Recursive Cube of Rings: A New Topology for Interconnection Networks

Yuzhong Sun, Paul Y.S. Cheung, *Senior Member*, *IEEE*, and Xiaola Lin, *Member*, *IEEE*

Abstract—In this paper, we introduce a family of scalable interconnection network topologies, named Recursive Cube of Rings (RCR), which are recursively constructed by adding ring edges to a cube. RCRs possess many desirable topological properties in building scalable parallel machines, such as fixed degree, small diameter, wide bisection width, symmetry, fault tolerance, etc. We first examine the topological properties of RCRs. We then present and analyze a general deadlock-free routing algorithm for RCRs. Using a complete binary tree embedded into an RCR with expansion-cost approximating to one, an efficient broadcast routing algorithm on RCRs is proposed. The upper bound of the number of message passing steps in one broadcast operation on a general RCR is also derived.

Index Terms—Scalable computer systems, recursive cube of rings (RCR), plane property, embedding, message routing.

1 Introduction

A computer system is scalable if it can scale up its resources to accommodate ever-increasing performance and functionality demand. In a parallel computer system with distributed-memory architecture, the design of the interconnection network topology is critical to the performance and scalability of the system. A general scalable network topology should match as closely as possible to the general communication patterns of various practical parallel applications to achieve low network latency and high throughput.

To satisfy the scalability requirement for interconnection networks, it is desirable that an interconnection network has a fixed degree, small diameter, wide bisection width, symmetric nodes, and fault tolerance. In most existing interconnection networks, these requirements are often in conflict with each other. For example, although an $N \times N$ mesh and torus have fixed degree, their diameters are $2N$ and N, respectively (hence, relatively large). The node degree of an n-cube (hypercube) increases logarithmically with the size of the network though the diameter of hypercube is small.

Recently, many new topologies have been proposed. Taking the product of two classical topologies is a prospective method of constructing new interconnection networks [1]. Construction of such a product network requires first choosing a base reference, such as de Bruijn networks [2], shuffle-exchange networks [3], and complete binary trees [4]. The base elements may be different [6], [8]. The cross product of interconnection networks outperforms traditional topologies such as mesh and hypercube in diameter, degree, and matching size [5]. Linear recursive networks are networks that are produced by a linear recurrence of the form:

$$X_n = a_1.X_{n-1} + a_2.X_{n-2} + \cdots + a_k.X_{n-k}$$

where $a_i, 1 \leq i \leq k$, are nonnegative integers and $a_k \neq 0$ [11]. In each recurrence, the subscript n corresponds to the dimension of the network X_n, while the parameter a_i indicates the number of occurrences of a lower dimensional network X_{n-i} within the n-dimensional network. The degree of linear recursive networks increases logarithmically with the network scale. Considering the increasing difficulty in layout and packaging, if the degree of a network expands with the processor size [5], the benefit of scalability from taking the product of interconnection networks may be greatly diminished in practical applications. A network with fixed node degree is therefore greatly desirable. A cube of rings (COR) network is a new proposed network that offers a balance between scalability and hardware overhead [10]. A cube of ring network is constructed to replace each node of a hypercube with a ring of the same size. It differs from cube-connected-cycle in the way of determining the cube neighbors of each node. The cube of rings has a fixed node degree and small diameter but, as will be shown later, the network size that may be chosen is very limited.

In this paper, we propose a new family of interconnection networks, named *recursive cube of rings* (RCR) network. An RCR is constructed by recursive expansion on a given generation seed (GS). A GS for an RCR consists of a number of rings interconnected in a cube-like fashion. It can be created according to certain criteria such as the desirable size of the network. RCRs possess many desirable topological properties in building scalable parallel machines, such as fixed degree, small diameter, high bisection width, and symmetry. Ring, hypercube, and cube-connected cycles are special forms of the RCRs. In addition, we show that RCR

• *The authors are with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: {ysun, cheung, xlin}@eee.hku.hk.*

Manuscript received 4 May 1998; revised 25 Feb. 1999; accepted 22 Apr. 1999.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 106804.

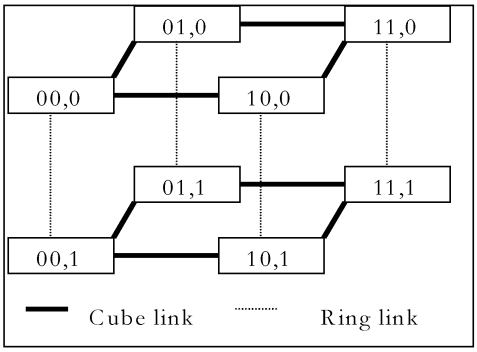

Fig. 1. Generation seed GS(2, 2) for RCR.

possess *plane* property, in which each node of an RCR network is located on at least one *cube plane*. All cube planes are connected by *ring planes*. This property may greatly simplify the routing algorithm design and improve the embedding capability. For example, we may use the symmetry and plane property of RCR to easily implement a broadcast algorithm. Using a complete binary tree embedded in an RCR with expansion-cost approximating one, we develop an efficient broadcast routing algorithm with very low upper bound of the number of message passing. A general deadlock-free routing algorithm for the RCR is also presented and analyzed.

The paper is organized as follows: We first describe the proposed RCR topology and its recursive generation method in Section 2. In Sections 3, we examine the topological properties of RCR. The implementation of broadcast operations based on a complete binary tree, as well as a general message routing algorithm, are presented and analyzed in Section 4, followed by a conclusion in Section 5.

2 Recursive Cube of Rings (RCR)

A general RCR consists of a number of rings interconnected by some links, called *cube links*. The nodes within a ring are connected by links called *ring links*. An RCR is denoted by RCR(k, r, j), where k is the dimension of the cube, r is the number of nodes on a ring, and j is the number of the expansions from the generation seed.

A function f similar to modulo is defined for the representation of node addresses and analysis of RCRs properties. The definition of f is different from the modulo in the case of $0 \le a \le b$, which is defined as follows:

$$f(a,b) = \begin{cases} b-a, & 0 \le a \le b \\ a - \left\lfloor \frac{a}{b} \right\rfloor \cdot b, & a \le b \end{cases}, \text{where } a,b,c \in I.$$

The address of a node in an RCR is specified as $[a_{m-1}a_{m-2}\cdots a_0, b]$, where $m = k + j$; a_i is a binary bit, $0 \le i < m$, and $0 \le b < r$. A node with address $[a_{m-1}a_{m-2}\cdots a_0, b]$ has k cube neighbors with addresses $[a_{m-1}a_{m-2}\cdots a_{f(b\times j+x,k+j)}\cdots a_0, b]$, $1 \le x \le k$, and two ring neighbors with addresses $[a_{m-1}a_{m-2}\cdots a_0, f(b+1,r)]$ and $[a_{m-1}a_{m-2}\cdots a_0, f(b-1,r)]$.

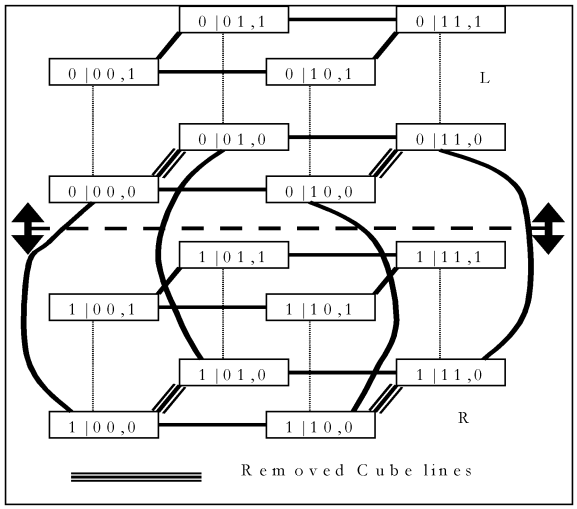

Fig. 2. RCR$(2,2,1)$ after one expansion from GS$(2,2)$.

