Optimal All-to-All Personalized Exchange in a Class of Optical Multistage Networks

Yuanyuan Yang*
Dept. of Elec. & Computer Eng.
State Univ. of New York at Stony Brook
Stony Brook, NY 11794, USA
yang@ece.sunysb.edu

Jianchao Wang
GTE Laboratories
40 Sylvan Road
Waltham, MA 02454, USA
jwang@gte.com

Abstract

All-to-all personalized exchange is one of the most dense collective communication patterns and occurs in many important parallel computing/networking applications. In this paper, we look into the issue of realizing all-to-all personalized exchange in optical multistage networks. 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 multistage networks 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. In this paper, we will provide an optimal scheme for realizing all-to-all personalized exchange in a class of unique-path, self-routing optical multistage networks crosstalk-free. The basic idea of realizing all-to-all personalized exchange in such a multistage network is to transform it to multiple *semi-permutations*, each of which can be realized crosstalk-free in a single pass, and take advantage of pipelined message transmission in consecutive passes. As can be seen, the time complexity of our all-to-all personalized exchange algorithms matches the lower bound of the communication delay in this type of network.

1 Introduction

All-to-all personalized exchange is one of the most dense collective communication patterns, in which every processor in a processor group sends a distinct message to every other processor in the group. All-to-all personalized exchange occurs in many important parallel computing/networking applications, such as matrix transposition and fast Fourier transform (FFT) [1]-[2]. There has been much work for all-to-all personalized exchange in various networks; see, for example [3]-[9].

In this paper, we look into the issue of realizing all-to-all personalized exchange in optical multistage networks. A basic element of optical switching networks is a directional coupler with two inputs and two outputs (hereafter referred to simply as switches). Depending on the control voltage applied to it, an input optical signal is coupled to either of the two outputs, setting the switch to either the *parallel* or the *crossing* state. A class of topologies that can be used to construct optical networks is multistage interconnection networks (MINs) [10], which interconnect their inputs and outputs via several stages of switches.

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 multistage networks hold great promise and have demonstrated advantages over their electronic counterpart, they also introduce new challenges such as how to deal with the unique problem of avoiding *crosstalk* in the optical switches, which occurs when two signal channels in a switch interact with each other. There are two ways in which optical signals 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 may experience some undesired coupling from one path to another within a switch. Although it is possible to use the optical switches that do not have serious crosstalk properties, these switches are still too expensive for many applications. Luckily, first order crosstalk can be eliminated by ensuring that only one signal passes through a switch at a time, which provides a cost-effective solution to the crosstalk problem.

There has been some work in the literature on crosstalk-free routing in an optical multistage network; see, for example [11, 12, 13]. In [12], a concept called *semi-permutation* was introduced as a useful tool for designing crosstalk-free routing in an optical multistage network. In this paper, we will further explore the properties of semi-permutations in the context of crosstalk-free routing and develop all-to-all personalized exchange algorithms in optical multistage networks based on crosstalk-free semi-permutations. In general, the basic idea of realizing all-to-all personalized exchange in an optical multistage network is to transform it to multiple semi-permutations each of which can be realized crosstalk-free in a single pass, and take advantage of pipelined message transmission in consecutive passes.

Notice that any permutation can be realized in a Benes network in a single pass (in the electronic version), and any permutation can be realized in a Benes network in two passes crosstalk-free (in the optical version) [12]. However, the switch setting in a Benes network is complex and expensive, which may discourage people to choose this type of network for a high speed communication environment. Moreover, a full permutation (semi-permutation) capability may not be necessary for all-to-all personalized exchange. In this paper, we consider a class of unique-path, self-routing multistage networks such as base-

*Research supported in part 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.

line, omega, banyan networks, and their reverse networks [10]. A typical network structure for this class of network is that each network has $n (= 2^m)$ inputs and outputs and $\log n = m$ stages, with each stage consisting of $\frac{n}{2}$ 2×2 switches and any two adjacent stages connected by n interstage links. Figure 1(a), (b) and (c) illustrate an 8×8 baseline network, omega network and banyan network, respectively.

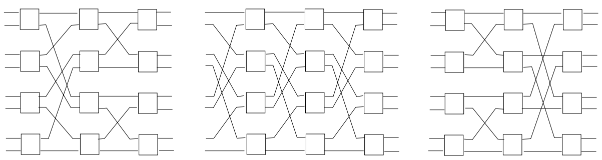

Figure 1: Three typical multistage networks.

This type of network has advantages of less hardware cost (almost half of that of a Benes network), and fast switch setting (self-routing). On the other hand, the limitations of such networks are that not every permutation is admissible to such a network, and not every admissible permutation to the network can be decomposed to semi-permutations which can be realized in a single pass crosstalk-free in the network. However, such limitations do not prohibit us to find some special permutations and semi-permutations which can be used for efficiently realizing all-to-all personalized exchange in such networks. In this paper, we will present efficient algorithms for all-to-all personalized exchange in this type of network. As can be seen later, the time complexity of our algorithms matches the lower bound of the communication delay in these networks.

2 Permutations and Semi-Permutations

A *permutation* is a one-to-one mapping between the network inputs and outputs. For an $n \times n$ network, suppose there is a one-to-one mapping ρ which maps input i to output a_i (i.e. $\rho(i) = a_i$), where $a_i \in \{0, 1, \ldots, n-1\}$ for $0 \leq i \leq n-1$, and $a_i \neq a_j$ for $i \neq j$. Let $\rho = \begin{pmatrix} 0 & 1 & \ldots & n-1 \\ a_0 & a_1 & \ldots & a_{n-1} \end{pmatrix}$ denote this permutation. In particular, when $\rho(i) = i$ for $0 \leq i \leq n-1$, we refer to this permutation as an *identity permutation* and denote it as I. A permutation which is realizable by the multistage network is referred to as an *admissible permutation* of the network.

