Parallel Routing and Wavelength Assignment for Optical Multistage Interconnection Networks

Enyue Lu and S. Q. Zheng
Department of Computer Science
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, TX 75083-0688, USA
{enyue, sizheng}@utdallas.edu

Abstract

Multistage interconnection networks (MINs) are among the most efficient switching architectures in terms of the number of switching elements (SEs) used. For optical MINs (OMINs), two I/O connections with neighboring wavelengths cannot share a common SE due to crosstalk. In this paper, we focus on the wavelength dilation approach, in which the I/O connections sharing a common SE will be assigned different wavelengths with enough wavelength spacing. We first study the permutation capacity of OMINs, then propose fast parallel routing and wavelength assignment algorithms for OMINs. By applying our permutation decomposition and graph coloring techniques, the proposed algorithms can route any permutation without crosstalk in wavelength-rearrangeable space-strict-sense Banyan networks and wavelength-rearrangeable space-rearrangeable Benes networks in polylogarithmic time using a linear number of processors.

1. Introduction

The explosive growth of Internet is driving an increased demand for transmission rate and faster switching technologies. Optical communications with photonic switching promise to meet high bandwidth, low error probability, and large transmission capacity. The networks using optical transmission and maintaining optical data paths can be used to remove the expensive optic-electro and electro-optic conversions. The electronic parallel processing for controlling such networks are capable, in principle, of meeting future high data rate requirements.

Nonblocking networks have been favored in switching systems since they can set up any one-to-one I/O mapping. For a nonblocking space-division-multiplexing network, it can be *strictly nonblocking (SNB)*, or *rearrangeable nonblocking (RNB)* [2, 8]. In SNB networks, a connection can be established from any idle input to any idle output without disturbing existing connections while in RNB networks the connection can be established if the rearrangement of existing connections is allowed. With wavelength-division-multiplexing (WDM) technology, the concept of SNB and RNB in space-division switching can be extended to wavelength-division switching. Depending on whether wavelengths can be reassigned, this extension results in four combinations: wavelength-rearrangeable space-rearrangeable *(WRSR)*, wavelength-rearrangeable space-strict-sense *(WRSS)*, wavelength-strict-sense space-rearrangeable *(WSSR)*, and wavelength-strict-sense space-strict-sense *(WSSS)*. It has been shown that using both wavelength and space multiplexing techniques in a fully dynamic manner, networks can achieve higher bandwidth and higher connectivity [19].

To build a large IP router with capacity of 1 Tb/s and beyond, optical multistage interconnection networks (OMINs) will be used. An OMIN usually comprises a number of 2×2 switching elements (SEs) grouped into several stages interconnected by a set of optical links (e.g. [6, 7, 22]). One of the problems with such OMINs is *crosstalk* at optical SEs, i.e., if more than one signal with the neighboring wavelengths share the same SE, they interfere with each other[1]. In electronic switching networks, there is only link conflict, i.e., two active inputs intend to be connected with the same output. The crosstalk in photonic switching networks adds a new type of blocking, called *wavelength conflict*.

In order to minimize wavelength conflicts in photonic switching networks, three approaches, *space dilation*, *time*

1 In this paper, we only consider non-filterable first-order SE crosstalk[12, 13], and different wavelengths are referred to the wavelengths with enough wavelength spacing so that no crosstalk will be generated when such wavelengths passing through the same SE

Proceedings of the 2004 International Conference on Parallel Processing (ICPP'04)

dilation and *wavelength dilation*, have been proposed. In space and time dilations, crosstalk can be avoided by ensuring at most one connection passing through an SE. More specifically, in space dilation crosstalk can be avoided by increasing the number of SEs in a switching network (e.g. [14, 23, 24]), while in time dilation a set of conflicting connections is partitioned into subsets so that the connections in each subset can be established simultaneously without conflicts (e.g. [15, 17, 21, 26]). In wavelength dilation, the crosstalk between two signals passing through the same SE is suppressed by routing to ensure the wavelengths to be far apart (e.g. [5, 20, 25]), or by using wavelength converters (e.g. [18]). Since the connections with neighboring wavelengths do not share any SE, the wavelength dilation approach is also useful for establishing a set of connections that would normally cause link conflicts in blocking space-division-multiplexing OMINs such as Banyan networks.

In this paper, we focus on the wavelength dilation approach, and consider the problem of quickly configuring an OMIN and assigning each connection a wavelength for realizing a permutation without crosstalk. In wavelength dilation, if there are wavelength converters available, we can convert the input signals with the neighboring wavelengths entering into the same SE to different ones. Thus, two wavelengths are necessary plus the costs of the wavelength converters. The use of wavelength converters will increase hardware cost and configuration time. If there is no wavelength converter available, i.e. each connection will be assigned the same wavelength, then we need to find a wavelength assignment for connections plus a setting of SEs so that there is no crosstalk in OMINs.

Through this paper, we assume that no wavelength converter is available in OMINs and assure the wavelengths in the same SE to be different by routing. The switch model used in this paper follows [16, 20]. The OMINs under such switch model can be built up using 2×2 multi-wavelength SEs, in which each input/output is capable of receiving/transmitting optical signals of a set of wavelengths and each wavelength is switched independently in SEs [20]. Such a multi-wavelength SE has an independently controllable state, straight or cross as shown in Fig. 1 (a), for each wavelength. Figure 1 (b) shows a signal transmission in a multi-wavelength SE, where the connections for the wavelength λ_2 in the upper input and the wavelength λ'_2 in the lower input are in cross state and all other connections are in straight state.

