Recursive Dual-Net: A New Universal Network for Supercomputers of the Next Generation

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 new universal network, called recursive dual-net (RDN), as a potential candidate for the interconnection network of supercomputers with very large scale. The recursive dual-net is based on a recursive dual-construction of a base network. A k-level dual-construction for $k > 0$ creates a network containing $(2m)^{2^k}/2$ nodes with node-degree $d + k$, where m and d are the number of nodes and the node-degree, respectively, of the base network. The recursive dual-net is node and edge symmetric and can contain huge number of nodes with small node-degree and short diameter. For example, we can construct a symmetric 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 recursive dual-net and compare it to that of other networks including 3D torus, WK-recursive network, hypercube, cube-connected-cycle, and dual-cube. We also establish the basic routing and broadcasting algorithms for the proposed recursive dual-net.

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 containing hundreds of thousands of nodes [11]. For example, the IBM Blue Gene/P system in Argonne National Laboratory contains 163,840 processors. It was predicted that the MPPs of the next decade will contain 10 to 100 millions of notes [3]. For such a very-large-scale parallel computer, the traditional interconnection networks can 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 of the interconnection network will be the critical measures of the effectiveness of the network. The node-degree is limited by the hardware technologies and the diameter affects directly all kind of communication schemes. Other measures include bisection bandwidth, scalability, and efficient routing and broadcasting algorithms.

In this paper, we propose a new interconnection network, called *Recursive Dual-Net* (RDN). A recursive dual-net is based on a recursive dual-construction of a base network. The dual-construction extends a 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 future MPPs 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, an $RDN(25, 2)$ can connect more than 3-million nodes that has only 6 links per node and its diameter equals to 22. For supercomputers 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 simple base networks of small size. Then we compare them to other networks such as three-dimensional torus used in IBM Blue Gene/L [1], WK-recursive network [13], hypercube [10], CCC (cube-connected-cycle) [9], and dual-cube [7, 8]. We also establish the basic routing and broadcasting 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. Section 4 gives the routing and broadcasting algorithms. Section 5 concludes the paper and presents some future research directions.

2 Recursive Dual-Net

A recursive dual-net $RDN(m, k)$ can be recursively defined as follows:

1. $RDN(m, 0)$ is a symmetric, regular graph with m nodes, called *base network*;
2. For $k > 0$, $RDN(m, k)$ is constructed from $RDN(m, k-1)$ by a dual-construction as explained below (also see Fig. 1).

Let $RDN(m, k-1)$ be referred to as a *cluster* of level k and $n = |RDN(m, k-1)|$. An $RDN(m, k)$ is a graph that contains $2n$ clusters of level k as subgraphs. These clusters are divided into two disjoint sets of clusters with each set containing n clusters.

Each cluster in one set is said to be of *type* 0, denoted as C_i^0, where $0 \leq i \leq n-1$ is the cluster ID. Each cluster in the other set is of *type* 1, denoted as C_j^1, where $0 \leq j \leq n-1$ is the cluster ID. At level k, each node in every cluster has a new link to a node in a distinct cluster of the other type. We call this link *cross-edges* of level k. By following this rule, for each pair of clusters C_i^0 and C_j^1, there is an unique cross-edge connecting a node in C_i^0 and a node in C_j^1.

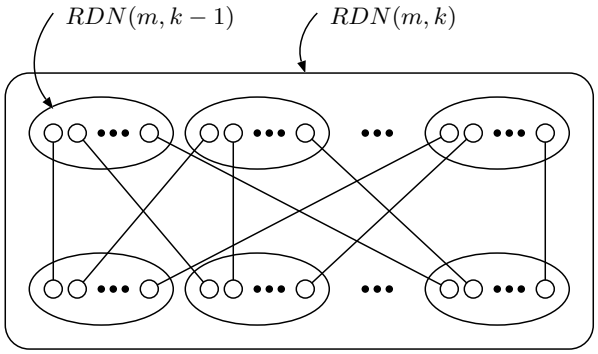

Fig. 1. The recursive dual-construction

We give two examples of recursive dual-nets with $k = 1$ and 2, and the base network is a ring with 3 nodes, in Fig. 2 and Fig. 3, respectively. Fig. 2 depicts an $RDN(3, 1)$ network with a 3-node ring as its base network. Fig. 3 shows the $RDN(3, 2)$ constructed from the $RDN(3, 1)$ in Fig. 2. Notice that only those edges connecting to a cluster of type 0 are shown.