Given two permutations ρ_1 and ρ_2, a *composition* $\rho_1\rho_2$ of the two permutations is also a permutation, which maps i to $\rho_1(\rho_2(i))$. Clearly, $\rho I = I\rho = \rho$, but in general $\rho_1\rho_2 \neq \rho_2\rho_1$. However, the associative law does apply here. That is, $\rho_1(\rho_2\rho_3) = (\rho_1\rho_2)\rho_3$. Let ρ^i denote the composition of i permutations ρ's. Also, if $\rho_1\rho_2 = I$, we call ρ_1 is the inverse of ρ_2 and vice versa, and denote them as $\rho_1 = \rho_2^{-1}$ and $\rho_2 = \rho_1^{-1}$. In addition, for representational convenience, we use $a \xrightarrow{\rho} b$ to represent a mapping $\rho(a) = b$.

When realizing a permutation in an optical multistage network in a single pass, as discussed earlier, switches suffer from crosstalk. *Semi-permutation* was thus introduced in [12] as a useful tool for crosstalk-free routing in an optical multistage network consisting of 2×2 switches.

Given an even integer n, a semi-permutation is defined as a one-to-one mapping between some $\frac{n}{2}$ inputs $\{x_0, x_1, \ldots, x_{\frac{n}{2}-1}\}$ and some $\frac{n}{2}$ outputs $\{y_0, y_1, \ldots, y_{\frac{n}{2}-1}\}$ satisfying

$$\left\{ \left\lfloor \frac{x_0}{2} \right\rfloor, \left\lfloor \frac{x_1}{2} \right\rfloor, \ldots, \left\lfloor \frac{x_{\frac{n}{2}-1}}{2} \right\rfloor \right\} = \left\{ \left\lfloor \frac{y_0}{2} \right\rfloor, \left\lfloor \frac{y_1}{2} \right\rfloor, \ldots, \left\lfloor \frac{y_{\frac{n}{2}-1}}{2} \right\rfloor \right\} = \left\{ 0, 1, \ldots, \frac{n}{2} - 1 \right\} \quad (1)$$

where $x_i, y_i \in \{0, 1, \ldots, n-1\}$, $y_i \neq y_j$ for $0 \leq i \neq j \leq \frac{n}{2} - 1$, and $x_0 < x_1 < \cdots < x_{\frac{n}{2}-1}$. The semi-permutation is denoted as

$$s = \begin{pmatrix} x_0 & x_1 & \ldots & x_{\frac{n}{2}-1} \\ y_0 & y_1 & \ldots & y_{\frac{n}{2}-1} \end{pmatrix}, \quad (2)$$

and the input set $\{x_0, x_1, \ldots, x_{\frac{n}{2}-1}\}$ is denoted as $InputSet(s)$ and the output set $\{y_0, y_1, \ldots, y_{\frac{n}{2}-1}\}$ is denoted as $OutputSet(s)$.

For example, for $n = 8$, the one-to-one mapping $\begin{pmatrix} 0 & 3 & 4 & 6 \\ 1 & 5 & 3 & 7 \end{pmatrix}$ is a semi-permutation.

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 crosstalk-free. Of course, to ensure the entire network crosstalk-free, we need to eliminate crosstalk in the switches in the intermediate stages as well.

In [12], the following result regarding the relationship between permutations and semi-permutations was given.

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

Given two semi-permutations decomposed from a permutation, we call one semi-permutation the *twin* of the other. Clearly, let the two semi-permutations be s and t, respectively, we have

$$InputSet(s) \cup InputSet(t) = OutputSet(s) \cup OutputSet(t) = \{0, 1, \ldots, n-1\}$$

If a semi-permutation can be realized in a multistage network in a single pass crosstalk-free, it is simply called a *crosstalk-free semi-permutation* for this network.

Lemma 1 *For the two semi-permutations decomposed from an admissible permutation of a multistage network, if one is crosstalk-free for the network, so is the other.*

Proof. Omitted (see [15] for detailed proof). ■

A decomposition of a permutation is called a *crosstalk-free decomposition* to a multistage network if its semi-permutations are crosstalk-free realizable to the network.

We can also extend the concept of composition of permutations to semi-permutations. Given two semi-permutations s_1 and s_2 with $InputSet(s_1) = OutputSet(s_2)$, a *composition* $s_1 s_2$ is defined as the mapping that i is mapped to $s_1(s_2(i))$ for $i \in InputSet(s_2)$. Clearly, the composition of two semi-permutations s_1 and s_2 is also a semi-permutation with the input set being $InputSet(s_2)$ and the output set being $OutputSet(s_1)$. We can iteratively define the composition of more than one semi-permutations. Let $s_1 s_2$ be a composition of two semi-permutations s_1 and s_2, and let s_3 be the third semi-permutations satisfying $InputSet(s_1 s_2) = OutputSet(s_3)$, i.e. $InputSet(s_2) = OutputSet(s_3)$. Then $s_1 s_2 s_3$ is a composition defined as $(s_1 s_2) s_3$ with the input set being $InputSet(s_3)$ and the output set being $OutputSet(s_1)$. It can be easily verified that the associate law holds for the composition operation, that is $(s_1 s_2) s_3 = s_1 (s_2 s_3)$.

2.1 Stage and Interstage Permutations and Crosstalk-Free Decomposition

In the context of a multistage network, each stage in the network can be viewed as a shorter $n \times n$ network, and so does each set of interstage links. Let σ_i $(0 \leq i \leq m-1)$ denote the permutation represented by stage i, and π_i $(0 \leq i \leq m-2)$ denote the permutation represented by the set of interstage links between stage i and stage $i+1$. We refer to the permutation σ_i as a *stage permutation*, the permutation π_i as an *interstage permutation*. Clearly, an admissible permutation of a multistage network can be expressed by a composition of stage permutations and interstage permutations. For example, an admissible permutation of a baseline network can be expressed as

$$\sigma_{m-1}\pi_{m-2}\sigma_{m-2}\ldots\pi_0\sigma_0 \tag{3}$$

In general, interstage permutations π_i's are fixed by the network topology. For a baseline network, suppose the binary representation of a number $a \in \{0,1,\ldots,n-1\}$ is $p_{m-1}p_{m-2}\ldots p_1p_0$. Then the permutation π_i represents the following mapping

$$p_{m-1}p_{m-2}\ldots p_1p_0 \xrightarrow{\pi_i} p_{m-1}p_{m-2}\ldots p_{m-i}p_0p_{m-i-1}\ldots p_2p_1 \tag{4}$$

This mapping corresponds to a 1-bit circular-right-shift among the $m-i$ least significant bits while keeping the i most significant bits unchanged.

