Optimal Routing and Channel Assignments for Hypercube Communication on Optical Mesh–like Processor Arrays. *

X. Yuan and R. Melhem
Department of Computer Science
The University of Pittsburgh
Pittsburgh, PA 15260
{xyuan, melhem}@cs.pitt.edu

Abstract

This paper considers optimal routing and channel assignment (RCA) schemes to realize hypercube communication on optical mesh–like networks. Specifically, we identify lower bounds on the number of channels required to realize hypercube communication on top of array and ring topologies and develop optimal RCA schemes that achieve the lower bounds on these two topologies. We further extend the schemes to mesh and torus topologies and obtain RCA schemes that use at most 2 more channels than the optimal for these topologies.

1. Introduction

In an optical interconnection network, each physical link can be multiplexed to create several *virtual channels* via either *Time–division multiplexing* (TDM) or *wavelength–division multiplexing* (WDM) [3, 6, 8]. A direct optical connection, also called a *lightpath* [3], between two nodes can be established using the virtual channels. To achieve *all–optical* communication without optical/electrical and electrical/optical conversions at intermediate nodes, the same channel on all links along the path must be used to establish the lightpath. This requirement is called the *channel–continuous* requirement [2].

Due to the channel–continuous requirement, traditional embedding techniques, which typically minimize the *congestion* for a given communication pattern, are not sufficient to minimize the number of virtual channels needed to realize the communication in an optical network. *Routing and channel assignment* (RCA) tries to minimize the number of channels to realize a communication requirement by

taking into consideration both routing options and channel assignment options. The RCA problem can be described as follows. Given a set of all–optical connections, the problem is to (a) find routes from the source nodes to their respective destinations, and (b) assign channels to these routes so that the same channel is assigned to all the links of a particular route. The goal of RCA is to minimize the number of assigned channels. We show in this paper that, using traditional XY shortest path routing, (nearly) optimal RCA schemes for realizing hypercube communication on mesh–like topologies can be achieved.

Realizing hypercube communication efficiently on optical mesh–like networks is important because of the following reasons. (1) There exist many parallel algorithms that use the hypercube communication pattern [7]. (2) Realizing a virtual hypercube topology, whose diameter is $O(lg(N))$, where N is the network size, on top of a mesh–like topology with a larger diameter, reduces the average number of relays when messages are routed on the virtual hypercube rather than on the physical mesh–like topology. The number of relays per message transmission is particularly important in optical networks where a relay requires optical/electronic and electronic/optical conversions. An extensive study of the maximum throughput and the message delay of routing dynamic traffic on the virtual hypercube described in this paper is performed in [12], where it is shown that routing messages on the logical hypercube, in general, improves the communication efficiency over routing messages on the physical mesh–like topologies.

In this paper, we study optimal RCA schemes for hypercube communication on optical mesh–like networks. Specifically, given networks of size N, we prove that $\lfloor \frac{2N}{3} \rfloor$ and $\lfloor \frac{N}{3} + \frac{N}{4} \rfloor$ channels are the minimum required to realize hypercube communication on array and ring topologies, respectively. We develop routing and channel assignment schemes that achieve these minimum requirements, which

indicates that the bounds are tight and the schemes are optimal. We further extend these schemes to mesh and torus topologies. We prove that for a $2^k \times 2^{r-k}$ ($k \geq r-k$) mesh or torus, $\lfloor \frac{2 \times 2^k}{3} \rfloor$ and $\lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor$ channels are the minimum required for realizing hypercube communication, respectively. We obtain routing and channel assignment schemes that use at most 2 more channels than the optimal to realize hypercube communication on these topologies.

Many researchers have studied the RCA problem in the context of either WDM networks or TDM networks. Heuristic RCA algorithms for arbitrary networks have been developed [2, 4, 10, 13] for WDM networks. In [2, 10], the optimal RCA problem is formulated as the *multicommodity 0–1 flow* problem which is NP–hard. Realizing hypercube communication is a subproblem of the general optimal RCA problem in [2, 10]. However, our algorithms provide exact solutions to the problem without the need to exploit exponential search domains. Heuristic algorithms for channel assignments on mesh topologies are studied in [11]. In [9], RCA for permutation communication patterns in mesh–like networks is considered. In [8], optimal schemes for realizing all–to–all patterns in multi-stage networks are presented. In [5], message scheduling for all–to–all communication in mesh–like topologies is described. A large body of related embedding techniques are surveyed in [7]. However, as discussed earlier, the objective and the techniques of RCA are different from that of embedding. To the best of our knowledge, optimal routing and channel assignment for hypercube communication in mesh–like topologies has not been studied previously.

The rest of the paper is organized as follows. In section 2 defines the problem. Section 3 considers the array topology. Section 4 discusses the ring topology. Section 5 presents the scheme for meshes and section 6 considers the torus topology. Section 7 concludes the paper.

2 Problem definition

We model a network as a directed graph G(V, E), where nodes in V are switches and edges in E are links. Each node in a network is assigned a node number starting from 0. We assume that, in arrays and rings, the nodes are numbered from left to right in ascending order, and that the nodes are numbered in row major order for meshes and tori of size $n \times m$. That is, the node in the ith column and the jth row is numbered as $j \times m + i$. Fig. 1 shows a 4×4 torus topology. This paper focuses on studying the optimal RCA schemes for these traditional numbering schemes. Optimal node numbering (and its RCA) is a much more complex problem and is beyond the score of this paper. We assume
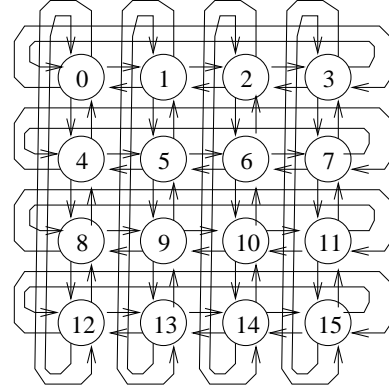
(d) a 4x4 torus

Figure 1. Node numbering in a torus topology

that the number of nodes in a network is $N = 2^r$. For a mesh or a torus to contain 2^r nodes, each row and column must contain a power of two number of nodes. Hence, we will denote sizes of meshes and tori as $N = 2^k \times 2^{r-k}$. We will use the notations $ARRAY(N)$ and $RING(N)$ to represent arrays and rings of size N respectively. The notations $MESH(2^k \times 2^{r-k})$ and $TORUS(2^k \times 2^{r-k})$ are used to represent meshes and tori of size $2^k \times 2^{r-k}$. Without losing generality, we always assume $k \geq r - k$.