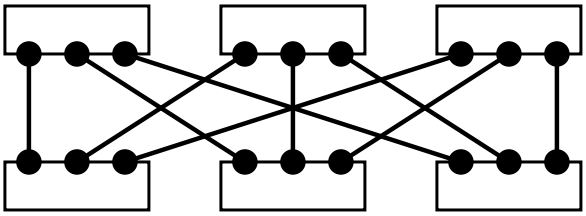

Fig. 2. The recursive dual-net $RDN(3, 1)$

It is easy to see from the above recursive dual-construction that $RDN(m, k)$ is a symmetric, regular network with node-degree $d_0 + k$, where d_0 is the node-degree of $RDN(m, 0)$, and the number of nodes $N_{m,k}$ in $RDN(m, k)$ satisfies the following recurrence:

1. $N_{m,0} = m$;
2. $N_{m,k} = 2N_{m,k-1}^2$ for $k > 0$.

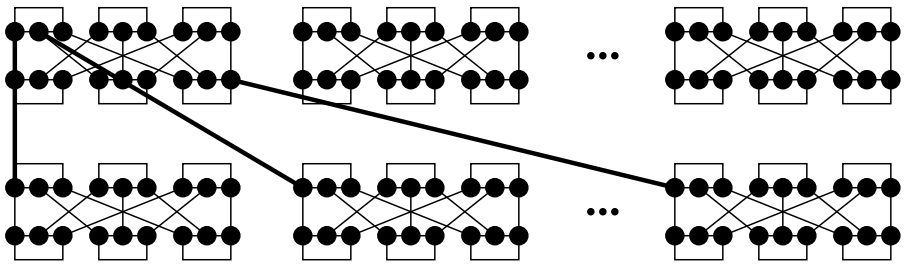

Fig. 3. The recursive dual-net $RDN(3,2)$

Solving the recurrence, we get $N_{m,k} = (2m)^{2^k}/2$.

Concerning the diameter D_k of $RDN(m,k)$, we know that the worst-case (the longest one) for the shortest path $P(u,v)$ connecting any two nodes u and v in $RDN(m,k)$ is as follow (also see Fig. 4): 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 $|u' \rightarrow w| = |w' \rightarrow v| = D_{k-1}$. Therefore, the diameter of $RDN(m,k)$ satisfies the recurrence $D_k = (1 + D_{k-1}) + (1 + D_{k-1})$ 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. We summarize the results into the following theorem.

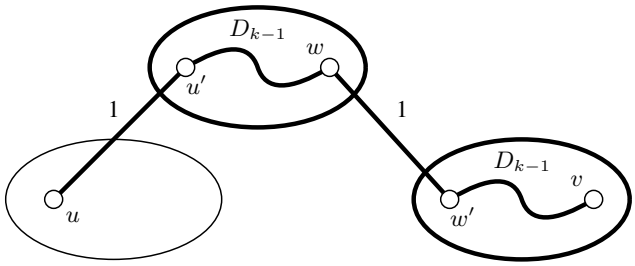

Fig. 4. The diameter of the RDN

The bisection bandwidth is important for fault-tolerance. Next, we investigate the bisection bandwidth of the $RDN(m,k)$ 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 $\lceil (2m)^{2^k}/8 \rceil$. Notice that if m is odd and $k = 1$ we should divide the RDN into two halves such that one half contains $\lfloor m/2 \rfloor$ (or $\lceil m/2 \rceil$) type 0 clusters and $\lceil m/2 \rceil$ (or $\lfloor m/2 \rfloor$) type 1 clusters. For example, the bisection bandwidth of $RDN(3,1)$ 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 is a symmetric, regular graph with m nodes, node-degree d_0, and the diameter D_0. Then, the node-degree, the number of nodes, the diameter and the bisection bandwidth of $RDN(m,k)$ are d_0+k, $(2m)^{2^k}/2$, $2^k * D_0 + 2^{k+1} - 2$ and $\lceil (2m)^{2^k}/8 \rceil$, respectively.*

