The Recursive Dual-net and its Applications

Yamin Li[1], Shietung Peng[1], and Wanming Chu[2]

[1] Department of Computer Science
Hosei University
Tokyo 184-8584 Japan
{yamin, speng}@k.hosei.ac.jp
[2] Department of Computer Hardware
University of Aizu
Aizu-Wakamatsu 965-8580 Japan
w-chu@u-aizu.ac.jp

Abstract. In this paper, we propose a universal network, called recursive dual-net (RDN). It can be used as a candidate of effective interconnection networks for massively parallel computers. The RDN is generated by recursively applying dual-construction on a base-network. Given a regular and symmetric graph of size n and node-degree d, the dual-construction generates a regular and symmetric graph of size $2n^2$ and node-degree $d+1$. The RDN has many interesting properties including low node-degree and small diameter. For example, we can construct an RDN connecting more than 3-million nodes with only 6 links per node and a diameter of 22. We investigate the topological properties of the RDN and compare it to other networks including 3D torus, WK-recursive network, hypercube, cube-connected-cycle, and dual-cube. We also describe an efficient routing algorithm for RDN.

Key words: Interconnection networks and routing algorithm

1 Introduction

In massively parallel processor (MPP), the interconnection network plays a crucial role on the issues such as communication performance, hardware cost, computational complexity, fault-tolerance, etc. Much research has been reported in the literatures for interconnection networks that can be used to connect parallel computers of large scale (see [2, 6, 12] for the review of the early work). The following two categories have attracted a great research attention. One is the hypercube-like family that has the advantage of short diameters for high-performance computing and efficient communication [5, 7–10]. The other is 2D/3D mesh or torus that has the advantage of small and fixed node-degrees and easy implementations. Traditionally, most MPPs in the history including those built by NASA, CRAY, FGPS, IBM, etc., use 2D/3D mesh or torus or their variations with extra diagonal links. The recursive networks also have been proposed as effective interconnection networks for parallel computers of large scale. For example, the WK-recursive network [4, 13] is a class of recursive scalable networks. It offers a high-degree of regularity, scalability, and symmetry and has a compact VLSI implementation.

Recently, due to the advance in computer technologies, the community of supercomputers rises competition to construct supercomputers of very-large scale that might contain millions of nodes [11]. For example, the IBM new Blue Gene system was proposed that will contain more than a million processors. It was predicted that the MPPs of the next decade will contain 10 to 100 millions of nodes [3]. For such a parallel computer of very-large scale, the traditional interconnection networks may no longer satisfy the requirements for the high-performance computing or efficient communication. For the future generation of MPPs with millions of nodes, the node-degree and the diameter will be the critical measures for the effectiveness of the interconnection networks. The node-degree is limited by the hardware technologies and the diameter affects directly all kind of communication schemes. Other important measures include bisection bandwidth, scalability, and efficient routing algorithms.

In this paper, we propose a set of networks, called *Recursive Dual-Net* (RDN). A recursive dual-net is based on the recursive dual-constructions of a regular base-network. The dual-construction extends a regular network with n nodes and node-degree d to a network with $2n^2$ nodes and node-degree $d+1$. The recursive dual-net is especially suitable for the interconnection network of the parallel computers with millions of nodes. It has the merits of regularity, scalability and symmetry and can connect a huge number of nodes with just a small number of links per node and very short diameters. For example, a 2-level RDN with $n = 25$ can connect more than 3-million nodes that has only 6 links per node and its diameter equals to 22. For parallel computers with millions of nodes, most of the known topologies will either require a large number of links per node (hypercube-like family) that is difficult to implement or have a large diameter (3D torus or WK-recursive network) that affects tremendously its performance.

We investigate the topological properties of the recursive dual-net and show some examples of recursive dual-net with rather simple base-networks. Then we compare them with other networks such as 3D torus [1], WK-recursive network [13], hypercube [10], CCC (cube-connected-cycle) [9], and dual-cube [7, 8]. We also propose efficient basic routing algorithms for the recursive dual-net.

The rest of this paper is organized as follows. Section 2 describes the recursive dual-net in details. Section 3 discusses the topological properties of the recursive dual-net. Sections 4 compares recursive dual-net with other networks. Section 5 gives a few examples of recursive dual-net for parallel computers of large-scale or very large-scale. Section 6 describes an efficient routing algorithm. Section 7 concludes the paper and presents some future research directions.