Given two parameters, an integer k and the number of nodes in the network, N, the generation seed GS(k, r) for RCR(k, r, j) is created as follows:

1. The GS(k, r) should have 2^k rings of r nodes. Each node has the address $[a_{k-1}a_{k-2}\cdots a_0, b]$, where $a_i \in \{0,1\}$, $0 \le i \le k-1$, $b \in \{0,1,\cdots,r-1\}$;
2. A node with address $[a_{k-1,}\cdots,a_0,b]$ has k cube neighbors with addresses $[a_{k-1,}\cdots,\bar{a}_0,b]$, $\ldots$, $[a_{k-1,}\cdots,\bar{a}_{f(i,k)},\cdots,a_0,b]$, $1 \le i \le k$. It has two ring nodes with addresses: $[a_{k-1,}\cdots,a_0,f(b+1,r)]$ and $[a_{k-1,}\cdots,a_0,f(b-1,r)]$.

In terms of the given k and N, the parameters r and j can be determined as follows. Then j expansions are conducted to obtain the desired network RCR(k, r, j).

Let N' be the desirable number of nodes in the network, and N the number of nodes in the generated RCR network. We may select a desired value of r, which in turn determines the value of j, such that N is closest to N' according to the following equation:

$$r = \begin{cases} \left\lceil \frac{N'}{2^{k+j}} \right\rceil & N > N' \\ \left\lfloor \frac{N'}{2^{k+j}} \right\rfloor & N \le N'. \end{cases} \quad (2.1)$$

Fig. 1 depicts a generation seed GS$(2,2)$. Fig. 2 shows an RCR$(2,2,1)$ obtained by one expansion from generation seed GS$(2,2)$, and Fig. 3 shows an RCR$(2,2,2)$ obtained from one more expansion from RCR$(2,2,1)$. At each expansion, the number of nodes is doubled and some new cube links must be added. At the same time, in order to keep the constant node degree, some cube links must be removed. For example, the node $[000,0]$ and node $[010,0]$ in RCR$(2,2,1)$ are mapped, respectively, to node $[0000,0]$ and node $[0010,0]$ in RCR$(2,2,2)$. The cube link ($[000,0]$, $[010,0]$) in the RCR$(2,2,1)$ is removed when it is expended to

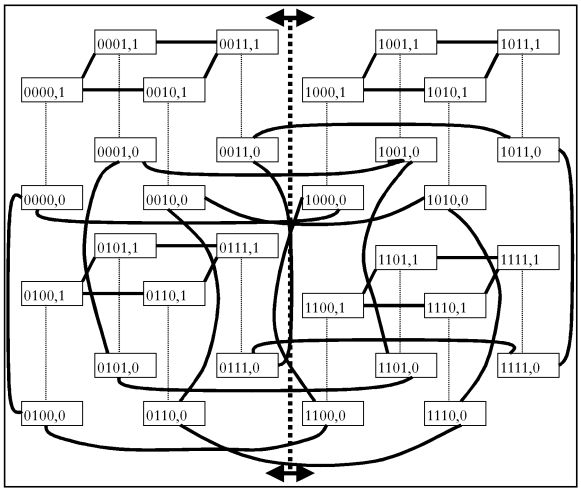

Fig. 3. The topology of RCR$(2,2,2)$.

RCR$(2,2,2)$, so that node $[0000,0]$ and node $[0010,0]$ in the RCR$(2,2,2)$ remain constant degree three when two new cube links are added to these two nodes during the expansion, as shown in Fig. 3. The algorithm for constructing an RCR is described in Fig. 4. It is important to note that, for the general cases, the actual number of nodes N in an RCR network may be different from the desired network size N'. For example, to build a network with 20,000 nodes ($N' = 20,000$), the size of the closest RCR network is 20,480, as will be explained in Section 3.

In the construction algorithm given in Fig. 4, for a node v, the address of the node, $d_v = [a_m a_{m-1} \cdots a_0, b] = [A, b]$ (a_i is a binary bit, $0 \le i \le m$), then $o^1(d_v) = [1a_m a_{m-1} \cdots a_0, b]$ and $o^0(d_v) = [0a_m a_{m-1} \cdots a_0, b]$ are the addresses of the next expansion of node v. In terms of the address, the expansion is done by concatenating a 0-bit or a 1-bit before the address. First, the GS is taken as the left graph G_L and its nodes are duplicated as the right graph G_R. The node set of a new graph G is the union of the node sets of the two graphs G_L and G_R. However, in the new graph G, the cube links at each node are rearranged. For example, in Fig. 2, a node v of GS with address $[a_{k-1} \cdots a_0, b]$ in the new graph has a new set of cube neighbors connected to the node v only by cube links. These neighbors have the addresses $[a_{k-1} \cdots a_{f(1\times b+x,1+k)} \cdots a_0, b]$, where $1 \le x \le k$ and i refers to the current index of expansion number.

3 Topological Properties of RCRs

In this section, we examine the major topological properties of the proposed RCR networks, such as an RCR size, degree, bisection width, diameter, size matching property, plane property, symmetry, and so forth.

3.1 General Topological Properties

An RCR(k,r,j) network is modeled as a graph $G = G(V,E)$, where a vertex in $V(G)$ corresponds to a node in the RCR network, and an edge in $E(G)$ corresponds to a link in the RCR network.

Property 1. *In an RCR(k,r,j), we have the following properties:*

P1.1. *The number of nodes, N, is $r \times 2^{k+j}$.*

P1.2. *All of the nodes in the RCR(k,r,j) have the same degree, and the degree d of the network is $k+2$ for $r > 2$, and $k+r-1$ for $1 \le r \le 2$.*

P1.3. *The number of edges of the network, E, is given as follows:*

$$E = \begin{cases} r \times 2^{k+j} \times (1 + \frac{k}{2}) & r > 2 \\ r \times 2^{k+j} \times (\frac{1}{2} + \frac{k}{2}) & r = 2 \\ r \times 2^{k+j} \times \frac{k}{2} & r = 1. \end{cases}$$

P1.4. *The bisection width of the network, B, is $Num(k,r,j) \times 2^{k+j-1}$ or $Num(k,r,j)/r \times N$, where N is the number of nodes of the network, and $Num(k,r,j)$ is defined as follows: For any integers x and y, $0 \le x \le r-1$ and $1 \le y \le k$, $Num(k,r,j)$ denotes the number of the x values satisfying $f(j \times x + y, j + k) = 1$.*

Proof. The topological properties P1.1, P1.2, and P1.3 follow directly from the GS architecture and the recursive construction algorithm of RCRs in Fig. 4. Therefore, it is only necessary to give the proof of P1.4. Given a node with address $[a_{m-1} \cdots a_0, b]$, $m = k + j$, all its k cube neighbor addresses are different only in one bit position, $\bar{a}_{f(j\times b+i,j+k)}, 1 \le i \le k$. For a given RCR$(k,r,j)$ network, the inverse bit position of a node is determined only by the value $b, b \in \{0,1,\ldots,r-1\}$. According to P1.1, for a value of b, there exist 2^{k+j} nodes with the same inverse bit-wise position. An RCR(k,r,j) network can be divided into two RCR$(k,r,j-1)$ networks. The link contributing to the bisection bandwidth of the RCR(k,r,j) network should have one endpoint of the link on one RCR$(k,r,j-1)$ network and the other endpoint on the other RCR$(k,r,j-1)$ network. According to the construction algorithm, the two nodes of such a link must have the same value b in their addresses. All 2^{k+j} nodes with such value b in their addresses can be divided into two groups, one in each of the lower order networks ($RCR(k,r,j-1)$). Therefore, the value b determines the number of links crossing the two RCR$(k,r,j-1)$ networks. According to the construction algorithm of RCR, the two nodes of the link that contribute to the bisection bandwidth differ in the highest bit position of the first parts in their addresses. That is, the possible value of b should satisfy $f(j \times b + y, j + k) = 1$. □

The following corollary can be derived directly from Property 1.

