Hamiltonian Connectedness of Recursive Dual-Net

Yamin Li and Shietung Peng
Department of Computer Science
Hosei University
Tokyo 184-8584 Japan
{yamin, speng}@k.hosei.ac.jp

Wanming Chu
Department of Computer Hardware
University of Aizu
Aizu-Wakamatsu 965-8580 Japan
w-chu@u-aizu.ac.jp

Abstract

Recursive Dual-Net (RDN) was proposed recently as an effective, high-performance interconnection network for supercomputers with millions of nodes. A Recursive Dual-Net $RDN(B)$ is recursively constructed on a base symmetric network B. At each iteration, the network is extended through dual-construction. The dual-construction extends a symmetric graph G into a symmetric graph G' with size $2n^2$ and node-degree $d+1$, where n and d are the size and the node-degree of G, respectively. Therefore, a k-level Recursive Dual-Net $RDN^k(B)$ contains $(2n_0)^{2^k}/2$ nodes with a node degree of d_0+k, where n_0 and d_0 are the size and the node-degree of the base network B, respectively. In this paper, we show that, if the base network B is hamiltonian, $RDN^k(B)$ is hamiltonian. We give an efficient algorithm for constructing a hamiltonian cycle in $RDN^k(B)$ for $k>0$. We also show that if the base network is hamiltonian connected, $RDN^k(B)$ is hamiltonian connected for any $k>0$.

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 a large number of nodes (see [2, 7, 16] 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 computation and efficient communication [5, 8, 10, 13, 14]. 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 [3, 17] 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 high-performance supercomputers containing hundreds of thousands of nodes [15]. In coming decades, supercomputers containing millions of nodes will be built. For such very-large-scale supercomputers, the traditional interconnection networks such as hypercube or mesh/torus [1] will have either large node degree or long diameter. New interconnection networks that have the merits of traditional networks such as node and edge symmetry and recursive structure etc., and also have small node degree and short diameter are critical for very large-scale supercomputers.

In this paper, we first introduce a new interconnection network, call Recursive Dual-Net [12], that was proposed as a candidate of interconnection networks for the supercomputers of the next generation. A Recursive Dual-Net $RDN(B)$ is based on a recursive dual-construction of a base network B. The dual-construction extends a network with n nodes and node-degree d to a network with $2n^2$ nodes and node-degree $d+1$. A k-level Recursive Dual-Net $RDN^k(B)$, also denoted as $RDN^k(B(n_0))$, can have $(2n_0)^{2^k}/2$ nodes with node degree of d_0+k where n_0 and d_0 are the number of nodes and node degree of the 0-level Recursive Dual-Net, or the base network B, respectively. We can see that the Recursive Dual-Net can connect a huge number of nodes with just a small number of links per node. Meanwhile, if the base network is a symmetric network, the Recursive Dual-Net is also a symmetric network.

Recursive Dual-Net has much shorter diameter than WK-recursive network, and smaller node-degree than hypercube. It is very flexible since its base network can be any popular symmetric network of small size. For example, if the base network $B(25)$ is a 5-ary, 2-cube with size 25 and node-degree 4 then $RDN^2(B(25))$ has 3,125,000 nodes with 6 links per node and a diameter of 22. Another example is $RDN^2(B(27))$, where the base network $B(27)$ is a 3-ary, 3-cube. It has 4,251,528 nodes with 8 links per node and a diameter of 18.

Linear array and ring are two fundamental networks and many algorithms were designed based on linear array and ring [6]. Thus embedding linear array or ring in networks is important for emulating those algorithms. A *hamiltonian cycle* of an undirected graph G is a simple cycle that contains every node in G exactly once. A *hamiltonian path* in a graph is a simple path that visits every node exactly once. A graph that contains a hamiltonian cycle is said to be *hamiltonian*. A graph is said to be *hamiltonian connected* if there is a hamiltonian path between *any* two distinct nodes in the graph.

Two algorithms for embedding a hamiltonian cycle in dual-cube and metacube were given in [9] and [11], respectively. Fu proved that the WK-recursive network is hamiltonian connected [4]. In this paper, we show that the Recursive Dual-Net is hamiltonian if the base network is hamiltonian and gives an efficient algorithm for the cycle construction. We also show that if the base network is hamiltonian connected, the Recursive Dual-Net is also hamiltonian connected.

The rest of the paper is organized as follows. Section 2 introduce the Recursive Dual-Net topology. Section 3 gives an efficient algorithm for constructing a hamiltonian cycle. Section 4 proves that if the base network is hamiltonian connected, then the Recursive Dual-Net is also hamiltonian connected. Section 5 concludes the paper.

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 Recursive Dual-Net $RDN(B)$, 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).

