has to be decoded in order to determine the switch state. In a hybrid MIN, this such a requirement usually means conversions from optical signals to electronic ones, which could be very costly. For this reason, circuit switching is usually preferred in optical MINs. In this paper, we assume circuit switching is used.

Wide-band optical signals can be switched under electronic control using directional couplers between Ti:$LiNbO_3$ waveguides on a planar $LiNbO_3$ crystal [11]. The basic switching element is a directional coupler with two active inputs and two active outputs. Depending on the amount of voltage at the junction of the two waveguides which carry the two input signals, either of the two inputs can be coupled to either of the two outputs. Many architectures have been proposed to construct an $n \times n$ MIN using the 2×2 directional coupler as the basic component. These architectures are essentially analogs to similar architectures for electronic switching and MINs.

2.2 Limitations of optical MINs

However, due to the difference in speeds of the electronic and optical switching elements, and the nature of optical signal, optical MINs also hold their own challenges. One problem is *path dependent loss*. In a large MIN, a substantial part of this path dependent loss is directly proportional to the number of couplers that the optical path passes through. Hence, it depends on the architecture used and its network size. Another problem is *optical crosstalk*. Optical crosstalk occurs when two signal channels interact with each other . There are two ways in which optical paths can interact in a planar switching network. The channels carrying the signals could cross each other in order to embed a particular topology. Alternatively, two paths sharing a switch will experience some undesired coupling from one path to another within a switch. For example, assume that the two inputs are y and z, respectively, the two outputs will have $ly + lxz$ and $lz + lxy$, respectively, where l is signal loss and x is signal crosstalk in a switch. Using the best device reported in the literature [9], $x = -35$ dB, and $l = 0.25$dB. For more practically available devices, it is more likely that $x = -20$ dB and $l = 1$dB [9]. Hence, when a signal passes many switches, the input signal will be distorted at the output due to the loss and crosstalk introduced on the path. Experimental results [11] show that it is possible to make the crosstalk from passive intersections of optical waveguides negligible by keeping the intersection angles above a certain minimum amount. Studies also indicate that the crosstalk problem is a more severe problem than the path dependent loss problem with current optical technology [8, 16]. Thus, switch crosstalk is the most significant factor which reduces the signal-to-noise ratio and limits the size of a network. Luckily, first order crosstalk can be eliminated by ensuring that a switch is not used by two input signals simultaneously. Once the major source of crosstalk disappears, crosstalk in an optical MIN will have very small effect on the signal-to-noise ratio and thus a large optical MIN can be built and effectively used in parallel computing systems. In the following discussion, switch crosstalk will be simply referred to as *crosstalk*.

3 Approaches to Solving Crosstalk Problems

One way to solve the crosstalk problem is a space domain approach, where a MIN is duplicated and combined to avoid crosstalk. Based on this idea, a dilated Benes network has been proposed [11] to reduce the crosstalk level by allowing only one signal to pass through a switch at any given time. The number of switches required for the same connectivity in a dilated Benes network is slightly larger than twice of the regular Benes network. The dilated slipped banyan network (DSB) proposed by Thompson [17] is another example using this approach. Clearly, this approach uses more than double of the original network hardware to achieve the same permutation capability.

Another way to implement this idea is a time domain approach [14]. A set of connections is partitioned into several subsets such that the connections in each subset can be established simultaneously in a network without crosstalk. These subsets can be used to determine a sequence of configurations that the network goes through in order to establish the set of connections. This approach makes sense in optical MINs for two reasons. First, most multiprocessors use electronic processors and optical MINs. There is a big mismatch between the slow processing speed in processors and the high communication speed in networks. Using time-division multiplexing, even though we need several passes to satisfy the connection request from the processors, the total time used to transfer all the messages is still small compared with the computation time of processors. Second, there is a mismatch between the routing control and the fast signal transmission speed. Calculating the routing bits on-fly is impossible for optical MINs. A general method is to generate these control bits before a computation through the analysis of the communication patterns in an application during its compiling. In the DSB network, both space domain and time domain approaches are used. Because a banyan network is a blocking network, even after its dilation, many permutations can still not be realized in one pass. In order to realize any permutation, a time domain approach is also used there. For example, Thompson [17] has shown that any connection between an input and an output in the DSB network is established once in every n time slots. In other words, the DSB emulates a fully-connected network in n passes. For undilated MINs, Qiao [12] has shown that an n node fully-connected network can be emulated in $2n$ passes in an undilated Omega network (or its equivalent). A method for diagnosing crosstalk in an optical Benes network has also been proposed by Qiao [13].

In this paper, we will consider how to realize permutations in an undilated optical MIN efficiently using the time domain approach. Because each switch can only pass one signal at a given time, any permutation in an optical MIN requires at least two passes. We will consider both blocking networks such as banyan networks and rearrangeable networks such as Benes networks under the avoiding crosstalk constraint that for any switch in the network, only one signal is allowed to pass through the switch at any time. Clearly, under this constraint, a 2×2 switch has the most efficient switch utilization. This is because that when a $k \times k$ switch is used ($k \geq 2$), since each time we allow only one pair of input and output active, the rest of $k-1$ inputs and outputs have to be idle. Therefore, in the rest of the paper we only discuss the multistage networks consist of 2×2 switches. For this type of network, we will introduce a new concept, *semi-permutation*, which is a special type of partial permutation and has the maximum potential to be realized in an optical network without crosstalk.