2 Recursive Dual-Net

Let G be an undirected graph. The size of G, denoted as $|G|$, is the number of vertices. A path from node s to node t in G is denoted by $s \rightarrow t$. The length of the path is the number of edges in the path. For any two nodes s and t in G, we denote $D(s,t)$ as the length of a shortest path connecting s and t. The diameter of G is defined as $D(G) = \max\{D(s,t)|s,t \in G\}$. For any two nodes s and t in G, if there is a path connecting s and t, we say G is a connected graph.

Suppose we have a symmetric connected graph B and there are n_0 nodes in B and the node degree is d_0. A k-level Recursive Dual-Net $RDN^k(B)$, also denoted as $RDN^k(B(n_0))$, can be recursively defined as follows:

1. $RDN^0(B) = B$ is a symmetric connected graph with n_0 nodes, called *base network*;
2. For $k > 0$, an $RDN^k(B)$ is constructed from $RDN^{k-1}(B)$ by a dual-construction as explained below (also see Figure 1).

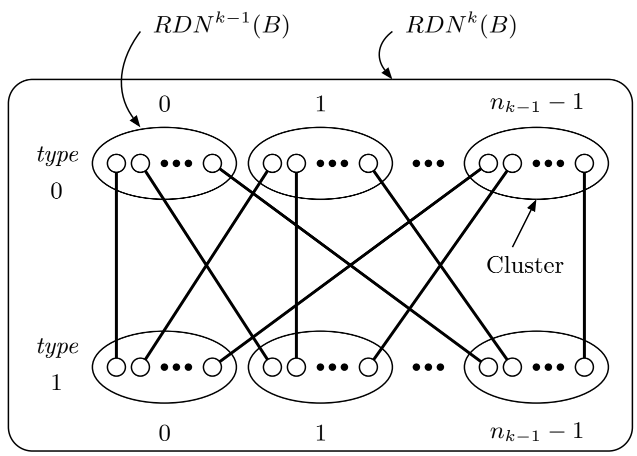

Fig. 1. Build an $RDN^k(B)$ from $RDN^{k-1}(B)$

Dual-construction: Let $RDN^{k-1}(B)$ be referred to as a *cluster* of level k and $n_{k-1} = |RDN^{k-1}(B)|$ for $k > 0$. An $RDN^k(B)$ is a graph that contains $2n_{k-1}$ clusters of level k as subgraphs. These clusters are divided into two sets with each set containing n_{k-1} clusters. Each cluster in one set is said to be of *type* 0, denoted as C_i^0, where $0 \leq i \leq n_{k-1} - 1$ is the cluster ID. Each cluster in the other set is of *type* 1, denoted as C_j^1, where $0 \leq j \leq n_{k-1} - 1$ is the cluster ID. At level k, each node in a cluster has a new link to a node in a distinct cluster of the other type. We call this link *cross-edge* of level k. By following this rule, for each pair of clusters C_i^0 and C_j^1, there is a unique edge connecting a node