If an SE can only receive/transmit one wavelength for each input/output, it is called a *basic* SE. The OMINs considered in this paper are WRSS Banyan networks and WRSR Benes networks, where the WRSR Benes networks only contain basic SEs. For an I/O permutation, if there is a setting of SEs to realize the permutation and a wavelength assignment of connections so that no two connections with the same wavelength share any SE or link, we called this setting and wavelength assignment a *crosstalk-free configuration* of the OMIN for the permutation. An algorithm that can find a crosstalk-free configuration for any permutation of an OMIN is called a *crosstalk-free routing and wavelength assignment* algorithm for the OMIN. In this paper, by applying graph edge and vertex coloring techniques, we present crosstalk-free routing and wavelength assignment algorithms that can route any permutation without crosstalk in $O(\log^2 N)$ time for a WRSS Banyan network using at most $2^{\lfloor \frac{\log N+1}{2} \rfloor}$ wavelengths and in $O(\log^3 N)$ time for a WRSR Benes network using at most $2\log N$ wavelengths, on a completely connected multiprocessor system of N processing elements (PEs). We also show that both routing and wavelength assignment algorithms can be implemented on a hypercube of $N/2$ PEs in $O(\log^4 N)$ time.

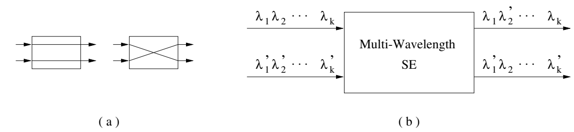

Figure 1. A 2×2 multi-wavelength SE. (a) Two states. (b) Signal transmission.

2. Definitions and Notations

Let $I = \{I_0, I_1, \cdots, I_{N-1}\}$ and $O = \{O_0, O_1, \cdots, O_{N-1}\}$ be the sets of inputs and outputs, respectively, of an $N \times N$ OMIN. Let $\pi : I \longmapsto O$ be a one-to-one I/O *mapping* that indicates connection requests from inputs to outputs. I_i and O_j are *active* if and only if there is a connection request from I_i to O_j, and in this case, $\pi(i) = j$ and $\pi^{-1}(j) = i$. The connection from input i to output $\pi(i)$ is denoted by i since it is a one-to-one mapping.

A one-to-one I/O mapping involving $K (\leq N)$ active inputs is called a *partial permutation*. A partial permutation with $K = N$ active inputs is also called a *permutation*. We are interested in a type of partial permutations that can be simultaneously connected through OMIN without crosstalk. Such partial permutations are called *crosstalk-free (CF) partial permutations*.

A special type of partial permutation, named *semi-permutation*, which ensures only one active input in every SE of the first and last stages of an OMIN at the same time, has the maximum potential to be simultaneously realized in OMIN without crosstalk. It was shown that any permutation can be decomposed into two semi-permutations and each semi-permutation can be routed in one pass in an optical Benes network without crosstalk [26]. In [10], we presented a parallel permutation decomposition algorithm to decompose a partial permutation into two partial semi-permutations and proved the following lemma[2].

Lemma 1 *For any partial permutation with $K(\leq N)$ active inputs, two partial semi-permutations can be computed in $O(\log K)$ time on a completely connected multiprocessor system of N PEs.*

The parallel decomposition algorithm of [10] is equivalent to an algorithm that finds a 2-edge coloring of a bipartite graph G with $\Delta(G) \leq 2$, where $\Delta(G)$ is the *degree* of G, the maximum number of edges incident at a vertex.

3. Parallel Routing and Wavelength Assignment in WRSS Banyan Networks

3.1. Banyan-type Networks

A class of multistage self-routing networks, *Banyan-type* networks, has received considerable attention. A network belonging to this class satisfies the following three properties:

i. It has $N = 2^n$ inputs, $N = 2^n$ outputs, n-stages and $N/2$ SEs in each stage.

ii. There is a unique path between each input and each output.

iii. Let u and v be two SEs in stage i, and let $S_j(u)$ and $S_j(v)$ be two sets of SEs to which u and v can reach in stage j, $0 < i + 1 = j \leq n$. Then $S_j(u) \cap S_j(v) = \emptyset$ or $S_j(u) = S_j(v)$ for any u and v.

Because of the above properties (short connection diameter, unique connection path, uniform modularity, etc.), Banyan-type networks are very attractive for constructing switching networks. Several well-known networks, such as *Banyan*, *Butterfly*, *Omega*, and *Baseline*, belong to this class. It has been shown that these networks are topologically equivalent [1]. In this paper, we use Baseline network as the representative of Banyan-type networks.

An $N \times N$ Baseline network, denoted by $BL(N)$, is constructed recursively. A $BL(2)$ is a 2×2 SE. A $BL(N)$ consists of a switching stage of $N/2$ SEs, and a shuffle connection, followed by a stack of two $BL(N/2)$'s. Thus a $BL(N)$ has $\log N$ stages labeled by $0, \cdots, n-1$ from left to right[3], and each stage has $N/2$ SEs labeled by $0, \cdots, N/2-1$ from top to bottom. Every SE has two inputs/outputs, each named *upper input/output* or *lower input/output* according to its relative position. The N links interconnecting two adjacent stages i and $i+1$ are called *output links* of stage i and *input links* of stage $i+1$ and labeled by $0, \cdots, N-1$ from top to bottom. The input (resp. output) links in the first (resp. last) stage of $BL(N)$ are connected with N inputs (resp. outputs) of $BL(N)$. To facilitate our discussions, the labels of stages, links and SEs are represented by binary numbers. Let $a_l a_{l-1} \cdots a_1 a_0$ be the binary representation of a. We use $\bar{a}$ to denote the integer that has the binary representation $a_l a_{l-1} \cdots a_1 (1 - a_0)$. Fig. 2 shows a $BL(16)$.