4 Semi-Permutations

We consider an $n \times n$ multistage interconnection network with n inputs and n outputs where $n = 2^m$. A *permutation* for a network is a pairing of its inputs and outputs such that each input appears in exactly one pair and each output appears also in exactly one pair. In other words, a permutation is a full one-to-one mapping between the network inputs and outputs. For an $n \times n$ network, suppose input x_i is mapped to output y_i, where $x_i = i$ and $y_i \in \{0, 1, \ldots, n-1\}$ for $i = 0, 1, \ldots, n-1$. We denote this permutation as

$$\begin{pmatrix} x_0 & x_1 & \ldots & x_{n-1} \\ y_0 & y_1 & \ldots & y_{n-1} \end{pmatrix}.$$

In addition, we will refer to a one-to-one mapping between n' network inputs and n' network outputs ($n' < n$) as a *partial permutation*.

Apparently, a crosstalk-free optical network can not realize a permutation in a single pass, since at least the two input links on an input switch or the two output links on an output switch cannot be active in the same pass. Then an interesting question we may ask is: What is minimum number of passes required for realizing a permutation in such a network? In other words, we are interested in what types of partial permutations could be possibly realized in an optical network under the constraint of avoiding crosstalk. For an optical network consisting of 2×2 switches, it is useful to introduce the following definition.

Definition 1 *A partial permutation*

$$\begin{pmatrix} x_0 & x_1 & \ldots & x_{\frac{n}{2}-1} \\ y_0 & y_1 & \ldots & y_{\frac{n}{2}-1} \end{pmatrix}$$

of an n-element set $\{0, 1, \ldots, n-1\}$, where n is an even integer, $x_i, y_i \in \{0, 1, \ldots, n-1\}$ and $x_0 < x_1 < \cdots < x_{\frac{n}{2}-1}$, is referred to as a semi-permutation of the n-element set, if

$$\left\{ \left\lfloor \frac{x_0}{2} \right\rfloor, \left\lfloor \frac{x_1}{2} \right\rfloor, \ldots, \left\lfloor \frac{x_{\frac{n}{2}-1}}{2} \right\rfloor \right\} = \left\{0, 1, \ldots, \frac{n}{2} - 1\right\}, \text{ and}$$

$$\left\{ \left\lfloor \frac{y_0}{2} \right\rfloor, \left\lfloor \frac{y_1}{2} \right\rfloor, \ldots, \left\lfloor \frac{y_{\frac{n}{2}-1}}{2} \right\rfloor \right\} = \left\{0, 1, \ldots, \frac{n}{2} - 1\right\}.$$

Example 1 *For $n = 8$, partial permutation*

$$\begin{pmatrix} 0 & 3 & 4 & 6 \\ 1 & 5 & 3 & 7 \end{pmatrix}$$

is a semi-permutation, since we have

$$\left\{ \left\lfloor \frac{0}{2} \right\rfloor, \left\lfloor \frac{3}{2} \right\rfloor, \left\lfloor \frac{4}{2} \right\rfloor, \left\lfloor \frac{6}{2} \right\rfloor \right\} = \{0, 1, 2, 3\}, \text{ and}$$

$$\left\{ \left\lfloor \frac{1}{2} \right\rfloor, \left\lfloor \frac{5}{2} \right\rfloor, \left\lfloor \frac{3}{2} \right\rfloor, \left\lfloor \frac{7}{2} \right\rfloor \right\} = \{0, 2, 1, 3\} = \{0, 1, 2, 3\}.$$

Clearly, a semi-permutation is a partial permutation that ensures that there is only one active link passing through each input switch and output switch, that is, it eliminates crosstalk in the first and last stages in the network, and thus it has the potential to be realized in an optical network under the constraint of avoiding crosstalk. Of course, to ensure the entire network crosstalk-free, we need to eliminate crosstalk in the switches in the intermediate stages as well.

4.1 Decomposition of a permutation into semi-permutations

In this subsection, we first show that there is a nice property for permutations that any permutation can be decomposed into semi-permutations. We then give an efficient algorithm for such decompositions.

4.1.1 Decomposability

Theorem 1 *Any permutation can be decomposed into two semi-permutations.*

Proof. We will prove the theorem by using a combinatorial theorem of P. Hall [2].

Let the permutation be

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 & \ldots & x_{n-2} & x_{n-1} \\ y_0 & y_1 & y_2 & y_3 & \ldots & y_{n-2} & y_{n-1} \end{pmatrix} \tag{1}$$

where $x_i = i$ for $0 \le i \le n-1$ and $\{y_0, y_1, \ldots, y_{n-1}\} = \{0, 1, \ldots, n-1\}$. Clearly, this permutation can be decomposed into $\frac{n}{2}$ partial permutations

$$\begin{pmatrix} x_0 & x_1 \\ y_0 & y_1 \end{pmatrix}, \begin{pmatrix} x_2 & x_3 \\ y_2 & y_3 \end{pmatrix}, \ldots, \begin{pmatrix} x_{n-2} & x_{n-1} \\ y_{n-2} & y_{n-1} \end{pmatrix}$$