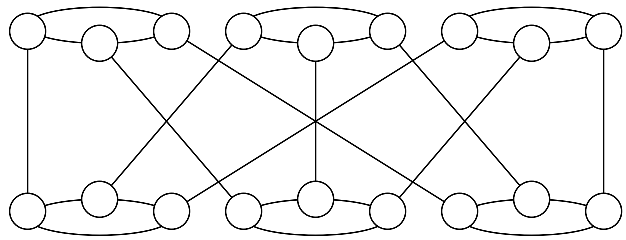

Fig. 2. A Recursive Dual-Net $RDN^1(B(3))$

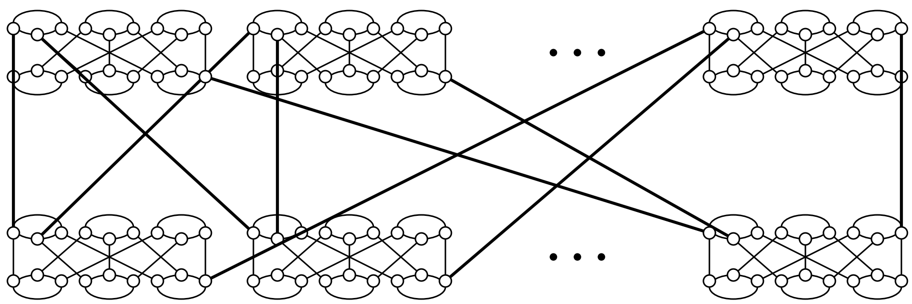

Fig. 3. A Recursive Dual-Net $RDN^2(B(3))$

in C_i^0 and a node in C_j^1, $0 \leq i, j \leq n_{k-1} - 1$. In Figure 1, there are n_{k-1} nodes within each cluster $RDN^{k-1}(B)$.

We give two simple examples of recursive dual-nets with $k = 1$ and 2, in which the base network is a ring with 3 nodes, in Figure 2 and Figure 3, respectively. Figure 2 depicts an $RDN^1(B(3))$ network. There are 3 nodes in the base network. Therefore, the number of nodes in $RDN^1(B(3))$ is 2×3^2, or 18. Figure 3 shows the $RDN^2(B(3))$ constructed from the $RDN^1(B(3))$ in Figure 2. We did not show all the nodes in the figure. The number of nodes in $RDN^2(B(3))$ is 2×18^2, or 648.

Similarly, we can construct an $RDN^3(B(3))$ containing 2×648^2, or 839,808 nodes with node-degree of 5 and diameter of 22. In contrast, the 839,808-node 3D torus machine (adopt by IBM Blue Gene/L [1]) configured as $108 \times 108 \times 72$ nodes, the diameter is equal to $54 + 54 + 36 = 144$ with a node degree of 6.

3 Topological properties of RDN

We can see from the recursive dual-construction described above that an $RDN^k(B)$ is a symmetric connected network with node-degree $d_0 + k$, where d_0 is the node-degree of the base network B. The number of nodes n_k in $RDN^k(B)$ satisfies the recurrence $n_k = 2n_{k-1}^2$ for $k > 0$. Solving the recurrence, we get $n_k = (2n_0)^{2^k}/2$.

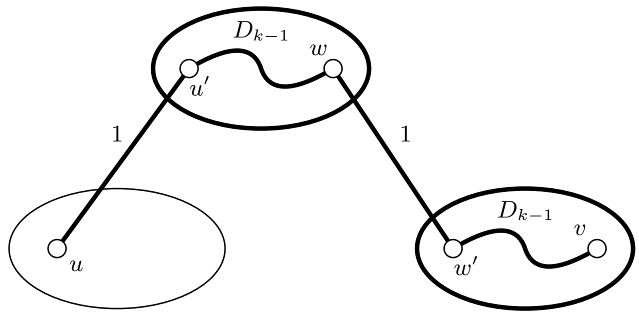

Fig. 4. The diameter of the Recursive Dual-Net

Concerning the diameter D_k of $RDN^k(B)$, we know that the worst-case (the longest one) for the shortest path $P(u, v)$ connecting any two nodes u and v in $RDN^k(B)$ is as follow: u and v are of the same type and path $P = u \rightarrow u' \rightarrow w \rightarrow w' \rightarrow v$, where $u \rightarrow u'$ and $w \rightarrow w'$ are cross-edges of level k, and $|u' \rightarrow w| = |w' \rightarrow v| = D_{k-1}$, as shown as in Figure 4. Therefore, the diameter of $RDN^k(B)$ satisfies the recurrence $D_k = 2D_{k-1} + 2$ for $k > 0$. Solving the recurrence, we get $D_k = 2^k D_0 + 2^{k+1} - 2$, where D_0 is the diameter of the base network.

The bisection bandwidth is important for fault-tolerance. Next, we investigate the bisection bandwidth of the $RDN^k(B)$ for $k \geq 1$. From the dual-construction, we know that there is no link between the clusters of level k that are of the same type. Therefore, the minimum number of links those removal will disconnect two halves occurs when both halves contain equal numbers of clusters of type 0 or 1. That is, the minimum number of links those removal will disconnect two halves equals to half of the total number of cross-edges of level k which is $\lceil (2n_0)^{2^k}/8 \rceil$.