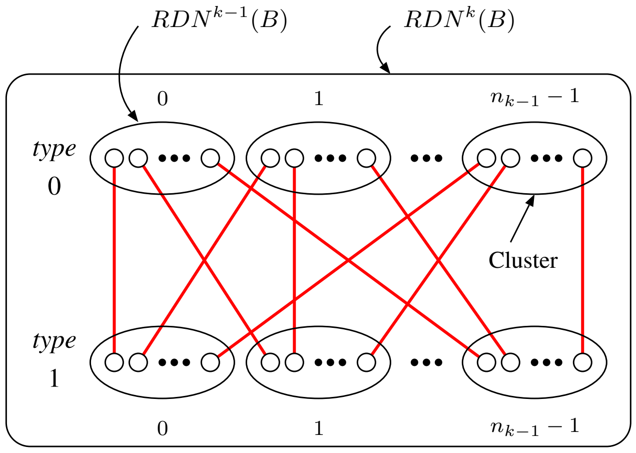


Figure 1. Build an $RDN^k(B)$ from $RDN^{k-1}(B)$

Dual-construction: Let $RDN^{k-1}(B)$ be referred to as a *cluster* of level k and $n_{k-1} = |RDN^{k-1}(B)|$. An $RDN^k(B)$ is a graph that contains $2n_{k-1}$ clusters of level k as subgraphs. These clusters are divided into two disjoint 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 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 $RDN^{k-1}(B)$.

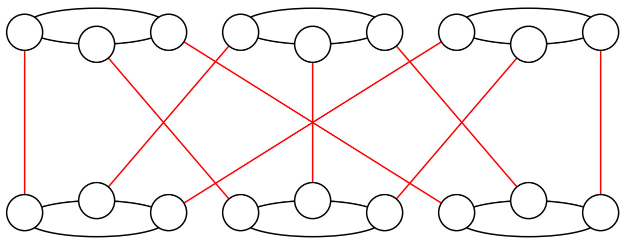


Figure 2. A Recursive Dual-Net $RDN^1(B(3))$

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

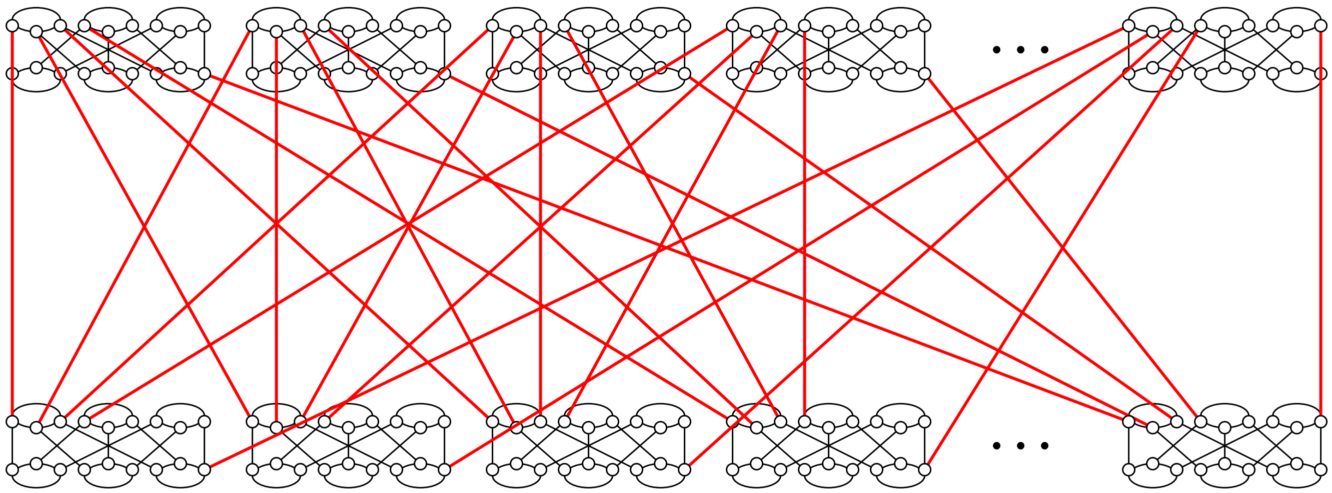


Figure 3. A Recursive Dual-Net $RDN^2(B(3))$

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.

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 $n_k = (2n_0)^{2^k}/2$.

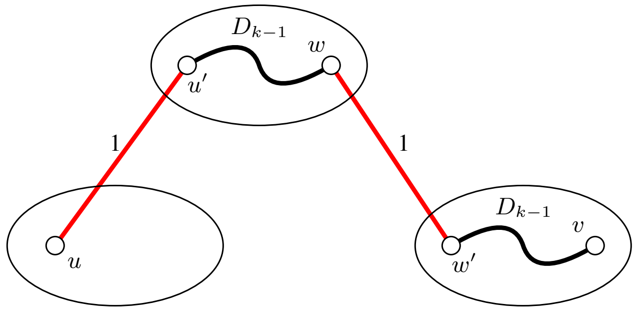


Figure 4. The diameter of the Recursive Dual-Net

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 $P = u \to u' \to w \to w' \to v$, where $u \to u'$ and $w \to w'$ are cross-edges of level k, and $|u' \to w| = |w' \to v| = D_{k-1}$, 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 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.*

3. Hamiltonian Cycle Embedding

In this section, we show how to construct a hamiltonian cycle in $RDN^k(B)$ recursively. First, we find a hamiltonian cycle in the base network $RDN^0(B)$ and assign i, $0 \leq i \leq n_0 - 1$, to each node in the cycle sequentially such that node i and node $((i+1) \mod n_0)$ are neighbors in the cycle.