Note that for the j^{th} partial permutation $\begin{pmatrix} x_{2j} & x_{2j+1} \\ y_{2j} & y_{2j+1} \end{pmatrix}$, where $0 \le j \le \frac{n}{2} - 1$, we have

$$\left\lfloor \frac{x_{2j}}{2} \right\rfloor = \left\lfloor \frac{x_{2j+1}}{2} \right\rfloor = j \tag{2}$$

Denote $\left\{ \left\lfloor \frac{y_{2j}}{2} \right\rfloor, \left\lfloor \frac{y_{2j+1}}{2} \right\rfloor \right\}$ as A_j for $j = 0, 1, \ldots, \frac{n}{2} - 1$. It is easy to see that there are two 0's, two 1's, …, two $(\frac{n}{2} - 1)$'s distributed in the $\frac{n}{2}$ 2-element sets $A_0, A_1, \ldots, A_{\frac{n}{2}-1}$. Now for any k sets ($1 \le k \le \frac{n}{2}$), $A_{i_1}, A_{i_2}, \ldots, A_{i_k}$, there are a total of $2k$ elements in the k sets, and these $2k$ elements form a multiset with the multiplicity of each element no more than two. Thus, the cardinality of the union of these k sets satisfies

$$|A_{i_1} \cup A_{i_2} \cup \cdots \cup A_{i_k}| \ge k \tag{3}$$

By Hall's distinct system representatives theorem [2], we know that (3) is the necessary and sufficient condition for these $\frac{n}{2}$ subsets to have a set of distinct system representatives, that is, there exist $a_0 \in A_0, a_1 \in A_1, \ldots, a_{\frac{n}{2}-1} \in A_{\frac{n}{2}-1}$ such that $a_i \ne a_j$ for any $i \ne j$ ($0 \le i, j \le \frac{n}{2} - 1$). This implies that

$$\{a_0, a_1, \ldots, a_{\frac{n}{2}-1}\} = \left\{0, 1, \ldots, \frac{n}{2} - 1\right\} \tag{4}$$

Since a_j is either $\left\lfloor \frac{y_{2j}}{2} \right\rfloor$ or $\left\lfloor \frac{y_{2j+1}}{2} \right\rfloor$, let a_j be $\left\lfloor \frac{y_{d_{j,1}}}{2} \right\rfloor$, and the other element in A_j be $\left\lfloor \frac{y_{d_{j,2}}}{2} \right\rfloor$. For the subscripts $d_{j,1}$ and $d_{j,2}$, we have

$$\{d_{j,1}, d_{j,2}\} = \{2j, 2j+1\} \tag{5}$$

Thus, the above permutation can be decomposed into two partial permutations

$$\begin{pmatrix} x_{d_{0,1}} & x_{d_{1,1}} & \ldots & x_{d_{\frac{n}{2}-1,1}} \\ y_{d_{0,1}} & y_{d_{1,1}} & \ldots & y_{d_{\frac{n}{2}-1,1}} \end{pmatrix} \tag{6}$$

and

$$\begin{pmatrix} x_{d_{0,2}} & x_{d_{1,2}} & \ldots & x_{d_{\frac{n}{2}-1,2}} \\ y_{d_{0,2}} & y_{d_{1,2}} & \ldots & y_{d_{\frac{n}{2}-1,2}} \end{pmatrix} \tag{7}$$

Next, we will verify that both (6) and (7) are semi-permutations. Notice that

$$\left\{ \left\lfloor \frac{x_{d_{0,1}}}{2} \right\rfloor, \left\lfloor \frac{x_{d_{1,1}}}{2} \right\rfloor, \ldots, \left\lfloor \frac{x_{d_{\frac{n}{2}-1,1}}}{2} \right\rfloor \right\} \overset{\text{by }(2)}{=} \left\{ \left\lfloor \frac{d_{0,1}}{2} \right\rfloor, \left\lfloor \frac{d_{1,1}}{2} \right\rfloor, \ldots, \left\lfloor \frac{d_{\frac{n}{2}-1,1}}{2} \right\rfloor \right\} \overset{\text{by }(5)}{=} \left\{0, 1, \ldots, \frac{n}{2} - 1\right\}$$

$$\left\{ \left\lfloor \frac{y_{d_{0,1}}}{2} \right\rfloor, \left\lfloor \frac{y_{d_{1,1}}}{2} \right\rfloor, \ldots, \left\lfloor \frac{y_{d_{\frac{n}{2}-1,1}}}{2} \right\rfloor \right\} = \{a_0, a_1, \ldots, a_{\frac{n}{2}-1}\} \overset{\text{by }(4)}{=} \left\{0, 1, \ldots, \frac{n}{2} - 1\right\} \tag{8}$$