Notice that if n_0 is odd and $k = 1$ we should divide the RDN into two halves such that one half contains $\lfloor n_0/2 \rfloor$ (or $\lceil n_0/2 \rceil$) type 0 clusters and $\lceil n_0/2 \rceil$ (or $\lfloor n_0/2 \rfloor$) type 1 clusters. For example, the bisection bandwidth of $RDN^1(B(3))$ is $\lceil 6^2/8 \rceil = \lceil 9/2 \rceil = 5$.

We summarize the discussion above about the fundamental properties of the Recursive Dual-Net in the following theorem.

Theorem 1 *Assume that the base network B is a symmetric graph with size n_0, node-degree d_0, and the diameter D_0. Then, the size, the node-degree, the diameter and the bisection bandwidth of $RDN^k(B)$ are $(2n_0)^{2^k}/2$, d_0+k, $2^k D_0 + 2^{k+1} - 2$, and $\lceil (2n_0)^{2^k}/8 \rceil$, respectively.*

4 Comparison to Other Interconnection Networks

An interconnection network is evaluated in terms of a number of parameters such as node-degree, diameter, bisection width, average distance, regularity, symmetry, etc. Let G be a regular, symmetric graph. There are trade-offs among the node-degree, the diameter, and the size of a graph G. It is not easy and maybe unfair to use a single parameter to compare the effectiveness of networks that have different topologies and sizes. However, it should be worth to have such a parameter that shows the combined effects of the topology on three important measures: node-degree, diameter and size. There might be an argument that the diameter is not an important issue if the system adopts the wormhole switching technique. However, for the MPPs with millions of nodes, it seems not possible to use wormhole switching technique since the whole system will occupy a big hall and the connection must be done with cables. Therefore, for the interconnection networks of MPPs, the diameter should play an important role for measuring the ability of high-performance computing and efficient communication.

In this paper, we introduce *cost ratio* $CR(G)$ as an important measure for the combined effects of the hardware cost and the software efficiency of an interconnection network presented as graph G. Let $|(G)|$, $d(G)$, and $D(G)$ be the number of nodes, the node-degree, and the diameter of G, respectively. We define $CR(G)$ as

$$CR(G) = (d(G) + D(G))/\lg|(G)|$$

The motivation here is that the node-degree and diameter should not increase faster than the logarithm of the size of of the graph. It should be considered as a basic rule for high-performance MPPs. The design of interconnection network should make effort to reduce the cost ratio, especially for an MPP with very large scale. The cost ratio of hypercube is a constant 2 for any size. One of the reasons that hypercube has been and will be still popular as an interconnection network of MPPs is that its node-degree and diameter grow logarithmically with its size. However, for an MPP with more than a million of nodes, the logarithmic growth rate of the node-degree is still too big for the current hardware technologies (each node requires more than 20 ports and channels)..

Other important measures for the performance of networks include the existence of simple and efficient routing and communication algorithms for certain communication patterns such as multicast or total exchange. We present a simple and efficient routing algorithm on RDN. The design of efficient algorithms for collective communication is beyond the scope of this paper. It should be an interesting subjects for the further research.

Table 1 summarizes the number of nodes, the node-degree, the diameter, and the cost ratio for 3D torus, hypercube, CCC, dual-cube, WK-recursive network and recursive dual-net. The *torus*, also called *wrap-around mesh* or a *toroidal mesh*, was adopt by IBM Blue Gene/L. This topology includes the p-ary, q-cube which is a q-dimensional torus with the restriction that each dimension is of the same size p. In a $CCC(n)$, each node in an n-cube is replaced with an n-node ring [9]. A dual-cube $DC(n)$ contains 2^n $(n-1)$-cubes called *clusters* [7]. Half of the clusters are of type 0 and the other half are of type 1. There is a unique link (cross-edge) connecting each pair of clusters of distinct types. $DC(n)$ is equal to $RDN(2^{n-1}, 1)$, where the base network is an $(n-1)$-cube.