However, stage permutation σ_i's are not fixed since each switch can be set to either parallel or cross.

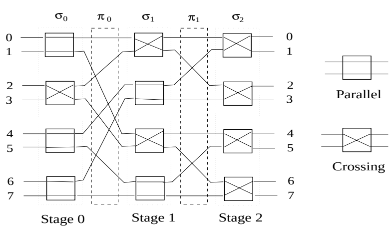

Figure 2: A routing example in an 8×8 baseline network.

Now let's look at an example shown in Figure 2. We have stage permutations σ_0, σ_1, and σ_2, which correspond to the permutations with only the second switch set to crossing, only the first and third switches set to crossing, and all four switches set to crossing, respectively, that is,

$$\sigma_0 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 1 & 3 & 2 & 4 & 5 & 6 & 7 \end{pmatrix},$$
$$\sigma_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 0 & 2 & 3 & 5 & 4 & 6 & 7 \end{pmatrix},$$
$$\sigma_2 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 0 & 3 & 2 & 5 & 4 & 7 & 6 \end{pmatrix}$$

and interstage permutations

$$\pi_0 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 4 & 1 & 5 & 2 & 6 & 3 & 7 \end{pmatrix}$$
$$\pi_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 1 & 3 & 4 & 6 & 5 & 7 \end{pmatrix}$$

For input 0, we can obtain the following transformation

$$0 \xrightarrow{\sigma_0} 0 \xrightarrow{\pi_0} 0 \xrightarrow{\sigma_1} 1 \xrightarrow{\pi_1} 2 \xrightarrow{\sigma_2} 3$$

that is, $0 \xrightarrow{\sigma_2\pi_1\sigma_1\pi_0\sigma_0} 3$.

After computing the transformation for every input, we can obtain the overall permutation for the switch settings in the network

$$\sigma_2\pi_1\sigma_1\pi_0\sigma_0 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 7 & 5 & 1 & 0 & 4 & 2 & 6 \end{pmatrix}$$

Notice that not every permutation is admissible to a multistage network like baseline, omega, banyan and their reverse network. Also not every admissible permutation of the network has a crosstalk-free decomposition, for which we will give a counterexample later. However, the following Lemma specifies the conditions of existence of a crosstalk-free decomposition from an admissible permutation.

Lemma 2 *Let an admissible permutation of a multistage network shown in (3) be denoted as $\rho_{2m-1}\rho_{2m-2}\ldots\rho_2\rho_1$. The admissible permutation can be decomposed into two crosstalk-free semi-permutations if and only if permutation ρ_i can be decomposed into two semi-permutations s_i and t_i for $1 \leq i \leq 2m-1$ such that*

$$\begin{aligned} InputSet(s_{2m-1}) &= OutputSet(s_{2m-2}) \\ InputSet(s_{2m-2}) &= OutputSet(s_{2m-3}) \\ &\vdots \\ InputSet(s_2) &= OutputSet(s_1). \end{aligned} \tag{5}$$

The two crosstalk-free semi-permutations obtained are $s_{2m-1}s_{2m-2}\ldots s_2s_1$ and $t_{2m-1}t_{2m-2}\ldots t_2t_1$, respectively.

Proof. Omitted (see [15] for detailed proof). ■

Needless to say, it is difficult to apply Lemma 2 to check the crosstalk-free property of a semi-permutation in practice. However, as a general guideline, it is useful in proving our further results concerning the crosstalk-free decompositions for a special set of admissible permutations for the class multistage networks we consider.

3 All-to-All Personalized Exchange in Multistage Networks

In this section, we first introduce an approach to realizing all-to-all personalized exchange in multistage networks without the constraint of crosstalk-free, i.e. in (ordinary) electronic multistage networks, then extend it to realizing all-to-all personalized exchange in optical multistage networks under the constraint of crosstalk-free.

Given n processors and an $n \times n$ multistage network, let processor i $(0 \leq i \leq n-1)$ be connected to input i and output i of the network. Processor i can send messages through input i of the network to other processors and receive messages from other processors through output i of the network.

3.1 Lower Bound for All-to-All Personalized Exchange

We have the following lemma concerning the lower bound on the maximum communication delay of all-to-all personalized exchange in a multistage network.

Lemma 3 *The maximum communication delay of all-to-all personalized exchange in an $n \times n$ network of $\log n$ stages is at least $\Omega(n + \log n)$.*

Proof. Omitted (see [15]). ■

Table I: All-to-all personalized exchange algorithm for a class of multistage networks

```
Algorithm ATAPE
begin
Step 1. for each processor j (0 ≤ j ≤ n − 1) do in parallel
    1.1 for each a_{i,j} (0 ≤ i ≤ n − 1) in the Latin square do
            prepare a personalized message from processor j
            to processor a_{i,j};
            insert the message into the message queue j;
Step 2. for each processor j (0 ≤ j ≤ n − 1) do in parallel
    2.1 for each message with destination address
            a_{i,j} (0 ≤ i ≤ n − 1) in the message queue j do
            send the message destined to a_{i,j} through
            input j of the network;
end;
```

3.2 All-to-All Personalized Exchange Algorithm Using a Latin Square

A *Latin square* [14] is defined as an $n \times n$ matrix

$$\begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{bmatrix}$$

in which the entries $a_{i,j}$'s are numbers in $\{0, 1, 2, \ldots, n-1\}$ and no two entries in a row (or a column) have the same value.

Now suppose that each row in the Latin square, $a_{i,0}$, $a_{i,1}$, $\ldots$, $a_{i,n-1}$, corresponds to a permutation

$$\begin{pmatrix} 0 & 1 & 2 & \cdots & n-1 \\ a_{i,0} & a_{i,1} & a_{i,2} & \cdots & a_{i,n-1} \end{pmatrix},$$

which is admissible to an $n \times n$ self-routing multistage network. We will show that by realizing the n admissible permutations corresponding to the Latin square, we can achieve all-to-all personalized exchange.