Therefore, by Definition 1, (6) is a semi-permutation. Since

$$\left\{ \left\lfloor \frac{y_{d_{0,1}}}{2} \right\rfloor, \ldots, \left\lfloor \frac{y_{d_{\frac{n}{2}-1,1}}}{2} \right\rfloor, \left\lfloor \frac{y_{d_{0,2}}}{2} \right\rfloor, \ldots, \left\lfloor \frac{y_{d_{\frac{n}{2}-1,2}}}{2} \right\rfloor \right\}$$

is a multiset on $\left\{1, 2, \ldots, \frac{n}{2} - 1\right\}$ with the multiplicity of each element equal to two, from (8) we have

$$\left\{ \left\lfloor \frac{y_{d_{0,2}}}{2} \right\rfloor, \left\lfloor \frac{y_{d_{1,2}}}{2} \right\rfloor, \ldots, \left\lfloor \frac{y_{d_{\frac{n}{2}-1,2}}}{2} \right\rfloor \right\} = \left\{0, 1, \ldots, \frac{n}{2} - 1\right\}.$$

Thus, (7) is also a semi-permutation. □

Example 2 *The decomposition of a permutation into two semi-permutations.*

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 0 & 7 & 5 & 3 & 4 & 6 & 2 \end{pmatrix} => \begin{pmatrix} 0 & 3 & 4 & 6 \\ 1 & 5 & 3 & 6 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 5 & 7 \\ 0 & 7 & 4 & 2 \end{pmatrix}$$

4.1.2 The algorithm

Theorem 1 guarantees the correctness of the decomposition. In the following, we describe a much simpler yet efficient algorithm to actually decompose a permutation into two semi-permutations.

Given a permutation of the form (1), in which input x_i is mapped to output y_i, where $0 \le i \le n-1$. Define $A_j^{[1]}$ and $A_j^{[2]}$ both as the two-element set $\{2j, 2j+1\}$ for $0 \le j \le \frac{n}{2} - 1$, where $A_j^{[1]}$ and $A_j^{[2]}$ correspond to the inputs and the outputs, respectively.

Now we construct an undirected bipartite graph $G = (V_1, V_2; E)$. The vertex sets of G are defined as

$$V_1 = \{A_0^{[1]}, A_1^{[1]}, \ldots, A_{\frac{n}{2}-1}^{[1]}\}, V_2 = \{A_0^{[2]}, A_1^{[2]}, \ldots, A_{\frac{n}{2}-1}^{[2]}\},$$

and the edge set E of G is defined as: for any one-pair mapping $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$ in the permutation (1), if $x_i \in A_{j_1}^{[1]}$ and $y_i \in A_{j_2}^{[2]}$, then there is an edge between vertex $A_{j_1}^{[1]}$ and vertex $A_{j_2}^{[2]}$ in E. We also assign each edge in E a label representing the corresponding one-pair mapping in the permutation. This bipartite graph has the following properties:

1. $|V_1| = |V_2| = \frac{n}{2}$ and $|E| = n$,
2. the graph may have parallel edges,
3. each vertex of the graph has degree two,
4. the graph may consist of more than one connected component.

From graph theory [3], we know that for a component of a graph in which each vertex has an even degree, there exists an Euler tour which traverses each edge exactly once. In particular, in our case since each vertex has degree two, the Euler tour becomes a cycle.

We are now in the position to give a high-level description of the decomposition algorithm.

Decomposition Algorithm:

Step 1: Construct a bipartite graph G for the given permutation.

Step 2: For each connected component of G, start from a vertex of this component in V_1, traverse through an unvisited edge to the neighbor vertex in V_2, back and forth until return to the starting vertex. (During the traversing, a visited edge is marked "forward" if the traverse direction on this edge is from V_1 to V_2; and marked "backward" if the direction is opposite.)

Step 3: Take all one-pair mappings corresponding to the edges marked with "forward", to form one semi-permutation; let the remaining one-pair mappings, corresponding to the edges marked with "backward", form another semi-permutation.

End

The above algorithm is correct because from the properties of the bipartite graph G listed above, the set of all edges marked with "forward" is a perfect matching of the bipartite graph G, and so is that marked with "backward." Also, it is easy to see that the time to construct the bipartite graph is proportional to the number of pairs in the permutation, i.e., $O(n)$, and the time to traverse all edges is $O(|E|) = O(n)$. Hence, the complexity of the decomposition algorithm is $O(n)$.

Let's apply the above algorithm to Example 2 in the last subsection. The bipartite graph and edge traverses are shown in Figure 1, where

$$e_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 2 \\ 7 \end{pmatrix}, e_3 = \begin{pmatrix} 3 \\ 5 \end{pmatrix},$$

$$e_4 = \begin{pmatrix} 4 \\ 3 \end{pmatrix}, e_5 = \begin{pmatrix} 5 \\ 4 \end{pmatrix}, e_6 = \begin{pmatrix} 6 \\ 6 \end{pmatrix}, e_7 = \begin{pmatrix} 7 \\ 2 \end{pmatrix}.$$

Then the "forward" pairs e_0, e_3, e_4, and e_6 form

$$\begin{pmatrix} 0 & 3 & 4 & 6 \\ 1 & 5 & 3 & 6 \end{pmatrix},$$

and the "backward" pairs e_1, e_2, e_5, and e_7 form

$$\begin{pmatrix} 1 & 2 & 5 & 7 \\ 0 & 7 & 4 & 2 \end{pmatrix},$$

which completes the decomposition.