To construct an $RDN^1(B)$ from $RDN^0(B)$, we prepare $2n_0$ clusters ($RDN^0(B)$) and divide them equally to two sets: type 0 and type 1. Each cluster in each type is assigned a number i, $0 \leq i \leq n_0 - 1$.

Then the ID of a node u in $RDN^1(B)$ is a triple (u_0, u_1, u_2), where u_0 is a 0 or a 1, we call it *typeID*; u_1 is the cluster number, we call it *clusterID*; and u_2 is the node number within a cluster, we call it *nodeID*.

With this ID presentation, (u, v) is a cross-edge of level 1 in $RDN^1(B)$ iff $u_0 \neq v_0$, $u_1 = v_2$, and $u_2 = v_1$. For example, there is a cross edge connecting node $(0, 2, 1)$ and node $(1, 1, 2)$. Figure 5 gives the presentation of the $RDN^1(B(3))$. Note that the "$*$" will be replaced with nodeID. For example, in the first cluster, the three nodes' IDs are $(0, 0, 0)$, $(0, 0, 1)$, and $(0, 0, 2)$; in the last cluster, the three nodes' IDs are $(1, 2, 0)$, $(1, 2, 1)$, and $(1, 2, 2)$. The ID of a node u in $RDN^1(B)$ can also be presented by a unique integer $i = u_0 n_0^2 + u_1 n_0 + u_2$, for $0 \leq i \leq 2n_0^2 - 1$. For example, the last node $(1, 2, 2)$ in Figure 5 where $n_0 = 3$ has a number $9 + 6 + 2 = 17$.

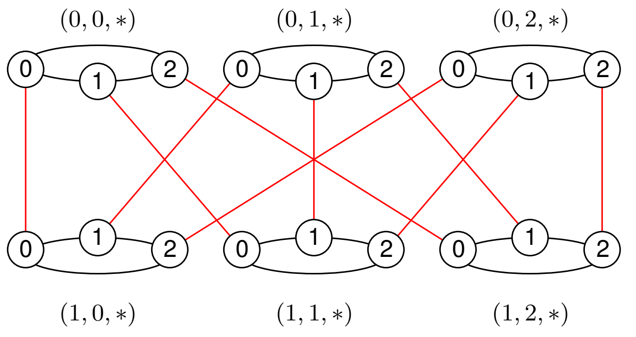


Figure 5. Presentation of $RDN^1(B(3))$

For $RDN^2(B)$, the ID of node u is also in the format of (u_0, u_1, u_2) where u_0 is a 0 or a 1 but u_1 and u_2 are IDs defined in $RDN^1(B)$. For example, the ID of the first node in an $RDN^2(B(3))$ is $(0, (0, 0, 0), (0, 0, 0))$; the ID of the last node is $(1, (1, 2, 2), (1, 2, 2))$. In general, ID_i, $1 \leq i \leq k$, can be defined recursively as follows: $ID_i = (T, ID_{i-1}, ID_{i-1})$, where $T = 0$ or 1. Fig. 6 (in the next page) gives the presentation of the $RDN^2(B(3))$.

Now, we describe how to construct a hamiltonian cycle in $RDN^1(B)$. We first constructs a *virtual hamiltonian cycle*. A virtual hamiltonian cycle is a cycle that connects all the clusters with cross edges and each cluster contributes two nodes u and v and one edge (u, v), as shown as in Figure 7.

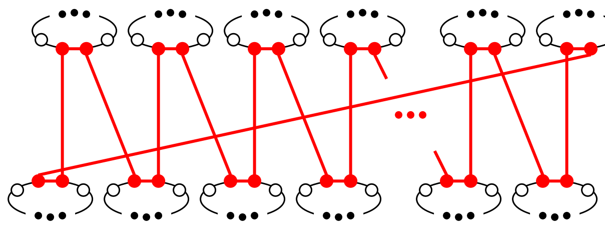
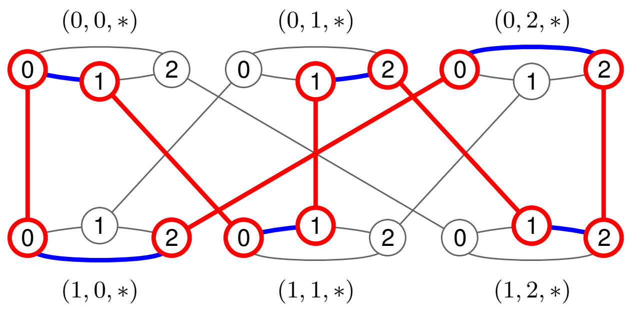
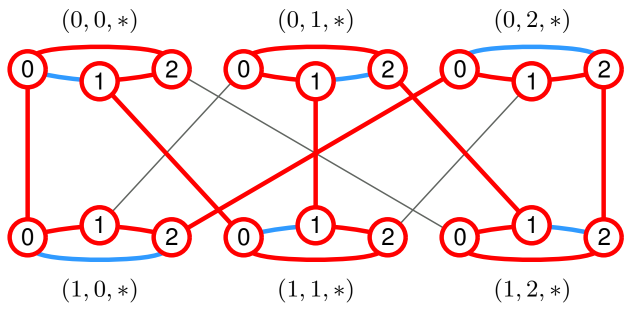