We also assume that every message has the same length so that the message transmission at each stage is synchronized. A higher-level description of the algorithm ATAPE is given in Table I.

Note that the network under consideration is a self-routing network, where each switch is set automatically by the routing tag contained in the message passing that switch. Since we are considering admissible permutations, at any time two messages entering from the two inputs of a switch can pass the switch simultaneously without any conflicts. In addition, once the previous n messages leave the switches of the current stage, the next n messages can enter the switches of this stage. Thus, the sequential steps of 2.1 are actually performed in a pipelined fashion, which achieves a form of parallelism. Therefore, the time complexities of Step 1 and Step 2 are $O(n)$ and $O(n + \log n)$, respectively. The total time delay for the all-to-all personalized exchange algorithm is $O(n + \log n)$, which matches the low bound of the communication delay for this type of network.

Now let's consider realizing all-to-all personalized exchange in optical multistage networks under the constraint of crosstalk-free. If each admissible permutation to the network can be decomposed to two crosstalk-free semi-permutations, the all-to-all personalized exchange in an optical multistage network can be achieved by realizing these $2n$ semi-permutations in the network. The total delay is only twice of the electronic counterpart. Since this process can still be performed in a pipelined fashion, the proposed all-to-all personalized exchange algorithm achieves optimal in communication delay for optical multistage networks as well.

The problems remaining unsolved are how to construct the Latin square the all-to-all personalized exchange algorithm is based on, and for optical multistage networks we also require the admissible permutations can be decomposed to crosstalk-free semi-permutations. These are the main focus of the rest of the paper.

3.3 The Special Set of Admissible Permutations Which Form a Latin Square

As we know, not all permutations are admissible to a baseline network. However, in the following, we give a simple way to choose a special set of permutations, which are admissible to a self-routing multistage network and can form a Latin square.

First, we introduce a set of basic permutations used for constructing a Latin Square. For an $n \times n$ mapping, where $n = 2^m$, we define m basic permutations ϕ_i $(1 \le i \le m)$ as follows. Let the binary representation of a number $a \in \{0, 1, \ldots, n-1\}$ be $p_{m-1}p_{m-2}\ldots p_1p_0$. Then

$$p_{m-1}p_{m-2}\ldots p_ip_{i-1}p_{i-2}\ldots p_1p_0 \xrightarrow{\phi_i} p_{m-1}p_{m-2}\ldots p_i\bar{p}_{i-1}p_{i-2}\ldots p_1p_0 \quad (6)$$

The permutation ϕ_i is actually the operation flipping the i^{th} bit of a binary number.

In describing the construction of all-to-all personalized exchange Latin square in the following, we are especially interested in ϕ_1, and other ϕ_i's are used in the proof of Theorem 2 in [9]. Clearly, the stage permutation of a stage in the network is ϕ_1 if and only if the switches in this stage are all set to crossing, and the stage permutation of a stage in the network is I if and only if the switches in this stage are all set to parallel. In [9] the following result was given.

Theorem 2 *Let the stage permutation of each stage in a baseline network take either ϕ_1 or I, The admissible permutations corresponding to all possible such switch settings form a Latin square.*

In Figure 3, we list all possible such switch settings in an 8×8 baseline network, and the corresponding Latin square is L in (7).

$$L = \begin{bmatrix} 0 & 4 & 2 & 6 & 1 & 5 & 3 & 7 \\ 1 & 5 & 3 & 7 & 0 & 4 & 2 & 6 \\ 3 & 7 & 1 & 5 & 2 & 6 & 0 & 4 \\ 2 & 6 & 0 & 4 & 3 & 7 & 1 & 5 \\ 6 & 2 & 4 & 0 & 7 & 3 & 5 & 1 \\ 7 & 3 & 5 & 1 & 6 & 2 & 4 & 0 \\ 5 & 1 & 7 & 3 & 4 & 0 & 6 & 2 \\ 4 & 0 & 6 & 2 & 5 & 1 & 7 & 3 \end{bmatrix} \quad (7)$$

We can draw the same conclusions as Theorem 2 for other self-routing multistage networks such as omega and banyan networks. For details see [9].

4 Decomposition of the Special Admissible Permutations

Notice that not every admissible permutation to the network can be decomposed into semi-permutations which can be realized in the

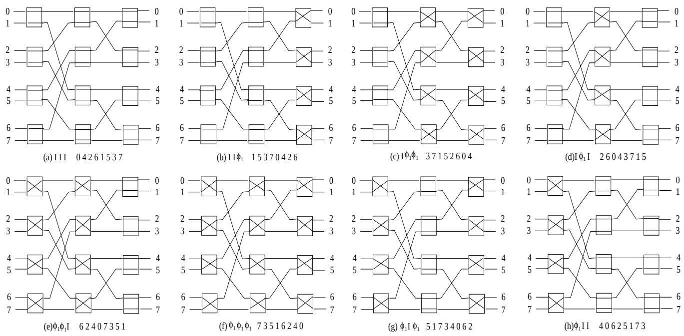

Figure 3: All possible switch settings, in which each stage is set to either ϕ_1 or I, in an 8×8 baseline network and the corresponding overall permutations realized.

network crosstalk-free. For example, the permutation

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

is admissible to an 8×8 baseline network. However, this permutation does not have a crosstalk-free decomposition. In fact, applying the decomposition algorithm to this permutation, we can obtain a unique decomposition of semi-permutations

$$\begin{pmatrix} 0 & 2 & 5 & 7 \\ 2 & 0 & 5 & 7 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 3 & 4 & 6 \\ 4 & 6 & 1 & 3 \end{pmatrix}$$

However, it can be easily verified that none of them can pass the network crosstalk-free.

Fortunately, we can prove that any permutation in the special set of admissible permutations which form a Latin square described in the last section does have a crosstalk-free decomposition of semi-permutations. Thus, all-to-all personalized exchange can be realized in $2n$ passes in an $n \times n$ self-routing optical multistage network crosstalk-free.