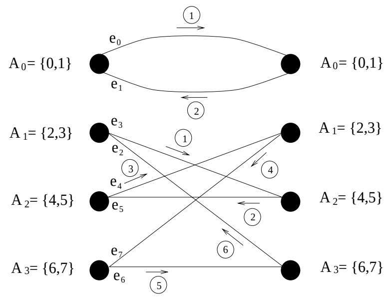

Figure 1: The bipartite graph and edge traverses of the decomposition.

4.2 The number of semi-permutations

Another interesting question we may ask for semi-permutations is: What is the total number of semi-permutations for an even integer n?

Theorem 2 *For any even integer n, the total number of semi-permutations for an n-element set is* $2^n \cdot (\frac{n}{2})!$.

Proof. Let the n-element set be $\{0, 1, \ldots, n-1\}$. Define a two-element set A_j $(0 \leq j \leq \frac{n}{2}-1)$ as $\{2j, 2j+1\}$. Consider a permutation for $\frac{n}{2}$-element set $\{A_0, A_1, \ldots, A_{\frac{n}{2}-1}\}$

$$\begin{pmatrix} A_0 & A_1 & \ldots & A_{\frac{n}{2}-1} \\ A_{i_0} & A_{i_1} & \ldots & A_{i_{\frac{n}{2}-1}} \end{pmatrix} \qquad (9)$$

where indexes $(i_0 \ i_1 \ \ldots \ i_{\frac{n}{2}-1})$ are a permutation of $\frac{n}{2}$-element set $\{0, 1, \ldots, \frac{n}{2}-1\}$. Take any $x_j \in A_j$ and $y_j \in A_{i_j}$ $(0 \leq j \leq \frac{n}{2}-1)$ in (9), then

$$\begin{pmatrix} x_0 & x_1 & \ldots & x_{\frac{n}{2}-1} \\ y_0 & y_1 & \ldots & y_{\frac{n}{2}-1} \end{pmatrix}$$

forms a semi-permutation for n-element set $\{0, 1, \ldots, n-1\}$. We have a total of $2^{\frac{n}{2}} \cdot 2^{\frac{n}{2}} = 2^n$ different such semi-permutations corresponding to one permutation of the form (9). On the other hand, any semi-permutation for n-element set $\{0, 1, \ldots, n-1\}$ corresponds to one permutation for $\frac{n}{2}$-element set $\{A_0, A_1, \ldots, A_{\frac{n}{2}-1}\}$ of some form (9). Note that there are a total of $(\frac{n}{2})!$ permutations of the form (9). Therefore, there are a total of $2^n \cdot (\frac{n}{2})!$ semi-permutations for n-element set $\{0, 1, \ldots, n-1\}$. □

Now, the problem of realizing a permutation in a crosstalk-free network can be transformed into the problem of realizing semi-permutations in the crosstalk-free network.

However, it should be pointed out that introducing the semi-permutation concept in a network composed of 2×2 switches can only guarantee crosstalk-free in the switches in the input stage and the output stage of the network. In fact, realizing a semi-permutation in a single pass implies that there is only one active input on each switch in the input stage and only one active output on each switch in the output stage. To ensure the entire network crosstalk-free, we need to know if there exists a proper routing that can eliminate crosstalk in the switches in the intermediate stages along different active paths. In the subsequent sections, we will look into this issue for two different types of networks.

5 Realizing Semi-Permutations in a Banyan Network

Banyan networks were first introduced by Goke and Lipovski [6] using graph methods. In a simplifying approach, any multistage network for which there is a unique path from each network input to each network output is called a banyan network. The banyan network considered in this paper is an $n \times n$ network composed of 2×2 switches, where $n = 2^m$. The network has m stages of switches and the switches in the consecutive stages are linked by recursively applying butterfly interconnection patterns. Figure 2 shows a 16×16 banyan network composed of 2×2 switches. Figure 3 illustrates the recursive construction of an $n \times n$ banyan network.

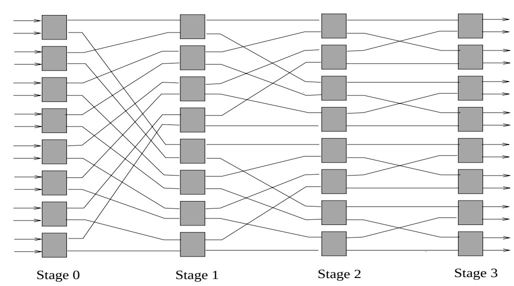

Figure 2: A 16×16 banyan network composed of 2×2 switches.

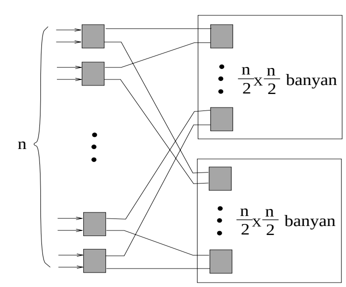

Figure 3: Recursive construction of an $n \times n$ banyan network.

Due to the unique path nature of a banyan network, a semi-permutation is routed through the network in a fixed switch setting. Consequently, some semi-permutations can be realized in a banyan network in a single pass while others cannot. In Figure 4, we give examples of such semi-permutations for an 8×8 banyan network. Figure 4(a) shows that semi-permutation $\begin{pmatrix} 0 & 2 & 4 & 6 \\ 3 & 5 & 7 & 1 \end{pmatrix}$ can be realized in a single pass, and Figure 4(b) shows that semi-permutation $\begin{pmatrix} 0 & 2 & 5 & 6 \\ 4 & 6 & 0 & 2 \end{pmatrix}$ cannot be realized in a single pass.