Figure 7. Virtual hamiltonian cycle



(a) A virtual hamiltonian cycle



(b) A hamiltonian cycle

Figure 8. A hamiltonian cycle $in RDN^1(B(3))$

Then, a virtual hamiltonian cycle could be

$(0, 0, 0) \rightarrow (0, 0, 1) \rightarrow (1, 1, 0) \rightarrow (1, 1, 1) \rightarrow$
$(0, 1, 1) \rightarrow (0, 1, 2) \rightarrow (1, 2, 1) \rightarrow (1, 2, 2) \rightarrow$
$(0, 2, 2) \rightarrow (0, 2, 3) \rightarrow (1, 3, 2) \rightarrow (1, 3, 3) \rightarrow$
$\cdots \rightarrow$
$(0, m-2, m-2) \rightarrow (0, m-2, m-1) \rightarrow$
$(1, m-1, m-2) \rightarrow (1, m-1, m-1) \rightarrow$
$(0, m-1, m-1) \rightarrow (0, m-1, 0) \rightarrow$
$(1, 0, m-1) \rightarrow (1, 0, 0) \rightarrow (0, 0, 0)$.

Finally, in each cluster, we replace the edge (u, v) with a hamiltonian path $u \rightarrow v$ so that we get a hamiltonian cycle containing all the $2n_0^2$ nodes in the $RDN^1(B)$. Figure 8(b)