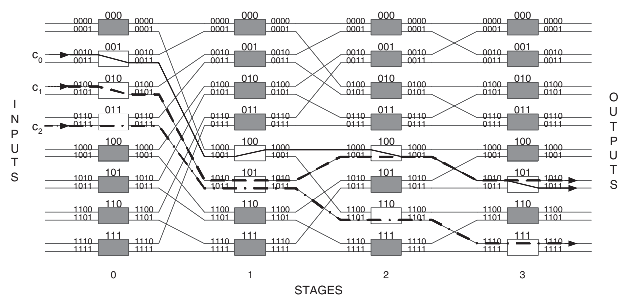

Figure 2. The self-routing of connections c_0, c_1, and c_2 in $BL(16)$.

Self-routing in $BL(N)$ is decided by the destination of each connection. Routing from inputs to outputs of $BL(N)$, if the $(n-i)$-th bit, d_{n-i-1}, of the destination equals to 0 (resp. 1), the input of the SE through which the connection passes in stage i is connected to the SE's upper (resp. lower) output. Fig. 2 shows three connection paths for connections c_0, c_1 and c_2. Connections c_0 and c_1 share two links, output link 1001 in stage 2 and input link 1010 in stage 3, and two SEs, SE 4 in stage 2 and SE 5 in stage 3; connections c_1 and c_2 share SE 5 in stage 1. Clearly, Banyan networks are blocking space-division-multiplexing networks. In the next subsection, we will show by using wavelength dilation, Banyan networks can be WRSS networks.

3.2. Routing and Wavelength Assignment algorithm

The idea of our crosstalk-free routing and wavelength assignment algorithm for WRSS Banyan networks is as follows. We partition a set of connections into subsets so that the connections in the same subset don't share any SE or link, and then assign the connections in different subsets with different wavelengths and the connections in the same subset with the same wavelength. Each of these subsets is called a *crosstalk-free (CF) subset*. Clearly, this wavelength assignment will not cause crosstalk in any SEs. Since

2 The algorithms discussed in this paper are all based on a completely connected multiprocessor system consisting of a set of N PEs connected in such a way that there is a direct connection between every pair of PEs. We assume that each PE can communicate with at most one processor during a communication step. The presented algorithms run on a completely connected multiprocessor system can be easily transformed to algorithms on more realistic multiprocessor systems as talked in Section 5.

3 In this paper, we assume $N = 2^n$ ($n = \log N$) and all logarithms are in base 2.

$BL(N)$ is a self-routing network, the routing for each connection can be easily done following the self-routing rule. We only need to consider how to partition a set of connections into CF subsets and assign the connections in different subsets with different wavelengths.

In order to find CF subsets, we need to study the permutation capacity of $BL(N)$ first. For $BL(N)$, the k-th *modulo-g input group* comprises inputs $I_{(k-1)g}, I_{(k-1)g+1}, \cdots, I_{kg-1}$, and the k-th *modulo-g output group* comprises outputs $O_{(k-1)g}, O_{(k-1)g+1}, \cdots, O_{kg-1}$, where $g = 2^i$ with $0 \leq i \leq n$ and $1 \leq k \leq N/g$. We say that two connections *share* a modulo-g input (resp. output) group if their sources (resp. destinations) are in the same modulo-g input (resp. output) group. The following lemma is proved in [13].

Lemma 2 *Given a partial permutation π of $BL(N)$, if any two connections in π do not share any modulo-$2^{\lfloor \frac{n+1}{2} \rfloor}$ input group and also do not share any modulo-$2^{\lfloor \frac{n+1}{2} \rfloor}$ output group, then π can be routed in $BL(N)$ simultaneously without crosstalk.*

We assume $g = 2^{\lfloor \frac{n+1}{2} \rfloor}$ in the rest of this section. By Lemma 2, if we assign different wavelengths to the connections in π with sources (resp. destinations) sharing the same modulo-g input (resp. output) group, then we can route π in $BL(N)$ without crosstalk. This wavelength assignment problem can be reduced to the edge coloring of a bipartite graph as follows.

Given any partial permutation π with K active inputs for $BL(N)$, we construct a graph $G(\pi, g)$, named *I/O mapping graph*, as follows. The vertex set consists of two parts, V_1 and V_2. Each part has N/g vertices, i.e., each modulo-g input (resp. output) group is represented by a vertex in V_1 (resp. V_2). There is an edge between vertex $\lfloor i/g \rfloor$ in V_1 and vertex $\lfloor j/g \rfloor$ in V_2 if $j = \pi(i)$. Thus, $G(\pi, g)$ is a bipartite graph with N/g vertices in each of V_1 and V_2 and K edges, where at most g edges are incident at any vertex, and the degree of $G(\pi, g)$ equals to g. Since there may be more than one connection from a modulo-g input group to the same modulo-g output group, $G(\pi, g)$ may have parallel edges between two vertices.