4.1 Self-Mapping and Opposite-Mapping Semi-Permutations

Given a semi-permutation s as in (2), if the input set $\{x_0, x_1, \ldots, x_{\frac{n}{2}-1}\}$ and the output set $\{y_0, y_1, \ldots, y_{\frac{n}{2}-1}\}$ are equal, the semi-permutation is said *self-mapping*; if the intersection of the two sets are empty (in other words, the union of the two sets is the whole set $\{0, 1, \ldots, n-1\}$), the semi-permutation is said *opposite-mapping*. The following are examples of self-mapping and opposite-mapping semi-permutations in an 8×8 network:

$$\text{self-mapping: } \begin{pmatrix} 0 & 3 & 5 & 6 \\ 0 & 5 & 6 & 3 \end{pmatrix}$$

$$\text{opposite-mapping: } \begin{pmatrix} 0 & 3 & 5 & 6 \\ 1 & 4 & 7 & 2 \end{pmatrix}$$

As can be seen later, the special set of admissible permutations which form the all-to-all personalized exchange Latin square in the last section can be decomposed to these two types of semi-permutations. We now discuss some properties of self-mapping and opposite-mapping semi-permutations.

Lemma 4 *If a semi-permutation is self-mapping, so is its twin semi-permutation. If a semi-permutation is opposite-mapping, so is its twin semi-permutation.*

Proof. Let a semi-permutation and its twin semi-permutation in an $n \times n$ network be s and t respectively. If s is self-mapping, we must have that $InputSet(s) = OutputSet(s)$, which yields $InputSet(t) = OutputSet(t) = \{0, 1, \ldots, n-1\} - InputSet(s)$. Thus, t is self-mapping. If s is opposite-mapping, we must have that $InputSet(s) = \{0, 1, \ldots, n-1\} - OutputSet(s)$, which is equivalent to $\{0, 1, \ldots, n-1\} - InputSet(t) = OutputSet(t)$. Hence, t is opposite-mapping. ■

Corollary 1 *Suppose semi-permutations s_1 and s_2 can form a composition $s_1 s_2$. If both s_1 and s_2 are self-mapping or both of them are opposite-mapping, then $s_1 s_2$ is self-mapping. If one of them is self-mapping and the other is opposite-mapping, then $s_1 s_2$ is opposite-mapping.*

For the special permutations I and ϕ_1, we have the following remarks.

Remark 1 *Any semi-permutation decomposed from I is self-mapping, and any semi-permutation decomposed from ϕ_1 is opposite-mapping.*

Remark 2 *Given any set $\mathcal{C}$ such that $|\mathcal{C}| = \frac{n}{2}$ and $\{\lfloor \frac{x}{2} \rfloor | x \in \mathcal{C}\} = \{0, 1, \ldots, \frac{n}{2}\}$, let $\mathcal{D} = \{0, 1, \ldots, n-1\} - \mathcal{C}$. Then the permutation I can be decomposed to two self-mapping semi-permutations I_s and I_t such that $InputSet(I_s) = \mathcal{C} = OutputSet(I_s)$ and $InputSet(I_t) = \mathcal{D} = OutputSet(I_t)$; and the permutation ϕ_1 can be decomposed to two opposite-mapping semi-permutations $\phi_{1,s}$ and $\phi_{1,t}$ such that $InputSet(\phi_{1,s}) = \mathcal{C} = OutputSet(\phi_{1,t})$ and $OutputSet(\phi_{1,s}) = \mathcal{D} = InputSet(\phi_{1,t})$.*

Remark 3 *For any permutation p which is decomposed to two semi-permutations s and t, I can be decomposed to I_s and I_t such that both sI_s and tI_t are compositions of semi-permutations (in fact, $sI_s = s$ and $tI_t = t$, and we can see here that I and its decomposition can always be ignored); and ϕ_1 can be decomposed to $\phi_{1,s}$ and $\phi_{1,t}$ such that both $s\phi_{1,s}$ and $t\phi_{1,t}$ are compositions of semi-permutations, which implies $p\phi_1$ is decomposed to $s\phi_{1,s}$ and $t\phi_{1,t}$, in particular, if s and t are self-mapping (or opposite-mapping) then $s\phi_{1,s}$ and $t\phi_{1,t}$ are opposite-mapping (or self-mapping).*

4.2 Decomposition of Interstage Permutations

In this subsection, we look at how to decompose interstage permutations in a multistage network to self-mapping semi-permutations. Let $Bits(x)$ be a function of integer x that returns the number of 1's in the binary representation of x. For an $n \times n$ permutation where $n = 2^m$, we define the following sets

$$\begin{aligned} \mathcal{A} &= \{x | x \in \{0, 1, \ldots, n-1\} \text{ and } Bits(x) \text{ is even }\}, \\ \mathcal{B} &= \{x | x \in \{0, 1, \ldots, n-1\} \text{ and } Bits(x) \text{ is odd }\}. \end{aligned} \quad (8)$$

The following Lemma reveals some properties of the sets $\mathcal{A}$ and $\mathcal{B}$.

Lemma 5 *For the sets defined in (8), we have $\mathcal{A} \bigcap \mathcal{B}$ is empty, $|\mathcal{A}| = |\mathcal{B}| = \frac{n}{2}$, and*

$$\left\{ \left\lfloor \frac{x}{2} \right\rfloor | x \in \mathcal{A} \right\} = \left\{ 0, 1, \ldots, \frac{n}{2} \right\} = \left\{ \left\lfloor \frac{x}{2} \right\rfloor | x \in \mathcal{B} \right\} \quad (9)$$

Proof. Omitted (see [15] for detailed proof). ■

Theorem 3 *Any interstage permutation π_i, $0 \le i \le \log n - 2$, in an $n \times n$ baseline network can be decomposed to two self-mapping semi-permutations with the input sets equal to $\mathcal{A}$ and $\mathcal{B}$ respectively.*