Table 1. CR of recursive dual-net and the other networks

Network	Number of nodes	Node-degree	Diameter
p-ary, 3-cube	p^3	6	$3p/2$
n-cube	2^n	n	n
$CCC(n)$	$n * 2^n$	3	$2n + \lfloor n/2 \rfloor - 2$
$DC(n)$	2^{2n-1}	n	$2n$
$WK(n,t)$	n^t	n	$2^t - 1$
$RDN^k(B)$	$n_k = (2n_0)^{2^k}/2$	$d_0 + k$	$2^k * D_0 + 2^{k+1} - 2$

Network	CR
p-ary, 3-cube	$(6 + 3p/2)/3 \lg p$
n-cube	2
$CCC(n)$	$(2n + \lfloor n/2 \rfloor + 1)/(n + \lg n)$
$DC(n)$	$3n/(2n-1)$
$WK(n,t)$	$(n + 2^t - 1)/\lg n^t$
$RDN^k(B)$	$(d_0 + k + D_k)/\lg n_k$

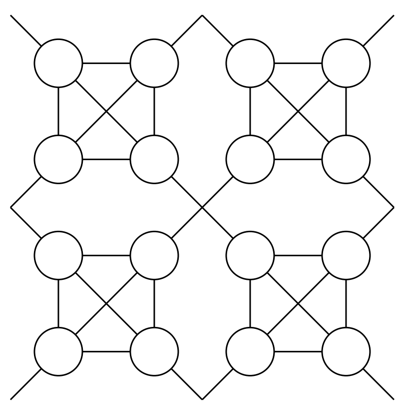

Fig. 5. A WK-recursive network $WK(4,2)$

A WK-recursive network of level t denoted as $WK(n,t)$ can be constructed recursively as follows [13]. $WK(n,1)$ is an n-node complete graph augmented with n open links each at a node. Each node of $WK(n,t)$ is incident with $n-1$ substituting links and one flipping link (or open link). The substituting links are those within basic building blocks, and the j-flipping links are those connecting two embedded $WK(n,j)$. Figure 5 shows a WK-recursive network with $n = 4$ and $t = 2$.

5 Samples of RDN for Massively Parallel Computers

In this section, we describe some selections of base-networks such that the corresponding recursive dual-net will be the candidate as an effective interconnection network for MPPs of different sizes. A good choice for the base-network is p-ary, q-cube. The p-ary, q-cube has many nice properties and is suitable as an interconnection network for parallel computers of small sizes. For example, a 5-ary, 2-cube or a 3-ary, 3-cube can be easily built into a 2D or 3D chip. The second choice for the base-network is a WK-recursive network with $n = 4$ and $t = 2$ or 3. The nature of WK-recursive network makes it easily to be implemented on a 2D chip. The selection of value k for recursive dual-net depends on the sizes of the MPPs. For the MPPs of large-scale (thousands of nodes), $k = 1$ is a good choice, while for the MPPs of very large-scale (millions of nodes), we can set $k = 2$ that applies dual-construction twice. We list below a few examples of the RDN as candidates of interconnection networks for MPPs based on the discussion above.

1. MPPs of large-scale:
   - $RDN^1(B(25))$, where $B(25)$ is a 5-ary, 2-cube: Since $n_0 = 25, d_0 = 4$, and $D_0 = 4$, this network has 1250 nodes. its node-degree, diameter and cost ratio are 5, 10, and 1.46, respectively.
   - $RDN^1(B(27))$, where $B(27)$ is a 3-ary, 3-cube: Since $n_0 = 27, d_0 = 6$, and $D_0 = 3$, this network has 1458 nodes. its node-degree, diameter and cost ratio are 7, 8 and 1.43, respectively.
   - $RDN^1(B(16))$, where $B(16)$ is a $WK(4, 2)$: Since $n_0 = 16, d_0 = 4$, and $D_0 = 3$, this network has 512 nodes. its node-degree, diameter and cost ratio are 5, 8 and 1.44, respectively.
2. MPPs of very large-scale:
   - $RDN^2(B(25))$, where $B(25)$ is a 5-ary, 2-cube: This network has 3,125,000 nodes. its node-degree, diameter and cost ratio are 6, 22 and 1.30, respectively.
   - $RDN^2(B(27))$, where $B(27)$ is a 3-ary, 3-cube: This network has 4,251,528 nodes. its node-degree, diameter and cost ratio are 8, 18 and 1.18, respectively.
   - $RDN^2(B(16))$, where $B(16)$ is a $WK(4, 2)$: This network has 524,288 nodes. its node-degree, diameter and cost ratio are 6, 18 and 1.26, respectively.