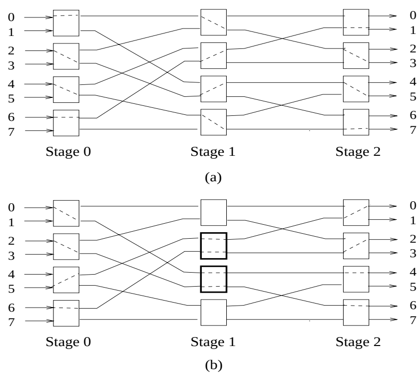

Figure 4: Realizing semi-permutations in an 8×8 banyan network. (a) A semi-permutation that can be realized in a single pass. (b) A semi-permutation that cannot be realized in a single pass. Bold boxes indicate the switches with crosstalk.

Then, it would be interesting to know how many semi-permutations can be realized in a banyan network in a single pass under the constraint of avoiding crosstalk.

Theorem 3 *The number of semi-permutations that can be realized in an $n \times n$ banyan network in a single pass under the constraint of avoiding crosstalk is $2^{\frac{3}{4}n} \cdot n^{\frac{n}{4}}$.*

Proof. Let $S(n)$ be the total number of semi-permutations that can be realized in an $n \times n$ banyan network without crosstalk. Let $F(n)$ be the total number of semi-permutations for the given fixed active inputs in the $\frac{n}{2}$ switches in the input stage. By the symmetry of semi-permutations, we have

$$S(n) = 2^{\frac{n}{2}} \cdot F(n) \tag{10}$$

Consider the recursive definition of an $n \times n$ banyan network (Figure 3). For the $\frac{n}{2}$ switches in input stage $u_0, u_1, \ldots, u_{\frac{n}{2}-1}$, we pair them off as follows:

$$\{u_0, u_1\}, \{u_2, u_3\}, \ldots, \{u_{\frac{n}{2}-2}, u_{\frac{n}{2}-1}\}.$$

There are a total of $\frac{n}{4}$ such pairs. Now, let's consider the interstage connections between the first stage of the network and the rest of the network. For each of such a pair, the upper outputs of both switches are linked to a switch in the upper $\frac{n}{2} \times \frac{n}{2}$ banyan network, the lower outputs of both switches are linked to a switch in the lower $\frac{n}{2} \times \frac{n}{2}$ banyan network, and only input switches in the pair are linked to those two switches in the next stage (one is in the upper sub banyan network, and the other is in the lower sub banyan network). To guarantee crosstalk-free, partial permutations on the upper and lower sub banyan networks must be semi-permutations. There are two possible ways for each input switch pair to be linked to (some fixed inputs) in the next stage, that is, letting the output of a switch go to the upper sub-net and the output of the other switch go to the lower sub-net, or verse visa. Since there are a total of $\frac{n}{4}$ such pairs, we immediately obtain a recurrence for $F(n)$

$$F(n) = 2^{\frac{n}{4}} \left[F\left(\frac{n}{2}\right) \right]^2 \tag{11}$$

Also note that $F(2) = 2$. Therefore,

$$F(n) = 2^{k \cdot \frac{n}{4}} \left[F\left(\frac{n}{2^k}\right) \right]^{2^k} = 2^{(\log n - 1) \cdot \frac{n}{4}} \cdot 2^{\frac{n}{2}} = 2^{\frac{n}{4}} \cdot n^{\frac{n}{4}}$$

Then from (10), we have $S(n) = 2^{\frac{3n}{4}} \cdot n^{\frac{n}{4}}$. □

By comparing Theorem 3 and Theorem 2, we can see that there are a substantial amount of semi-permutations that cannot be realized in a banyan network, especially when n gets larger.

6 Realizing Semi-Permutations in a Benes Network

A Benes network can be constructed by concatenating a banyan network and a reverse banyan network with the center stages overlapped. Figure 5 is an example of a 16×16 Benes network. It is symmetric about the central stage. A Benes network can also be defined in a recursive fashion as shown in Figure 6. In an $n \times n$ Benes network, the upper or lower output of each input switch is linked to an input of the upper or lower $\frac{n}{2} \times \frac{n}{2}$ Benes network, respectively, and the upper or lower input of each output switch is linked from an output of the upper or lower $\frac{n}{2} \times \frac{n}{2}$ Benes network, respectively. Electronic Benes networks are well known for being capable of realizing all possible permutations [1]. In the following, we will show that Benes networks also have good properties to support permutations in optical networks.

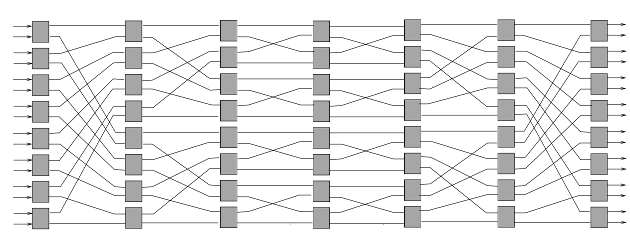

Figure 5: A 16×16 Benes network composed of 2×2 switches.