It has been proved that any bipartite graph G has a $\Delta(G)$-edge coloring [3]. Hence, $G(\pi, g)$ has a g-edge coloring since $G(\pi, g)$ is bipartite and $\Delta(G(\pi, g)) = g$. Thus, if we can find a g-edge coloring of $G(\pi, g)$, then we can assign wavelength i to the connections corresponding to the edges with the color i, $0 \leq i \leq g - 1$. By Lemma 2, we know this wavelength assignment will not cause any crosstalk in $BL(N)$.

An efficient algorithm for finding a g-edge coloring of a bipartite graph can be found in [11], from which we have the following lemma.

Lemma 3 *For any partial permutation π with K active inputs, a g-edge coloring of the I/O mapping graph $G(\pi, g)$ can be found in $O(\log g \cdot \log K)$ time using a completely connected multiprocessor system of N PEs.*

By the above discussion and Lemma 3, the following Theorem is clear since $O(\log g) = O(\log N)$.

Theorem 1 *For any partial permutation π with $K (\leq N)$ active inputs, a crosstalk-free routing and wavelength assignment of π for a WRSS $BL(N)$ can be found in $O(\log N \cdot \log K)$ time using at most $2^{\lfloor \frac{n+1}{2} \rfloor}$ wavelengths on a completely connected multiprocessor system of N PEs.*

It is easy to verify that $2^{\lfloor \frac{n+1}{2} \rfloor}$ wavelengths are also necessary for a WRSS $BL(N)$ since there exist permutations with $2^{\lfloor \frac{n+1}{2} \rfloor}$ connections sharing a common SE.

4. Parallel Routing and Wavelength Assignment in WRSR Benes Networks

4.1. Benes Networks

The Benes network [2] is one of the most efficient switching architectures in terms of the number of 2×2 SEs used. We denote an $N \times N$ Benes network by $B(N)$, which can be constructed from $BL(N)$ by concatenating the mirror image of the first $\log N - 1$ stages of a $BL(N)$ to the back of the $BL(N)$. Thus, a $B(N)$ consists of $2 \log N - 1$ stages labeled by $0, 1, \cdots, 2n - 2$ from left to right. Each $B(N)$ contains 2 $B(N/2)$s from stage 1 to stage $2n - 3$, respectively named *upper subnetwork* and *lower subnetwork*, each having 2 $B(N/4)$s from stage 2 to $2n - 4$, named upper subnetwork and lower subnetwork of $B(N/4)$ respectively, and so on. Fig. 3 shows a $B(8)$, which contains 2 $B(4)$s within dashed boxes, each containing 2 $B(2)$s within dotted boxes.

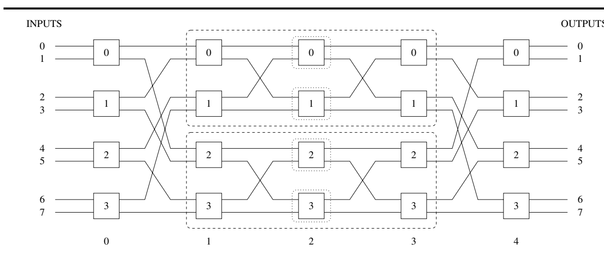

Figure 3. A $B(8)$.

Benes networks are space-division-multiplexing rearrangeable nonblocking. By [10, 26], we know that each permutation can be decomposed into two crosstalk-free partial permutations so that each CF partial permutation can be routed in an optical Benes network simultaneously. Hence, if we assign the same wavelength to the connections in the same CF partial permutation and assign the different wavelengths to the connections in different CF partial permutations, two wavelengths are sufficient for a WRSR $B(N)$ in which SEs may contain non-basic states. In the following two subsections, we will show the case that WRSR Benes networks only contain basic SEs, which has the reduced hardware complexity [16].

4.2. Upper Bound for the Number of Wavelengths

In order to find an upper bound for the number of wavelengths needed for crosstalk-free routing, we need to consider routing a permutation in an OMIN. We model the wavelength assignment for a permutation in an OMIN as the vertex coloring of a graph G_ω, where the vertex set $V(G_\omega) = \{\text{connections}\}$ and the edge set $E(G_\omega) = \{\{u, v\}|\text{two connections } u \text{ and } v \text{ conflict with each other}\}$. We call G_ω a *wavelength conflict graph*. Although finding the minimum number of wavelengths and assigning the wavelengths to the connections are equivalent to finding the minimum number of colors and assigning the colors to the vertices respectively, which are both NP-complete for general graphs, we can find an upper bound for the number of wavelengths needed for realizing any permutation in WRSR Benes networks.

Theorem 2 *For any permutation of a WRSR $B(N)$,*

$$\omega \leq \begin{cases} 2\log N, & \textit{if } N \leq 4 \\ 2\log N - 1, & \textit{otherwise} \end{cases}$$

where ω is the number of wavelengths needed for the crosstalk-free routing of a permutation in $B(N)$.