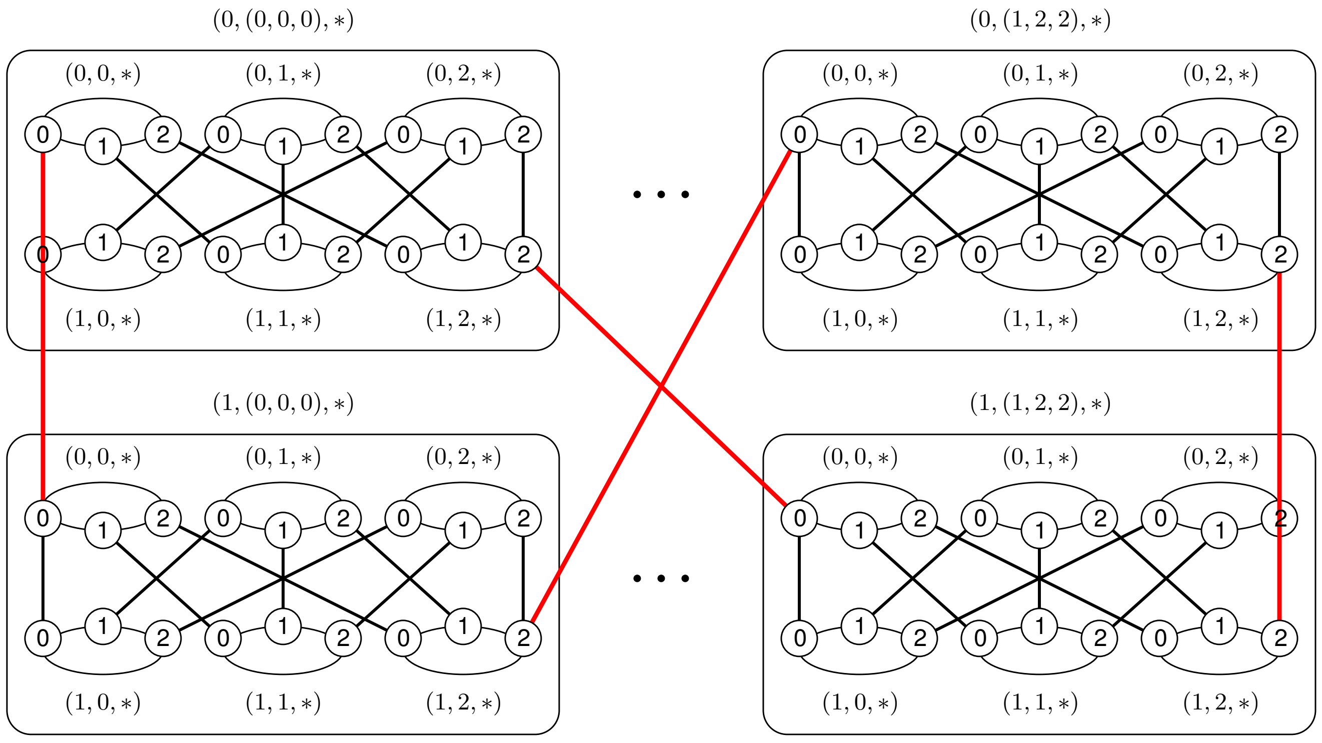


Figure 6. Presentation of $RDN^2(B(3))$

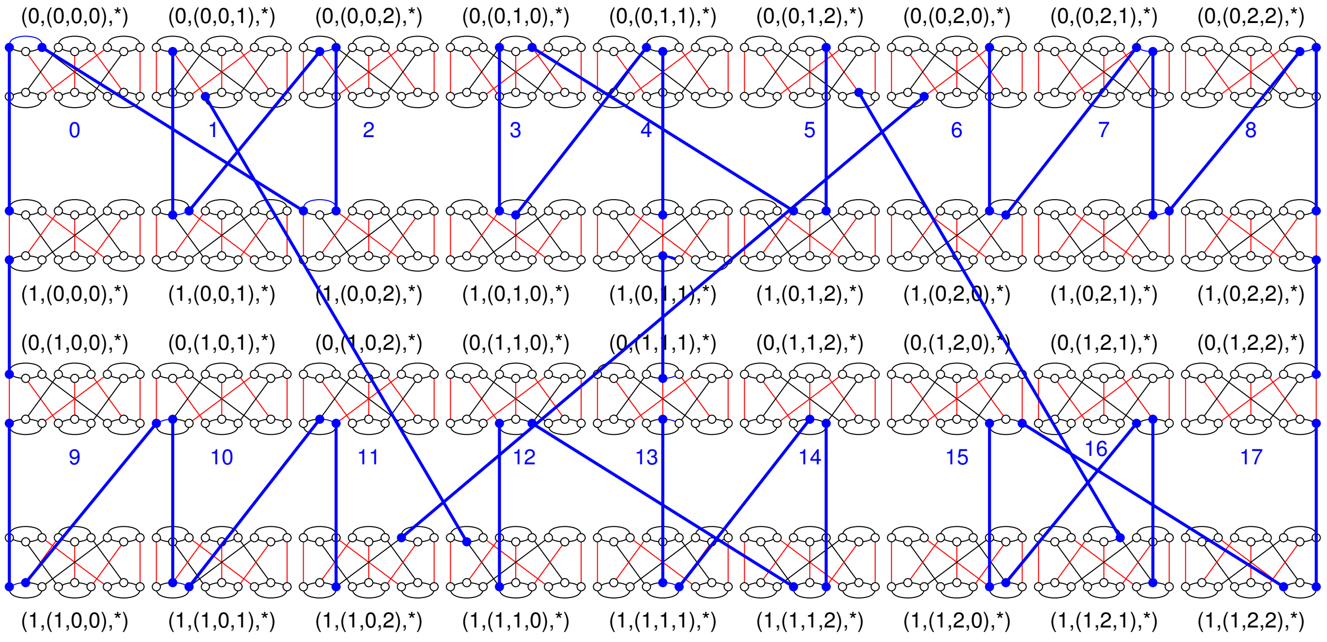


Figure 9. A hamiltonian cycle in $RDN^2(B(3))$

shows an example of a hamiltonian cycle in $RDN^2(B(3))$. Figure 8(a) is the virtual hamiltonian cycle.

Then the hamiltonian cycle in an $RDN^1(B)$ is used for constructing a hamiltonian cycle in an $RDN^2(B)$. The virtual hamiltonian cycle in $RDN^2(B)$ can be constructed along with the hamiltonian cycle in $RDN^1(B)$. For example, by following the node sequence shown in Figure 8(b), we can construct a virtual hamiltonian cycle in $RDN^2(B(3))$:

(0,(0,0,0),(0,0,0)) → (0,(0,0,0),(0,0,2)) → (1,(0,0,2),(0,0,0)) → (1,(0,0,2),(0,0,2)) →

(0,(0,0,2),(0,0,2)) → (0,(0,0,2),(0,0,1)) → (1,(0,0,1),(0,0,2)) → (1,(0,0,1),(0,0,1)) →

(0,(0,0,1),(0,0,1)) → (0,(0,0,1),(1,1,0)) → (1,(1,1,0),(0,0,1)) → (1,(1,1,0),(1,1,0)) →

(0,(1,1,0),(1,1,0)) → (0,(1,1,0),(1,1,2)) → (1,(1,1,2),(1,1,0)) → (1,(1,1,2),(1,1,2)) →

(0,(1,1,2),(1,1,2)) → (0,(1,1,2),(1,1,1)) → (1,(1,1,1),(1,1,2)) → (1,(1,1,1),(1,1,1)) →

(0,(1,1,1),(1,1,1)) → (0,(1,1,1),(0,1,1)) → (1,(0,1,1),(1,1,1)) → (1,(0,1,1),(0,1,1)) →

(0,(0,1,1),(0,1,1)) → (0,(0,1,1),(0,1,0)) → (1,(0,1,0),(0,1,1)) → (1,(0,1,0),(0,1,0)) →