We denote by (src, dst) the connection from node src to node dst. A *communication pattern* is a collection of connections. The *hypercube* communication pattern contains a connection (src, dst) if and only if the binary representations of src and dst differ in precisely one bit. A connection in the hypercube communication pattern is called a *dimension l connection* if it connects two nodes that differ in the lth bit position. In a network of size $N = 2^r$, the set, DIM_l, where $0 \leq l \leq r-1$, is defined as the set of all dimension l connections and H_r is defined as the hypercube communication pattern. That is

$$DIM_l = \{(i, i+2^l) \mid i \bmod 2^{l+1} < 2^l\} \cup$$
$$\{(i, i-2^l) \mid i \bmod 2^{l+1} \geq 2^l\}$$

$$H_r = \cup_{l=0}^{r-1} DIM_l$$

It can be easily proven that removing any DIM_l, for any $l \leq r-1$, from H_r leaves two disjoint sets of connections, each of which being a hypercube pattern on $\frac{N}{2}$ nodes. For example, removing DIM_0 from H_r results in an H_{r-1} on the even–numbered nodes and another H_{r-1} on the odd–numbered nodes when the nodes are properly renumbered. Next, we introduce some definitions and summarize the results of this paper.

Definition : $P(x, y)$ is a directed path in G from node x

to node y. It consists of a set of consecutive edges beginning at x and ending at y.

Definition : Given a network G and a communication pattern I, A *routing R(I)* of I is a set of directed path $R(I) = \{P(x,y)|(x,y) \in I\}$.

Definition : Given a network G, a communication pattern I and a routing R(I) for the communication pattern, the congestion of an edge $\alpha \in E$, denoted as $\pi(G, I, R(I), \alpha)$, is the number of paths in R(I) containing α. The *congestion* of G in the routing R(I), denoted as $\pi(G, I, R(I))$, is the maximum congestion of any edge of G in the routing R(I), that is $\pi(G, I, R(I)) = max_\alpha\{\pi(G, I, R(I), \alpha)\}$. The *congestion* of G for a communication pattern I, denoted as $\pi(G, I)$, is the minimum congestion of G in any routing R(I) for I, that is $\pi(G, I) = min_R\{\pi(G, I, R(I))\}$.

Definition : Given a network G and a routing $R(I)$ for communication pattern I, an *assignment function A* : $R \to INT$, is a mapping from the set of paths to the set of integer INT, where an integer corresponds to a channel. A *channel assignment* for a routing $R(I)$ is an assignment function A that satisfies the following conditions:

1. $A(P(x_1, y_1)) \neq A(P(x_2, y_2))$ if $P(x_1, y_1)$, $P(x_2, y_2)$ are different paths that share a common edge.
2. $A(P(x, y_1)) \neq A(P(x, y_2))$ and $A(P(x_1, y)) \neq A(P(x_2, y))$.

The condition (1) ensures that each channel on one link can only be assigned to one connection. The condition (2) ensures that each node can only use one channel to send to or receive from one other node. A channel assignment that violates condition (1) is said to have *link conflicts*, and a channel assignment that violates condition (2) is said to have *node conflicts*. We denote by $A(R)$ the set of channels assigned to the paths in R and by $|A(R)|$ the size of $A(R)$. We denote by $w(G, I, R)$ the minimum number of channels for the routing R, thus $w(G, I, R) = min_A\{|A(R)|\}$. We denote by $w(G, I)$ the smallest $w(G, I, R)$ over all R, i.e. $w(G, I) = min_R\{w(G, I, R)\}$

lemma 1: $w(G, I) \geq \pi(G, I)$.
　　Proof: Straight forward from the definitions. □

In the following sections, we will show that

$$w(ARRAY(N), H_r) = \pi(ARRAY(N), H_r) = \lfloor \tfrac{2N}{3} \rfloor$$
$$w(RING(N), H_r) = \pi(RING(N), H_r) = \lfloor \tfrac{N}{3} + \tfrac{N}{4} \rfloor$$

We further give a lower bound for $\pi(MESH(2^k \times 2^{r-k}), H_r)$ and $\pi(TORUS(2^k \times 2^{r-k}), H_r)$ and show that

$$w(MESH(2^k \times 2^{r-k}), H_r) \leq \lfloor \tfrac{2 \times 2^k}{3} \rfloor + 2$$
$$\leq \pi(MESH(2^k \times 2^{r-k}), H_r) + 2$$

$$w(TORUS(2^k \times 2^{r-k}), H_r) \leq \lfloor \tfrac{2^k}{3} + \tfrac{2^k}{4} \rfloor + 2$$
$$\leq \pi(TORUS(2^k \times 2^{r-k}), H_r) + 2$$

3 Hypercube connections on linear arrays

Since routing in a linear array is fixed, the RCA problem is reduced to a channel assignment problem. Given a linear array of size $N = 2^r$, we prove that $\lfloor \frac{2N}{3} \rfloor$ channels is the lower bound to realize the hypercube communication by showing that $\pi(ARRAY(N), H_r) \geq \lfloor \frac{2N}{3} \rfloor$. We then develop a channel assignment scheme that uses $\lfloor \frac{2N}{3} \rfloor$ channels for the hypercube communication. This proves that the bound is a tight lower bound and that the channel assignment scheme is optimal.

3.1 A lower bound

Using Lemma 1, we will obtain a lower bound by proving that there exists a link in the linear array that is used $\lfloor \frac{2N}{3} \rfloor$ times when realizing H_r. The following lemmas establish the bound.

Lemma 2: In a linear array of size $N = 2^r$, where $r \geq 2$, there are 2^{r-1} connections in $DIM_{r-1} \cup DIM_{r-2}$ that use the link $(n, n+1)$ for any specific n satisfying $2^{r-2} \leq n \leq 2^{r-1} - 1$.