Proof. Each connection conflicts with at most $2\log N - 1$ connections since it passes through total $2\log N - 1$ basic SEs. Thus $\Delta(G_\omega) \leq 2\log N - 1$. By Brooks theorem (see a proof in [3]), if G_ω is neither a complete graph nor an odd cycle, then we need at most $\Delta(G_\omega)$ colors to color $V(G_\omega)$ such that any two adjacent vertices have different colors; otherwise $\Delta(G_\omega) + 1$ colors are sufficient. Clearly, for any permutation of an OMIN with $N > \Delta(G_\omega) + 1$, G_ω is neither a complete graph nor an odd cycle since $\Delta(G_\omega) < N - 1$ and N is even. Therefore, the theorem holds. □

The simple proof of an upper bound on the number of required wavelengths does not directly lead to a wavelength assignment algorithm. In the next subsection, we utilize the properties of our permutation decomposition and the structure of Benes network to obtain a fast parallel crosstalk-free routing and wavelength assignment algorithm for a WRSR $B(N)$ using no more than $2\log N$ wavelengths.

4.3. Routing and Wavelength Assignment Algorithm

Our routing and wavelength assignment algorithm uses the permutation decomposition algorithm of [10] as a subalgorithm and the vertex coloring technique similar to that of [4]. Conceptually, this algorithm has $\log N$ iterations from iteration 0 to iteration $\log N - 1$. In each iteration i, if $i < \log N - 1$, the algorithm decides the setting of SEs in stage i and stage $2\log N - 2 - i$ and uses at most $2i + 3$ wavelengths to ensure that there is no wavelength conflict in stage j for any $j \in \{0, \cdots, i\} \cup \{2\log N - 2 - i, \cdots, 2\log N - 2\}$; if $i = \log N - 1$, the algorithm decides the setting of SEs in stage $\log N - 1$ and uses at most $2\log N$ wavelengths to ensure that there is no wavelength conflict in $B(N)$.

We define a *wavelength class* as the set of connections assigned the same wavelength. A wavelength λ is called a *free* wavelength for a connection c if λ is not assigned to any connection conflicting with c.

Each PE_i is associated with connection i, and maintains one variable $\lambda(i)$, and two arrays C_i and W_i, $0 \leq i < N - 1$. For any $0 \leq i \leq N - 1$, C_i consists of $2\log N - 1$ entries $C_i[j]$, $0 \leq j \leq 2\log N - 2$, and W_i consists of $2\log N$ entries $W_i[k]$, $0 \leq k \leq 2\log N - 1$. $\lambda(i)$, $C_i[j]$, and $W_i[k]$ are used to record the assigned wavelength, the new conflicting connections generated in iteration $\lfloor j/2 \rfloor$, and the number of conflicting connections with wavelength k, respectively, for connection i. We call C_i and W_i *connection conflict array* and *wavelength conflict array* of connection i, respectively. The other variables are all working variables. Initially, let $\lambda(i) := 0$, $C_i[j] := \infty$, and $W_i[k] := 0$, for $i \in \{0, \cdots, N - 1\}$, $j \in \{0, \cdots, \leq 2\log N - 2\}$, and $k \in \{0, \cdots, 2\log N - 1\}$, respectively. We use operator ":=" to denote an assignment local to a PE or to the control unit, and use operator "←" to denote an assignment requiring some interprocessor communication. In our parallel routing and wavelength assignment algorithm, each iteration i consists of the following steps:

Step 1-Permutation Decomposition: decompose a (partial) permutation of each subnetwork $B(N/2^i)$ into two parts, each named upper or lower partial permutation, satisfying that two active inputs (resp. outputs) in an SE in the first (resp. last) stage of $B(N/2^i)$ are in different parts.

Step 2-Setting SEs: set the SEs in the first and last stages of each $B(N/2^i)$ in such a way that (i) if $i \neq \log N - 1$, the active inputs and outputs in the upper (resp. lower) partial permutation are connected with an upper (resp. lower) subnetwork $B(N/2^{i+1})$; (ii) if $i = \log N - 1$, each active input is connected with its mapped output.

The above two steps decide the routing for the given permutation. The following steps are used to find a wavelength assignment for the routing solution. For all PE_c, $0 \leq c \leq N - 1$, do in parallel:

Step 3-Recording Conflicting Connections: (i) if there is a connection c' so that c and c' pass through the same SE in stage i and $c' \neq C_c[j]$ for all $0 \leq j < 2i$, then $C_c[2i] := c'$; (ii) if $i \neq \log N - 1$ and there is a connection c" so that c and c'' pass through the same SE in stage $2\log N - 2 - i$ and $c'' \neq C_c[j]$ for all $0 \leq j < 2i+1$, then $C_c[2i+1] := c''$.

Step 4-Reassigning Wavelengths: if connection c is in a lower partial permutation, $\lambda'(c) := \lambda(c)$ and $\lambda(c) := \lambda(c) + (2i+1)$.

Step 5-Updating Conflicting Wavelengths: update wavelength conflicts by (i) adding new conflicts and (ii) updating existing conflicts, where (ii) consists of two substeps: (ii-1) clearing old wavelengths and (ii-2) adding updated wavelengths. The detailed implementation of this step is given in Algorithm 1.

```
if i ≠ log N − 1, j' := 2i + 1; otherwise, j' := 2i;
for all PE_c, 0 ≤ c ≤ N − 1, do
  t(c) := ∞;
  for j = 2i to j' do
    if C_c[j] ≠ ∞ and λ(c) ≤ 2 log N − 1 then
      t(C_c[j]) ← λ(c);
    end if
    if t(c) ≠ ∞ then
      W_c[t(c)] := W_c[t(c)] + 1;  /* (i): adding new conflicts */
      t(c) := ∞;
    end if
  end for
  if connection c is in a lower partial permutation and i ≠ 0 then
    for j = 0 to 2i − 1 do
      if C_c[j] ≠ ∞ then
        t(C_c[j]) ← λ'(c);
      end if
      if t(c) ≠ ∞ then
        W_c[t(c)] := W_c[t(c)] − 1;  /* (ii-1): clearing old wavelengths */
        t(c) := ∞;
      end if
      if C_c[j] ≠ ∞ and λ(c) ≤ 2 log N − 1 then
        t(C_c[j]) ← λ(c);
      end if
      if t(c) ≠ ∞ then
        W_c[t(c)] := W_c[t(c)] + 1;  /* (ii-2): adding updated wavelengths */
        t(c) := ∞;
      end if
    end for
  end if
end for
```