Proof. As shown in (4), the mapping π_i only shifts some bits and does not change the total number of 1's. Thus, for any $x \in \mathcal{A}$ we have $\pi_i(x) \in \mathcal{A}$, that is, π_i is an one-to-one self-mapping of $\mathcal{A}$. Meanwhile, let $\mathcal{A} = \{x_0, x_1, \ldots, x_{\frac{n}{2}}\}$ with $x_0 < x_1 < \cdots < x_{\frac{n}{2}}$, we have $\{\pi_i(x_0), \pi_i(x_1), \ldots, \pi_i(x_{\frac{n}{2}})\} = \mathcal{A}$. Also from (9),

$$s_i = \begin{pmatrix} x_0 & x_1 & \ldots & x_{\frac{n}{2}-1} \\ \pi_i(x_0) & \pi_i(x_1) & \ldots & \pi_i(x_{\frac{n}{2}-1}) \end{pmatrix}$$

is a semi-permutation with $InputSet(s_i) = OutputSet(s_i) = \mathcal{A}$. Similarly,

$$t_i = \begin{pmatrix} y_0 & y_1 & \ldots & y_{\frac{n}{2}-1} \\ \pi_i(y_0) & \pi_i(y_1) & \ldots & \pi_i(y_{\frac{n}{2}-1}) \end{pmatrix}$$

is a semi-permutation with $InputSet(t_i) = OutputSet(t_i) = \mathcal{B}$, where $\mathcal{B} = \{y_0, y_1, \ldots, y_{\frac{n}{2}}\}$ with $y_0 < y_1 < \cdots < y_{\frac{n}{2}}$. Therefore, the permutation π_i is decomposed to self-mapping semi-permutations s_i and t_i with the input sets (also the output sets) being $\mathcal{A}$ and $\mathcal{B}$ respectively. ■

For example, when $n = 8$, we have that

$$\pi_0 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 4 & 1 & 5 & 2 & 6 & 3 & 7 \end{pmatrix} \text{ decomposed to}$$

$$\begin{pmatrix} 0 & 3 & 5 & 6 \\ 0 & 5 & 6 & 3 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 2 & 4 & 7 \\ 4 & 1 & 2 & 7 \end{pmatrix}.$$

We have the same conclusion for the class of multistage networks.

Corollary 2 *Any interstage permutation in an $n \times n$ network, which could be a baseline, omega, banyan, or their reverse networks, can be decomposed to two self-mapping semi-permutations with the input sets being $\mathcal{A}$ and $\mathcal{B}$ respectively.*

Finally, we have the following remark concerning the possibility of decomposition of interstage permutation to opposite-mapping semi-permutations.

Remark 4 *There does not exist any decomposition of the interstage permutation to opposite-mapping semi-permutations for the class of multistage networks. The reason is that any such interstage permutation maps 0 to 0 itself.*

4.3 Crosstalk-Free Decomposition of the Special Admissible Permutations

In this subsection, we give a crosstalk-free decomposition of the special admissible permutations which form the all-to-all personalized exchange Latin square described in the last section.

Theorem 4 *Let the stage permutation σ_i of each stage in a baseline network take either ϕ_1 or I. Then any admissible permutation shown in (3) has a crosstalk-free decomposition of semi-permutations.*

Proof. First we consider the case that $\sigma_i = I$ for any $0 \le i \le m-1$. Notice that by the first part of Remark 3, we only need to consider the permutation $\pi_{m-2}\pi_{m-3}\ldots\pi_1\pi_0$. By Theorem 3 each π_i can be decomposed to two self-mapping semi-permutations s_i and t_i with $InputSet(s_i) = OutputSet(s_i) = \mathcal{A}$ and $InputSet(t_i) = OutputSet(t_i) = \mathcal{B}$. By Lemma 2, $\pi_{m-2}\pi_{m-3}\ldots\pi_1\pi_0$ can be decomposed to two crosstalk-free semi-permutations $s_{m-2}s_{m-3}\ldots s_1 s_0$ and $t_{m-2}t_{m-3}\ldots t_1 t_0$, which are self-mapping.

For the general case, we only construct one crosstalk-free semi-permutation, and its twin can be obtained symmetrically. First take a semi-permutation $\sigma_{0,s}$ decomposed from σ_0 with the input set $\mathcal{A}$ regardless $\sigma_0 = I$ or ϕ_1. Then take a semi-permutation $\pi_{0,s}$ decomposed from π_0 with the input set $\mathcal{A}$ or $\mathcal{B}$ depending on $\sigma_0 = I$ or ϕ_1. In step i, take a semi-permutation $\sigma_{i,s}$ decomposed from σ_i with the input set equal to $OutputSet(\pi_{i-1,s}\sigma_{i-1,s}\ldots\pi_{0,s}\sigma_{0,s})$, which is either $\mathcal{A}$ or $\mathcal{B}$, and then take a semi-permutation $\pi_{i,s}$ decomposed from π_i with the input set equal to $OutputSet(\sigma_{i,s}\pi_{i-1,s}\sigma_{i-1,s}\ldots\pi_{0,s}\sigma_{0,s})$, which is either $\mathcal{A}$ or $\mathcal{B}$, and so on. Finally, we obtain $\sigma_{m-1,s}\pi_{m-2,s}\sigma_{m-2,s}\ldots\pi_{0,s}\sigma_{0,s}$ which is a decomposed crosstalk-free semi-permutation, and is self-mapping (or opposite-mapping) if the number of σ_i's which are equal to ϕ_1 is even (or odd). ■

For example, in an 8×8 baseline network, the admissible permutation with switch setting I at each stage, which corresponds to the first row of Latin square (7),

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 4 & 2 & 6 & 1 & 5 & 3 & 7 \end{pmatrix} = \pi_1\pi_0$$

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

is decomposed to

$$\begin{pmatrix} 0 & 3 & 5 & 6 \\ 0 & 3 & 6 & 5 \end{pmatrix} \begin{pmatrix} 0 & 3 & 5 & 6 \\ 0 & 5 & 6 & 3 \end{pmatrix} = \begin{pmatrix} 0 & 3 & 5 & 6 \\ 0 & 6 & 5 & 3 \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 2 & 4 & 7 \\ 2 & 1 & 4 & 7 \end{pmatrix} \begin{pmatrix} 1 & 2 & 4 & 7 \\ 4 & 1 & 2 & 7 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 & 7 \\ 4 & 2 & 1 & 7 \end{pmatrix}$$

and the admissible permutation with switch settings I, ϕ_1, I, which corresponds to the fourth row of Latin square (7),