Corollary 1. *For an RCR(k,r,j) network, the bisection bandwidth B is $2^{k+j-1} B 2^{k+j-1} \times \lfloor j \times (r-1)/(j+k) \rfloor$.*

It is straightforward to verify the following property.

Input: design parameters, k and N', where N' is the desired number of nodes of the network to be constructed;
Output: a network RCR(k,r,j), the number of nodes of RCR(k,r,j) is N.

Procedure:

1. Select parameters, k, r, j, such that N is closest to N' and construct GS(k,r).
2. /* initialize G_L and G_R */

   Set $G_L(V,E)$ = GS(k,r), and $G_R(V,E)$ = GS(k,r).
3. For i = 1 to j, do the following:

- /* expand the network by concatenating a 0-bit to the most significant bit of the node addresses of G_L */

  For any node v in G_L, expands its address $d_v = o^0(d_v)$;
- /* generate the next higher-order RCR, G, by the union of G_L and G_R */

  Set $G = G_L \cup G_R$;
- /* complete the generation of the next higher-order RCR by the appropriate link insertion and deletion */

  Insert and delete the appropriate links in G, such that each node of G_L or G_R with address $[a_m a_{m-1} \cdots a_0, b]$ has its cube neighbors, $m = k + j - 1$, $[a_m \cdots \overline{a}_{f(i\times b+1,i+k)} \cdots a_0, b]$, ... , $[a_m \cdots \overline{a}_{f(i\times b+x,i+k)} \cdots a_0, b]$, ... , $[a_m \cdots \overline{a}_{f(i\times b+k,i+k)} \cdots a_0, b]$, where $1 \le x \le k$. The ring neighboring relation of each node remains unchanged.
- Set $G_L = G$ and $G_R = G$.

4. Set RCR(k,r,j) = G.

Fig. 4. Algorithm to construct an RCR.

Property 2. *The following topologies are special forms of RCRs:*

P2.1. Ring networks are RCR$(0, r, j)$ for any r and j.
P2.2. An n-dimensional hypercube is RCR$(2, 1, j)$.

In Table 1, we compare the node degree, diameter, and bisection width of RCRs with some popular topologies such as n-cube, n-dimensional cube-connected cycles (CCC), and CORs. It shows that RCRs have constant degree and relatively short diameter. It also shows that RCRs have larger bisection width than the other topologies except hypercube, but hypercube does not have constant degree.

The actual size N of a network with a given topology may be different from an arbitrary size N' of the desired network. The ratio N/N' is called the match ratio. When the match ratio is closer to one, we say that it has better size matching. The sizes for an n-cube, n-dimensional CCC, and COR, and RCRs are 2^n, $n2^n$, $r2^{kr}$, and $r2^{k+j}$, respectively. Compared with the other three topologies, apparently, the RCRs can better match a given size by selecting proper parameters, r, k, and j. For example, for $N' = 20,000$, the size of an n-cube is 16,384 for $n = 14$ $(N/N' = 0.82)$ and 32,768 for $n = 15$ $(N/N' = 1.64)$. By selecting proper r, k, and j, an RCR with size 18,422 $(N/N' = 0.92)$ or 20,480 $(N/N' = 1.02)$ can be constructed. It is easy to check that RCRs also have a better match ratio than that of CCC or COR in this example.

3.2 Plane Property

RCRs also possess a special *plane* property such that an RCR(k, r, j) can be taken as the combination of two different types of planes, *cube-plane* and *ring-plane*, to be defined below. This property can be used to develop efficient routing algorithms.

Definition 1. *A* cube plane *(CP) is a subgraph of an* RCR(k, r, j) *such that the CP is connected and all links of the CP are cube links. In a CP, node* $[A, b]$ *and node* $[A', b']$ *are connected by a cube link if and only if* $|A \oplus A'| = 1$ *and* $b = b'$.

Definition 2. *A* ring plane *(RP) is a subgraph of an* RCR(k, r, j) *such that the RP is connected and all links of*

TABLE 1
Comparisons of Degree, Diameter, and Bisection Width (N Is the Number of Nodes in the Networks)

Topology	Degree	Diameter	Bisection width
RCR(2,r,j) , $r>2$	4	$\log N - \log r + \lceil \frac{r-1}{2} \rceil - 1$	$N/4$
Hypercube	$\log N$	$\log N$	$N/2$
Cube-connected cycles	3	$O(\log N)$	$O(N/\log N)$
Butterfly	4	$O(\log N)$	$O(N/\log N)$
DeBruijn	4	$O(\log N)$	$O(N/\log N)$
Mesh-connected computer	4	$2\sqrt{N}$	$\sqrt{N}$
Torus	4	$\sqrt{N}$	$2\sqrt{N}$
Honeycomb torus	3	$0.81\sqrt{N}$	$2.04\sqrt{N}$

the RP are ring links. An RP consists of r nodes $[A,0]$, $[A,1]$, $\ldots$, $[A,r-1]$, *where A is certain binary number.*

For example, the RCR$(2,2,2)$ has 16 RPs and eight CPs, as shown in Fig. 5. Fig. 5 also illustrates the connection relations among CPs through RPs. For instance, a CP denoted by [- -00, 0] is comprised of the four nodes, $[0000,0]$, $[1000,0]$, $[0100,0]$, and $[1100,0]$. The CP, [- -00, 0], joins the CP, [11- -, 1] by the two nodes $[1100,0]$ and $[1100,1]$. "+1" between the two CPs denotes increase in b from the CP [- -00, 0] to the CP [11- -, 1]. The connections among the eight CPs of an RCR$(2,2,2)$ are shown. Each CP is referred to as a super node. Therefore, we can construct a new graph in which a link between two super nodes exists if and only if two nodes, respectively, from the super nodes are located in the same ring. The constructed graph is also a *contraction* with *4-partition* in graph theory [17]. Such a contraction in graph theory can give rise to an efficient broadcast algorithm, described in detail in Section 4.

Property 3. *For an* RCR(k,r,j), *let $G=G(V,E)$ denote the corresponding undirected graph of the network. The graph G consists of $C_k^2 \times 2^{k+j-2}$ CPs with four nodes. All CPs are connected by RPs.*

Proof. According to the neighboring definition of an RCR(k,r,j), the ring neighbor relations of each node with the other nodes remain in the whole expansion construction of the RCR because each node with the address $[a_{k+j-1}\ldots a_0,b]$ should have two ring neighbors with the addresses $[a_{k+j-1}\ldots a_0, f(b+1,r)]$ and $[a_{k+j-1}\ldots a_0, f(b-1,r)]$ that are independent of the parameter j. Therefore, each node must be located in the same ring plane. In other words, in the whole construction process, all the RPs in the RCR remain unchanged. The ring neighborhood relations between nodes are reserved in the expansions.