We show the comparisons of the RDN and other networks for MPPs of large-scale and very large-scale in Table 2 and Table 3, respectively. It can be seen from the tables that the RDN with properly selected base-networks are superior to other networks.

Finally, concerning the physical layout of an MPP with recursive dual-net, it can be described briefly as follows. The base-network that is a 5-ary, 2-cube, or a 3-ary, 3-cube, or an $WK(4, 2)$ can be built on a 2D or 3D chip. The MPP of large-scale that contains clusters of level 1 can be packed into a dual-rack

Table 2. CR for MPPs of large-scale

Network	n	d	D	CR
10-ary 3-cube	1,000	6	15	2.11
10-cube	1,024	10	10	2.00
CCC(8)	2,048	3	18	1.91
$WK(8,3)$	512	8	7	1.67
$DC(6)$	2,048	6	12	1.64
$RDN^1(B(25))$	1,250	5	10	1.46
$RDN^1(B(27))$	1,458	7	8	1.43
$RDN^1(B(16))$	512	5	8	1.44

Table 3. CR for MPPs of very large-scale

Network	n	d	D	CR
100-ary 3-cube	1,000,000	6	150	7.83
20-cube	1,048,576	20	20	2.00
CCC(16)	1,048,576	3	38	2.05
$WK(8,7)$	2,097,152	8	127	6.43
$DC(11)$	2,097,152	11	22	1.57
$RDN^2(B(25))$	3,125,000	6	22	1.30
$RDN^2(B(27))$	4,251,528	8	18	1.18
$RDN^2(B(16))$	524,288	6	18	1.26

that connects to sets of clusters face-to-face. The MPP of very large-scale can be built and displayed in a big hall with dual-racks connected through cables. With the advance of technologies, the above configuration of an MPP with the recursive dual-net might become a reality.

6 An Efficient Routing Algorithm in RDN

The problem of finding a path from a source s to a destination t and forwarding a message along the path is known as the basic routing problem. In this section, we present efficient algorithms for the basic routing in RDN.

In order to describe the routing algorithm, we first give a presentation for $RDN^k(B)$ that provides an unique ID to each node in $RDN^k(B)$. Let the IDs of nodes in B, denoted as ID_0, be $i,\ 0 \leq i \leq n_0 - 1$. The ID_k of node u in $RDN^k(B)$ for $k > 0$ is a triple (u_0, u_1, u_2), where u_0 is a 0 or 1, u_1 and u_2 belong to ID_{k-1}. We call u_0, u_1, and u_2 typeID, clusterID, and nodeID of u, respectively.

More specifically, $ID_i,\ 1 \leq i \leq k$, can be defined recursively as follows: $ID_i = (b, ID_{i-1}, ID_{i-1})$, where $b = 0$ or 1. The ID of a node u in $RDN^k(B)$

can also be presented by an unique integer $i,\ 0 \le i \le (2n_0)^{2^k}/2-1$, where i is the lexicographical order of the triple (u_0, u_1, u_2). For example, the ID of node $(1,1,2)$ in $RDN^1(B)$ is $1*3^2+1*3+2=14$. It can be verified easily that the definition is consistent with the definition of the recursive dual-net in Section 2.

With this ID presentation, (u,v) is a cross-edge of level k in $RDN^k(B)$ iff $u_0 \ne v_0$, $u_1 = v_2$, and $u_2 = v_1$.

Assume that a routing algorithm for the base network B is available. The proposed routing algorithm that routes node u to node v in $RDN^k(B)$ for $k > 0$ is a recursive one. If u and v are in the same cluster of level k then just call itself for $k-1$. Otherwise, we assume that u and v has distinct typeID (for the case $u_0 = v_0$, we simply route u to w via a cross-edge of level k then we treat w as u). We route u to u' with $u'_2 = v_1$ and v to v' with $v'_2 = u_1$ inside the clusters of level k where u and v belong to. This can be done by recursive calls for $k-1$. Then we can route u' to v' in 1 hop since there is a cross-edge of level k from u' to v'. The proposed routing algorithm is described formally as Algorithm 1.