(0,(0,1,0),(0,1,0)) → (0,(0,1,0),(0,1,2)) → (1,(0,1,2),(0,1,0)) → (1,(0,1,2),(0,1,2)) →

(0,(0,1,2),(0,1,2)) → (0,(0,1,2),(1,2,1)) → (1,(1,2,1),(0,1,2)) → (1,(1,2,1),(1,2,1)) →

(0,(1,2,1),(1,2,1)) → (0,(1,2,1),(1,2,0)) → (1,(1,2,0),(1,2,1)) → (1,(1,2,0),(1,2,0)) →

(0,(1,2,0),(1,2,0)) → (0,(1,2,0),(1,2,2)) → (1,(1,2,2),(1,2,0)) → (1,(1,2,2),(1,2,2)) →

(0,(1,2,2),(1,2,2)) → (0,(1,2,2),(0,2,2)) → (1,(0,2,2),(1,2,2)) → (1,(0,2,2),(0,2,2)) →

(0,(0,2,2),(0,2,2)) → (0,(0,2,2),(0,2,1)) → (1,(0,2,1),(0,2,2)) → (1,(0,2,1),(0,2,1)) →

(0,(0,2,1),(0,2,1)) → (0,(0,2,1),(0,2,0)) → (1,(0,2,0),(0,2,1)) → (1,(0,2,0),(0,2,0)) →

(0,(0,2,0),(0,2,0)) → (0,(0,2,0),(1,0,2)) → (1,(1,0,2),(0,2,0)) → (1,(1,0,2),(1,0,2)) →

(0,(1,0,2),(1,0,2)) → (0,(1,0,2),(1,0,1)) → (1,(1,0,1),(1,0,2)) → (1,(1,0,1),(1,0,1)) →

(0,(1,0,1),(1,0,1)) → (0,(1,0,1),(1,0,0)) → (1,(1,0,0),(1,0,1)) → (1,(1,0,0),(1,0,0)) →

(0,(1,0,0),(1,0,0)) → (0,(1,0,0),(0,0,0)) → (1,(0,0,0),(1,0,0)) → (1,(0,0,0),(0,0,0)) →

By replacing every edge in clusters with a path, we can get a hamiltonian cycle. Figure 9 shows a hamiltonian cycle in $RDN^2(B(3))$ in which a cluster is an $RDN^1(B(3))$ given in Figure 8(b). The solid cycles are the nodes in the virtual hamiltonian cycle. The algorithm for embedding a hamiltonian cycle in an $RDN(B)$ is given in Algorithm 1.

In Algorithm 1, there are two arrays, HC and VHC, to store node IDs in a hamiltonian cycle and in a virtual hamiltonian cycle, respectively. For simplicity, if there are n nodes in a cycle, we store $n+1$ nodes' IDs in HC: The last node ID is the same as the first node ID. The algorithm first constructs a virtual hamiltonian cycle VHC in $RDN^1(B)$ based on the hamiltonian cycle HC in $RDN^0(B)$. Then each pair-node in VHC is replaced with a hamiltonian path in $RDN^0(B)$. We got a hamiltonian cycle in $RDN^1(B)$. Let the cycle be HC, repeat the same process until the hamiltonian cycle in $RDN^k(B)$ is constructed.

```
Algorithm 1: RDN_HC($n_0, k$)
begin
  for $i \leftarrow 0$ to $n_0$ do                    /* in the base nrtwork */
    $HC[i] \leftarrow i$;                         /* $HC$: hamiltonian cycle */
  endfor
  $HC[n_0] \leftarrow 0$;                       /* last node = first node */
  for $j \leftarrow 0$ to $k-1$ do                          /* $k$ levels */
    $n \leftarrow (2n_0)^{2^j}/2$;          /* # of nodes in $RDN^{k-1}(B)$ */
    $VHC \leftarrow \emptyset$;       /* $VHC$: virtual hamiltonian cycle */
    for $i \leftarrow 0$ to $n-1$ do                        /* $n$ nodes */
      $u \leftarrow HC[i]$;
      $v \leftarrow HC[i+1]$;
      $VHC \leftarrow VHC \cup (0,u,u)$;   /* node $N_0$, type 0 */
      $VHC \leftarrow VHC \cup (0,u,v)$;   /* node $N_1$, type 0 */
      $VHC \leftarrow VHC \cup (1,v,u)$;   /* node $N_2$, type 1 */
      $VHC \leftarrow VHC \cup (1,v,v)$;   /* node $N_3$, type 1 */
    endfor
    for each pair nodes $(u,v)$ in $VHC$ do
      Replace $(u,v)$ with a hamiltonian path $u \rightarrow v$;
    endfor
    $u \leftarrow HC[0]$;
    $HC \leftarrow VHC \cup (0,u,u)$;  /* last node = first node */
  endfor
end
```

Theorem 2 *If the base network B is hamiltonian then $RDN^k(B)$ is hamiltonian for any $k > 0$.*