On the one hand, the cube relations of one node with other nodes may often change after each expansion according to the construction algorithm. Now, considering the intermediate derived network RCR(k,r,q), where $1 \le q \le j$, let p_0 with the address $[a_{q+k-1}\ldots a_0,b]$ be an arbitrary node in the network RCR(k,r,q). We try to find a ring in a CP starting with node p_0. According to the construction algorithm, the k cube neighbors of p_0 should be

$$[a_{q+k-1}\ldots \bar{a}_{f(q\times b,q+k)}\ldots a_0,b],\ldots,$$
$$[a_{q+k-1}\ldots \bar{a}_{f(q\times b+i,q+k)}\ldots a_0,b],\ldots,$$
$$[a_{q+k-1}\ldots \bar{a}_{f(q\times b+k-1,q+k)}\ldots a_0,b].$$

We repeat the same procedure for each next node until no new node can be found. Then, we find a cycle from the node p_0 to the same node p_0. The cycle forms a CP with length four. On the other hand, any two neighboring nodes should have only one bit different in the first parts of their addresses. Each node should simultaneously be located on the C_k^2 CPs. The number of CPs is $C_k^2 \times r \times 2^{k+j-2}$. Thus, we can conclude that all CPs are connected by RPs. □

3.3 RCR as a Cayley Graph

A network is symmetric if the network topology is the same looking from any node in the network. A symmetric interconnection network may simplify the design of the routers and interfaces, and thus reduce the cost of the networks. Cayley graphs have been proved to be symmetric

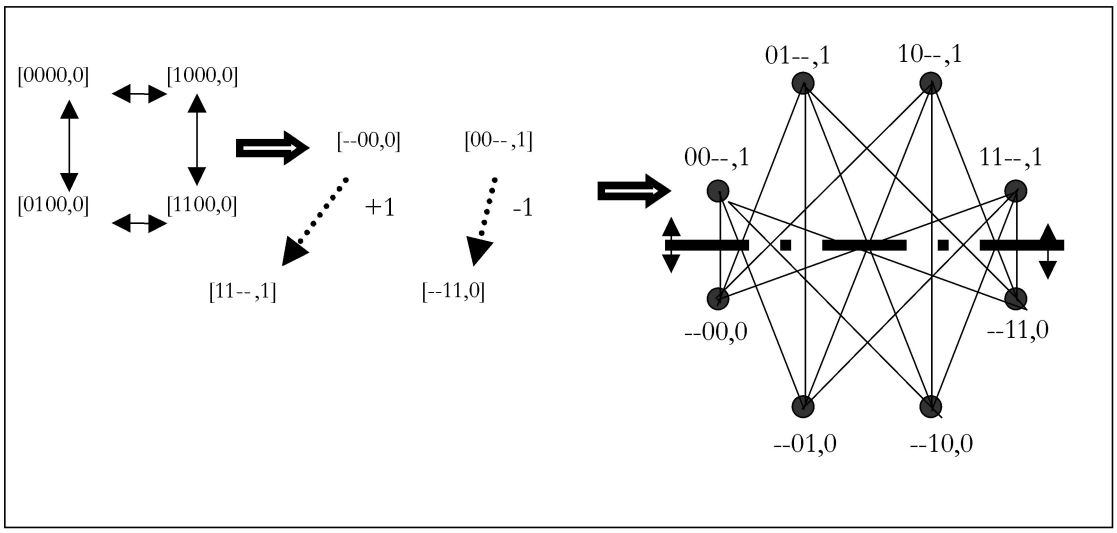

Fig. 5. Plane property of RCR(2,2,2).

graphs [13]. We show that RCRs are Cayley graphs, and therefore they are symmetric. The following definitions are directly from [13].

Definition 3. *A group G consists of a set of elements and an associative binary operator $a \bullet b$. The operator has an identity element i (for all $a \in G, a \bullet i = i \bullet a = a$), exhibits the closure property (for all $a, b \in G, a \bullet b \in G$), and every element a in G has an inverse $a^{-1} \in G (a \bullet a^{-1} = a^{-1} \bullet a = i$, for all $a \in G$).*

Definition 4. *Let G be a finite group with associative operator $\bullet$. A* Cayley set *H is a subset of G such that the identity element of G is not in H, and that if group element $g \in G$ is in H, so is the inverse of g.*

Definition 5. *A* Cayley graph *(G, H) is a graph defined on a finite group G and the associated* Cayley set *H whose nodes are the elements of G and whose edges are the pairs $g \in G$ and $h \in H$. We say the graph (G, H) is the Cayley graph derived from G and H.*

Given a set of generators of a finite group with an identity element such that it is closed under inverses, a Cayley graph can be obtained by taking the elements of the group as vertices and connecting by an edge every pair of elements x and y if, and only if, y is obtained from x by applying one of the group generators [14], [10].

Property 4. *An* RCR(k, r, j) *is a Cayley graph and therefore is symmetric.*

Proof. First, we construct a finite group G and define an associative operator $*$ on it:

$$[A, b] * [X, y] = [A \times C^{k+j} + X, +y],$$

where $[A, b]$ and $[X, y]$ are the two elements of G. Let Q_2^n denote the set of all Boolean n-tuples under bitwise addition modulo 2, and Q_r denote the integer set $\{0, 1, \ldots, r-1\}$ under addition modulo r. The two sets Q_2^n and Q_r are proven to be Cayley graphs [10]. Then, we construct a finite group through ordered pairwise $Q_2^n \times Q_r$, where $n = k + j + 1$. For the associate operator $*$, bit-wise addition modulo 2 is used for the first entry, and addition modulo r is used for the second entry, where C is an Boolean matrix:

$$\begin{pmatrix} 000\cdots\cdots01 \\ 100\cdots\cdots00 \\ 010\cdots\cdots00 \\ \cdots \\ 000\cdots\cdots10 \end{pmatrix}.$$

Second, we construct a Cayley set H using the same method in [10], which consists of nodes

$$[0\ldots0,1], [0\ldots0,r-1], [0\ldots1,0],$$
$$[0\ldots010,1], \ldots, [\underbrace{0\ldots0}_{k-1}10\ldots,0].$$

Note that the inverse element of each element of H also belongs to H. Therefore, H is a Cayley set.

Finally, we can construct the RCR(k, r, j) in the group G based on H and the associative operator. For any node $[A, b]$ of the RCR(k, r, j) and each element from H, we derive one edge to each neighbor of $[A, b]$ in the RCR(k, r, j) as follows:

- $([A,b], [0\cdots0,1] * [A,b]) = ([A,b], [A,b+1])$
- $([A,b], [0\cdots r-1,1] * [A,b]) = ([A,b], [A,b-1])$
- $([A,b], [0\cdots1,1] * [A,b]) = ([A,b], [a_{k+j-1}\cdots\bar{a}_{bj+1}\cdots a_0, b])$
- $([A,b], [0\cdots10,1] * [A,b]) = ([A,b], [a_{k+j-1}\cdots\bar{a}_{bj+2}\cdots a_0, b])$
- $([A,b], [0\cdots1\underbrace{0\cdots0}_{k},1] * [A,b])$
  $= ([A,b], [a_{k+j-1}\cdots\bar{a}_{bj+k}\cdots a_0, b])$.