Algorithm 1: RDN_routing($RDN^k(B), u, v$)
begin
  if $k = 0$ **then** RDN_routing($RDN(m,0), u, v$)
  else
    Case 1:$u_0 = v_0$ and $u_1 = v_1$
      RDN_routing($RDN^{k-1}_{u_0,u_1}(B), u_2, v_2$);
      /* $RDN^{k-1}_{u_0,u_1}(B)$ is the cluster with typeID = u_0
        and clusterID = u_1. */
    Case 2: $u_0 \ne v_0$
      RDN_routing($RDN^{k-1}_{u_0,u_1}(B), u_2, v_1$);
      $u' = (u_0, u_1, v_1)$;
      RDN_routing($RDN^{k-1}_{v_0,v_1}(B), v_2, u_1$);
      $v' = (v_0, v_1, u_1)$;
      connect u' and v' via a cross-edge of level k;
    Case 3: $u_0 = v_0$ and $u_1 \ne v_1$
      route u to w via the cross-edge of level k;
      route node w to node v as in Case 2;
  endif
end

Example (also see Fig. 6):
$k = 2:$
  $u = (u_0, u_1, u_2) = (0, (0,0,0), (0,0,0))$
  $v = (v_0, v_1, v_2) = (1, (1,2,2), (0,2,2))$
  $u_0 = 0,\ u_1 = (0,0,0),\ u_2 = (0,0,0)$
  $v_0 = 1,\ v_1 = (1,2,2),\ v_2 = (0,2,2)$
  $u_0 \ne v_0$ (Case 2, cross-edge):
  $u' = (u_0, u_1, v_1) = (0, (0,0,0), (1,2,2))$

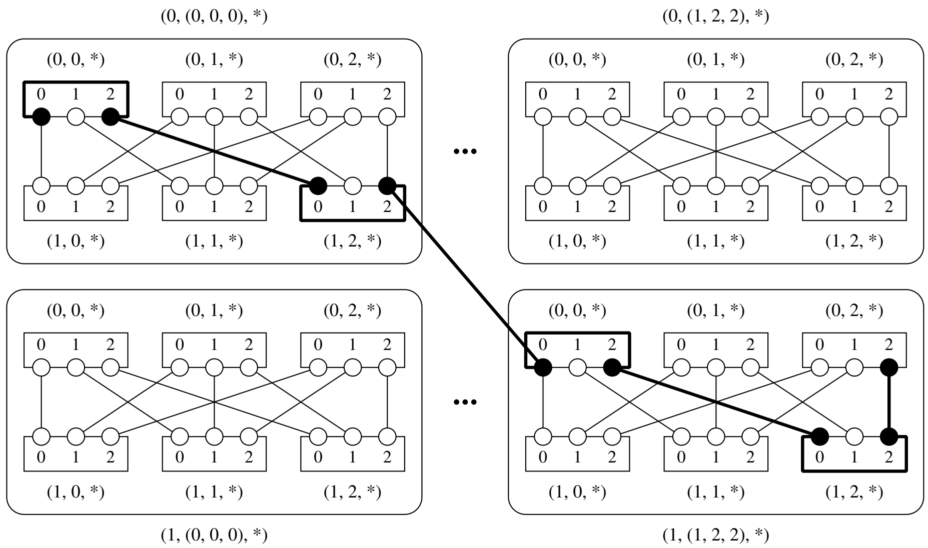

Fig. 6. Routing in $RDN^2(B)$

$v' = (v_0, v_1, u_1) = (1, (1, 2, 2), (0, 0, 0))$
$u_2 = (0, 0, 0) \rightarrow v_1 = (1, 2, 2)$, see $k = 1$ (1)
$v_2 = (0, 2, 2) \rightarrow u_1 = (0, 0, 0)$, see $k = 1$ (2)
$k = 1$ (1): in cluster $(0, (0, 0, 0), *)$
$u = (u_0, u_1, u_2) = (0, 0, 0)$
$v = (v_0, v_1, v_2) = (1, 2, 2)$
$u_0 = 0,\ u_1 = 0,\ u_2 = 0$
$v_0 = 1,\ v_1 = 2,\ v_2 = 2$
$u_0 \neq v_0$ (Case 2, cross-edge):
$u' = (u_0, u_1, v_1) = (0, 0, 2)$
$v' = (v_0, v_1, u_1) = (1, 2, 0)$
$u_2 = 0 \rightarrow v_1 = 2$, (Case 1, $k = 0$)
$v_2 = 2 \rightarrow u_1 = 0$, (Case 1, $k = 0$)
$k = 1$ (2): in cluster $(1, (1, 2, 2), *)$
$u = (u_0, u_1, u_2) = (0, 2, 2)$
$v = (v_0, v_1, v_2) = (0, 0, 0)$
$u_0 = 0,\ u_1 = 2,\ u_2 = 2$
$v_0 = 0,\ v_1 = 0,\ v_2 = 0$
$u_0 = v_0$ and $u_1 \neq v_1$ (Case 3)
$w = (w_0, w_1, w_2) = (1, 2, 2)$
Let $u = w$, then do similarly in $k = 1$ (1).