Theorem 4 *Any semi-permutation can be realized in a Benes network in a single pass under the constraint of avoiding crosstalk.*

Proof. Consider the recursive definition of an $n \times n$ Benes network (Figure 6). Let the $\frac{n}{2}$ input switches

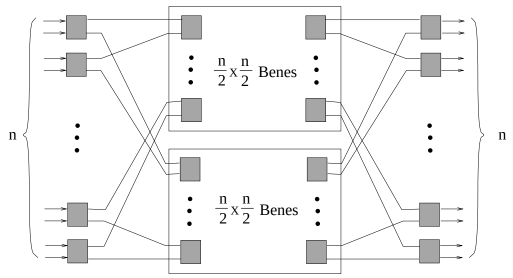

Figure 6: Recursive construction of an $n \times n$ Benes network.

be $u_0, u_1, \ldots, u_{\frac{n}{2}-1}$, and the $\frac{n}{2}$ output switches be $v_0, v_1, \ldots, v_{\frac{n}{2}-1}$. We pair all input switches into $\frac{n}{4}$ pairs as $A_j^{[1]} = \{u_{2j}, u_{2j+1}\}$ for $0 \le j \le \frac{n}{4} - 1$. Similarly, we pair all output switches into $\frac{n}{4}$ pairs as $A_j^{[2]} = \{v_{2j}, v_{2j+1}\}$ for $0 \le j \le \frac{n}{4} - 1$.

Since a Benes network consists of a banyan network and a reverse banyan network, the observations for banyan networks in the proof of Theorem 3 are applicable here. That is, both input switches in $A_j^{[1]}$ have links to some two switches in the next stage (one is to the upper sub Benes network and the other is to the lower sub Benes network), and no other input switches have links to any of those two switches in the next stage. Symmetrically, both output switches in $A_j^{[2]}$ have links from some two switches in the previous stage (one is from the upper sub Benes network and the other is from the lower sub Benes network), and no other output switches have links from any of those two switches in the previous stage.

A pair of input/output mapping can be realized by starting from the corresponding input switch via either the upper or the lower sub network and ending at the corresponding output switch. To ensure crosstalk-free in the entire network, a proper arrangement is needed to provide the upper and lower sub networks with semi-permutations, so that the routing can continue in a recursive fashion. To achieve this, for the pair of input switches in $A_j^{[1]}$, we connect the only active output of one switch to the upper sub network and the only active output of the other switch to the lower sub network. The arrangement for the pair of output switches in $A_j^{[2]}$ is similar.

Not surprisingly, the idea of the decomposition algorithm described in Section 4.1.2 can also serve for our purpose here. To do so, we build a bipartite graph $G = (V_1, V_2; E)$ as follows. The vertex sets are defined as

$$V_1 = \{A_0^{[1]}, A_1^{[1]}, \ldots, A_{\frac{n}{4}-1}^{[1]}\}, V_2 = \{A_0^{[2]}, A_1^{[2]}, \ldots, A_{\frac{n}{4}-1}^{[2]}\},$$

and the edge set E is defined as: for any one-pair mapping $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$ in the semi-permutation, if the input switch corresponding to x_i belongs to $A_{j_1}^{[1]}$ and the output switch corresponding to y_i belongs to $A_{j_2}^{[2]}$, then there is an edge between vertex $A_{j_1}^{[1]}$ and vertex $A_{j_2}^{[2]}$ in E. We assign each edge in E a label representing the corresponding one-pair mapping in the semi-permutation. Note that the bipartite graph is of size $|V_1| = |V_2| = \frac{n}{4}$ and $|E| = \frac{n}{2}$. The idea here is that after running the decomposition algorithm, for one-pair mapping $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$ marked with "forward", we make the input x_i pass the corresponding input switch, link to the upper sub network, then link to the corresponding output switch, and finally reach the output y_i; on the other hand, for one-pair mapping marked with "backward", we make the connection via the lower sub network.

Now the routing algorithm for a semi-permutation in an $n \times n$ Benes network can be obtained by slightly modifying the decomposition algorithm described in Section 4.1.2.

Routing Algorithm (integer n, SEMI_PERM $semi\_perm$):

Step 1: If n is 2, make the connection in the 2×2 switch according to $semi\_perm$; exit.

Step 2: Construct the bipartite graph $G = (V_1, V_2; E)$ corresponding to $semi\_perm$ in the $n \times n$ Benes network.

Step 3: Same as the Step 2 in Decomposition Algorithm in Section 4.1.2.

Step 4: Take all one-pair mappings corresponding to the edges marked with "forward" to form one semi-permutation $semi\_perm\_upper$, and for each of these one-pair mapping make a connection pass through the corresponding input switch and to the upper sub network, and from the upper sub network to the corresponding output switch and to the output; take all one-pair mappings corresponding to the edges marked with "backward" to form a semi-permutation $semi\_perm\_lower$, and for each of these one-pair mapping make a connection pass through the corresponding input switch and to the lower sub network, and from the lower sub network to the corresponding output switch and to the output.

Step 5: Recursively call Routing Algorithm ($\frac{n}{2}$, $semi\_perm\_upper$) in the upper sub Benes network.

Step 6: Recursively call Routing Algorithm ($\frac{n}{2}$, $semi\_perm\_lower$) in the lower sub Benes network.