Therefore, RCR(k, r, j) is a Cayley graph. Then, according to [13], we conclude that the RCR(k, r, j) is symmetric. □

Property 5. *The diameter D of an* RCR(k, r, j) *is less than or equal to* $\lceil \frac{r-1}{2} \rceil + k + j - 1$.

Proof. According the definition of an RCR, the longest shortest path in an RCR is from node [0...0, 0] (node S) to node $\left[1 \cdots 1, \frac{r-1}{2}\right]$ (node D). The distance between the S and D, $dis(S, D)$, is the diameter of the RCR. We first construct a sequence of nodes, $p_0 p_1 \ldots p_{m-1}$, such that $m = k + j$; $p_0 = S$, $p_{m-1} = D$, and $p_i \oplus p_{i-1} = 1$. Let $pi = [Ai, bi]$, $0 \le i \le m-1$, we can then construct a path from p_i to p_{i+1}:

$$[Ai, bi], [Ai, bi+1], \ldots, [Ai, bi+1], [Ai+1, bi+1].$$

Note that $b_0 = 0$ and $bm - 1 = \lceil \frac{r-1}{2} \rceil$. Therefore, we have

$$\begin{aligned} dis(S, D) &= \sum_{i=0}^{m-2} dis(P_i, p_{i+1}) = \sum_{i=0}^{m-2} (b_{i+1} - b_i + 1) \\ &= b_{m-1} - b_0 + m - 1 = \lceil \tfrac{r-1}{2} \rceil + k + j - 1. \end{aligned}$$ □

According to the above analysis, k and r are often determined with respect to various implementation requirements. It is reasonable to assume that the two parameters, k and r, are fixed. We then consider the effect of the parameter j on network size N and diameter D. According to Property 1 and Property 5, it is concluded that $D = \log N - \log r + \lceil \frac{r-1}{2} \rceil - 1$, shown in Table 1.

It is proven that an RCR(k, r, j) network is a regular graph, in which every vertex has a fixed degree. A regular graph of degree d can have fault tolerance at most $d-1$ such that the graph remains connected when at most $d-1$ fault nodes or links are incurred simultaneously. A regular graph of degree d with fault tolerance $d-1$ is said to be optimally fault-tolerant [14]. The following property proves that RCR networks are optimally fault tolerant.

Property 6. *An* RCR(k, r, j) *is optimally fault tolerant.*

Proof. First, we consider the fault tolerance of the GS(k, r). According to definitions of the GS(k, r) in Section 2 and Cube-of-Rings(COR) [10], it is derived that the GS(k, r) is equal to the COR(k, r). COR networks have been proved to be optimally fault tolerant. Therefore, the GS(k, r) is also optimally fault-tolerant.

Second, we consider the expansion graph G based on the GS(k, r). It is noted that G_L or G_R have changed after the expansion, which are denoted by G_L^E and G_R^E in G, respectively. Let $V(Y)$ denote the node set of a graph Y. According to the generation algorithm in Fig. 4, we have $V(G_L) = V(G_L^E)$ and $V(G_R) = V(G_R^E)$. Now, we consider how to expand cube links in G_L or G_R to construct G_L^E and G_R^E. With respect to symmetry of RCRs, we only consider how a changed cube link (a_i, a_j) in G_L is mapped in the new graph G. After the expansion, the cube link (a_i, a_j) is removed in G_L^E, and two new cube links from the two nodes a_i and a_j are introduced in G_L^E, denoted by (a_i, a_j^E) and (a_j, a_j^E), respectively, a_i^E and $a_j^E \in V(G_R^E)$. Note that there is a cube link between a_j, a_j^E in G_R and it is removed in G_R^E. Also, a_i^E and a_j^E are in two rings in G_R^E. There exist other cube links connecting the two rings. We can construct a path from a_i to a_j in G to replace the cube link (a_i, a_j). Each changed cube link in G_L and G_R can be expanded in the same way. Then, we can construct the G_L^E and G_R^E after the expansion. The rings related to each changed cube link in G_L and G_R are exclusive to other changed cube links. That is, G_L^E and G_R^E remain optimally fault-tolerant.

Comparing to G_L, G_L^E replaces a changed cube link (a_i, a_j) in G_L with some cube and ring links in the new graph G while the two endpoints a_i and a_j are mapped to a_i^E and a_j^E in G_L^E, respectively. These cube and ring links in the new graph G whose nodes are in G_R^E (except a_i and a_j) form a path connecting the two nodes a_i^E and a_j^E. The a_i^E and a_j^E in the G_L^E are reduced to the a_i and a_j in the G_L if the highest bit of the A in the address of each of a_i^E and a_j^E is removed. Note that the path joining a_i and a_j is disjointed with the paths containing certain changed cube links in G_L due to expansion. With respect to symmetry, the new graph after each time of expansion remains optimally fault-tolerant. □

An n-star is also a *Cayley* graph [24]. It is an undirected graph consisting of $n!$ nodes. Each node in an n-star graph is assigned a unique label $x_0 x_1 \ldots x_{n-1}$, which is a per–mutation of n symbols $\{0, 1, \ldots, n-1\}$. Each permutation is connected to every other permutation that can be obtained from it by interchanging the first symbol with any of the other symbols. Obviously, the degree of the graph is $n-1$ and the diameter is $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$. n-star graphs become an increasingly attractive alternative to n-cube due to its certain desirable topological features, such as vertex- and edge-symmetries, fault tolerance, small diameter, high bisection bandwidth, good embedding capability, and low degree [23], [24]. The proposed RCRs also possess these features. However, compared with n-star graphs, RCRs are more suitable for the interconnection networks in scalable computer systems. First, an RCR($2, r, j$), $r < 2$, has a constant degree of four, while an n-star graph has variable degree $n-1$. Second, the number of nodes of an RCR is $r \cdot 2^{k+j}$, while the number nodes of an n-star is $n!$. It is easy to see that an RCR may have much better match ratio by selecting proper integer parameters r, k, and j. For example, if we want to build a network with number of nodes $N = 20,000$, we may get an RCR with number of nodes $N = 18,422$ or 20,480, as shown earlier. However, an n-star graph with $n = 7$ contains only 5,040 nodes ($N = 5,040$) but, when $n = 8$, it has 40,320 nodes ($N = 40,320$)! Obviously, the match ratio N/N' of RCRs is much better (closer to 1) than that of n-star graphs. Also, more efficient

Input: current node c with address $[A_c, b_c]$, source node S with address $[A_s, b_s]$ and destination node D with address $[A_d, b_d]$.

Output: an outgoing virtual channel (c,v) used to forward the message towards to D (v is a neighboring node of c).

Procedure:

1. If $A_c = A_d$
   - If $b_a = b_c$, then c is D, forward the message to the local processor, and stop.
   - If $b_a \neq b_c$, then select neighboring node v such that $|b_v - b_d| \leq |b_c - b_d|$.
   - If $b_c < b_v \leq b_d$ (or $b_c < b_v \geq b_d$), then select $vc1$(or $vc2$) as the outgoing channel ($vc1$ and $vc2$ are virtual channels from c to v), and stop.
2. If $A_c \neq A_d$, then use e-cube routing with A_c as the local node address and A_d as the destination address to find A_v. If $[A_v, b_v]$ is a neighboring node of c, then select any outgoing virtual channel from c to v, and stop.
3. Select neighboring node v such that $|b_v - b_d| \leq |b_c - b_d|$.
4. If $b_c < b_v \leq b_d$ (or $b_c < b_v \geq b_d$), then select $vc1$(or $vc2$) as the outgoing channel($vc1$ and $vc2$ are virtual channels from c to v), and stop.