Proof: The connections in DIM_{r-1} and DIM_{r-2} can be represented by

$$DIM_{r-1} = \{(i, i + \tfrac{N}{2})|0 \leq i < \tfrac{N}{2}\}$$
$$\cup \{(i, i - \tfrac{N}{2})|\tfrac{N}{2} \leq i < N\}$$

$$DIM_{r-2} = \{(i, i + \tfrac{N}{4})|0 \leq i < \tfrac{N}{4} \text{ or } \tfrac{N}{2} \leq i < \tfrac{3N}{4}\}$$
$$\cup \{(i, i - \tfrac{N}{4})|\tfrac{N}{4} \leq i < \tfrac{N}{2} \text{ or } \tfrac{3N}{4} \leq i < N\}$$

Consider the connections in DIM_{r-1}. All connections $(i, i + \tfrac{N}{2})$ with $0 \leq i \leq n$ use link $(n, n+1)$, where $2^{r-2} \leq n \leq 2^{r-1} - 1$. Hence, as shown in Fig. 2 (a), there are n+1 connections in DIM_{r-1} that use link $(n, n + 1)$. Similarly, in DIM_{r-2}, all connections $(i, i + \tfrac{N}{4})$, where

3

$n < i + \frac{N}{4} < \frac{N}{2}$, use link $(n, n+1)$. As shown in Fig. 2 (b), there are $2^{r-1} - n - 1$ such connections. Hence, there are a total of $n + 1 + 2^{r-1} - n - 1 = 2^{r-1}$ connections in DIM_{r-1} and DIM_{r-2} that use link $(n, n+1)$. □

Lemma 3: In a linear array of size $N = 2^r$, there exists a link $(n, n+1)$ such that at least $\lfloor \frac{2N}{3} \rfloor$ connections in H_r use that link.

Proof: Let $T_i(2^r)$ be the number of connections in H_r that use link $(i, i+1)$ and let $T(2^r) = max_i(T_i(2^r))$. Thus $T(2^0) = 0$, $T(2^1) = 1$. From Lemma 2, we know that for $2^{r-2} \leq n \leq 2^{r-1} - 1$, link $(n, n+1)$ is used 2^{r-1} times by connections in DIM_{r-1} and DIM_{r-2}. That is, the links in the second quarter of the array (from node 2^{r-2} to node $2^{r-1} - 1$) are used 2^{r-1} times by dimension $r - 1$ and dimension $r - 2$ connections. By the definition of hypercube communication, we known that dimension 0 to dimension $r - 3$ connections form a hypercube on this quarter of the array. Thus, Lemma 2 can be recursively applied and the following inequality is obtained.

$$T(2^r) \geq 2^{r-1} + T(2^{r-2})$$

It can be proven by induction that the above inequality and the boundary conditions $T(2^0) = 0$, $T(2^1) = 1$, imply that $T(N) = T(2^r) \geq \lfloor \frac{2N}{3} \rfloor$. Hence, there exists a link which is used at least $\lfloor \frac{2N}{3} \rfloor$ times by connections in H_r. □

The proof of Lemma 3 is constructive in the sense that the link that is used at least $\lfloor \frac{2N}{3} \rfloor$ times can be found. By recursively considering the second quarter of the linear array, we conclude that the source node, n, of the link $(n, n+1)$ that is used at least $\lfloor \frac{2N}{3} \rfloor$ times in H_r is $n = \frac{N}{4} + \frac{N}{16} + \frac{N}{64} + .. = \lfloor \frac{N}{3} \rfloor$. Hence, the link that is used at least $\lfloor \frac{2N}{3} \rfloor$ times in H_r is $(\lfloor \frac{N}{3} \rfloor, \lceil \frac{N}{3} \rceil)$.

Corollary 3.1 Give an array of size $N = 2^r$, if the nodes in the array are partitioned into 2 sets $S_1 = \{i | 0 \leq i \leq n\}$ and $S_2 = \{i | n + 1 \leq i \leq N\}$, where $n = \lfloor \frac{N}{3} \rfloor$, then there are at least $\lfloor \frac{2N}{3} \rfloor$ connections in H_r from S_1 to S_2 and $\lfloor \frac{2N}{3} \rfloor$ connections from S_2 to S_1. □

Theorem 1: $\pi(ARRAY(N), H_r) \geq \lfloor \frac{2N}{3} \rfloor$.

Proof: Directly from Lemma 3. □

3.2 An optimal channel assignment scheme

By the definition of hypercube communication, connections in H_r can be partitioned into three sets, DIM_0, $EVEN_r$ and ODD_r. DIM_0 contains the dimension 0 connections, $EVEN_r$ contains connections between nodes with even node numbers, and ODD_r contains connections

(a) DIM$_0$

(b) DIM$_1$

Channel 1

Channel 2

(c) realizing DIM$_0$ and DIM$_1$ using 2 channels
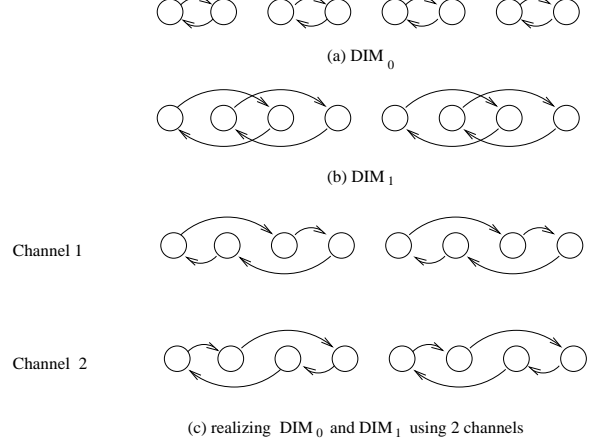
Figure 3. Realizing $DIM_0 \cup DIM_1$ of H_3