Proof: Suppose that nodes $0, 1, \ldots, n_0 - 1$ form a hamiltonian cycle C_B in the base network B. That is, node i and node $(i+1) \mod n_0$ are neighbors. For $k = 1$, the virtual hamiltonian cycle constructed in Algorithm 1 is a cycle that consists of $4n_0$ nodes and travels through all the $2n_0$ clusters: Since nodes u and v are two neighbors in the hamiltonian cycle in B, node N_0 with ID $(0,u,u)$ and node N_1 with ID $(0,u,v)$ are two neighbor nodes of type 0 in C_B; node N_2 with ID $(1,v,u)$ and node N_3 with ID $(1,v,v)$ are two neighbor nodes of type 1 in C_B; and N_1 and N_2 are connected with a cross edge. Replacing nodes (N_0, N_1) and (N_2, N_3) with $(N_0 \rightarrow N_1)$ and $(N_2 \rightarrow N_3)$ inside C_B, respectively, and doing the same for all the pair nodes in C_B, a cycle that contains $2n_0^2$ nodes is constructed.

To embed a hamiltonian cycle in $RDN^k(B)$ for $k > 1$, the algorithm takes the hamiltonian cycle constructed in $RDN^{k-1}(B)$ as the base network and apply the same process as in $RDN^1(B)$. ❑

Corollary 1 *If the base network B is hamiltonian then there are $\lfloor (2n_0)^{k-1}/4 \rfloor$disjoint virtual hamiltonian cycles in $RDN^k(B)$ where n_0 is the number of nodes in B.*

Proof: Consider the $RDN(B)$ with $k = 1$. Notice that only two adjacent nodes of the hamiltonian cycle in the base network B appear in the virtual hamiltonian cycle. Algorithm 1 constructs a virtual hamiltonian cycle from node $(0,0,0)$. If the starting node is $(0,0,2i)$ for $i = 0, 1, \ldots, n_0/2 - 1$, then the $n_0/2$ virtual hamiltonian cycles are disjoint. Since we can use a hamiltonian cycle constructed in $RDN^{k-1}(B)$ as the base network, the argument can be expanded to any level of k. ❑

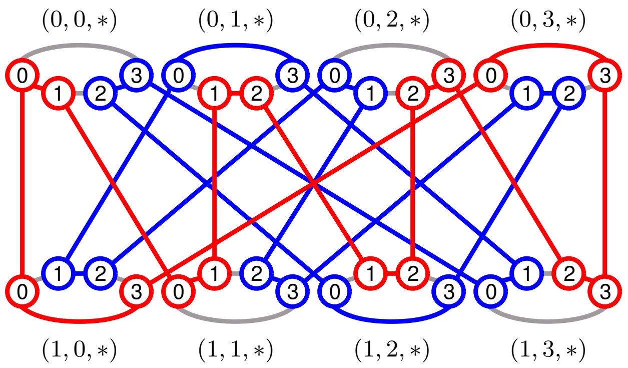


Figure 10. Two disjoint virtual hamiltonian cycles

As an example, Figure 10 shows two disjoint virtual hamiltonian cycles in an $RDN^1(B(4))$ where $B(4)$ is a 4-node ring. One virtual hamiltonian cycle starts from node $(0,0,0)$, the other starts from node $(0,0,2)$.

4. Hamiltonian Connectedness

If there is a hamiltonian path between *any* two distinct nodes in a graph, we say that this graph is *hamiltonian connected*. We show the hamiltonian connectedness of the Recursive Dual-Net. Note that a 3-node ring is a hamiltonian connected graph but a 4-node ring is not. If the base network is hamiltonian connected, any two distinct nodes in the base network can be used when we construct the virtual hamiltonian cycle because there is always a hamiltonian path in the base network between the two nodes.

Theorem 3 *If the base network B is hamiltonian connected, the $RDN^k(B)$ is also hamiltonian connected for any $k > 0$.*

Proof: Let the k-level cluster containing node u be C_u. We prove the theorem by induction on k. Assume that $RDN^{k-1}(B)$ is hamiltonian connected. For any two distinct nodes u and v in $RDN^k(B)$, we want to show that there is a hamiltonian path $u \to v$. We have three cases.

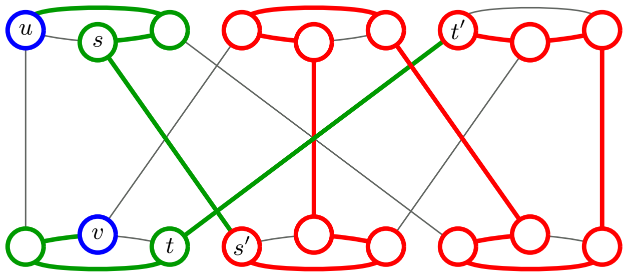


Figure 11. Case 1: u and v are of different types

Case 1: u and v are of different types. This is the simplest case. Let $s \in C_u$ and $s \neq u$, $t \in C_v$ and $t \neq v$. There are cross edges including (s, s') and (t, t') that form a *virtual hamiltonian path $VHP(s,t)$*. Then the hamiltonian path $P(u,v)$ in $RDN(B)$ can be $u \to s$, $s' \to t'$, $t \to v$ where $s' \to t'$ is a path connecting all the nodes in the other clusters than C_u and C_v. This is illustrated in Figure 11.