End

The correctness of the algorithm is clear. Since Steps 1-4 take $O(n)$ steps and Steps 5-6 recursively call the same algorithm for $\frac{n}{2}$, it is easy to see that the time complexity of the routing algorithm is $O(n \log n)$. □

In the proof of the above theorem, we have actually given an efficient routing/switch setting up algorithm.

7 Realizing Permutations in a Benes Network

Finally, we consider how many passes are required to realize a permutation in a Benes network.

Theorem 5 *Any permutation can be realized in a Benes network in two passes under the constraint of avoiding crosstalk.*

Proof. By Theorem 1, we know that any permutation can be decomposed into two semi-permutations. Then by Theorem 4, each semi-permutation can be realized in a Benes network in a single pass. Thus, in two passes we can realized any permutation without crosstalk. □

It should be pointed out that a permutation requires at least two passes in any $n \times n$ optical MIN due to the constraint of avoiding crosstalk in the input stage of switches. In other words, two is the lower bound on the number of passes for any optical permutation networks under the constraint of avoiding crosstalk. Theorem 5 indicates that an undilated Benes network reaches this lower bound and realizes permutations optimally.

8 Conclusions

In this paper, we have analyzed the permutation capability of optical MINs. We introduced a new concept, semi-permutation, which ensures that there is only one active link passing through each input switch and output switch, and thus has the potential to be realized without crosstalk in an optical network. An efficient algorithm to decompose a permutation into two semi-permutations has also been developed. For the blocking banyan network, we have shown that not all semi-permutations are realizable in one pass, and given the number of realizable semi-permutations. For the rearrangeable nonblocking Benes network, we have shown that any semi-permutation is realizable in one pass and any permutation is realizable in two passes under the constraint of avoiding crosstalk. A routing algorithm for realizing a semi-permutation in a Benes network has been given. We believe that the concepts and analytical methods developed in this paper can also be used to solve the crosstalk problem in other optical MINs.

References

[1] V.E. Benes, "Heuristic remarks and mathematical problems regarding the theory of switching systems," *The Bell System Technical Journal,* vol. 41, pp.1201-1247, 1962.

[2] K.P. Bogart, *Introductory Combinatorics, 2nd Edition,* Harcourt Brace Jovanovich, 1990.

[3] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications,* American Elsevier Publishing Co., 1976.

[4] C. Clos, "A study of non-blocking switching networks," *The Bell System Technical Journal,* vol. 32, pp. 406-424, 1953.

[5] J. Erickson and H. Hinton, "Implementing a Ti:LiNbO_3 4×4 nonblocking interconnection network," *SPIE Integrated Optical Circuit Engineering,* vol. 578, pp. 201-206, 1985.

[6] L.R. Goke and G.J. Lipovski, "Banyan networks for partitioning multiprocessor systems," *Proc. of the First Annual Symposium on Computer Architecture*, 1973, pp. 21-28.

[7] A. Himeno, and M. Kobayashi, "4×4 optical-gate matrix switch," *J. Lightwave Technology,* vol. 3, pp. 230-235, Apr. 1985.

[8] H. Hinton, "A non-blocking optical interconnection network using directional couplers," *Proc. IEEE Global Telecom. Conference,* pp. 885-889, Nov. 1984.

[9] D. K. Hunter and I. Andonovic, "Guided wave optical switch architectures," *International Journal of Optoelectronics,* vol. 9, no. 6, pp. 477-487, 1994.

[10] K. Li, Y. Pan, and S.Q. Zheng, *Parallel Computing Using Optical Interconnections*, Kluwer Academic Publishers, Boston, USA to be published in 1998.

[11] K. Padmanabhan and A. N. Netravali, "Dilated networks for photonic switching," *IEEE Trans. Communications,* vol. 35, no. 12, pp. 1357-1365, Dec. 1987.

[12] C. Qiao, *A high-speed interconnection paradigm and its applications to optically interconnected networks,* PhD Dissertation, Department of Computer Science, University of Pittsburgh, 1993.

[13] C. Qiao, "A two-level process for diagnosing crosstalk in photonic dilated Benes network," *Journal of Parallel and Distributed Computing,* vol. 41, no. 1, pp. 53-66, 1997.

[14] C. Qiao, and R. Melhem, "A time domain approach for avoiding crosstalk in optical blocking multistage interconnection networks," *J. Lightwave Technology,* vol. 12. no. 10, pp. 1854-1862, Oct. 1994.

[15] A.A. Sawchuck, B.K. Jenkins, C.S. Raghavendra and A. Varma, "Optical crossbar networks," *Computer,* vol. 20, no. 6, pp. 50-60, June 1987.

[16] R. Spanke, "Architectures for large non-blocking optical switches," *IEEE J. Quantum Electro.,* vol. 22, no. 6, pp. 964-967, June 1986.

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

[18] Y. Yang and G.M. Masson, "Broadcast ring sandwich networks," *IEEE Trans. Computers,* vol. C-44, no. 10, pp. 1169-1180, 1995.

[19] Y. Yang and G.M. Masson, "Nonblocking broadcast switching networks," *IEEE Trans. Computers,* vol. C-40, no. 9, pp. 1005-1015, 1991.