Fig. 6. Routing algorithm for RCR.

collective communication operations such as broadcast on RCRs can be developed by using the special plane property of RCRs, as will be shown in the next section.

4 Message Routing in RCR

Efficient routing algorithms are essential for any interconnection networks. In this section, we present efficient unicast and broadcast message routing algorithms for RCR networks. We will assume that wormhole switching technique is adopted in the RCR networks. Virtual channels will be introduced to avoid the deadlock, as shown in [15], [16].

4.1 Unicast Communication

The basic idea of the routing algorithm is similar to the well-known e-cube routing algorithm for binary cubes [15]. It is proven that each node of an RCR(k, r, j) is on certain CPs. In the case of $k = 2$, each node is located only on a single CP. In one CP, the addresses $[A, b]$s of nodes show a regular change of bit patterns such that the same k bit positions in A of each node differ and the other bits remain the same. An appropriate neighboring cube plane can be chosen in the way similar to the e-cube routing. The exit node of such a chosen CP is the node closest to the destination. We call such a routing algorithm in Fig. 6 a hop-plane routing algorithm. The hop-plane routing algorithm can always find a shortest path from any source node to any destination node in RCRs.

To prevent the occurrences of deadlock, two virtual channels are set up on a physical link [15]. A node $[A, b]$ is assigned an integer number $A \times r + b$. The nodes in the network can then be ordered with the assigned numbers as the keys. One of the two virtual channels, denoted by $vc1$, is used when a message traverses a link in ascending order from one node to another. The other virtual channel denoted by $vc2$ is used when the message traverses a link in descending order, regardless of cube links or ring links. Let (a, c) or $([A_a, b_a], [A_c, b_c])$ denote a link. For a cube link, A_a differs from A_c in one bit while $b_a = b_c$. For a ring link, $|b_a - b_c| = 1$ while $A_a = A_c$. It can be shown that the message routing algorithm shown in Fig. 6 is deadlock-free.

Theorem 1. *For an* RCR(k, r, j), *the message routing algorithm in Fig. 6 can always find a shortest path from the source node to the destination node, and it is deadlock-free.*

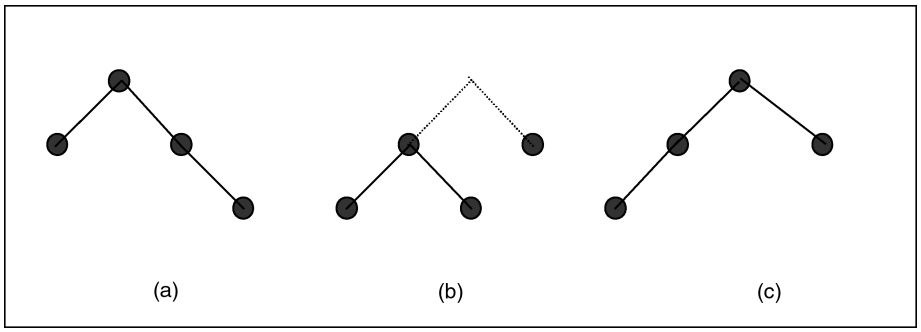

Fig. 7. Three subsets with four nodes.

Proof. From the proof of Property 5, there is a shortest path between any two nodes. It is easy to verify that the routing algorithm can always find the shortest path in terms of the definition of RCRs. According to the routing algorithm in Fig. 6, two virtual channels denoted by $vc1$ and $vc2$ are incident to each physical link. With respect to the order of the endpoints of a link, a message traverses the link along different virtual channels. In case of ascending order, the message travels along $vc1$. Otherwise, the message trips along $vc2$. Assume that a deadlock incurred among a sequence of messages. According to deadlock conditions for adaptive routing algorithms [15], [16], a cyclic dependence graph would be constructed among these messages. Therefore, there would exist certain messages in the sequence, which traverses a $vc1$ ($vc2$) in descending order (ascending order). This situation is not permitted in our algorithm. Therefore, the routing algorithm is deadlock-free. □

4.2 Broadcast Communication

The embedding of a complete binary tree $T(h)$ to an RCR(k, r, j) will be used as a broadcast tree to implement a broadcast in which the same message from a source is sent to all other nodes in the RCR network. A similar idea has been widely used in wormhole-routed 2D-mesh, torus, and hypercubes [18], [20], [21], [9], [19]. We assume that an all-port model is adopted in which each node can simultaneously send (and receive) as many messages as possible to its neighbors in the network [22]. The message passing steps introduced in [18] will be used to measure the temporal cost of a broadcast operation in RCRs. In this subsection, we first show how to embedding a binary tree $T(h)$ into an RCR(k, r, j). We then present a broadcast algorithm based on such an embedding tree.

An embedding, $F : V \rightarrow V'$, of a guest graph, $G' = (V', E')$, into a host graph, $G = (V, E)$, is a one-to-one mapping of V' into V such that each vertex of G' is mapped to a distinct vertex of G. Two parameters, expansion-cost and dilation-cost, measure the embedding capability of a network. The ration of the numbers of the nodes in the host graph to that of the guest graph is defined as the expansion-cost of the embedding. The maximum distance in G between $F(x)$ and $F(y)$ for any two adjacent nodes x and y of G' is defined as the dilation-cost of the embedding.

The basic idea of embedding a tree $T(h)$ with height h into an RCR(k, r, j) is based on the contraction embedding [17]. The two basic concepts used in finding an embedding of a $T(h)$ into an RCR(k, r, j) are introduced as follows: The objective of our embedding F is to obtain as small an expansion-cost as possible with respect to the desired dilation-cost. For an RCR(k, r, j), the $T(h)$ with the maximum level $h = [\log_2(r \times 2^{k+j} + 1)]$ can be embedded into an RCR(k, r, j). In this case, the expansion-cost of F can be as low as one. We introduce an embedding F with the maximum dilation-cost being three.

Definition 6. *A contraction of degree w of $G(V, E)$ is a graph $G'(V', E')$ obtained by a w-partition of G, then by replacing (contracting) each vertex subset V_i by a single vertex v'_i of V', and by connecting pairs of vertices (v'_i, v'_j) in G' by an edge if and only if there is at least one link between any vertex in the subset V_i, and any vertex in the subset V_j in G.*

Definition 7. *A bounded contraction of degree w from G to G' is a contraction of degree w in which the degree of each vertex $v'_i \in V'$ is not greater than the number of vertices in the subset V_i of G which were contracted to form v'_i.*

A bounded 4-partition of $T(h)$, called $P_{T(h)}$, is first obtained. We partition a $T(h)$ into as many distinct and exclusive subsets as possible according to the combination of depth and width priorities. In each row, we partition as many nodes into subsets as possible (with two brother nodes) according to the width priority. In this case, it is possible that a subset only has a single node. Between rows, from up to down, we combine the partitioned subsets on neighboring rows into a greater subset with four nodes according to depth priority. The rules are that, 1) all nodes in such a subset have a direct parent-child relation; 2) as many subsets of complete binary tree as possible exist. Three kinds of the subsets with four nodes, as shown in Fig. 7, can partition a $T(h)$. The objective of our partitioning is to embed a subtree of a $T(h)$ to a CP of an RCR(k, r, j). It is obvious that a lot of nodes in an RCR(k, r, j) cannot be included by all such subsets.