between nodes with odd node numbers. Each of $EVEN_r$ and ODD_r forms a $r - 1$ dimensional hypercube communication, H_{r-1}, if only the nodes involved in communications are considered and that the nodes are renumbered accordingly. Thus, channel assignment schemes for H_{r-1} can be extended to realize H_r as shown in the following lemma.
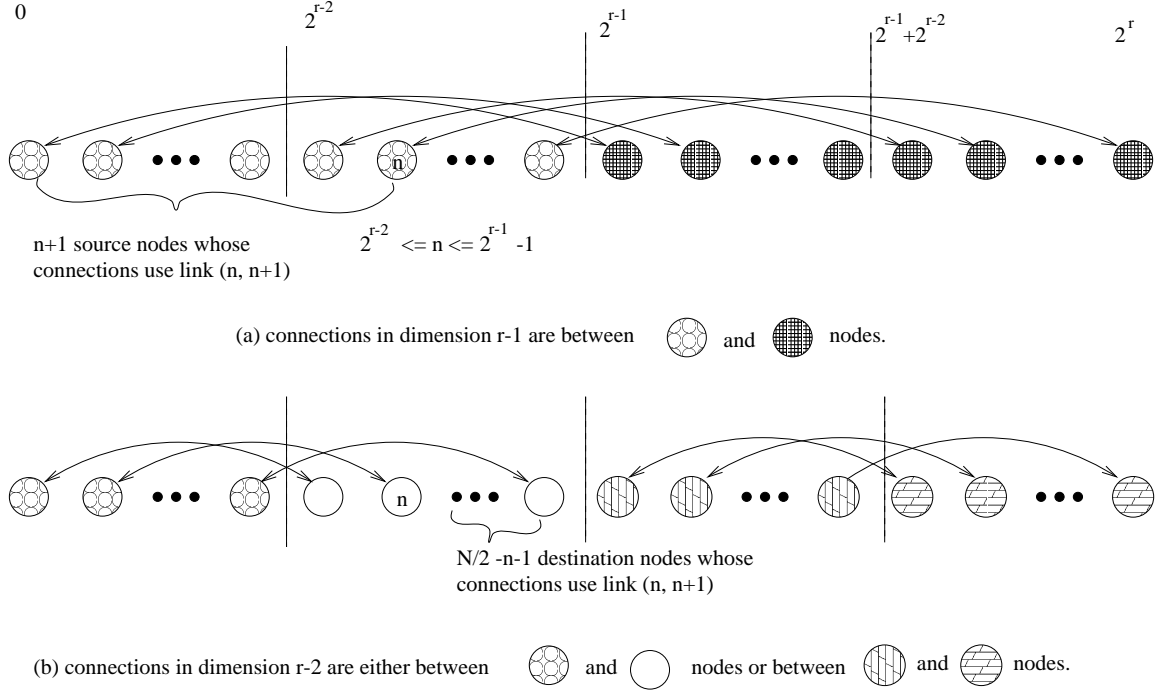
Lemma 4: Assuming that H_{r-1} can be realized on an array of size 2^{r-1} using K channels, then H_r can be realized on an array of size 2^r using $2K + 1$ channels.

Proof: $H_r = EVEN_r \cup ODD_r \cup DIM_0$. From the above discussion and the assumption, $EVEN_r$ and ODD_r are H_{r-1} (when nodes are properly renumbered), K channels can be used to realize $EVEN_r$ or ODD_r. Since it can easily be proven that DIM_0 can be realized with one channel, a total of $2K + 1$ channels can be used to realize H_r. □

Let $D(N)$ be the number of channels needed for H_r on an array of size $N = 2^r$. If we use a channel assignment scheme that is in accordance with the proof of Lemma 4, we can obtain the equation, $D(N) = 2D(N/2) + 1$. Given that no channel is needed to realize hypercube communication on a 1–node array, D(1) = 0. Solving for $D(N)$ results in $D(N) = N - 1$, which is not optimal. The following lemma improves this simple channel assignment scheme.

Lemma 5: Assuming that H_{r-2} can be realized on an array of size 2^{r-2} using K channels, then H_r can be realized on an array of size 2^r using $4K + 2$ channels.

Proof: Consider H_r without dimension 0 and dimension 1 connections. By the definition of H_r,
$H_r - (DIM_0 \cup DIM_1) = DIM_2 \cup ... \cup DIM_{r-1}$ forms four hypercube patterns, each being an H_{r-2} pattern on nodes $\{n \mid n \bmod 4 = i\}$ (with proper node renumbering), denoted by $subarray_i$, for $i = 0, 1, 2$ or 3. From the hypothesis, H_{r-2} can be realized on an array of size

4

(a) connections in dimension r-1 are between ⊛ and ▦ nodes.

(b) connections in dimension r-2 are either between ⊛ and ○ nodes or between ▦ and ⬓ nodes.

Figure 2. Dimension $r-1$ and $r-2$ connections

Algorithm 1: Assign_array($N = 2^r$)
(1)   If $(r = 0)$ then return ϕ
(2)   If (r is odd) then
(3)   /* applying Lemma 4 */
(4)   recursively apply Assign_array($N/2 = 2^{r-1}$) for $EVEN_r$.
(5)   recursively apply Assign_array($N/2 = 2^{r-1}$) for ODD_r.
(6)   assign connections in DIM_0 to one channel.
(7)   Else /* r is even, apply Lemma 5 */
(8)   For i = 0, 1, 2, 3
(9)   apply Assign_array($N/4 = 2^{r-2}$) for $subarray_i$.
(10)   assign connections in $DIM_0 \cup DIM_1$ to 2 channels.

Figure 4. The channel assignment algorithm
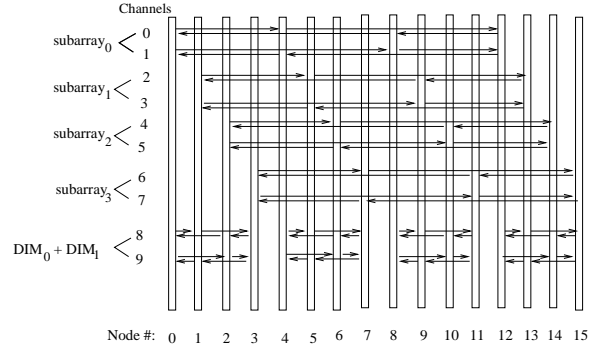
Figure 5. Optimal channel assignment for H_4

2^{r-2} using K channels. The four sub–cube patterns can be realized in $4K$ channels. The remaining connections to be considered are those in DIM_0 and DIM_1. It can easily be proven that connections in DIM_0 and DIM_1 can be assigned to 2 channels as shown in Fig. 3. Hence, the hypercube communication H_r can be realized using a total of $4K + 2$ channels. □

The channel assignment algorithm, *Algorithm 1*, is depicted in Fig. 4. For the base case, when $N = 2^0 = 1$, the hypercube pattern contains no connection. To assign channels to connections in an array of size $N = 2^r$, $r > 0$, there are two cases. If r is even, then Lemma 5 is applied to use $4K + 2$ channels for the hypercube pattern, where K is the

number of channels needed to realize a hypercube pattern on an array of size $2^{r-2} = N/4$. If r is odd, Lemma 4 is applied to use $2K + 1$ channels to realize the hypercube pattern, where K is the number of channels needed to realize a hypercube pattern in an array of size $2^{r-1} = N/2$. The example of using this algorithm to schedule H_4 in an array of size 16 is shown in Fig. 5.