Table 1. CR of Recursive dual-net and the others

Network	Number of nodes	Node-degree	Diameter
3D Torus	$x * y * z$	6	$(x+y+z)/2$
n-cube	2^n	n	n
$CCC(n)$	$n * 2^n$	3	$2n$
$DC(n)$	2^{2n-1}	n	$2n$
$WK(n,t)$	n^t	n	$2^t - 1$
$RDN(m,k)$	$N_k = (2m)^{2^k}/2$	$d_0 + k$	$D_k = 2^k * D_0 + 2^{k+1} - 2$
Network	CR		
3D Torus	$(6 + (x+y+z)/2)/\lg(x * y * z)$		
n-cube	2		
$CCC(n)$	$(2n+3)/(n + \lg n)$		
$DC(n)$	$3n/(2n-1)$		
$WK(n,t)$	$(n + 2^t - 1)/\lg n^t$		
$RDN(m,k)$	$(d_0 + k + D_k)/\lg N_k$		

Table 2. CR for Recursive dual-net of size between 2M and 50M

Network	Number of nodes	Node-degree	Diameter	CR
3D-torus	2,097,152	6	192	9.43
$WK(8,7)$	2,097,152	8	127	6.43
21-cube	2,097,152	21	21	2.00
$CCC(17)$	2,228,224	3	34	1.75
$DC(11)$	2,097,152	11	22	1.57
$RDN(5^2,2)$	3,125,000	6	22	1.30
$RDN(3^3,2)$	4,251,528	8	18	1.18
$RDN(5,3)$	50,000,000	5	30	1.37

3 Topological Properties and Cost Ratio

Let G be a undirected graph. The size of G denoted as $|G|$ is the number of nodes in G. The node-degree of G denoted as $d(G)$ is the maximum number of edges incident on any node in G. 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 denoted as $D(G)$ is $\max\{D(s,t)|s,t \in G\}$.

A topology is evaluated in terms of a number of parameters such as node-degree, diameter, bisection width, average distance for any two nodes, 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 graphs 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 of the future generation, the diameter should play an important role for measuring the ability of high-performance computing and efficient communication.

In this paper, we propose a key parameter, called *cost ratio* $CR(G)$, to measure the cost 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

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

It is clear from the definition of $CR(G)$ that the smaller the value of $CR(G)$ the better the graph G as a candidate for interconnection network of an MPP. Other important measures for the performance of networks include the existence of simple and efficient routing and communication algorithms. From Theorem 1, we get $CR(RDN(m,k)) = ((d_0 + k) + (2^k * D_0 + 2^{k+1} - 2))/\lg((2m)^{2^k}/2)$, where $m = |(RDN(m,0))|$, $d_0 = d(RDN(m,0))$, and $D_0 = D(RDN(m,0))$, respectively. 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.