Algorithm 1: Updating Conflicting Wavelengths

By the above five steps, it is easy to know the wavelength assignment in each iteration will not result in any conflict in the SEs that have been set up so far. However, we can reduce the number of wavelengths by reassigning new wavelengths in $\{0, \cdots, 2(i+1)\}$ to the connections with wavelengths in $\{2(i+1)+1, \cdots, 2(2i+1)-1 = 4i+1\}$ without resulting in any wavelength conflict. (The correctness for the reassignment of wavelengths will be proved in Lemma 4.) This is done as follows: for $\lambda^* = 2(i+1)+1$ to $4i+1$, if $\lambda(c) = \lambda^*$, then perform the following two steps:

Step 6-Adjusting Wavelengths: find a free wavelength $j \in \{0, 1, \cdots, j'+1\}$ such that $W_c[j] = 0$ by checking the values in $\{W_c[0], \cdots, W_c[j'+1]\}$, and $\lambda'(c) := \lambda(c)$ and $\lambda(c) := j$. (The value of j' in this step and next step is the same as that in Algorithm 1.)

Step 7-Updating Conflicting Wavelengths: for $k = 0$ to j', do (i) if $C_c[k] \neq \infty$ and $\lambda'(c) \leq 2\log N - 1$, then decrease $W_{C_c[k]}[\lambda'(c)]$ by 1; and (ii) if $C_c[k] \neq \infty$, then increase $W_{C_c[k]}[\lambda(c)]$ by 1. (The detailed implementation is similar to Algorithm 1)

Lemma 4 *After iteration i,* $0 \leq i \leq \log N - 1$*, of our parallel routing and wavelength assignment algorithm, there is no wavelength conflict in stage j, for any* $j \in \{0, \cdots, i\} \cup \{2\log N - 2 - i, \cdots, 2\log N - 2\}$*, and at most* ω_i *wavelengths are used, where*

$$\omega_i \leq \begin{cases} 2(i+1), & \textit{if } i = 0, \log N - 1 \\ 2(i+1)+1, & \textit{otherwise} \end{cases}$$

Proof. The proof is done by induction on iteration i. If $i = 0$, it is true since two connections passing though the same SE in the first or last stage are assigned different wavelengths and $\omega_0 = 2$. Now we assume that it is true for any $i < k \leq \log N - 1$. In iteration k, by assumption, we know that there is no wavelength conflict in stage j, for any $j \in \{0, \cdots, k-1\} \cup \{2\log N - 1 - k, \cdots, 2\log N - 2\}$, using ω_{k-1} wavelengths. By Step 4, two connections passing though the same SE in stage k and stage $2\log N - 2 - k$ are assigned different wavelengths using $2 \cdot \omega_{k-1}$ wavelengths. Hence, there is no wavelength conflict in stage j for any $j \in \{0, \cdots, k\} \cup \{2\log N - 2 - k, \cdots, 2\log N - 2\}$, using $2 \cdot \omega_{k-1}$ wavelengths. In the following, we show that $2 \cdot \omega_{k-1}$ wavelengths are too much for the case that $2 \cdot \omega_{k-1} > 2(k+1)+1$ if $k \neq \log N - 1$ or the case that $2 \cdot \omega_{k-1} > 2\log N$ if $k = \log N - 1$. For iteration k, each connection conflicts with at most $2(k+1)$ connections if $k \neq \log N - 1$ and at most $2\log N - 1$ if $k = \log N - 1$. This is because for iteration j, if $j \leq k < \log N - 1$, we need to consider wavelength conflicts in two stages, stages j and $2\log N - 2 - j$; if $j = k = \log N - 1$, we only need to consider wavelength conflict in stage $\log N - 1$ since stage j and stage $2\log N - 2 - j$ are the same. Thus, in Step 6, a free wavelength of index no greater than $2(k+1)$ for $k < \log N - 1$ and $2\log N - 1$ for $k = \log N - 1$ can always be found. Furthermore, the connections in the same wavelength class have no wavelength conflict so that we can do wavelength adjustment for these connections at the same time without resulting in any new conflict. □

Theorem 3 *For any (partial) permutation, a routing and wavelength assignment for a WRSR $B(N)$ can be found in $O(\log^3 N)$ time using at most $2 \log N$ wavelengths on a completely connected multiprocessor system of N PEs.*

Proof. By the recursive structure of $B(N)$ and by applying our permutation decomposition algorithm recursively, we can find a setting of SEs in $B(N)$ so that any permutation can be realized. By Lemma 4, we know that the wavelength assignment assures no wavelength conflict for the routing solution. Now, we analyze the time complexity. In each iteration, Steps 2 and 4 take $O(1)$ time and each of other steps takes $O(\log N)$ time. Iteration i has at most $\omega_{i-1}(\leq 2i+1)$ wavelength classes to be adjusted, and thus, Steps 6 and 7 in iteration i are executed at most $\omega_{i-1}(\leq 2i+1) = O(\log N)$ times. Since there are $\log N$ iterations, the total time complexity of our routing and wavelength assignment algorithm is $O(\log^3 N)$. □

5. Implementation on Realistic Multiprocessor Systems

The presented algorithms running on a completely connected multiprocessor system can be easily transformed to algorithms on more realistic multiprocessor systems. As an example, in this section, we show how to implement our algorithms on a hypercube of $N/2$ PEs such that any (partial) permutation can be routed without crosstalk in a WRSS $BL(N)$ and a WRSR $B(N)$ in $O(\log^4 N)$ time.