Theorem 2:   *Algorithm 1* uses $\lfloor \frac{2N}{3} \rfloor$ channels for H_r on a linear array with $N = 2^r$ nodes, thus $w(ARRAY(N), H_r) \leq \lfloor \frac{2N}{3} \rfloor$.

Proof: Let $D_{odd}(2^r)$ and $D_{even}(2^r)$ denote the number of channels needed when r is odd and even, respectively. The

number of channels for the hypercube pattern using *Algorithm 1* can be formulated as follows,

$$D_{odd}(2^r) = 2D_{even}(2^{r-1}) + 1, \text{ when } r \text{ is odd.}$$

$$D_{even}(2^r) = 4D_{even}(2^{r-2}) + 2, \text{ when } r \text{ is even.}$$

Using the boundary condition $D_{even}(1) = D_{even}(2^0) = 0$, it can be proven by induction that $D_{odd}(N) = \frac{2N}{3} - \frac{1}{3}$ and $D_{even}(N) = \frac{2N}{3} - \frac{2}{3}$. Hence, $D_{odd}(N)$ and $D_{even}(N)$ are equal to $\lfloor \frac{2N}{3} \rfloor$. $w(ARRAY(N), H_r) \le \lfloor \frac{2N}{3} \rfloor$. $\square$

Theorem 3:
$w(ARRAY(N), H_r) = \pi(ARRAY(N), H_r) = \lfloor \frac{2N}{3} \rfloor$, and Algorithm 1 is optimal.

Proof: Straight forward from Theorem 1, Theorem 2 and Lemma 1. $\square$

4 Hypercube connections on rings

By having links between node 0 and node $N - 1$, two paths can be established from any node to any other node on a ring. It has been shown [1] that even for a fixed routing, general optimal channel assignment problem is NP–complete. In this section, we focus on the specific problem of optimal RCA for H_r on ring topologies. We obtain a lower bound on the number of channels needed to realize H_r and develop optimal routing and channel assignment that achieves this lower bound and is, thus, optimal.

Lemma 6: $\pi(RING(N), H_r) \ge \lfloor \frac{N}{3} + \frac{N}{4} \rfloor$.

Proof: This lemma is proven by showing that there exist two cuts on a ring that partition the ring into two sets, S_1 and S_2, such that $2 \times \lfloor \frac{N}{3} + \frac{N}{4} \rfloor$ connections in H_r originate at nodes in S_1 and terminate at nodes in S_2. Since there are only 2 links connecting S_1 to S_2, one of the 2 links must be used at least $\lfloor \frac{N}{3} + \frac{N}{4} \rfloor$ times, regardless of which routing scheme is used. Consider H_r on a ring of size $N = 2^r$. The connections in $DIM_0 \cup ...DIM_{r-2}$ form two $r - 1$ dimensional hypercube patterns in two *arrays* of size 2^{r-1}. The first array, denoted by $subarray_1$, contains nodes 0, .., $2^{r-1} - 1$ and the second array, denoted by $subarray_2$, contains nodes $2^{r-1},..., 2^r - 1$. From Corollary 3.1, we know that there exists a link in each 2^{r-1} node array such that $\lfloor \frac{N}{3} \rfloor$ connections in the hypercube pattern use that link in each direction. From the discussion in previous section, the link is $(\lfloor \frac{N}{3} \rfloor, \lceil \frac{N}{3} \rceil)$ in $subarray_1$ and $(\lfloor \frac{N}{3} \rfloor + 2^{r-1}, \lceil \frac{N}{3} \rceil + 2^{r-1})$ in $subarray_2$. These two links partition the ring into two sets $S_1 = \{i | 0 \le i \le \lfloor \frac{N}{3} \rfloor\} \cup \{i | 2^{r-1} + \lfloor \frac{N}{3} \rfloor + 1 \le i \le 2^r - 1\}$ and $S_2 = \{i | \lfloor \frac{N}{3} \rfloor + 1 \le i \le 2^{r-1} + \lfloor \frac{N}{3} \rfloor\}$. Hence, there are $\lfloor \frac{N}{3} \rfloor$ connections from $S_1 \cap subarray_1$ to $S_2 \cap subarray_1$ and $\lfloor \frac{N}{3} \rfloor$ connections from $S_1 \cap subarray_2$
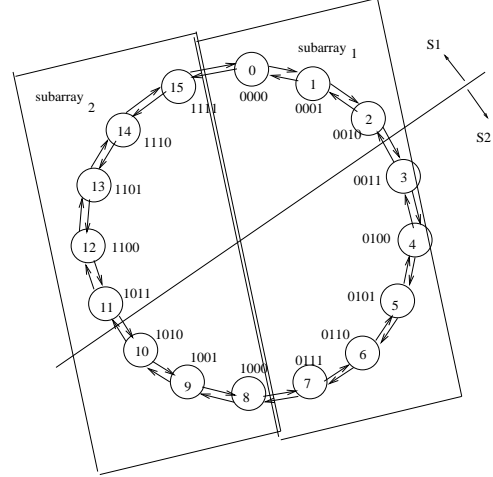

Figure 6. Hypercube on a ring

to $S_2 \cap subarray_2$ in $DIM_0 \cup ...DIM_{r-2}$. Thus, there are $2 \times \lfloor \frac{N}{3} \rfloor$ connections in $DIM_0 \cup .. \cup DIM_{r-2}$ originating at nodes in S_1 and terminating at nodes in S_2. Fig. 6 shows the cuts on a 16–node ring. The remaining connections of H_r are in DIM_{r-1}. By partitioning the ring into S_1 and S_2, each nodes in S_1 has a dimension $r - 1$ connection to a node in S_2. Hence, there are $N/2$ connections in DIM_{r-1} between S_1 and S_2. Therefore, a total of $2 \times \lfloor \frac{N}{3} \rfloor + N/2 = 2 \times \lfloor \frac{N}{3} + \frac{N}{4} \rfloor$ connections in H_r are from S_1 to S_2. Thus, $\pi(RING(N), H_r) \ge \lfloor \frac{N}{3} + \frac{N}{4} \rfloor$. $\square$

Our RCA scheme uses an odd–even shortest path routing. Given a ring of size $N = 2^r$, an odd–even shortest path routing works as follows. A connection between two nodes is established using a shortest path. Connections that have two shortest paths are of the forms $(i, i + 2^{r-1})$ and $(i, i - 2^{r-1})$. For these connections, the clockwise path is used if i is even and the counter–clockwise path if i is odd.