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 6 & 0 & 4 & 3 & 7 & 1 & 5 \end{pmatrix} = \pi_1\phi_1\pi_0$$

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

is decomposed to

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

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

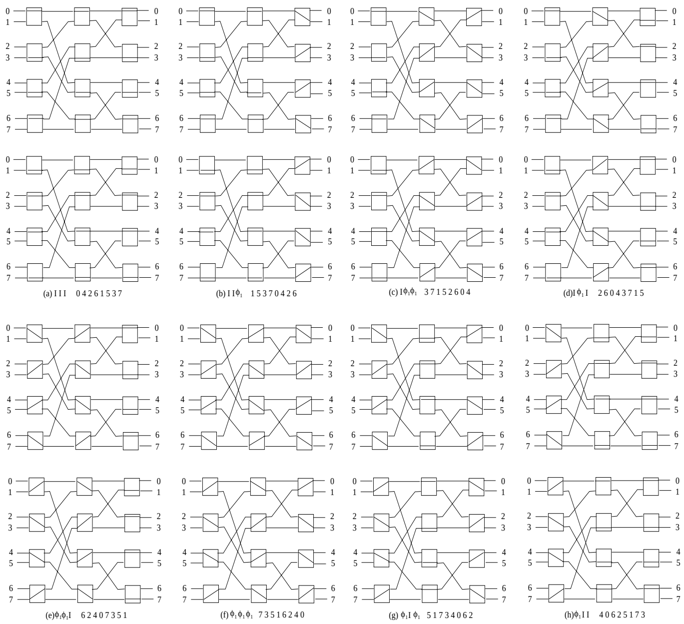

Figure 4: All crosstalk-free routings corresponding to the semi-permutations decomposed from the permutations which form a Latin square, in an 8×8 baseline network.

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

In Figure 4, we list all crosstalk-free routings corresponding to the semi-permutations decomposed from permutations which form a Latin square in (7), in an 8×8 baseline network. Clearly, we can have $2n$ passes of crosstalk-free routings to realize all-to-all personalized exchange in an $n \times n$ baseline network and other similar networks.

4.4 An Efficient Decomposition Algorithm

We have already given the crosstalk-free decomposition for a special set of permutations which correspond to the all-to-all personalized exchange Latin square for a class of multistage network. Since these networks are self-routing and have a unique path between each pair of input and output, we actually only need to know the overall crosstalk-free semi-permutations, without being concerned too much about the details in the internal stages. This is because that the results established in the previous sections already guarantee the correctness of the routing. Furthermore, for any of such permutations, their semi-permutations have input set and output set equal to either set $\mathcal{A}$ or set $\mathcal{B}$. Therefore, after we obtain $\mathcal{A}$ and $\mathcal{B}$ as defined in (8), we can easily generate two semi-permutations for each permutation corresponding to one row in the all-to-all personalized exchange Latin square, by masking the permutation by $\mathcal{A}$ or $\mathcal{B}$ respectively.

The problem remains unsolved is how to generate sets $\mathcal{A}$ and $\mathcal{B}$ for $n \times n$ networks. A naive approach is to calculate them directly by definition (8), which takes at least $\Omega(nm) = \Omega(n \log n)$ time because we need to count each bit of a number, i.e. calculate $Bits(\cdot)$. In the following, we give an $O(n)$ time algorithm to generate sets $\mathcal{A}$ and $\mathcal{B}$.

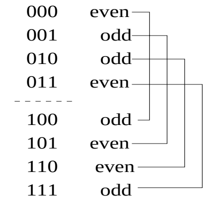

Figure 5: The odd-even list of consecutive numbers $0, 1, \ldots, 2^{k-1}$ for $k = 3$.

Table II: Decomposition algorithm

```
Algorithm GenSemiSets(Set A, Set B, int n = 2^m)
{
A = NULL;
B = NULL;
List taglist = GenTag(n);
for (i = 0; i < n; i++) {
    if (taglist[i] == 0)
        A = A + {i};
    else /* taglist[i] == 1 */
        B = B + {i};
end for;
}

Function List GenTag(int n = 2^m)
{
    if (n == 2)
        return List {0, 1};
    List tmplist1 = GenTag(n/2);
    List tmplist2 = Negate(tmplist1);
    return List Concat(tmplist1, tmplist2);
}
```

For each number i in $\{0, 1, \ldots, n-1\}$ where $n = 2^m$, we assign a tag $taglist(i)$ to it such that $taglist(i) = 0$ if $Bits(i)$ is even, and $taglist(i) = 1$ if $Bits(i)$ is odd. Clearly, any number with its tag being 0 will be put in the set $\mathcal{A}$, otherwise be put in $\mathcal{B}$. However we need to avoid directly computing $Bits(\cdot)$. To see how to efficiently assign a tag for each number, we observe the following fact. For any $k \geq 1$, for consecutive numbers $0, 1, \ldots, 2^k - 1$, divide them from the middle into two consecutive parts, the k^{th} bits of numbers in the first part are all 0's, and those in the second part are all 1's. Thus the odd-even list of the second part is negated from that of the first part. Figure 5 shows the case for $k = 3$. Now we have the algorithm as shown in Table II.

The algorithm $GenSemiSets(\cdot)$ calls a recursive function $GenTag(\cdot)$ to generate a tag list of 0's and 1's, then builds the sets $\mathcal{A}$ and $\mathcal{B}$ based on the tag list. The function $Negate(\cdot)$ used in $GenTag(\cdot)$ returns a new tag list with each tag negated from corresponding tag in the input tag list; while the function $Concat(\cdot)$ returns a concatenation of two tag lists. For example, for $n = 16$, the resulting tag list is $\{0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0\}$ and thus $\mathcal{A} = \{0, 3, 5, 6, 9, 10, 12, 15\}$ and $\mathcal{B} = \{1, 2, 4, 7, 8, 11, 13, 14\}$ which correspond to the 0's and 1's in the tag list respectively. To analyze the complexity of the algorithm, let $T(n)$ denote the time for $GenTag(n)$. we have $T(n) = T(\frac{n}{2}) + n$. Thus $T(n) = O(n)$, and the time for $GenSemiSets(\cdot)$ is $O(n)$ as well.