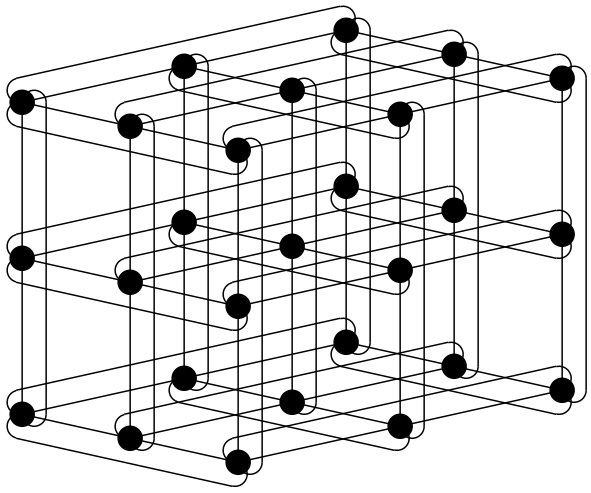

Fig. 5. A 3-ary 3-cube base network

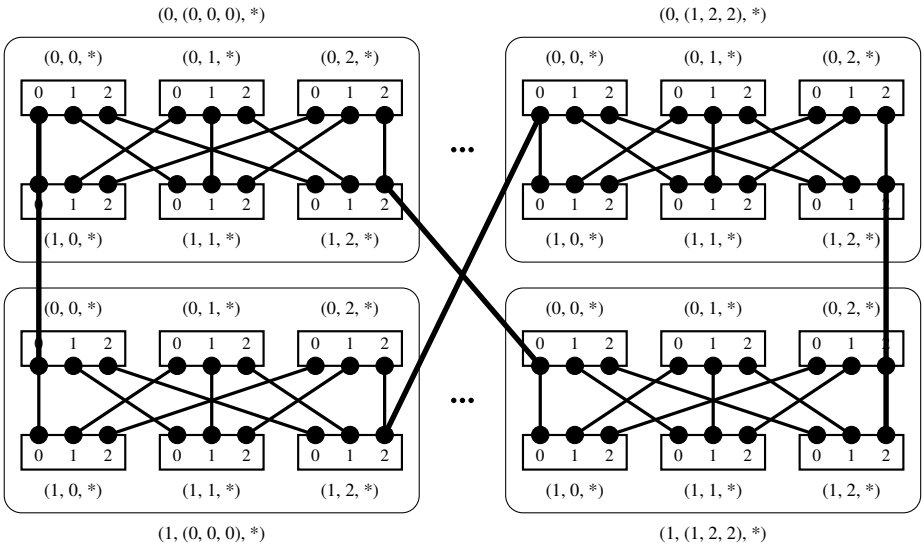

Fig. 6. Presentation of $RDN(3, 2)$ including IDs of nodes

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 $WK(n, j)$. 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. 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. Fig. 5 shows a 3-ary 3-cube network, which can be used in recursive dual-nets as a base network.

For consideration to be used as an effective interconnection network of MPPs of the future generations, we use p-ary, q-cube as the base network. To construct recursive dual-net with small diameter, we assign $p = 3$ or 5. The selection of value q depends on k. We assign $q = 2$ or 3, and 1 for $k = 2$ and 3, respectively, since for $k = 3$ the RDN will contain 50 m nodes with a base network of size 3. In Table 2, based on the values of p and q assigned above, we calculate the values of $CR(RDN(p^q, k))$ and compare these values with that of 3D torus, WK-recursive network, hypercube, CCC, and dual-cube of sizes around 2M. The range of the sizes of the selected $RDN(p^q, k)$ is between 3M and 50M. For the 2M-node 3D torus machine (Blue Gene/L) configured as 128 * 128 * 128 nodes (128-ary 3-cube), the diameter is equal to 64 + 64 + 64 = 192 hops. From the table, we can see that the recursive dual-nets are more effective than all other networks measured by the cost ratio. The most effective one is $RDN(3^3, 2)$

whose diameter is only 18 and the number of nodes is up to about 4M with 8 links per node.

The proposed recursive dual-nets with a small number of recursions ($k \leq 3$) are suitable for MPPs for the reasons explained above. In addition, concerning the physical layout of an MPP with $RDN(5^2, 2$ or $RDN(3^3, 2)$, it can be described briefly as follows: a 2D or 3D torus of with 5×5 or 3×3×3 nodes can be embedded on a 2D or 3D chip. The clusters of level 1 can be packed into a dual-rack that connects to sets of clusters face-to-face. The whole system can be displayed in a big hall with dual-racks connected through cables. We believe that the advance of new technologies will bring such a configuration of an MPP with the recursive dual-net into reality.