The channel assignment algorithm is derived from Lemma 6. There are two parts in the algorithm, channel assignment for connections in DIM_{r-1} and channel assignment for connections in $DIM_0 \cup .. \cup DIM_{r-2}$. Channel assignment for connections in $DIM_0 \cup .. \cup DIM_{r-2}$ is equivalent to channel assignment for two H_{r-1} in two disjoint arrays, thus, using the channel assignment scheme (for array) described in the previous section, $\lfloor \frac{N}{3} \rfloor$ channels can be used to realize these connections. For the connections in DIM_{r-1}, using odd–even shortest path routing, four connections in DIM_{r-1}, $(i, i + 2^{r-1})$, $(i + 2^{r-1}, i)$, $(i + 1, i + 2^{r-1} + 1)$, $(i + 2^{r-1} + 1, i + 1)$, can be realized using one channel. We denote by $CONFIG_i$ these four connections. Since the union of all $CONFIG_i$, where $i = 0, 2, 4, ..., N/2 - 2$ is equal to DIM_{r-1}, $N/4$ channels are sufficient to realize DIM_{r-1}. Fig. 7 shows the channel

Algorithm 2: Assign_ring($N = 2^r$)

(1)   Apply Assign_array($N/2 = 2^{r-1}$) on $subarray_1$.
(2)   Apply Assign_array($N/2 = 2^{r-1}$) on $subarray_2$.
      Since $subarray_1$ and $subarray_2$ are disjoint,
      channels can be reused in steps (1) and (2).
(3)   for i = 0, N/2-2, step 2
         Assign a channel to connections $(i, i + 2^{r-1})$, $(i + 2^{r-1}, i)$,
         $(i + 1, i + 2^{r-1} + 1)$ and $(i + 2^{r-1} + 1, i + 1)$

Figure 7. The channel assignment for rings

assignment algorithm for ring topologies.

Theorem 4: *Algorithm 2* uses $\lfloor \frac{N}{3} + \frac{N}{4} \rfloor$ channels to realize H_r in a ring of size $N = 2^r$.

Proof: Straight forward from above discussion. □

Theorem 5: $w(RING(N), H_r) = \pi(RING(N), H_r) = \lfloor \frac{N}{3} + \frac{N}{4} \rfloor$, and the odd–even shortest path routing with Algorithm 2 is an optimal RCA scheme for hypercube connection on rings.

Proof: Straight forward from Lemma 1, Lemma 6 and Theorem 4.□

5   Hypercube connections on meshes

Given a $2^k \times 2^{r-k}$ mesh, realizing the hypercube connections on the mesh is equivalent to realizing H_k in each row and H_{r-k} in each column. The following lemma gives the lower bound on the number of channels required to realize hypercube communication patterns on meshes.

Lemma 7: $\pi(MESH(2^k \times 2^r - k), H_r) \geq \lfloor \frac{2 \times 2^k}{3} \rfloor$, assuming $k \geq r - k$.

Proof: The hypercube pattern on the mesh contains 2^{r-k} k–dimensional hypercube patterns on 2^k arrays in the 2^{r-k} rows. Consider a cut in edges $(\lfloor \frac{2^k}{3} \rfloor, \lceil \frac{2^k}{3} \rceil)$ in every row, which partitions the mesh into two parts. From Corollary 3.1, we know that for each row there are $\lfloor \frac{2 \times 2^k}{3} \rfloor$ connections from the left of the cut to the right of the cut, hence, there are a total of $2^{r-k} \times \lfloor \frac{2 \times 2^k}{3} \rfloor$ connections crossing the cut. Since there are 2^{r-k} edges in the cut, there exists at least one edge that is used at least $\lfloor \frac{2 \times 2^k}{3} \rfloor$ times. Thus, $\pi(MESH(2^k \times 2^r - k), H_r) \geq \lfloor \frac{2 \times 2^k}{3} \rfloor$. □

Given a mesh of size $2^k \times 2^{r-k}$, we will denote the hypercube communication pattern in each row by H_k^{row} and the hypercube communication pattern in each column by H_{r-k}^{col}. Our RCA scheme uses X–Y shortest path routing.
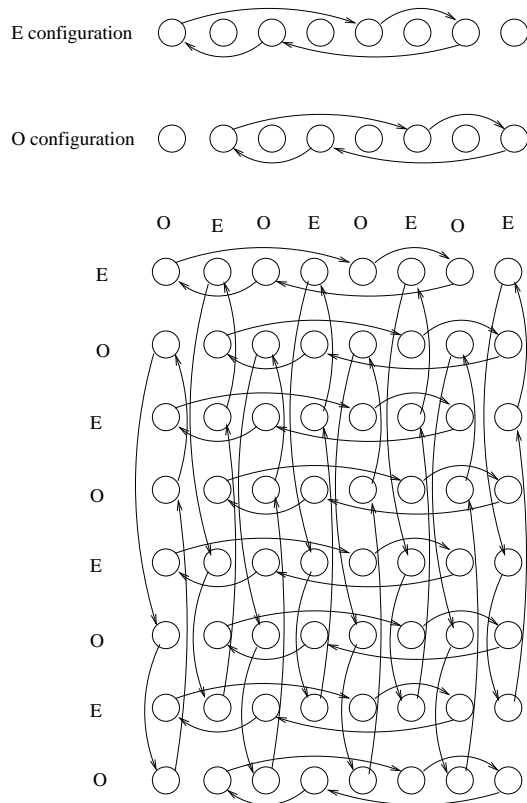

Figure 8. a Mesh configuration

Since we already know the optimal channel assignment for H_k^{row} and H_{r-k}^{col}, the challenge here is to reuse channels on connections in two dimensions efficiently. Let us define an *array configuration* as the set of connections in a linear array that are assigned to the same channel. *Ring*, *mesh* and *torus configurations* are defined similarly. Using the definition of configurations, a mesh configuration can be obtained by combining array configurations in the rows and the columns. For example, if an array configurations in x dimension and an array configuration in y dimension can be combined into a mesh configuration, the two array configurations can be realized in the mesh topology using one channel. Notice that, while there is no link conflict when assigning channels to row and column connections, node conflicts may occur and must be avoided.