A node or a ring- or cube- plane not belonging to a partitioning is called to be *idle*. In our partitioning, it is possible that some CPs are idle while the partitioning cannot continue because the current leaves are not directly connected to these idle CPs. This phenomenon may incur CP conflicts such that the two nodes in different CPs

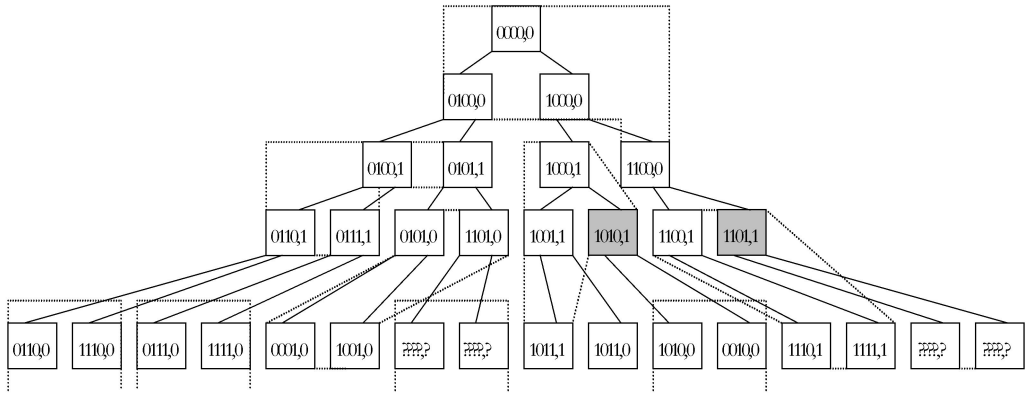

Fig. 8. A 4-partition of T(4) with two conflicts.

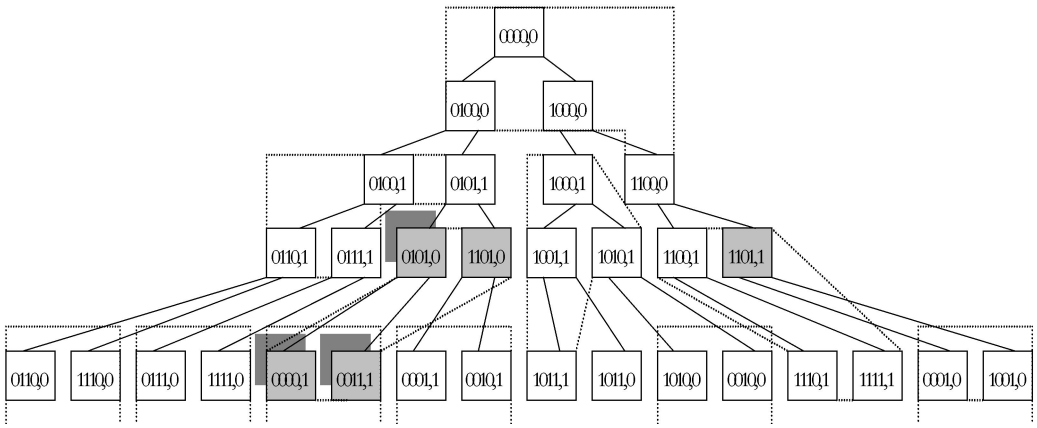

Fig. 9. A 4-partition of T(4).

without any idle nodes wish the nodes in the same CP as their direct child nodes. In this partitioning, we hope the dilation-cost should be less than or equal to three. It is obvious that all CPs with node size of four contain at least one cycle. We may adopt a transitive strategy to solve the problem such that two neighboring idle nodes in one CP are selected and then translated to the conflicting CP along one path. Note that each node of the CPs in the path may change its direct parents or children in the RCR(k, r, j). The partitioning of an RCR(k, r, j) has dilation-cost of less than or equal to three. Fig. 8 illustrates the two conflicts. Fig. 9 shows the final 4-partition of the RCR(2, 2, 2) after the transitive exchange.

Lemma 1. *For an* RCR(k, r, j) *network, a* $T(h)$ *with height* h *can be embedded to the RCR, where* $h = \lfloor \log_2(r \times 2^{k+j} + 1) \rfloor$, *with the dilation-cost less than three and expansion-cost one.*

Proof. According to the construction of the 4-partition of the $T(h)$ described above, we have $h = \lfloor \log_2(r \times 2^{k+j} + 1) \rfloor$. The $T(h)$ can be embedded to the RCR(k, r, j) network. Because the RCR(k, r, j) network is a connected graph, we can transfer two idle nodes on a CP to a conflict by a path. Along the path, we guarantee that the first two occupied nodes (nonidle nodes) closest to the two idle nodes are replaced by the two idle nodes in the 4-partition, such that their maximum distance remains less than three. The procedure is repeated. Obviously, the dilation-cost cannot be greater than three. The $T(h)$ has $2^h - 1$ nodes, and the RCR(k, r, j) has $r \cdot 2^{k+j}$ nodes. Considering the RCR(k, r, j) network size, the difference between the numbers of nodes of the $T(h)$ and the RCR(k, r, j) is so small that it can be neglected. Thus, the expansion-cost of the embedding should be as low as one. □

The broadcast algorithm for an RCR(k, r, j) is to construct an embedded complete binary tree $T(h)$, and first broadcast the message along $T(h)$. For those nodes not belonging to $T(h)$, they must be in certain RPs. In this embedding, at least one node in each RP must belong to $T(h)$. In the worst case, an RP may have only one node in $T(h)$. The message can then be sent from that node to the rest of nodes in the RP, and apparently it takes $\lfloor \frac{r}{2} \rfloor$ steps. The upper bound of message passing steps of the algorithm is analyzed in Theorem 2.

Theorem 2 *Given an* RCR(k, r, j) *network, the upper bound of the number of message passing steps in a broadcast operation is* $k + j + \lceil \log_2 r \rceil + \lfloor \frac{r}{2} \rfloor$.

Proof. According to Lemma 1, a complete binary tree $T(h)$ can be embedded to an RCR(k, r, j), where height $h = \lfloor \log_2(r \times 2^{k+j} + 1) \rfloor$, with dilation-cost less than three and expansion-cost as low as one. The $T(h)$ can

be used as the major part of the a broadcast tree. The needed number of message passing steps is $h = \lfloor \log_2(r \times 2^{k+j} + 1) \rfloor$, in which the same message from a root is sent to all nodes in the $T(h)$, with h as the depth of the tree. According to the above broadcast algorithm description, the message can be sent to those nodes not in $T(h)$ from the nodes in $T(h)$ using at most $\lfloor \frac{r}{2} \rfloor$ steps. Thus, the total steps for a broadcast operation is the sum of the two parts, h and $\lfloor \frac{r}{2} \rfloor$. □

5 Conclusion

We have proposed a class of new topologies for an interconnection network, named recursive cube of rings, which are recursively constructed by adding ring edges to a cube. We have proven that RCRs possess many desirable topological properties in building scalable parallel machines, such as fixed degree, small diameter, plane property, wide bisection width, and symmetry. We have also presented and analyzed a general deadlock-free routing algorithm for RCRs, and developed an efficient broadcast routing algorithm using a complete binary tree embedded into an RCR with expansion-cost approximating to one.