Theorem 2 *In $RDN^k(B)$, routing from source s to destination t can be done in at most $2^k * D_0 + 2^{k+1} - 2$ steps, where D_0 is the diameter of the base network.*

Proof: The correctness of the algorithm 1 can be proved easily by induction on k. The worst-case for the length of the routing path is Case 3. In Case 3, the length of routing path $d(u, v)$ satisfies the inequality $d(u, v) \leq d(w, w') + d(v, v') + 2$ for $k > 0$, where $d(w, w') \leq D_{k-1}$ and $d(v, v') \leq D_{k-1}$. Therefore, we have $d(u, v) \leq 2^k * D_0 + 2^{k+1} - 2$, where D_0 is the diameter for the base network. ❑

7 Conclusion

In this paper, we described a universal network, recursive dual-net, that can be used as an effective interconnection network of an MPP with very large scale (having millions of nodes). If the base-network is properly selected, the recursive dual-net has many attractive properties including small and flexible node-degree, short diameter, recursive structure, and efficient routing algorithms. We studied the topological properties of the recursive dual-net. We also described an efficient routing algorithm in $RDN^k(B)$ for $k > 0$. To design efficient algorithms for collective communications, parallel prefix computation, sorting, and numerical computations in recursive dual-net are certainly worth of the further research. The other direction of the future work includes the study of architectural aspects of the proposed network.

References

1. N. R. Adiga, M. A. Blumrich, D. Chen, P. Coteus, A. Gara, M. E. Giampapa, P. Heidelberger, S. Singh, B. D. Steinmacher-Burow, T. Takken, M. Tsao, and P. Vranas. Blue gene/l torus interconnection network. *IBM Journal of Research and Development, http://www.research.ibm.com/journal/rd/492/tocpdf.html*, 49(2/3):265–276, 2005.
2. S. G. Aki. *Parallel Computation: Models and Methods*. Prentice-Hall, 1997.
3. P. Beckman. Looking toward exascale computing, keynote speaker. In *International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'08)*, University of Otago, Dunedin, New Zealand, December 2 2008.
4. G. H. Chen and D. R. Duh. Topological properties, communication, and computation on wk-recursive networks. *Networks*, 24(6):303–317, 1994.
5. K. Ghose and K. R. Desai. Hierarchical cubic networks. *IEEE Transactions on Parallel and Distributed Systems*, 6(4):427–435, April 1995.
6. F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, 1992.
7. Y. Li and S. Peng. Dual-cubes: a new interconnection network for high-performance computer clusters. In *Proceedings of the 2000 International Computer Symposium, Workshop on Computer Architecture*, pages 51–57, ChiaYi, Taiwan, December 2000.
8. Y. Li, S. Peng, and W. Chu. Efficient collective communications in dual-cube. *The Journal of Supercomputing*, 28(1):71–90, April 2004.

9. F. P. Preparata and J. Vuillemin. The cube-connected cycles: a versatile network for parallel computation. *Commun. ACM*, 24:300–309, May 1981.
10. Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, July 1988.
11. TOP500. *Supercomputer Sites*. http://top500.org/, Jun. 2008.
12. A. Varma and C. S. Raghavendra. *Interconnection Networks for Multiprocessors and Multicomputers: Theory and Practice*. IEEE Computer Society Press, 1994.
13. G. Vicchia and C. Sanges. A recursively scalable network vlsi implementation. *Future Generation Computer Systems*, 4(3):235–243, 1988.