Let us first take a deeper look at the array configurations for arrays of size $N = 2^k$. Following the channel assignment algorithm, *Algorithm 1*, array configurations can be classified into three categories; E configurations that contain only connections between even–numbered nodes, O configurations that contain only connections between odd–numbered nodes, and EO configurations that contain dimension 0 (and/or) dimension 1 connections As discussed in Section 3, if k is odd, there is only one EO configuration

for connections in DIM_0, $(\lfloor \frac{2N}{3} \rfloor - 1)/2$ E configurations for connections in $EVEN_k$, and $(\lfloor \frac{2N}{3} \rfloor - 1)/2$ O configurations for connections in ODD_k. Similarly, if k is even, there are two EO configurations, $(\lfloor \frac{2N}{3} \rfloor - 2)/2$ E configurations and $(\lfloor \frac{2N}{3} \rfloor - 2)/2$ O configurations. The following lemma shows that E and O configurations in the rows and columns of the mesh can be combined.

Lemma 8: Given an E configuration, E_x, and an O configuration, O_x, in the x direction and an E configuration, E_y, and an O configuration, O_y, in the y direction, E_x and O_x in all rows and E_y and O_y in all columns can be realized in two mesh configurations.

Proof: The proof is by constructing the two mesh configurations. In the first mesh configuration, let all odd numbered rows realize O_x and all even numbered row realize E_x. In this case, no connection starts or terminates at an odd numbered node in an even column or at an even numbered node in an odd column. Thus, in the same mesh configuration, E_y can be realized in odd columns and O_y can be realized in even columns. The second mesh configuration realizes E_x on odd numbered rows, O_x on even numbered rows, E_y on even numbered columns and O_y on odd numbered columns. These two mesh configurations realize E_x and O_x in all rows and E_y and O_y in all columns. Fig. 8 shows the construction of a mesh configuration. $\square$

Lemma 8 lays the foundation for the channel assignment algorithm. Let a be the number of E and O configurations in H_k^{row}, b be the number of EO configurations in H_k^{row}, c be the number of E and O configurations in H_{r-k}^{col}, d be the number of EO configurations in H_{r-k}^{col}. By assumptions, we have $k \geq r - k$, $a \geq c$, $a + b = \lfloor \frac{2 \times 2^k}{3} \rfloor$ and $d \leq 2$. By combining E and O, row and column configurations into mesh configurations, all the E and O configurations in each row and all the E and O configurations in each column can be realized using a mesh configurations. Using an individual mesh configuration for each EO configuration in the rows and the columns, a total of $a + b + d \leq \lfloor \frac{2 \times 2^k}{3} \rfloor + 2$ configurations are sufficient to realize the hypercube connections.

Theorem 4: H_r can be realized on a $2^k \times 2^{r-k}$ mesh, where $k \geq r - k$, using $\lfloor \frac{2 \times 2^k}{3} \rfloor + 2$ channels. $\square$

Corollary 4.1: $w(MESH(2^k \times 2^{r-k}), H_r) \leq \lfloor \frac{2 \times 2^k}{3} \rfloor + 2 \leq \pi(MESH(2^k \times 2^{r-k}), H_r) + 2$. $\square$

6 Hypercube connections on tori

As in the case of realizing H_r on a mesh, H_r can be realized on a $2^k \times 2^{r-k}$ torus by realizing H_k^{row} in each row and H_{r-k}^{col} in each column. The following lemma gives a lower bound on the number of channels required to realize H_r on a torus.

Lemma 9: $\pi(TORUS(2^k \times 2^r - k), H_r) \geq \lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor$, assuming $k \geq r - k$.

Proof: The hypercube pattern on the torus contains 2^{r-k} k–dimensional hypercube patterns on 2^k rings in the 2^k rows. Considered 2 cuts in edges ($\lfloor \frac{2^{k-1}}{3} \rfloor, \lceil \frac{2^{k-1}}{3} \rceil$) and $(\lfloor \frac{2^{k-1}}{3} \rfloor + 2^{k-1}, \lceil \frac{2^{k-1}}{3} \rceil + 2^{k-1})$ in every row which partition the torus into two parts. Following the same reasoning as in the proof of lemma 6, we know that for each row there are $2 \times \lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor$ connections from one part to the other part, hence, there are a total of $2^{r-k} \times 2 \times \lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor$ connections crossing the two parts. Since there are $2 \times 2^{r-k}$ edges in the cut, regardless of the routing scheme used, there exist at least one edge that is used at least $\lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor$ times. Thus, $\pi(TORUS(2^k \times 2^r - k), H_r) \geq \lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor$. $\square$

We use X–Y routing between dimensions and odd–even shortest path routing within each dimension to develop the RCA scheme. Here, we need to consider how to combine ring configurations into torus configurations. As in the case of rings, given a $2^k \times 2^{r-k}$ torus, we partition the connections in H_r into two sets. The first set includes all connections in $DIM_0 \cup .. \cup DIM_{k-2}$ in each row and all connections in $DIM_0 \cup .. \cup DIM_{r-k-2}$ in each column. The second set includes the connections in DIM_{k-1} in each row and the connections in DIM_{r-k-1} in each column. The connections in $DIM_0 \cup .. \cup DIM_{k-2}$ in each row and the connections in $DIM_0 \cup .. \cup DIM_{r-k-2}$ in each column form four hypercube patterns on four disjoint $2^{k-1} \times 2^{r-k-1}$ sub–meshes in the torus. A straight forward extension of the channel assignment scheme in the previous section can be used to assign channels to these connections with at most $\lfloor \frac{2^k}{3} \rfloor + 2$ channels.