4.5 Uniqueness of the Decomposition

In this subsection, we discuss the uniqueness of crosstalk-free decomposition of the special admissible permutations for a baseline network, and the same conclusion can be drawn for other multistage networks. We start with the decomposition of interstage permutation π_i $(0 \leq i \leq m-2)$. Notice that the self-mapping decomposition of π_i for $i > 0$ is not unique. For example, in an 8×8 network, the permutation

$$\pi_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 1 & 3 & 4 & 6 & 5 & 7 \end{pmatrix}$$

has two self-mapping decompositions, one is that shown right after Theorem 3

$$\begin{pmatrix} 0 & 3 & 5 & 6 \\ 0 & 3 & 6 & 5 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 2 & 4 & 7 \\ 2 & 1 & 4 & 7 \end{pmatrix}, \text{ and}$$

$$\text{another is } \begin{pmatrix} 0 & 3 & 4 & 7 \\ 0 & 3 & 4 & 7 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 2 & 5 & 6 \\ 2 & 1 & 6 & 5 \end{pmatrix}.$$

However, for the permutation π_0, we have the following result.

Theorem 5 *The permutation π_0 has a unique self-mapping decomposition.*

Proof. Omitted (see [15] for detailed proof). ■

The impact of the uniqueness of π_0's self-mapping decomposition on the special admissible permutations is shown in the following theorem.

Theorem 6 *The special admissible permutation described in the last section has a unique crosstalk-free decomposition such that each of semi-permutations decomposed from stage permutations and interstage permutations is self-mapping or opposite-mapping.*

Proof. First, by Remark 4 we know that any interstage permutation does not have opposite-mapping decomposition. Then by Theorem 5, interstage permutation π_0 has a unique self-mapping decomposition. Notice that in the proof of Theorem 4, the resulting crosstalk-free semi-permutation is a composition of semi-permutations decomposed from those stage permutations and interstage permutations including π_0. Therefore, the special admissible permutation has a unique crosstalk-free decomposition. ■

5 Conclusions

We have studied the issue of realizing all-to-all personalized exchange in optical multistage networks, which are a class of unique-path, self-routing networks. The approach to realizing all-to-all personalized exchange we developed in this paper is first to transform it to some special admissible permutations of the network which form a Latin square, and then to decompose each of such a permutation to semi-permutations, each of which can be realized crosstalk-free in a single pass in the network. Thus $2n$ crosstalk-free semi-permutations are used to realize all-to-all personalized exchange in an $n \times n$ network. We have established the general results for crosstalk-free decomposition, and also introduced self-mapping and opposite-mapping semi-permutations to simplify the crosstalk-free decomposition of a special set of permutations. By taking advantage of fast self-routing switch setting and pipelined message transmission in multistage networks, the proposed all-to-all personalized exchange algorithm has $O(n)$ time complexity which matches the lower bound of the communication delay in this type of network. Finally, notice that the Latin square and the input sets of self-mapping semi-permutation can be viewed as system parameters. Therefore, the algorithms of Latin square construction and input sets ($\mathcal{A}$ and $\mathcal{B}$) generation are off-line, and run only once at the time the network is built.

References

[1] D. Gannon and J.V. Rosendale, "On the Impact of Communication Complexity in the Design of Parallel Numerical Algorithms," *IEEE Trans. Computer,* vol. 33, pp. 1180-1194, 1984.

[2] S.L. Johnsson, "Communication Efficient Basic Linear Algebra Computations on Hypercube Architectures," *Journal of Parallel Distributed Computing,* vol. 4, pp. 133-172, 1987.

[3] D.S. Scott, "Efficient All-to-All Communication Patterns in Hypercube and Mesh Topologies," *Proc. of 6th Distributed Memory Computing Conference,* pp. 398-403, 1991.

[4] R. Thakur and A. Choudhary, "All-to-All Communication on Meshes with Wormhole Routing," *Proc. of IPPS'94*, pp. 561-565, April 1994.

[5] Y.-C. Tseng and S. Gupta, "All-to-All Personalized Communication in a Wormhole-Routed Torus," *IEEE Trans. Parallel and Distributed Systems*, vol. 7, no. 5, pp. 498-505, 1996.

[6] Y.-C. Tseng, T.-H. Lin, S. Gupta, and D.K. Panda, "Bandwidth-Optimal Complete Exchange on Wormhole Routed 2D/3D Torus Networks: A Diagonal-Propagation Approach," *IEEE Trans. Parallel and Distributed Systems,* vol. 8, no. 4, pp. 380-396, 1997.

[7] Y.J. Suh and S. Yalamanchili, "All-to-All Communication with Minimum Start-up Costs in 2D/3D Tori and Meshes," *IEEE Trans. Parallel and Distributed Systems,* vol. 9, no. 5, pp. 442-458, 1998.

[8] Y.J. Suh and K.G. Shin, "Efficient All-to-All Personalized Exchange in Multidimensional Torus Networks," *Proc. of ICPP'98,* pp. 468-475, August 1998.

[9] Y. Yang and J. Wang, "Optimal All-to-All Personalized Exchange in Self-Routable Multistage Networks," to appear in *IEEE Trans. Parallel and Distributed Systems*.

[10] G.S. Almasi and A. Gottlieb, *Highly Parallel Computing,* The Benjamin/Cummings, 1994.

[11] Y. Pan, C. Qiao and Y. Yang, "Optical Multistage Interconnection Networks: New Challenges and Approaches," *IEEE Communications*, vol. 37, no. 2, pp. 50-56, Feb. 1999.

[12] Y. Yang, J. Wang and Y. Pan, "Permutation Capability of Optical Multistage Interconnection Networks," *Journal of Parallel and Distributed Computing*, vol. 60, no. 1, pp. 72-91, Jan. 2000.

[13] C. Qiao, R. Melhem, D. Chiarulli and S. Levitan, "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.

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

[15] Y. Yang and J. Wang, "Optimal all-to-all personalized exchange in a class of optical multistage networks," *ECE Dept. Technical Report, SUNY at Stony Brook*, 2000.