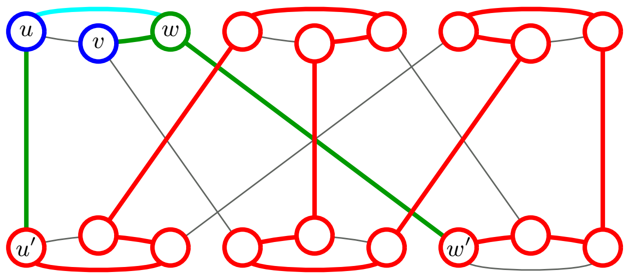


Figure 12. Case 2: u and v are in a same cluster

Case 2: u and v are in a same cluster. This is also simple. Find a hamiltonian path u, $w \to v$ within the cluster C_u. Then there is a virtual hamiltonian path $VHP(u,w)$. The hamiltonian path $P(u,v)$ in $RDN^k(B)$ can be u, $u' \to w'$, $w \to v$ where $u' \to w'$ is a path connecting all the nodes in the other clusters than C_u. This is illustrated in Figure 12.

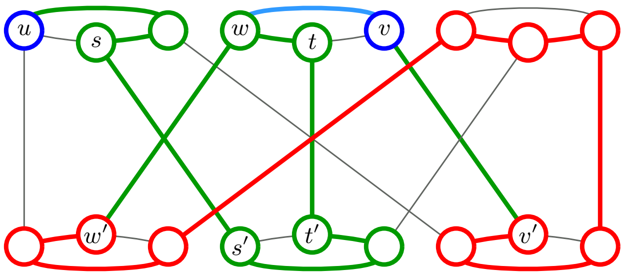


Figure 13. Case 3: u and v are in different clusters of the same type

Case 3: u and v are different clusters of the same type. This case is a little bit complex. In C_u, find a hamiltonian path $u \to s$ where $s \neq u$. In $C_{s'}$, find a hamiltonian path $s' \to t'$ where $t \in C_v$. In C_v, find a hamiltonian path v, $w \to t$. Then the hamiltonian path $P(u,v)$ in $RDN^k(B)$ can be $u \to s$, $s' \to t', t \to w$, $w' \to v'$, v where $w' \to v'$ is a path connecting all the nodes in the other clusters than C_u, C_v, and $C_{t'}$. This is illustrated in Figure 13. ❏

5. Concluding Remarks

The Recursive Dual-Net can connect a large number of nodes with a small node-degree and a short diameter. It is a potential candidate for the interconnection network of the supercomputers of the next generation that have more than one million of nodes.

We can select any of popular networks of small sizes that are symmetric as the base network and then connect multiple base modules with cross links (cables) to construct a Recursive Dual-Net of very large scale. The base networks can be implemented in a NoC VLSI and line cables may be used as the cross links to connect PCB modules in cabinets.

We showed that the Recursive Dual-Net keeps the properties of the base network including the hamiltonian connectivities and presented an efficient algorithm to embed a hamiltonian cycle in the Recursive Dual-Net. Fault-tolerant cycle embedding in the Recursive Dual-Net may be interesting for further research.

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] G. H. Chen and D. R. Duh. Topological properties, communication, and computation on wk-recursive networks. *Networks*, 24(6):303–317, 1994.

[4] J. S. Fu. Hamiltonian-connectedness of the wk-recursive network. In *Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN04)*, pages 569–574, 2004.

[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 and Hypercubes*. Morgan Kaufmann Pub, 1992.

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

[8] 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.

[9] Y. Li, S. Peng, and W. Chu. Hamiltonian cycle embedding for fault tolerance in dual-cube. In *Proceedings of the IASTED International Conference on Networks, Parallel and Distributed Processing, and Applications (NPDPA 2002)*, pages 1–6, Tsukuba, Japan, October 2002.

[10] Y. Li, S. Peng, and W. Chu. Efficient collective communications in dual-cube. *The Journal of Supercomputing*, 28(1):71–90, April 2004.

[11] Y. Li, S. Peng, and W. Chu. An algorithm for constructing hamiltonian cycle in metacube networks. In *Proceedings of the International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'07)*, pages 285–292, Adelaide, Australia, Dec. 2007. IEEE Press.

[12] Y. Li, S. Peng, and W. Chu. Recursive dual-net: A new universal network for supercomputers. 2009. to be published.

[13] F. P. Preparata and J. Vuillemin. The cube-connected cycles: a versatile network for parallel computation. *Commun. ACM*, 24:300–309, May 1981.

[14] Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, July 1988.

[15] TOP500. *Supercomputer Sites*. http://top500.org/, Jun. 2008.

[16] A. Varma and C. S. Raghavendra. *Interconnection Networks for Multiprocessors and Multicomputers: Theory and Practice*. IEEE Computer Society Press, 1994.

[17] G. Vicchia and C. Sanges. A recursively scalable network vlsi implementation. *Future Generation Computer Systems*, 4(3):235–243, 1988.