To realize the connections in DIM_{k-1} in each row and the connections in DIM_{r-k-1} in each column, we follow the same partitioning for the ring topology discussed in section 4. Specifically, we construct the following configurations in rows and columns respectively

$row_i = \{(i, i + 2^{k-1}), (i + 2^{k-1}, i), (i + 1, i + 1 + 2^{k-1}), (i + 1 + 2^{k-1}, i + 1)\}$

$column_j = \{(j, j + 2^{r-k-1}), (j + 2^{r-k-1}, j), (j + 1, j + 1 + 2^{r-k-1}), (j + 1 + 2^{r-k-1}, j + 1)\}$

DIM_{k-1} is composed of the configurations row_i, for $i = 0, 2, ..., 2^{k-1} - 2$ and DIM_{r-k-1} is composed of the configurations $column_j$ for $j = 0, 2, ..., 2^{r-k-1} - 2$.

Lemma 10 For any i_1, i_2, where $i_1 \neq i_2$, row_{i_1} and row_{i_2} in each row and $column_{i_1}$ and $column_{i_2}$ in each column

8

can be realized in two torus configurations.

Proof: Similar to the proof of Lemma 8, omitted. $\square$

Theorem 5: H_r can be realized on a $2^k \times 2^{r-k}$ torus, where $k \geq r - k$, using $\lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor + 2$ channels.

Proof: As discussed above, $\lfloor \frac{2^k}{3} \rfloor + 2$ channels are sufficient to realize all connections in H_r, except the connections in DIM_{k-1} in each row and DIM_{r-k-1} in each column, by realizing four hypercube communication patterns on the four disjoint sub–meshes. From Lemma 10, configurations row_i, $i = 0, 2, ..., 2^{r-k-1} - 2$ and configurations $column_j$, $j = 0, 2, ..., 2^{r-k-1} - 2$ can be realized in 2^{r-k-2} torus configurations. Since $2^{k-2} - 2^{r-k-2}$ torus configurations can be used to realize row_i, $i = 2^{r-k-1}, 2^{r-k-1} + 2, .., 2^{k-1} - 2$, all the dimension $k - 1$ connections in each row and dimension $r - k - 1$ connections in each column can be realized in 2^{k-2} torus configurations. Hence, H_r can be realized by a total of $\lfloor \frac{2^k}{3} \rfloor + 2 + 2^{k-2} = \lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor + 2$ configurations. $\square$

Corollary 5.1: $w(TORUS(2^k \times 2^{r-k}), H_r) \leq \lfloor \frac{2^k}{3} + \frac{2^k}{4} \rfloor + 2 \leq \pi(TORUS(2^k \times 2^{r-k}), H_r) + 2.$ $\square$

7 Conclusion

In this paper, we studied optimal schemes to realize hypercube connections on mesh–like optical networks. We prove that $\lfloor \frac{2N}{3} \rfloor$ and $\lfloor \frac{N}{3} + \frac{N}{4} \rfloor$ are tight lower bounds of the number of channels needed to realize hypercube connections on linear arrays and rings of size N, respectively. We develop optimal RCA algorithms that achieve these lower bounds. Also, we study the mesh and torus topologies and develop RCA algorithms that use at most 2 more channels than the optimal. Our results can be used to efficiently establish virtual hypercube topologies on optical mesh–like physical networks. They can also be used to realize hypercube communications efficiently for parallel algorithms that involve such communication.

References

[1] B. Beauquier, J. Bermond, L. Gargano, P. Hell, S. Perennes and U. Vaccaro "Graph Problems Arraying from Wavelength–Routing in All–Optical Networks." *Workshop on Optics and Computer Science*, 1997.

[2] C. Chen and s. Banerjee, "A New Model for Optimal Routing and Wavelength Assignment in Wavelength Division Multiplexed Optical Networks," *Proc. IEEE Infocom'96*, pages 164–171, 1996.

[3] I. Chlamtac, A. Ganz and G. Karmi "Lightpath Communications: An Approach to High Bandwidth Optical WAN's," *IEEE Trans. on Communications*, Vol. 40, No. 7, July 1992.

[4] I.Chlamtac, A. Ganz and G. Karmi "Lightnets: Topologies for High–Speed Optical Networks." *Journal of Lightwave Technology*, Vol. 11, No. 5/6, pages 951—961, May/June 1993.

[5] S. Hinrichs, C. Kosak, D.R. O'Hallaron, T. Stricker and R. Take. "An Architecture for Optimal All–to–All Personalized Communication." In *6th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 310-319, June 1994.

[6] S. Lee, A. D. Oh and H.A. Choi "Hypercube Interconnection in TWDM Optical Passive Star Networks", *Proc. of the 2nd International Conference on Massively Parallel Processing Using Optical Interconnections.* San Antonio, Oct. 1995.

[7] F. T. Leighton, "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, Inc., 1992.

[8] C. Qiao and R. Melhem, "Reconfiguration with Time Division Multiplexed MIN's for Multiprocessor Communications." *IEEE Trans. on Parallel and Distributed Systems*, Vol. 5, No. 4, April 1994.

[9] C. Qiao and Y. Mei, "On the Multiplexing Degree Required to Embed Permutations in a Class of Networks with Direct Interconnects." *IEEE Symposium on High Performance Computer Architecture (HPCA-2)*, pp. 118–129, Feb. 1996.

[10] R. Ramaswami and K. Sivarajan, "Optimal Routing and Wavelength Assignment in All–Optical Networks." *IEEE INFOCOM'94*, pp 970–979, 1994.

[11] X. Yuan, R. Melhem and R. Gupta "Compiled Communication for All–optical TDM Networks", *Supercomputing'96*, Pittsburgh, PA, Nov. 1996.

[12] X. Yuan, R. Melhem and R. Gupta "Performance of Multihop Communication Using Logical Topologies on Torus Networks." Tech. Report, Department of Computer Science, University of Pittsburgh, 1998.

[13] Z. Zhang and A. Acampora, "A Heuristic Wavelength Assignment Algorithm for Multihop WDM Networks with Wavelength Routing and Wavelength Reuse." *Proc. IEEE Infocom'94*, pp 534-543, June 1994.