4 Routing and Broadcasting

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 routing problem. The broadcasting task is to send a message from a source to all other nodes. Routing and broadcasting are the basic communication problems for interconnection networks. In this section, we will describe routing and broadcasting algorithms for the recursive dual-net.

In order to describe the routing algorithm, we first give a presentation for $RDN(m, k)$ that provides an unique ID to each node in $RDN(m, k)$. Let the IDs of nodes in $RDN(m, 0)$, denoted as ID_0, be $i,\ 0 \leq i \leq m-1$. The ID_k of node u in $RDN(m, k)$ 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. Fig. 6 gives the presentation of the $RDN(3, 2)$. The ID of a node u in $RDN(m, k)$ can also be presented by an unique integer $i,\ 0 \leq i \leq (2m)^{2^k}/2 - 1$, 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 $RDN(3, 1)$ 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 $RDN(m, k)$ iff $u_0 \neq v_0$, $u_1 = v_2$, and $u_2 = v_1$.

Assume that a routing algorithm RDN_routing $(RDN(m, 0), u, v)$ for the base network is available. The proposed routing algorithm that routes node u to node v in $RDN(m, k)$ is a recursive one for $k > 0$. 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.

Example (also see Fig. 7):
$k = 2$:
$u = (u_0, u_1, u_2) = (0, (0, 0, 0), (0, 0, 0))$
$v = (v_0, v_1, v_2) = (1, (1, 2, 2), (0, 2, 2))$

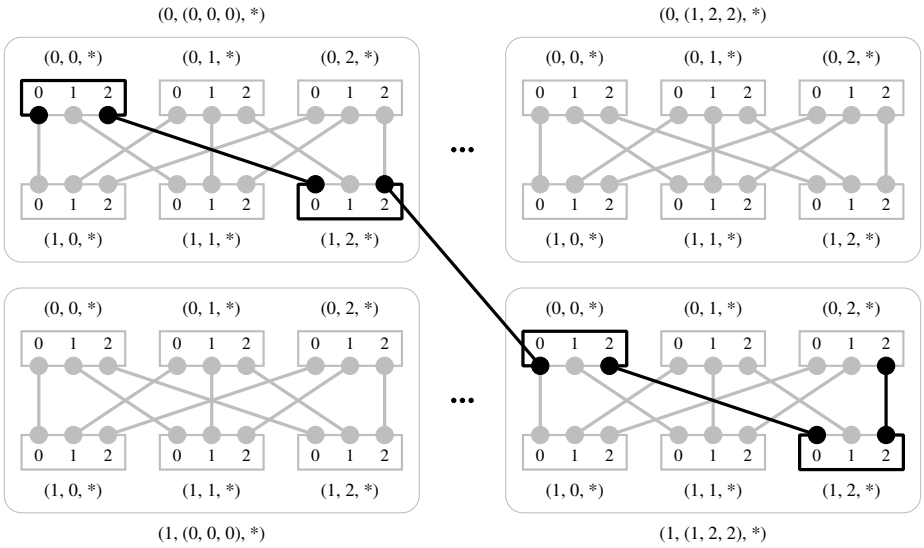

Fig. 7. Routing in $RDN(3, 2)$

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

$u_0 = 0$, $u_1 = (0, 0, 0)$, $u_2 = (0, 0, 0)$
$v_0 = 1$, $v_1 = (1, 2, 2)$, $v_2 = (0, 2, 2)$
$u_0 \neq v_0$ (Case 2, cross-edge):
$u' = (u_0, u_1, v_1) = (0, (0, 0, 0), (1, 2, 2))$
$v' = (v_0, v_1, u_1) = (1, (1, 2, 2), (0, 0, 0))$
$u_2 = (0, 0, 0) \rightarrow v_1 = (1, 2, 2)$, 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 $RDN(m, k)$, 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. ❑

The broadcasting process should satisfy some desirable properties: (1) A node should not send (receive) the message simultaneously to (from) more than one of its neighbors; (2) A node receives the message exactly once for the whole duration of the broadcasting process. We show an efficient broadcasting algorithm which completes broadcasting under the above two conditions.