With respect to incremental scalability, the proposed RCR networks may not reach the level of scalability of the incrementally scalable incomplete star graphs proposed in [25], in which the gap between consecutive sizes can be fully deleted. However, comparing to the other existing topologies such as n-star graph and hypercube, the RCR networks obviously have better incremental scalability as shown in Section 3. Our future work is to develop a new topology based on the RCR networks that can achieve the level of the incremental scalability of the incomplete graph while preserving it to be Cayley graphs.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments and suggestions that helped us to improve the quality of this paper. Partial funding was provided by Hong Kong RGC 87-96E and Hong Kong University CRCG grant 337-062-0012.

References

[1] K. Efe and A.O. Fernandez, "Products of Networks with Logarithmic Diameter and Fixed Degree," *IEEE Trans. Parallel and Distributed Systems,* vol. 6, no. 9, pp. 963-975, Sept. 1995.

[2] A.L. Rosenberg, "Product-Shuffle Networks: Toward Reconciling Shuffles and Butterflies," *Discrete Applied Mathematics,* vol. 37-38, pp. 465-488, July 1992.

[3] R.B. Panwar and L.M. Patnaik, "Solution of Linear Equations on Shuffle-Exchange and Modified Shuffle Exchange Networks," *Proc. 26th Allerton Conf.,* pp. 1,116-1,125, 1988.

[4] K. Efe and A. Fernandez, "Mesh Connected Tree: A Bridge between Grids and Meshes of Trees," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, no. 12, pp. 1,283-1,293, Dec. 1996.

[5] K. Day and A.-E. Al-Ayyoub, "The Cross Product of Interconnection Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 8, no. 2, pp. 109-118, Feb. 1997.

[6] E. Ganesan and D.K. Pradhan, "The Hyper-Debruijn Networks: Scalable Versatile Architecture," *IEEE Trans. Parallel and Distributed Systems,* vol. 4, no. 9, pp. 962-978, Sept. 1993.

[7] S.K. Das and A.K. Banerjee, "Hyper Petersian Networks: Yet Another Hypercube-Like Topology," *Proc. Fourth Symp. Frontiers of Massively Parallel Computation,* pp. 270-277, Oct. 1992.

[8] A.S. Youssef and B. Narahari, "The Banyan-Hypercube Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 1, pp. 160-169, 1990.

[9] X. Lin, P.K. McKinley, and L.M. Ni, "Deadlock-Free Multicast Wormhole Routing in 2-D Mesh Multicomputers," *IEEE Trans. Parallel and Distributed Systems,* vol. 5, no. 8, pp. 793-804, Aug. 1994.

[10] T.J. Cortina and Z. Xu, "The Cube of Rings Interconnection Networks," *Int'l J. Foundations of Computer Science,* 1997.

[11] W.-J. Hsu, M.-J. Chung, and A. Das, "Linear Recursive Networks and Their Applications in Distributed Systems," *IEEE Trans. Parallel and Distributed Systems,* vol. 8, no. 7, pp. 673-680, July 1997.

[12] I. Stojmenovic, "Honeycomb Networks: Topological Properties and Communication Algorithms," *IEEE Trans. Parallel and Distributed Systems,* vol. 8, no. 10, pp. 1,036-1,042, Oct. 1997.

[13] S.B. Akers and B. Krishnamurthy, "A Group-Network Theoretical Model for Symmetric Interconnection Networks," *IEEE Trans. Computers,* vol. 38, no. 4, pp. 555-566 Apr. 1989.

[14] B. Alspach, "Cayley Graphs with Optimal Fault Tolerance," *IEEE Trans. Computers,* vol. 41, pp. 1,337-1,339, 1992.

[15] W.J. Dally and C.L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Nnetworks," *IEEE Trans. Computers,* vol. 36, no. 5, pp. 547-553, May 1987.

[16] W.J. Dally, "Virtual channel flow control," *IEEE Trans. Computers,* vol. 3, no. 3, pp. 194-205, Mar. 1992.

[17] A. Barak and E. Schenfeld, "Embedding Classical Communication Topologies in the Scalable OPAM Architecture," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, no. 9, pp. 979-992, Sept. 1996.

[18] Y.-J. Tsai and P.K. McKinley, "A Broadcast Algorithm for All-Port Wormhole-Routed Torus Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, no. 8, pp. 876-885, Aug. 1996.

[19] P.K. McKinley, Y.-J. Tsai, and D. Robinson, "Collective Communication in Wormhole-Routed Massively Parallel Computers," *Computer,* vol. 28, no. 12, pp. 39-50, Dec. 1995.

[20] C.-T. Ho and M. Kao, "Optimal Broadcast in All-Port Wormhole-Routed Hypercubes," *Proc. 1994 Int'l Conf. Parallel Processing,* vol. III, pp. 167-171 Aug. 1994.

[21] T.-S. Chen, Y.-C. Tseng, and J.-P. Sheu, "Balanced Spanning Trees in Complete and Incomplete Star Graphs," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, no. 7, pp. 717-723, July 1996.

[22] S.L. Hohnsson and C.-T. Ho, "Optimal Broadcasting and Personalized Communication in Hypercubes," *IEEE Trans. Parallel and Distributed Systems,* vol. 38, no. 9, pp. 1,249-1,268, Sept. 1989.

[23] S.B. Akers, D. Harel, and B. Krishnameurthy, "The Star Graph: An Attractive Alternative to the n-cube," *Proc. Int'l Conf. Parallel Processing,* pp. 393-400, 1987.

[24] S.B. Akers and B. Krishnameurthy, "A Group-Theoretic Model for Symmetric Interconnection Networks," *IEEE Trans. Computers,* vol. 38, no. 4, pp. 555-566, Apr. 1989.

[25] S. Latifi and N. Bagherzadeh, "Incomplete Star: An Incrementally Scalable Network Based on the Star Graph," *IEEE Trans. Parallel and Distributed Systems,* vol. 5, no. 1, pp. 97-102, Jan. 1994.

Yuzhong Sun received his PhD degree in computer engineering at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1997. Currently, he is a research fellow in the Department of Electrical and Electronic Engineering at the University of Hong Kong. His research interests include interconnection networking, architectures for parallel and distributed computation, cluster computing, and parallel algorithms.

Paul Y.S. Cheung received his BSc degree with first class honors in 1973 and his PhD degree in 1978, both in electrical engineering from the Imperial College of Science and Technology, University of London. After working for Queen's University of Belfast for two years as an engineer in charge of a laboratory, he returned to Hong Kong in 1978 to take up an academic position at the Hong Kong Polytechnic. He joined the University of Hong Kong as lecturer in 1980 and was promoted to senior lecturer and associate professor in 1987. He served as the associate dean of engineering from 1991-1994 and has been the Dean of Faculty of Engineering at the University of Hong Kong since 1994. He was the IEEE Asia Pacific Director in 1995-1996 and served as the IEEE secretary in 1997. His research interests include parallel computer architecture, internet computing, VLSI design, signal processing, and pattern recognition. Dr. Cheung is a senior member of the IEEE.

Xiaola Lin received the BS and MS degrees in computer science from Peking University, Beijing, China, in 1982 and 1985, respectively, and the PhD degree in computer science from Michigan State University, East Lansing, in 1992. He is currently with the Department of Electric and Electronic Engineering at the University of Hong Kong. His research interests include parallel and distributed computing, design and analysis algorithms, and high speed computer networks. Dr. Lin is a member of the IEEE.