In our presentation, the Benes network $B(N)$ is the back-to-back concatenation of two $BL(N)$'s. A *Butterfly network* (also known as Banyan-type network) of N inputs and N outputs, denoted by $BF(N)$, is isomorphic to $BL(N)$ (see Fig. 4 (a) and (b)). An n-dimensional *hypercube*, denoted by $H(2^n)$, is constructed recursively. $H(2)$ is an edge with two nodes. $H(2^n)$ is constructed from 2 $H(2^{n-1})$'s by adding 2^{n-1} edges, named *n-dimension edges*, that connect the corresponding 2^{n-1} nodes in 2 $H(2^{n-1})$'s. Butterfly networks are in the family of the hypercube [9] because $H(N/2)$ can be obtained from $BF(N)$ by merging all SEs in row i of $BF(N)$ as a node i of $H(N/2)$ and merging all links connecting SEs contained in two different nodes as an edge of $H(N/2)$. Fig. 4 (c) shows a $H(4)$, where k-dimension edges are labeled by k^*. Since each PE can communicate with at most one other PE in every communication step of our algorithms, in the following, we show how to implement one communication step of a completely connected multiprocessor system of N PEs by a set of one-to-one communications on a $H(N/2)$, in which each PE is responsible for a pair of connections i and $\bar{i}$.

The time complexity of our routing and wavelength assignment algorithm for a WRSS $BL(N)$ depends on g-edge coloring algorithm, which can be implemented in $O(\log^4 N)$ time on $H(N/2)$ [11], Thus, the routing and wavelength assignment algorithm for a WRSS $BL(N)$ takes $O(\log^4 N)$ on $H(N/2)$.

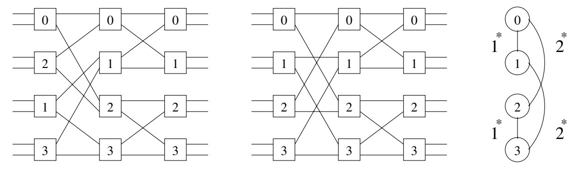

Figure 4. The relationship of Baseline network, Butterfly network and hypercube (a) $BL(8)$ **(b)** $BF(8)$ **(c)** $H(4)$

Considering our routing and wavelength assignment algorithm for a WRSR $B(N)$, we can see that the total time for routing on $H(N/2)$ only depends on the decomposition algorithm [10], which can be implemented in $O(\log^3 N)$ time on $H(N/2)$ since each pointer jumping step on a completely connected multiprocessor system can be implemented on $H(N/2)$ by a sorting operation, which takes $O(\log^2 N)$ time. Consequently, the routing on $H(N/2)$ takes $O(\log^4 N)$ time. For wavelength assignment, communications among PEs only occur in Step 5 and Step 7, in which PE_c needs to talk to PE_d if d is recorded in C_c (see "$\leftarrow$" operations in Algorithm 1). Fortunately, all conflicting connections of c are recorded in connection conflict array C_c in the order of SEs through which c passes from both sides, i.e. from a pair of outside stages i and $2\log N - 2 - i$ towards the center stage, stage $\log N - 1$. Thus, these conflicting connections can be located using this ordering via interstage connections in $B(N)$. Since the interstage interconnection pattern between stage i (resp. $2\log N - 2 - i$) and stage $i+1$ (resp. $2\log N - 3 - i$) of $B(N)$ corresponds to $(\log \frac{N}{2} - i)$-dimension edges of $H(N/2)$, the communication ordering defined by connection conflict arrays directly corresponds to a classic hypercube communication technique called *dimension ordering*. Thus, the total time for wavelength assignment on $H(N/2)$ remains unchanged. Therefore, when our routing and wavelength assignment algorithm for a WRSR $B(N)$ is implemented on $H(N/2)$, it has a slowdown factor of $O(\log N)$ and its time complexity is $O(\log^4 N)$.

6. Conclusion

In this paper, we studied the crosstalk problem in OMINs using wavelength dilation approach. We proposed parallel routing and wavelength assignment algorithms to route a partial permutation in optical WRSS Banyan networks and WRSR Benes networks so that there is no crosstalk in these networks. For an arbitrary partial permutation, it can be routed without crosstalk in a WRSS $BL(N)$ in $O(\log^2 N)$

time using at most $2^{\lfloor \frac{\log N+1}{2} \rfloor}$ wavelengths and in a WRSR $B(N)$ with only basic SEs in $O(\log^3 N)$ time using at most $2 \log N$ wavelengths, on a completely connected multiprocessor system with N PEs. The proposed algorithms run on a completely connected multiprocessor system can be easily transformed to algorithms on more realistic multiprocessor systems. For example, our routing and wavelength assignment algorithms for a WRSS $BL(N)$ and a WRSR $B(N)$ take $O(\log^4 N)$ time on a hypercube with $N/2$ PEs.