The algorithm first let node s broadcast the message to all other nodes within the cluster of level k which node s belongs to. Then, all the nodes in that cluster send the message via the cross-edges of level k and the nodes in other clusters of level k that receive the message broadcast the message to all other nodes in the clusters of level k which the nodes belong to. Finally, assuming that typeID of node s is 0, all nodes in clusters of level k with typeID = 1 (except the nodes in cluster $C^1_{s'}$ of level k, where s' has a cross-edge of level k to s) send message via cross-edges of level k.

Assume that a broadcasting algorithm RDN_broadcast ($RDN(m, 0), s$) for the base network is available. The algorithm for broadcasting from a source s in $RDN(m, k)$ is formally described as Algorithm 2.

Fig. 8 shows the broadcasting in $RDN(3, 2)$, where $RDN(3, 0)$ is a 3-node ring. The numbers in the right side of the figure are the numbers of steps during the broad-

Algorithm 2: RDN_broadcast($RDN(m,k), s$)
begin
/* Without loss of generality, assume that s is of type 0 */
if $k = 0$ **then** RDN_broadcast($RDN(m,0), s$)
else
RDN_broadcast($RDN^{s_0}_{s_1}(m,k-1), s_2$);
/* $RDN^{s_0}_{s_1}(m,k-1)$ is the cluster with typeID = s_0 and clusterID = s_1. */
for each u in $RDN^{s_0}_{s_1}(m,k-1)$ **do**
send message to u' via the cross-edge of level k;
endfor
for each u' **do**
RDN_broadcast($RDN^{u'_0}_{u'_1}(m,k-1), u'_2$);
endfor
for each v in $\cup C^1_i \setminus C^1_{s'}$ **do**
/* $C^1_{s'}$ is the cluster that contains s', where s' is connected to s via a cross-edge of level k */
send message to v' via the cross-edge of level k;
endfor
endif
end

```
1. broadcasting in RDN(3,0)           2
2. cross-edging via k1                1
3. do in parallel (*3):
     1. broadcasting in RDN(3,0)      2
     2. cross-edging via k1           1

4. cross-edging via k2 (18 nodes)     1

5. do in parallel (*18):
     1. broadcasting in RDN(3,0)      2
     2. cross-edging via k1           1
     3. do in parallel (*3):
          1. broadcasting in RDN(3,0) 2
          2. cross-edging via k1      1

     4. cross-edging via k2 (18*18)   1
```

Fig. 8. Broadcasting in $RDN(3,2)$

casting. From the above algorithm, since broadcasting in $RDN(3,1)$ is done in $2+1+2+1=6$ steps, the broadcasting in $RDN(3,2)$ is done in $6+1+6+1=14$ steps.

Theorem 3. *In $RDN(m,k)$, broadcasting from source node s can be done in at most $2^k * B_0 + 2^{k+1} - 2$ steps, where B_0 is the time complexity of $RDN\_broadcast(m,0)$, under the one-port communication model.*

Proof: The correctness of the algorithm 2 can be proved easily by induction on k. The time complexity B_k of $RDN(m,k)$ satisfies the following recurrence: $B_k = 2B_{k-1}+2$ for $k > 0$. Therefore, we have $B_k = 2^k * B_0 + 2^{k+1} - 2$, where B_0 is the time complexity of $RDN\_broadcast(m,0)$. It is easy to see that each node in $RDN(m,k)$ receives message only once. ❑

5 Conclusion

In this paper, we proposed a new network, called recursive dual-net, that can be used as an efficient interconnection network of a supercomputer of the future generation. The proposed network has many attractive properties including small and flexible node-degree, short diameter, recursive structure, efficient routing and broadcasting algorithms. We believe that it has great potential. To investigate algorithmic aspects of the proposed network in depth is certainly worth of the further research. The other direction of further work includes the study of architectural aspects of the proposed network such as system organization and layout etc.

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.