References

[1] D.P. Agrawal, "Graph Theoretical Analysis and Design of Multistage Interconnection Networks", *IEEE Transactions on Computers*, vol. C-32, no. 7, pp. 637-648, July 1983.

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

[3] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, Elsevier North-Holland, 1976.

[4] A.V. Goldberg, S.A. Plotkin, and G.E. Shannon, "Parallel Symmetry-Breaking in Sparse Graphs," *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pp. 315-323, 1987.

[5] Q.P. Gu and S. Peng, "Wavelengths Requirement for Permutation Routing in All-Optical Multistage Interconnection Networks", *Proceedings of 14th International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 761-768, May 2000.

[6] H. Hinton, "A Non-Blocking Optical Interconnection Network Using Directional Couplers", *Proceedings of IEEE Global Telecommunications Conference*, pp. 885-889, Nov. 1984.

[7] D.K. Hunter, P.J. Legg, and I. Andonovic, "Architecture for Large Dilated Optical TDM Switching Networks", *IEE Proceedings on Optoelectronics*, vol. 140, no. 5, pp. 337-343, Oct. 1993.

[8] F.K. Hwang, *The Mathematical Theory of Nonblocking Switching Networks*, World Scientific, 1998.

[9] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays · Trees · Hypercubes*, Morgan Kaufmann Publishers, 1992.

[10] E. Lu and S.Q. Zheng, "High-Speed Crosstalk-Free Routing for Optical Multistage Interconnection Networks", *Proceedings of the 12th IEEE International Conference on Computer Communications and Networks*, pp. 249-254, Oct. 2003.

[11] E. Lu and S.Q. Zheng, "Parallel Routing Algorithms for Nonblocking Electronic and Photonic Multistage Switching Networks", *Proceedings of IEEE International Parallel and Distributed Processing Symposium (IPDPS 2004), Workshop on Advances in Parallel and Distributed Computing Models*, April, 2004.

[12] G. Maier, A. Pattavina, and S.G. Colombo, "Control of Non-filterable Crosstalk in Optical-Cross-Connect Banyan Architectures", *Proceedings of IEEE Global Telecommunications Conference GLOBECOM*, vol. 2, pp. 1228-1232, Nov.-Dec. 2000.

[13] G. Maier and A. Pattavina, "Design of Photonic Rearrangeable Networks with Zero First-Order Switching-Element-Crosstalk", *IEEE Transactions on Communications*, vol. 49, no. 7, pp. 1268-1279, Jul. 2001.

[14] K. Padmanabhan and A. Netravali, "Dilated Network for Photonic Switching", *IEEE Transactions on Communications*, vol. COM-35, no. 12, pp. 1357-1365, Dec. 1987.

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

[16] G. Pieris and G. Sasaki, "A Linear Lightwave Benes Network", *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 441-445, Aug. 1993.

[17] C. Qiao, R. Melhem, D. Chiarulli, and S. Levitan, "A Time Domain Approach for Avoiding Crosstalk in Optical Blocking Multistage Interconnection Networks", *IEEE Journal Lightwave Technology*, vol. 12, no. 10, pp. 1854-1862, Oct. 1994.

[18] X. Qin and Y. Yang, "Nonblocking WDM Switching Networks with Full and Limited Wavelength Conversion", *IEEE Transactions on Communications*, vol. 50, no. 12, pp. 2032-2041, Dec. 2002.

[19] J. Sharony, S. Jiang, T.E. Stern, and K.W. Cheung, "Wavelength Rearrangeable and Strictly Nonblocking Networks", *IEEE Electronics Letters*, vol. 28, no. 6, pp. 536-537, Mar. 1992.

[20] J. Sharony, K.W. Cheung, and T.E. Stern, "The Wavelength Dilation Concept in Lightwave Networks-Implementation and System Considerations", *IEEE Journal of Lightwave Technology*, vol. 1, no. 5/6, pp. 900 -907, May-Jun. 1993.

[21] X. Shen, F. Yang, and Yi Pan, "Equivalent Permutation Capabilities between Time-Division Optical Omega Networks and Non-Optical Extra-Stage Omega Networks", *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 518-524, Aug. 2001.

[22] G.H. Song and M. Goodman, "Asymmetrically-Dilated Cross-Connect Switches for Low-Crosstalk WDM Optical Networks", *Proceedings of IEEE 8th Annual Meeting Conference on Lasers and Electro-Optics Society Annual Meeting*, vol. 1, pp. 212-213, Oct. 1995.

[23] F.M. Suliman, A.B. Mohammad, and K. Seman, "A Space Dilated Lightwave Network-a New Approach", *Proceedings of IEEE 10th International Conference on Telecommunications (ICT 2003)*, vol. 2, pp. 1675 -1679, 2003.

[24] M. Vaez and C.T. Lea, "Strictly Nonblocking Directional-Coupler-Based Switching Networks under Crosstalk Constraint", *IEEE Transactions on Communications*, vol. 48, no. 2, pp. 316-323, Feb. 2000.

[25] T.S. Wong and C.T. Lea, "Crosstalk Reduction Through Wavelength Assignment in WDM Photonic Switching networks", *IEEE Transactions on Communications*, vol. 49, no. 7, pp. 1280-1287, Feb. 2001.

[26] 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.